Capítulo 2

Conceitos de Mineração de Dados na Web

Rafael Santos

Resumo

Já não é mais possível apresentar a Web como uma novidade, comentando sobre suas características básicas – sua pervasividade e ubiqüidade a tornam uma ferramenta conhecida de todos, sendo praticamente o maior repositório de dados publicamente acessíveis na atualidade. Alguns aspectos e coleções de conteúdo da Web são parcialmente indexados, e existem mecanismos relativamente efetivos de fazer pesquisas em seus dados, mas a maioria destes são passivos ou reativos, dependendo de indexação manual por palavras-chave ou semi-automática por conteúdo (que podem ser enriquecidas por informações auxiliares) para oferecer resultados aceitáveis.

Outras técnicas mais eficientes e inteligentes podem ser usadas para aumentar o potencial de descoberta de conhecimento usando os dados existentes na Web. Algumas técnicas que tem sido investigadas e aplicadas com algum sucesso são técnicas de mineração de dados.

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 conhecimento pode ser na forma de regras descritivas dos dados, modelos que permitem a classificação de dados desconhecidos a partir de análise de dados já conhecidos, previsões, detecção de anomalias, visualização anotada ou dirigida, etc. Embora muitas destas técnicas tratem com dados tabulares, é possível extrair informações tabulares de dados estruturados de forma diferente (como encontrados na Web) ou mesmo usar algoritmos específicos para minerar dados da Web como conjuntos de links entre documentos.*

Este curso apresenta alguns conceitos básicos de mineração de dados e descoberta de conhecimento em bases de dados, com ênfase em dados estruturados como os da Web: textos (estruturados de diversas formas e em diversos graus, como hiperdocumentos, e-mail, arquivos XML e outros tipos), imagens e vídeos, registros de acesso a servidores, metadados (como redes ou grafos que representam ligações entre documentos e objetos como participantes de redes sociais), etc.

2.1. Introdução

Estamos nos afogando em informação mas com sede de conhecimento – John Naisbitt, Megatendências, 1982.

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 terabytes em computadores pessoais a um custo acessível. Uma quantidade quase incomensurável de informações de diversos tipos, origens, formatos e finalidades estão disponíveis na Internet, podendo ser acessadas a partir destes dispositivos comuns.

O baixo custo dos dispositivos e do acesso a redes de computadores fez também com que o número de usuários destes sistemas aumentasse consideravelmente. Novas ferramentas permitem que estes usuários criem conteúdo digital de forma relativamente simples e barata, o que só faz aumentar a quantidade de informações disponíveis para outros usuários.

Esta vasta quantidade de informações, embora facilmente acessível, nem sempre é facilmente localizável. Alguns sites na Internet indexam informações de determinadas categorias de forma controlada e organizada a um certo custo computacional e/ou humano, como, por exemplo, sites especializados como o *Internet Movie Database* (`www.imdb.com`) ou SourceForge (`sourceforge.net`). Outros indexam conteúdo externo permitindo a busca usando palavras-chave ou opções mais complexas de busca, como o Google (`www.google.com`) ou Bing (`www.bing.com`). Ainda outros funcionam como portais apresentando informações externas (em outros sites) de forma categorizada e com conteúdo personalizado.

Podemos observar então que existe um esforço considerável, em várias frentes e exercido de várias formas, de tentar organizar, indexar e categorizar informações já existentes na Internet. Um problema enfrentado por estes esforços é que as informações nem sempre são facilmente organizáveis (justamente e paradoxalmente por causa da facilidade com que podem ser coletadas e distribuídas; e por causa de sua própria estrutura e natureza).

Outro problema é a quantidade e variedade de informações que devem ser organizadas. Alguns exemplos mais específicos do volume de informações são apresentados a seguir[1]:

- De acordo com algumas estimativas[2], o site YouTube continha 45 terabytes de ví-

[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 *International Data Corporation*, 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://www.businessintelligencelowdown.com/2007/02/top_10_largest_.html

deos em 2006. O site Flickr tinha 2 bilhões de fotografias digitais[3] em 2007 (e um teste rápido mostrou que já podem ser ao menos 3.7 bilhões).

- 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[4].
- O site CiteSeerX (`citeseerx.ist.psu.edu`) indexa mais de 1.400.000 artigos científicos e 27.000.000 citações, e contém muitas informações adicionais, inclusive referências cruzadas.
- O site da editora Springer (`www.springerlink.com/content`) contém mais de 4.400.000 artigos científicos completos, também com muitas informações adicionais.
- O site de relacionamentos Facebook (`www.facebook.com`) contém 250 milhões de usuários que participam de alguns dos 45 milhões de grupos de interesse no site. O site recebe um bilhão de fotografias digitais por mês, e tem um bilhão de informações como notícias, links, blogs, etc. compartilhados por mês[5].
- O já mencionado Sourceforge contém 230.000 projetos de software aberto, cada um com código fonte, páginas, documentos, listas de e-mails etc. indexados e organizados.
- De acordo com uma estimativa da Nielsen (`www.blogpulse.com`), existiam, em Agosto de 2009, mais de 114 milhões de blogs, com quase 90 mil novos blogs criados por dia.
- O site *Internet Movie Database* contém informações categorizadas sobre quase 1.500.000 de filmes, mais de 3.000.000 de pessoas envolvidas com os filmes, e mais de 1.600.000 links para documentos relacionados.

É importante então ter ferramentas que possibilitem a procura de informação entre esta avalanche de dados. A distinção entre dado e informação é sutil mas importante: dados podem ser coletados de forma rápida, simples e automática, e armazenados em grande volume a baixo custo; informações são de nível semântico mais alto. De forma simplista podemos considerar texto como sendo dados, e o conteúdo deste texto como sendo informações.

Informações podem ser obtidas a partir de dados através de técnicas de interpretação, anotação, classificação, agrupamento, sumarização, etc. destes dados ou de técnicas que permitam a associação e correlação de outras informações (possivelmente de outras fontes). Várias destas técnicas fazem parte do conjunto de técnicas, ferramentas, procedimentos e algoritmos conhecidos comumente como Mineração de Dados (*Data Mining*), que por sua vez faz parte de um processo conhecido como Descoberta de Conhecimento em Bancos de Dados (KDD, *Knowledge Discovery in Databases*). Estes conceitos serão detalhados na seção 2.2.

[3] http://www.techcrunch.com/2007/11/13/2-billion-photos-on-flickr
[4] ftp://ftp.ncbi.nih.gov/genbank/gbrel.txt
[5] http://www.facebook.com/press/info.php?statistics

Técnicas de mineração de dados tem sido usadas com sucesso para resolver vários problemas relacionados com extração e representação de conhecimento a partir de documentos e estruturas da Web, como, por exemplo, extração de conteúdo a partir de hiperdocumentos [8, 31, 39, 73]; identificação de padrões nas estruturas dos hiperdocumentos [17, 48, 92]; aplicações em redes sociais e sistemas de recomendação [4, 26, 94, 100]; mineração de metadados como registros (*logs*) [7, 57, 77]; etc. Estas e outras aplicações serão detalhadas na seção 2.6.

O sucesso de algumas aplicações e a necessidade de soluções melhores, mais rápidas, mais escaláveis e mais precisas (ou simplesmente melhor desenhadas para determinada aplicação) para a extração de conhecimento a partir da Web motiva investimento de empresas, a participação de cientistas e o envolvimento de professores, estudantes e profissionais que usam a Web para coletar ou fornecer informações.

O restante deste capítulo é organizado da seguinte forma: a seção 2.2 apresenta com detalhes os conceitos de mineração de dados e descoberta de conhecimento em bancos de dados. A seção 2.3 apresenta as técnicas mais usadas e conhecidas de mineração de dados, que podem ser aplicadas a algumas categorias de dados da Web e a seção 2.4 mostra como estes dados da Web são representados e como dados podem ser extraídos para mineração. Outra categoria de dados da Web são dados estruturais, que frequentemente são representados como grafos; este tópico será apresentado na seção 2.5. A seção 2.6 apresenta vários exemplos de aplicação de técnicas de mineração de dados para aplicações na Web como modelagem de usuários, análise de conteúdo, etc. e outras possíveis áreas de aplicação. Finalmente a seção 2.7 apresenta algumas ferramentas para testes e prototipação de algoritmos.

2.2. Conceitos de Mineração de Dados

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 [35]). O processo de descoberta de conhecimentos em bancos de dados é ilustrado na Figura 2.1.

Ainda de acordo com [35], e usando a Figura 2.1 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 2.1 são destacados 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**);

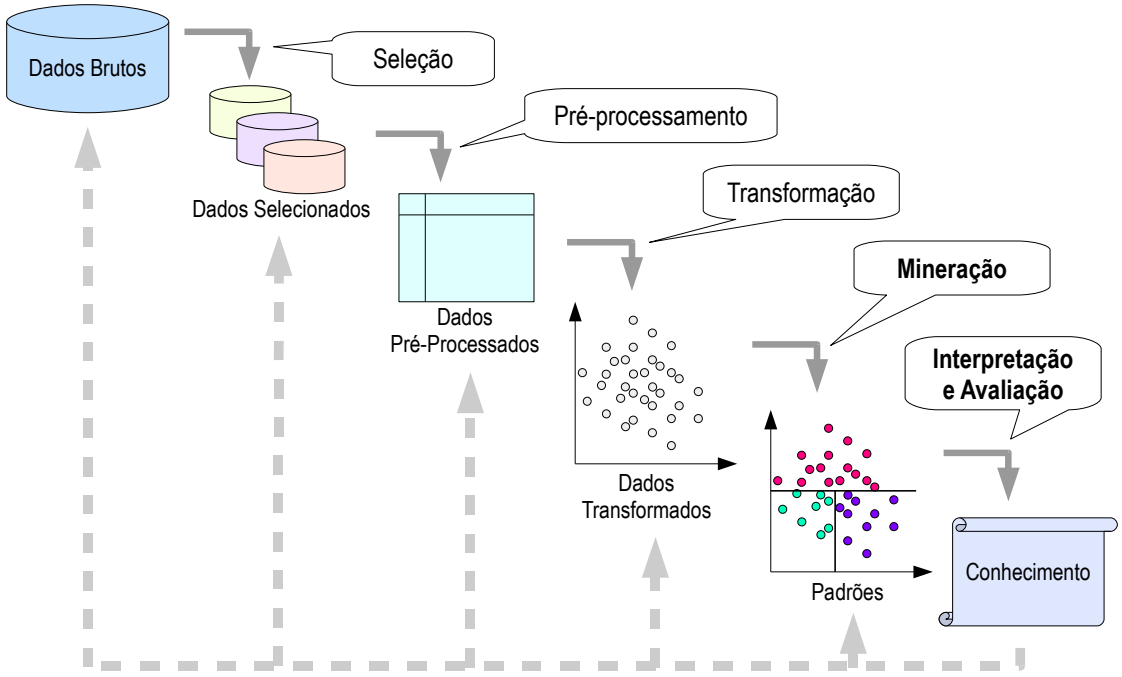

Figura 2.1. Processo de Descoberta de Conhecimento em Bancos de Dados (adaptado de [35])

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, extração de atributos numéricos de documentos e *logs*, 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.);

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 2.1 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 da Web, 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 [95]). De acordo com [47], 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.

2.3. Técnicas Básicas 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 a 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 [70, 75] – 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.

Um exemplo de conjunto de dados representado em tabela, que ilustra estes conceitos, é mostrado na Tabela 2.1 (com dados e atributos fictícios sobre um documento na

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

Tabela 2.1. Exemplo de dados para mineração em forma de tabela

Web). 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 ('I' ou 'P'). A classe é discreta, podendo assumir os valores "alto", "médio" ou "baixo". 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. Devemos, em um passo anterior ao da mineração, decidir se eliminamos o registro com informação incompleta, se eliminamos todo o atributo ou se tentamos derivar um valor usando uma técnica qualquer.

Um outro conceito importante usado em mineração de dados 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 facilmente visualizado se tivermos duas ou três dimensões (dados com mais de três dimensões podem ser visualizados com técnicas específicas). 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 [47]):

- **Classificação**: Descoberta de uma função preditiva que consegue classificar um dado em uma de várias classes discretas que são predefinidas ou conhecidas. Um exemplo (usando a Tabela 2.1) seria a classificação do conteúdo de um documento a partir de atributos medidos do mesmo, 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. O algoritmo de classificação aprenderá que testes e valores devem ser aplicados aos atributos para decidir por uma classe. 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 2.1 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; ou verificar se os dados para determinada classe formam grupos compactos e bem separados dos de outras classes.

- **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 2.1 e expressa com regras: documentos classificados como "alto" tem o valor do atributo A_2 maior do que 50 e documentos classificados como "médio" tem os valores de A_1 maiores que 100.
  É possível sumarizar os dados de uma base ou coleção através de técnicas de classificação, mas nem toda técnica de classificação cria modelos que descrevem os dados que podem ser facilmente interpretados.

- **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 [76] 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 [82]. 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 2.2.

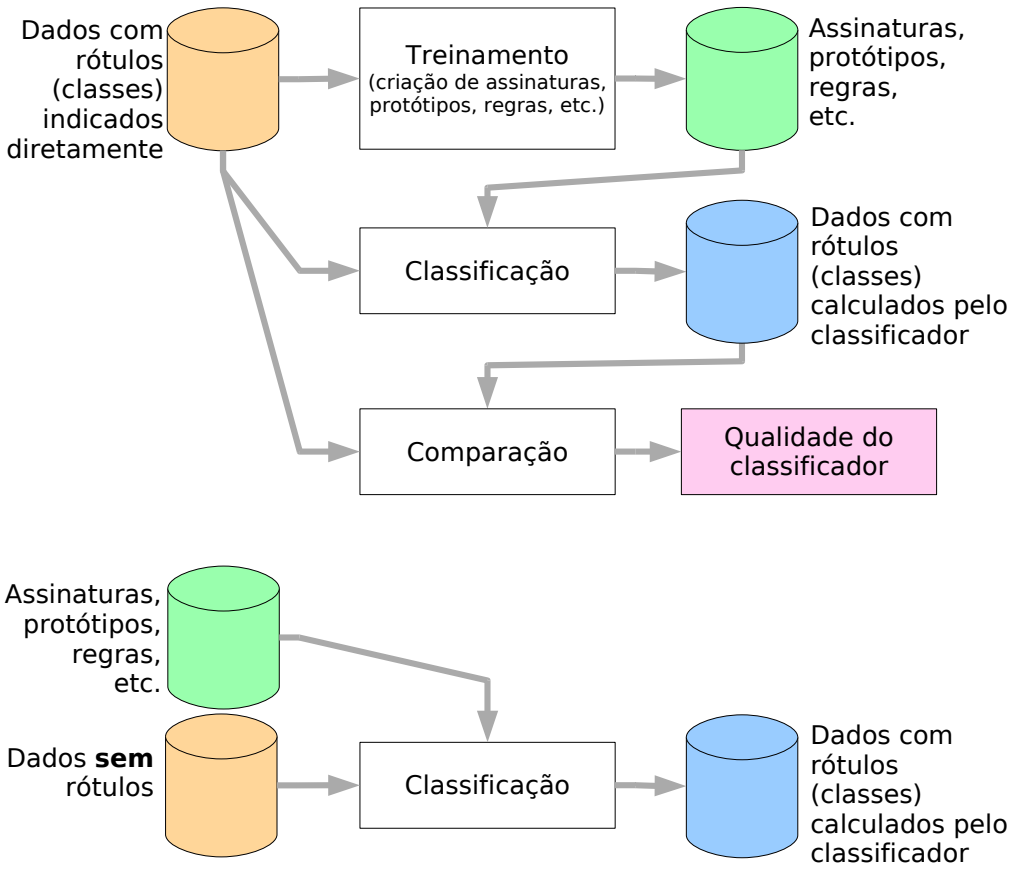

Figura 2.2. Processo de Classificação Supervisionada de Dados

Para a 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, como mencionado anteriormente, 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 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 2.2) 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 2.2): 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 [76, 80] e método do paralelepípedo [78, 91]. Sistemas especialistas [24, 45], 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 [10, 34, 56, 91] e *Support Vector Machines* [28, 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 2.3 ilustra esta diferença.

A Figura 2.3 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 2.3 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 2.3 temos uma classificação feita por uma

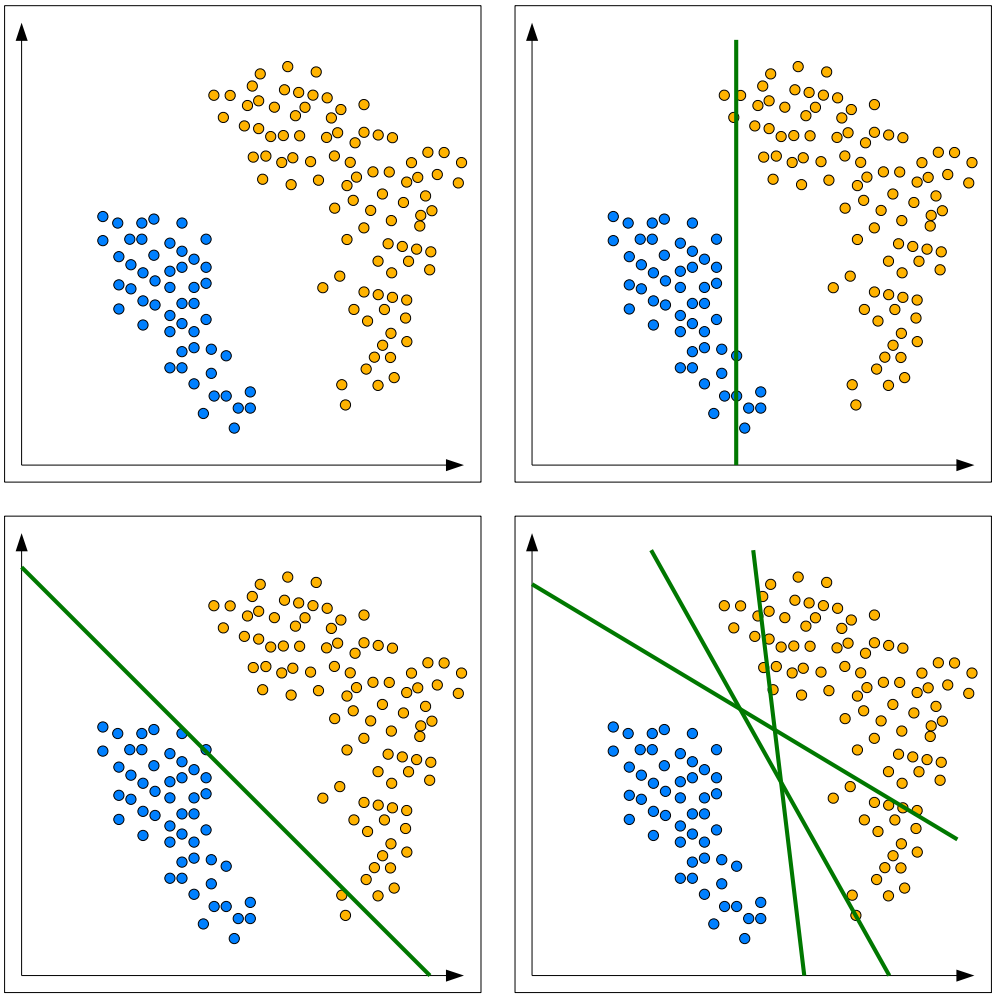

Figura 2.3. Separação de duas classes no espaço de atributos

rede neural com um único neurônio e com treinamento inadequado. 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 [56, 78, 91] ou máxima verossimilhança entre distribuições de classes [78, 91].

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 [46, 56, 80]. 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 2.4 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 2.4 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, 4 e 10. 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 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 [15], 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 [81, 82].

O algoritmo Fuzzy C-Médias tem muitas variantes [16, 29, 69, 83], 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 (mas que pode ser usado para detectar agrupamentos em dados numéri-

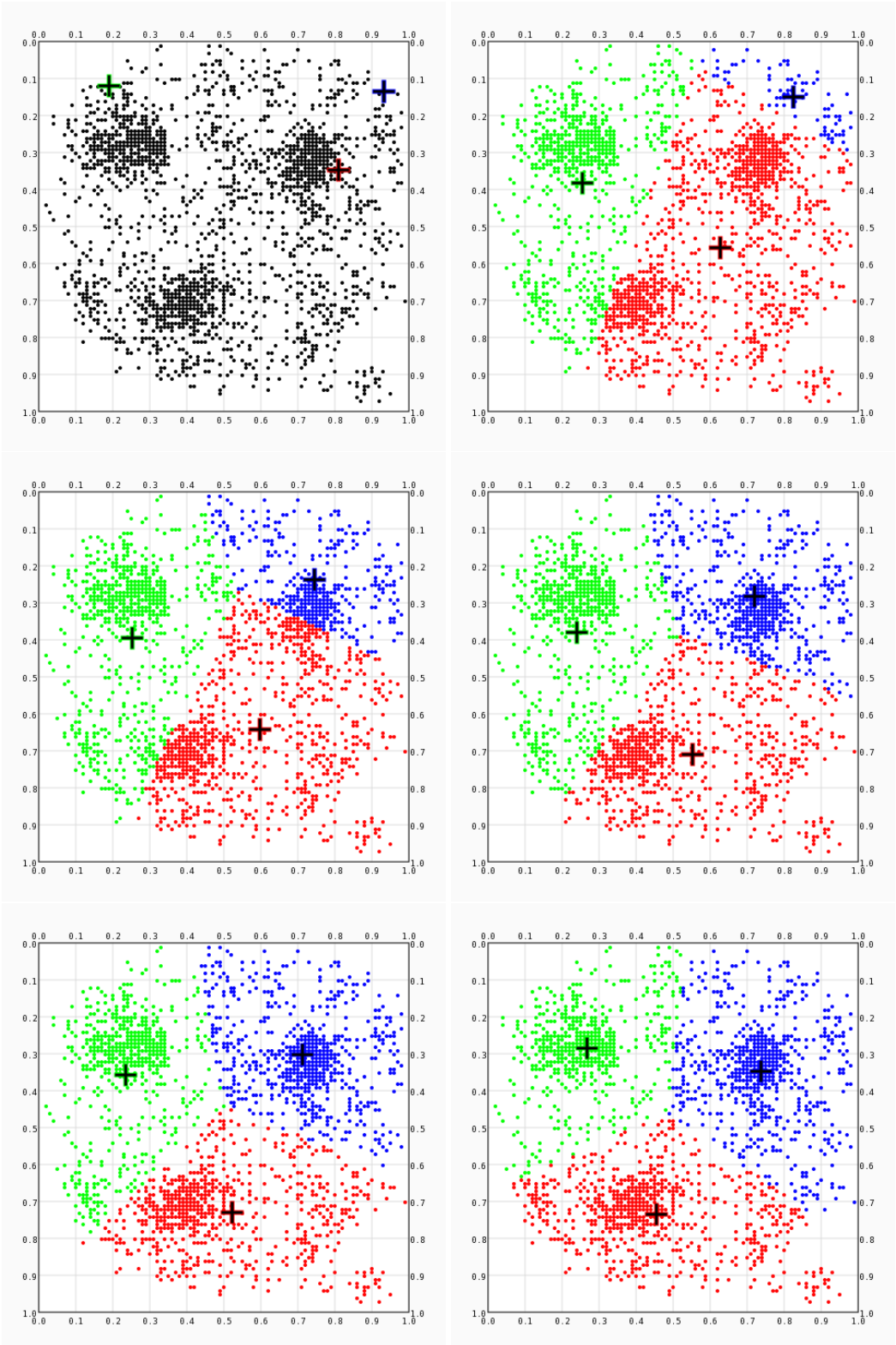

Figura 2.4. Passos do algoritmo K-Médias

cos), é o Isodata [46, 56]. 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.

Outras variantes do K-Médias, em particular uma usada para agrupamento de documentos, são descritas em [58], que também contém referências a outras técnicas de agrupamento e sua aplicação em documentos.

Os Mapas Auto-Organizáveis de Kohonen [34, 50] 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 [6, 46], 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 2.5 mostra um dendograma criado com dados genéticos de 938 indivíduos de 51 populações [53].

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 [76, 80], 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 2.6 (adaptada de [52]) 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 2.6 os atributos usados para a decisão são indicados dentro de elipses, as decisões (classificações) dentro de retângulos e as arestas entre elipses e retângulos 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

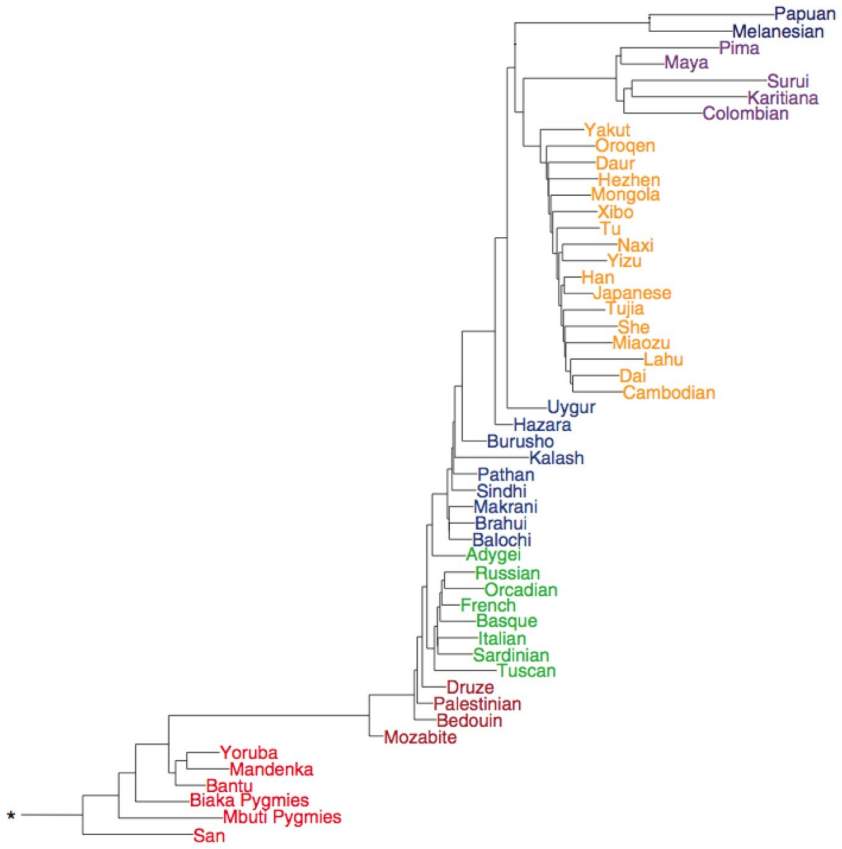

Figura 2.5. Dendograma criado com dados genéricos de populações humanas ([53])

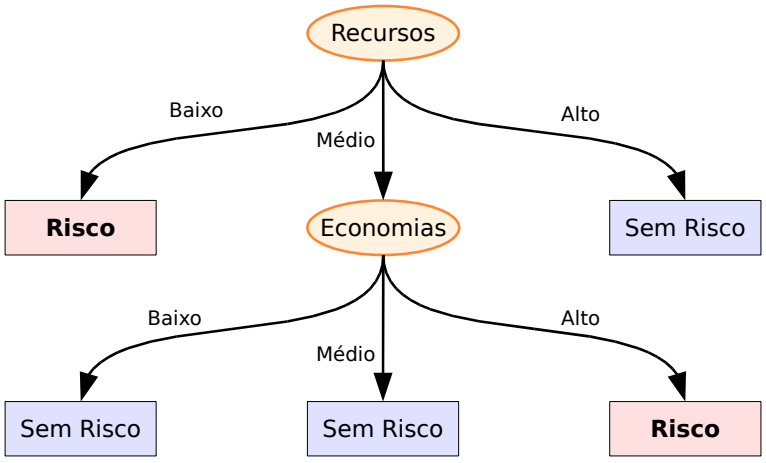

Figura 2.6. Árvore de decisão (adaptado de [52])

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 mon-

tar árvores de decisão a partir de dados rotulados podem ser encontrados em [52, 81].

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 [52, 81].

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 2.7.

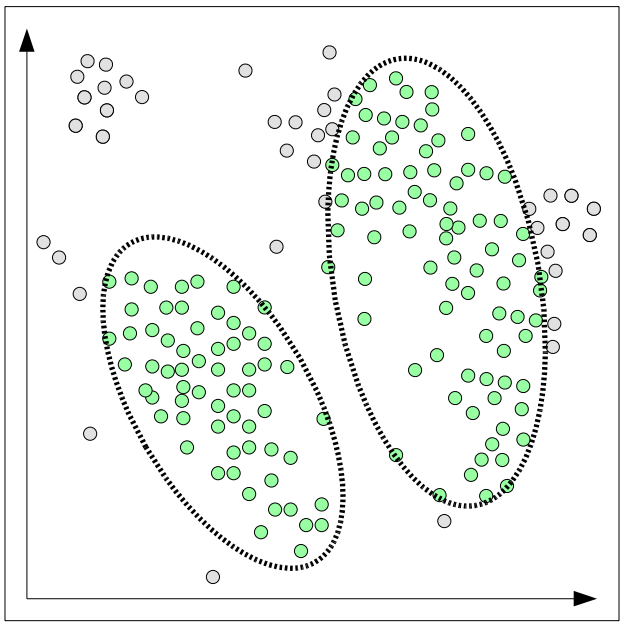

Figura 2.7. Possíveis *outliers*

Na Figura 2.7 (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*.

2.4. Representação e Processamento de Dados da Web para Mineração

Mineração de dados na Web tem três enfoques principais [40]:

- **Mineração de Conteúdo da Web**, que é o processo de extração de conhecimento do conteúdo de documentos e de seus metadados (descrição, informações sobre autores, palavras-chave, etc.). Este enfoque abrange principalmente documentos textuais (páginas em texto, HTML ou outros formatos; e-mails, listas de discussão, grupos de usuários, *blogs*, etc.), mas podemos também incluir mineração de dados multimídia na Web usando ou não dados textuais associados. Neste capítulo não veremos informações específicas sobre mineração de dados multimídia, que podem ser encontradas em [70, 80, 81, 85, 101].

- **Mineração de Estruturas da Web**, que é o processo de descoberta de conhecimento a partir da organização da Web, em especial através da ligação entre documentos na Web.

- **Mineração de Uso da Web**, que envolve a análise de dados coletados sobre o acesso a documentos na Web (em particular, *logs*), geralmente com a intenção de descobrir padrões de acesso a sites ou conjuntos de documentos para melhorar a qualidade da experiência do usuário ou para modelar o comportamento dos mesmos.

Os três enfoques não são mutuamente exclusivos: frequentemente usa-se um conjunto de dados como suporte a outro. Por exemplo, algumas abordagens (ex. [54, 92]) usam dados de conteúdo dos documentos e das ligações entre documentos para tarefas específicas de mineração, e outras (ex. [13, 90]) usam *logs* de servidores juntamente com as estruturas correspondentes dos sites para melhor caracterizar os padrões de acesso dos usuários.

A natureza dos dados que podem ser usados nos diferentes enfoques varia bastante: dados de conteúdo são geralmente textuais, com alguma estrutura, dependendo do formato (HTML, e-mails), que indica seções ou identifica metadados dos documentos. Dados sobre uso na Web, em geral são estruturalmente bem mais simples, representados como entradas temporais em uma base de dados textual (*logs*) que podem ser praticamente considerados como uma tabela de bancos de dados relacionais. Dados de estruturas da Web são representados como grafos onde vértices representam objetos na Web e arestas representam conexões entre estes objetos.

Os conceitos apresentados na seção 2.3 são comuns a muitas aplicações de mineração de dados, mas partem do pressuposto de que os dados a ser minerados estão estruturados de forma relacional, ou seja, em tabelas organizadas em linhas (dados) e colunas (atributos). Dados disponíveis na Web raramente seguem este padrão (exceto para tarefas de análise de *logs*); e diferentes categorias de dados tem diferentes padrões e formatações, e devem ser preprocessados para uso com os algoritmos tradicionais de mineração de dados. Nesta seção veremos algumas das técnicas aplicáveis a alguns tipos de dados.

Dados textuais (documentos sem marcações ou relações com outros) podem ser reduzidos a tabelas de ocorrência e posição de palavras usando uma técnica chamada *índice invertido* [20]. Com esta técnica, documentos podem ser indexados e eventualmente comparados, de forma superficial e simples, quanto ao conteúdo. Os passos para preprocessamento para eventual indexação ou mineração são:

1. Remoção de elementos não textuais (pontuação, marcação como HTML, etc.)
2. Transformação de palavras em radicais (por exemplo, remoção de plurais, transformação em letras minúsculas, uso do infinitivo do verbo, etc.)
3. Transformar as palavras em *tokens* indexados numericamente.

Com estes passos um documento é convertido em uma lista de ocorrência de palavras que pode, alternativamente, conter informações sobre a posição da palavra no documento. Esta lista pode ser facilmente transformada em um vetor em um espaço de N dimensões onde N é o conjunto total de palavras sendo consideradas, e técnicas simples

de mineração de dados que usam distâncias podem calcular a distância entre dois vetores destes. Outras informações podem também ser armazenadas: se os documentos forem em HTML pode-se preservar em tabelas separadas trechos identificáveis como títulos do documento ou de seções. Documentos podem ser comparados usando a quantidade ou simples existência de palavras em comum. Variantes deste conceito são o uso de combinação de palavras, substituição de sinônimos, eliminação de palavras frequentes mas não essenciais, etc. [12] Mais informações sobre estas técnicas de indexação, inclusive variantes e contexto histórico, podem ser encontradas em [20].

A transformação de documentos em vetores de ocorrência de palavras permite também o uso de técnicas de agrupamento, para a criação de grupos de documentos que são semelhantes entre si. Em grandes coleções é possível fazer agrupamentos em lote, para evitar a execução de um algoritmo computacionalmente caro cada vez que um novo documento for inserido na coleção [12]. Alguns exemplos e extensões a algoritmos tradicionais são descritos em [62, 93, 98, 102].

Apesar da praticidade e aplicabilidade, é importante observar que indexação inversa não leva em conta o conteúdo real do documento, pois nenhuma informação semântica é criada ou preservada: o algoritmo somente assume que documentos são semelhantes se eles contém as mesmas palavras. Apesar da simplicidade este algoritmo pode ser usado com sucesso para indexar e comparar documentos de domínios específicos, como, por exemplo, artigos científicos.

Outro problema relevante é que para um corpo extensivo de palavras o vetor de atributos (ocorrência das palavras) para a maioria dos documentos pode ser bastante esparso, isto é, conter muitos zeros, afinal nem todos os documentos contém todas as palavras encontradas em toda a coleção de documentos. Técnicas especiais como redução para atributos relevantes, hierarquização dos termos ou redução de dimensionalidade podem ser aplicadas nestes casos [51, 61].

Uma desvantagem óbvia deste algoritmo simples é que a tabela gerada pode ser muito grande, mesmo para documentos pequenos. Dependendo da aplicação pode-se optar por reduzir o conjunto de palavras, mas isto sempre pode causar perda de capacidade de recuperação de documentos a partir dos termos. Outro problema é que para tarefas de recuperação de informação o algoritmo não indica qual documento é mais relevante para determinada busca – sem esta ordenação, usando somente presença ou não de palavras-chave, sistemas de busca falham em face à enorme quantidade de páginas na Internet. Por exemplo, uma busca recente (Agosto de 2009) no Google por *"data mining"* retorna mais de 9 milhões de páginas, seria impraticável procurar uma a uma pela informação desejada.

Para isto precisamos de um mecanismo de *ranking* ou ordenação dos resultados. Hipoteticamente a melhor forma de obter o *ranking* de documentos por termos é através da classificação da qualidade do resultado por usuários. Se muitos usuários concordam que um determinado documento atende à busca feita por eles, e que este determinado documento é mais relevante do que outros, este documento deve ser associado mais fortemente aos termos da busca feita. Na realidade isto é pouco prático, pois dependeria da boa vontade dos usuários em classificar a adequação do documento e possibilitaria a corrupção do *ranking* por ferramentas automatizadas com finalidades escusas (ex. *spamming*).

Por outro lado usuários podem se interessar por resultados de busca ordenados de acordo com algum critério autoritativo, sendo esta a razão principal do sucesso de sistemas de recomendação.

Uma forma de ordenar páginas encontradas por relevância foi criada com o algoritmo *PageRank* [12, 19, 55], usado no Google. Este algoritmo basicamente avalia páginas baseado na quantidade de ligações a ela feitas por outras páginas consideradas importantes. Uma variante do algoritmo diferencia tipos de páginas entre autoridades e distribuidoras: autoridades são páginas que são indicadas por várias outras, distribuidoras são páginas que contém muitas ligações para páginas consideradas autoridades [49]. Uma variante do algoritmo *PageRank* que considera páginas mais recentes com pesos maiores é descrita em [55], que também mostra mais detalhes sobre o funcionamento do algoritmo (inclusive com uma simulação da execução do mesmo).

O conjunto de ligações entre hiperdocumentos pode ser considerado como um grafo (e outras formas de estrutura na Web, como ligações entre participantes de uma rede social, citações em artigos científicos e mesmo co-participações em trabalhos [3]). Algoritmos que podem ser usados para minerar grafos como os de ligação entre hiperdocumentos são descritos na seção 2.5.

As duas áreas principais de aplicação de mineração de uso da Web são coleta de métricas de uso e extração de conhecimento para personalização [5]. Mineração de uso da Web envolve, frequentemente, preparação dos dados que são coletados automaticamente. Embora praticamente todos os servidores de dados na Web permitam a coleta de *logs* de acesso, esta informação por si só não é suficiente para caracterizar usuários – geralmente é preciso extrair as sessões de navegação dos usuários e frequentemente é desejável enriquecer esta informação com dados auxiliares, que nem sempre estão disponíveis no próprio servidor.

Técnicas para reorganizar a informação de acesso de forma que possa ser minerada podem ser classificadas em proativas (onde o servidor Web cuida da coleta detalhada de dados para a finalidade em vista) ou reativas (onde heurísticas tentam completar os dados *a posteriori*) [12]. Descrições de algumas destas técnicas são feitas em [13, 25, 74, 89]. Um estudo abrangente sobre técnicas de mineração de dados para personalização é apresentado em [64].

2.4.1. Mineração de Dados na Web Semântica

A Web Semântica[6] é uma iniciativa relativamente recente, inspirada por Tim Berners-Lee, que propõe o avanço da Web conhecida para que a mesma se torne um sistema distribuído de representação e processamento do conhecimento. O objetivo da Web Semântica não é somente permitir acesso a informação da Web em si através de sistemas de busca, mas também permitir o uso integrado de documentos na Web.

Algumas das características propostas para a Web Semântica são:

- **Formato padronizado**: A Web Semântica propõe padrões para uma linguagem descritiva de metadados uniforme, que além de servir como base para troca de da-

[6]Esta subseção foi adaptada de Berendt et al. [12].

dos, suporta representação do conhecimento em vários níveis. Por exemplo, texto pode ser anotado com uma representação formal que explicita conhecimento sobre o texto. O mesmo pode ser feito com imagens e possivelmente áudio e vídeo.

- **Vocabulário e conhecimento padronizados**: a Web Semântica encoraja e facilita a formulação de vocabulários e conhecimentos compartilhados na forma de *ontologias*, que podem ser disponibilizadas para modelagem de novos domínios e atividades. Com isto uma grande quantidade de conhecimento pode ser estruturada, formalizada e representada para possibilitar a automação do acesso e uso.

- **Serviços compartilhados**: além das estruturas estáticas, serviços na Web (os já conhecidos *web services*) podem ser usados para composição de aplicações que podem estar localizadas em sistemas diferentes, programados em linguagens diferentes e com acesso a dados especializados, usando a Internet para comunicação entre os módulos dos sistemas.

O formato padrão de dados, a popularidade dos documentos com metadados (anotações) sobre conteúdo e a ambição para formalização em grande escala do conhecimento propostos pela Web Semântica causa duas consequências para a área de mineração de dados da Web. A primeira é que a disponibilidade de informação melhor estruturada permitirá o uso mais amplo de métodos existentes de mineração de dados, já que muitos dos algoritmos poderão ser usados com apenas pequenas modificações. A segunda consequência é a possibilidade de uso do conhecimento formalizado através das ontologias. A combinação destas duas características possibilita o aprendizado onde o conhecimento é adquirido a partir dos dados já anotados e pode ser usado para mais anotações e enriquecimento do conhecimento.

Para realizar o objetivo da Web Semântica é necessário, primeiramente, que novos objetos na Web sejam anotados usando os padrões de formato, conhecimento e vocabulário para metadados. Mais complicada será a tarefa de converter a vasta quantidade de objetos já existentes para uso com as ferramentas da Web Semântica. Algumas abordagens para automação desta tarefa envolvem a anotação e classificação de acordo com ontologias pré-existentes [63] e até mesmo a reorganização de ontologias existentes [22].

2.5. Mineração de Grafos

Muitos dados na Web podem ser representados como grafos, em particular, hiperdocumentos e outros tipos de ligações (participantes de redes sociais, remetentes e destinatários de mensagens eletrônicas, co-autores e co-participantes em trabalhos acadêmicos, etc. – em resumo, praticamente qualquer tipo de relação entre objetos). A representação por grafos requer processamento especializado dos dados, portanto preprocessamento especializado é um passo indispensável para que possamos aplicar algoritmos de mineração de dados e descoberta de conhecimento tradicionais.

Um dos algoritmos de maior sucesso (tanto em uso acadêmico para modificações e extensões como em sucesso em implementações comerciais) é o *PageRank* [19, 55], usado para ordenar resultados de busca no Google. Este algoritmo usa a estrutura dos grafos correspondente à ligações de e para uma página para ter uma métrica de importância da página. Uma descrição detalhada, com exemplo simples, é mostrada em [55].

Uma revisão de técnicas de representação de documentos como grafos, dos algoritmos para processamento de grafos e de técnicas de inteligência computacional e mineração de dados aplicadas à mineração de grafos é feita em [84]. Parte do material desta seção é baseada nesta referência.

Algoritmos de mineração de dados usam métricas baseadas em similaridade e dissimilaridade para tomada de decisões, em particular para decidir se determinado dado é similar ou não a um protótipo ou se a distância entre um dado e protótipo é maior que um limiar (no caso de algumas técnicas de classificação) ou para verificar, em um grupo, qual é o par de dados mais próximos em um espaço qualquer (no caso de técnicas de agrupamento). É de se esperar, então, que algoritmos de mineração de grafos sejam também baseados em medidas de distância para indicar similaridade ou dissimilaridade.

Medidas de distância entre grafos devem seguir algumas propriedades simples: a distância d entre dois grafos G_1 e G_2 deve ser maior ou igual a zero, sendo igual a zero se $G_1 = G_2$. A distância deve ser simétrica: $d(G_1,G_2) = d(G_2,G_1)$ e a desigualdade triangular deve existir: $d(G_1,G_3) \leq d(G_1,G_2) + d(G_2,G_3)$.

As duas medidas mais simples são:

- **Distância de Edição**, que é usada para medir distâncias entre árvores e strings, e que é baseada em operadores que permitem a deleção, inserção e alteração de elementos em sequências. As operações tem um custo associado, e o mínimo custo necessário para transformar um grafo G_1 em G_2 é considerada a distância entre G_1 e G_2. Técnicas como programação dinâmica podem ser usadas para calcular este custo. A vantagem deste método é que é simples e direto; a desvantagem é que os custos devem ser determinados *a priori*.

- **Localização do maior subgrafo**: entre dois grafos, o maior subgrafo comum é localizado, o tamanho deste subgrafo é proporcional à distância entre os dois grafos usados.

Outras medidas são mencionadas em [84], que apresenta também técnicas para calcular a média e mediana de grafos e variantes de algoritmos conhecidos que usam estas métricas. Algumas técnicas específicas para mineração de subgrafos são apresentadas em [42, 68].

2.6. Exemplos de Aplicações de Mineração de Dados na Web

Nesta seção apresentamos alguns trabalhos recentemente publicados em diversas áreas relacionadas com mineração de dados na Web. Alguns dos trabalhos mencionados pertencem a mais de uma das categorias desta seção. Dada a vasta quantidade de publicações na área e o fato de que um artigo sobre mineração de dados relacionados à Web pode ser publicado em vários tipos de veículos, esta lista deve ser considerada como uma visão parcial e incompleta das técnicas e soluções existentes.

2.6.1. Mineração de Conteúdo

Gryc et al. [39] investigam algumas abordagens analíticas para tentar descobrir como inovação acontece com dados de discussão coletados de uma rede social limitada e "tem-

porária" (*Innovation Jam* da IBM). Os dados contém informações textuais (tópicos de discussão), a estrutura destes tópicos e as relações entre os participantes (a maioria sendo funcionários da IBM).

Durant e Smith [31] apresentam técnicas de mineração de dados que, usadas com alguns atributos específicos, conseguem estimar o sentimento político de blogs. A seleção de atributos melhora consideravelmente a qualidade da classificação obtida com algoritmos clássicos. Hayes et al. [41] também analisam blogs através do uso de métricas para caracterização de tópicos.

Abbassi e Mirrokni [1] apresentam algoritmos para criação de um sistema de recomendação para blogs com conteúdo similar. Bose et al. [18], em sua abordagem, diminuem a limitação que a maioria dos sistemas de recomendação de sites tem em não usar informações sobre características conceituais (hierárquicas) dos sites. Pierrakos et al. [72] descrevem um sistema de recomendação de comunidades na Web, que usa agrupamento de documentos para gerar a hierarquia de comunidades.

Outro problema potencial com sistemas de recomendação é a escassez de avaliações (e por conseguinte de correlações entre usuários) que ocorre em sistemas reais: Bergholz [14] apresenta uma solução para este problema que usa outros mecanismos para inferir correlações e agentes para simulação de usuários. Geyer-Schulz e Michael Hahsler [37] apresentam uma análise de algoritmos de recomendação baseados em descoberta de correlações. Geyer-Schulz et al. [38] também investigam a aplicabilidade de um modelo econômico teórico que trata da repetibilidade de compras para um sistema de recomendações.

Baeza-Yates et al. [8] apresentam um estudo interessante sobre reuso de conteúdo na Web, mostrando que o conteúdo de parte da Web usada no estudo é "reciclada" de outras páginas mais antigas, e comentam sobre a influência deste fato nos algoritmos de classificação de sistemas de busca.

Probst et al. [73] propõem um método de classificação semi-supervisionado que é capaz de extrair informações na forma de pares *atributo-valor* de páginas na Web, para criação de bancos de dados sobre produtos comerciais. Outra abordagem, por Ahmed et al. [2], usa algoritmos de mineração de padrões e heurísticas específicas para certos domínios para extrair informações de produtos a partir de catálogos on-line.

Linstead et al. [54] apresenta uma ferramenta que coleta, processa e armazena documentos em repositórios de software na Internet, criando métricas e descritores sobre autores, documentos, palavras e tópicos, que podem ser usadas para quantificação e análise do código e busca por similaridades, disparidades e competências.

Yang e Rahi [97] demonstram como melhorar os resultados de busca por documentos na Web através do agrupamento dos documentos por frases que contém as palavras procuradas. Wang et al. [93] apresentam um novo algoritmo para agrupar e rotular agrupamentos para uso em resultados de busca. Chen e Dong [21] criam rótulos descritivos para agrupamentos de documentos de forma automática.

Mladenić e Grobelnik [63] apresentam uma abordagem para automaticamente mapear páginas na Web para a ontologia de documentos usada no Yahoo!. Os documentos

foram preprocessados para representação como vetores de palavras, com sequências de até cinco palavras. A hierarquia de categorias é usada para criar um conjunto de classificadores independentes, cada um usado para prever a pertinência de um documento à respectiva categoria.

Piatetsky-Shapiro [71] usa os documentos do site KDNuggets.com para uma análise das mudanças de termos frequentes ao longo do tempo, identificando mudanças de comportamento como ofertas de emprego relacionadas com mineração de dados por indústrias e decréscimo do interesse por alguns termos (com explicações baseadas em experiência pessoal).

Outras técnicas e algoritmos de agrupamento de documentos e texto em geral podem ser vistas em [62, 96, 98, 102].

2.6.2. Mineração de Estruturas na Web

O exemplo mais conhecido (por seu enorme sucesso) de algoritmo de mineração de estruturas na Web é o *PageRank* [19, 55], implementado pelos criadores do Google. Este algoritmo foi mencionado na seção 2.5.

Kierfer et al. [48] apresentam métodos para monitoramento de tópicos na Web e para o estudo de co-visibilidade de tópicos, usando contagem de *hits* de sites de busca e redes semânticas, mostrando exemplos reais de aplicação.

Utard e Fürnkranz [92] mostram uma nova maneira de incorporar informações sobre o conteúdo de dois documentos na Web conectados por *hiperlinks*: ao invés de usar todo o texto ou um sumário dos documentos, eles usam parte das páginas próximas das declarações dos *hyperlinks*. Seu trabalho apresenta várias abordagens para identificar proximidade estrutural e textual entre documentos, e avalia estas abordagens.

Bhagat et al. [17] usam informações de relações entre blogs classificá-los através de uma abordagem de rotulação de grafos de forma semi-supervisionada. A técnica é demonstrada classificando blogs como semelhantes a alguns já rotulados usando atributos como idade, sexo e localização.

2.6.3. Mineração de Redes Sociais e Similares

Creamer et al. [26] apresentam uma técnica de mineração de ligações para extrair hierarquias sociais a partir de coleções de mensagens eletrônicas. A abordagem é demonstrada com dados reais (trocas de mensagens entre executivos da empresa Enron). A técnica pode ser usada para inferir hierarquias de outros domínios, como redes sociais, por exemplo.

Creamer e Stolfo [27] apresentam um algoritmo que pode ser aplicado a redes sociais corporativas (composta de diretores e analistas financeiros) para avaliação do impacto de parâmetros em ganhos e estratégias das empresas.

Zaïane et al. [100] considera que bases de dados bibliográficas podem ser usadas para abstrair redes sociais de pesquisadores, criando e analisando grafos de relações autor-conferência e autor-conferência-tópicos. A técnica pode ser usada para identificar áreas de atuação similares e recomendar colaborações entre pesquisadores. Semeraro et

al. [86] apresentam um sistema de descoberta de perfis de usuários que extrai as preferências do usuário a partir de bases de artigos científicos indexados semanticamente. Uma comparação entre técnicas para indução de perfis de usuários a partir de recomendações de produtos dos usuários (e consequentemente de suas preferências) é feita por Esposito et al. [33].

Williams et al. [94] apresentam um estudo sobre mecanismos que podem impedir ou minimizar o efeito de *ataques por injeção de perfis*, que são usados para prejudicar revisões em sistemas abertos de recomendação. Este trabalho extende um anterior ([65]) que apresenta as vulnerabilidades em sistemas colaborativos de recomendação e as técnicas que podem ser usadas para explorar estas vulnerabilidades.

Mobasher et al. [66] apresentam uma visão geral de sistemas de filtragem colaborativa e de suas variantes e características. Anand e Mobasher [4] usam modelos da memória humana (de curto e longo prazo) e informações contextuais como base para modelos de sistemas colaborativos de recomendação, e demonstram a aplicação em um site de compras.

Wang e Kabán [93] apresentam um modelo generativo para inferência de comunidades a partir de uma sequência temporal de eventos de interações entre membros de uma comunidade, em contraste à maioria das técnicas tradicionais de mineração de dados de comunidades, que usam redes ou grafos estáticos.

Shah et al. [87] usam técnicas para identificar padrões frequentes ou comuns de lances em um sistema de leilões eletrônico (eBay), e conseguem confirmar padrões já esperados e identificar novos nos dados coletados. Como parte da análise os autores apresentam possíveis motivações econômicas para alguns destes padrões e identificam possíveis tentativas de fraude.

2.6.4. Mineração de Registros de Acesso (*logs*) a Servidores e Similares

Anand et al. [5] apresentam uma visão geral do processo de mineração de registros de acesso, analisando várias métricas de eficiência propostas na literatura e propondo modelos de interação entre usuários e objetos em um site.

Chi et al. [23] apresentam um sistema automatizado que categoriza usuários de um servidor na Web através de análise de agrupamentos, e que apresenta performance e precisão melhor do que outros sistemas existentes. Outra abordagem para este problema, que usa modelos de Markov, é apresentada por Ypma e Heskes [99].

Baeza-Yates e Poblete [7] apresentam um modelo de mineração de sites que usa dados textuais entrados nos sistemas de busca no próprio site para identificar vários tipos de resultados, que podem sugerir reestruturação do site ou mesmo inclusão de novos tópicos para melhor atender às expectativas de seus usuários.

Técnicas de mineração de *logs* para caracterização de padrões de navegação pressupõem que podemos facilmente extrair as sessões de navegação de usuários para processamento. Berendt et al. [13] avaliam o impacto da estrutura do site e do ambiente do usuário na reconstrução de sessões de navegação, oferecendo heurísticas de reconstrução de sessões que podem ser usadas quando os sites estiverem em servidores distribuídos e quando os usuários não tiverem mecanismos de acompanhamento (ex. *cookies*) para

fornecer dados ao servidor. Outras heurísticas são apresentadas por Cooley et al. [25] e Spiliopoulou et al. [89].

Ainda outra abordagem para usar a estrutura do site para identificar padrões de navegação é apresentada por Tan e Kumar [90]. Esta abordagem tem a vantagem adicional de tentar identificar associações negativas (padrões inexistentes de uso).

Extração e consolidação de *logs* pode ser complexa e trabalhosa; Punin et al. [74] apresentam ferramentas baseadas em XML que facilitam a criação de grafos e relatórios que simplificam bastante a implementação de algoritmos de mineração deste tipo de dados.

Kim e Chan [77] mostram uma técnica para personalizar resultados de um sistema de buscas na Internet usando interesse pessoal dos usuários, representado através de seus marcadores (*bookmarks*) que indicam interesses em páginas e tópicos.

Masseglia et al. [60] apresentam a solução para um problema interessante: tradicionalmente *logs* são segmentados em períodos arbitrários (um determinado mês ou período para o qual existe um interesse explícito), o que faz com que a análise seja automaticamente tendenciosa e que impede a descoberta de picos sazonais em registros. A abordagem proposta pelos autores extrai automaticamente períodos "densos" de acesso e padrões de comportamento frequentes.

Frías-Martínez e Karamcheti [36] usam técnicas para analisar regras sequenciais e temporais para modelar o comportamento de usuários. Esta modelagem pode ser usada para, tentativamente, prever sequências de acessos futuros e possibilitar otimização de conteúdo para usuários.

Lu et al. [57] propõem uma técnica para gerar padrões significativos de uso (SUP, *significant usage patterns*) a partir de sessões e decisões dos usuários, e usa estes SUPs para determinar trilhas preferenciais de navegação dos usuários. Experiências com dados reais de um site de comércio demonstra que os comportamentos dos usuários são identificáveis e interpretáveis.

Outra abordagem para a identificação de padrões de navegação é apresentada por Hay et al. [40], que usam métodos baseados em alinhamento de sequências (inspirados em técnicas bastante usadas em bioinformática). Yang e Parthasarathy [97] também abordam o problema de modelar o comportamento do usuário analisando padrões de navegação com uma técnica que considera as ações do usuário de forma temporal, pressupondo que acessos mais recentes tem mais potencial de modelar interesse e comportamento futuro do que acessos menos recentes. Outra abordagem que usa informação temporal para modelar usuários, baseada em regras, é apresentada por Baron e Spiliopoulou [9].

Outras técnicas de caracterização de padrões temporais de navegação são apresentadas por Desikan e Srivastava [30].

Muitas outras técnicas podem ser usadas para modelar e identificar padrões de navegação. Alguns exemplos são: agrupamento de dados em matrizes (Oyanagi et al. [67]) ou em cubos tridimensionais (Huang et al. [43]), visualização de padrões com gráficos temporais de transição entre documentos (Berendt [11]), matrizes de atributos que permitem a troca entre precisão e escalabilidade (Shahabi e Banaei-Kashani [88]), etc.

2.6.5. Outros

Escudeiro e Jorge [32] apresentam uma metodologia de recuperação automática de conteúdo (coleções de documentos) da Web baseada em tópicos que é adaptativa e dinâmica (podendo mudar de acordo com mudanças de interesse do usuário). O artigo também apresenta uma detalhada análise de sistemas semelhantes desenvolvidos anteriormente, por outros autores.

Markov et al. [59] propõem o uso de informação estrutural e contextual para classificação de documentos, e mostram que o uso deste tipo de informação (ordem e proximidade das palavras, localização da palavra no documento, marcadores de texto como HTML) oferece resultados melhores do que os obtidos com classificadores que usam vetores de atributos dos textos.

2.7. Algumas Ferramentas para Testes e Prototipação de Algoritmos

A aplicação de técnicas de mineração de dados é um processo experimental, onde o interessado deve ter um papel ativo de investigação de métodos de preprocessamento, algoritmos, parâmetros, avaliação de resultados e eventualmente repetição de experimentos, conforme mostrado na seção 2.2.

Como referência básica para o leitor, nesta seção apresentamos dois dos softwares mais comuns para prototipação e exploração de algoritmos de mineração de dados. Os softwares foram selecionados por serem simples, oferecerem interfaces gráficas para facilitar o uso e por serem abertos e executáveis em qualquer sistema operacional.

O software Weka (*Waikato Environment for Knowledge Analysis*) [95] pode ser baixado de `www.cs.waikato.ac.nz/ml/weka/` e oferece várias interfaces para uso dos algoritmos implementados. A Figura 2.8 mostra a interface *Explorer* do software, que permite a carga, edição e manipulação, classificação, agrupamento e visualização dos dados de interesse. As operações são apresentadas em estruturas parecidas com menus.

A Figura 2.9 mostra outro ambiente de exploração do Weka: o *Knowledge Flow*, que permite que operadores de preprocessamento, classificação, etc. sejam representados visualmente como vértices de um grafo dirigido, que pode ser executado como uma aplicação visual.

Outro software bastante flexível, e que contém ainda mais algoritmos de mineração de dados é o RapidMiner (antigamente conhecido como Yale), e que pode ser copiado de `www.rapidminer.com`. A interface do RapidMiner é semelhante à do *Explorer* do software Weka, mas contém muito mais operadores do que este. Operadores no RapidMiner podem ser encadeados e descritos em um documento XML. A Figura 2.10 mostra a execução de um algoritmo de visualização que usa os Mapas Auto-Organizáveis de Kohonen [50]. A Figura 2.11 mostra a representação gráfica de uma árvore de decisão.

Além de permitir a prototipação visual, ambos softwares permitem a integração de seus algoritmos em outras aplicações escritas na linguagem Java. Alguns exemplos para o software Weka podem ser vistos em [79].

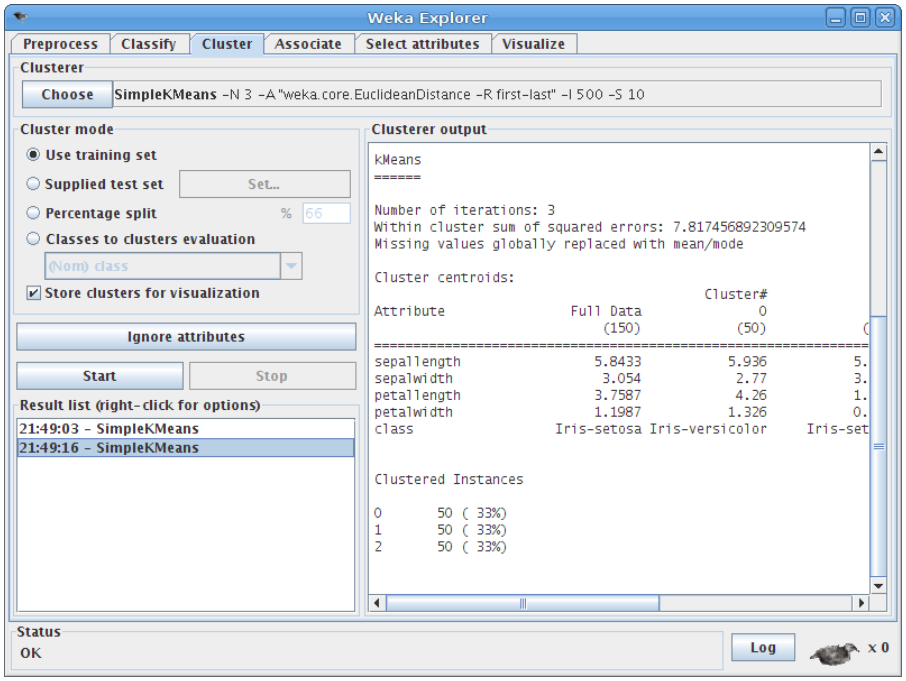

Figura 2.8. Interface *Explorer* do software Weka.

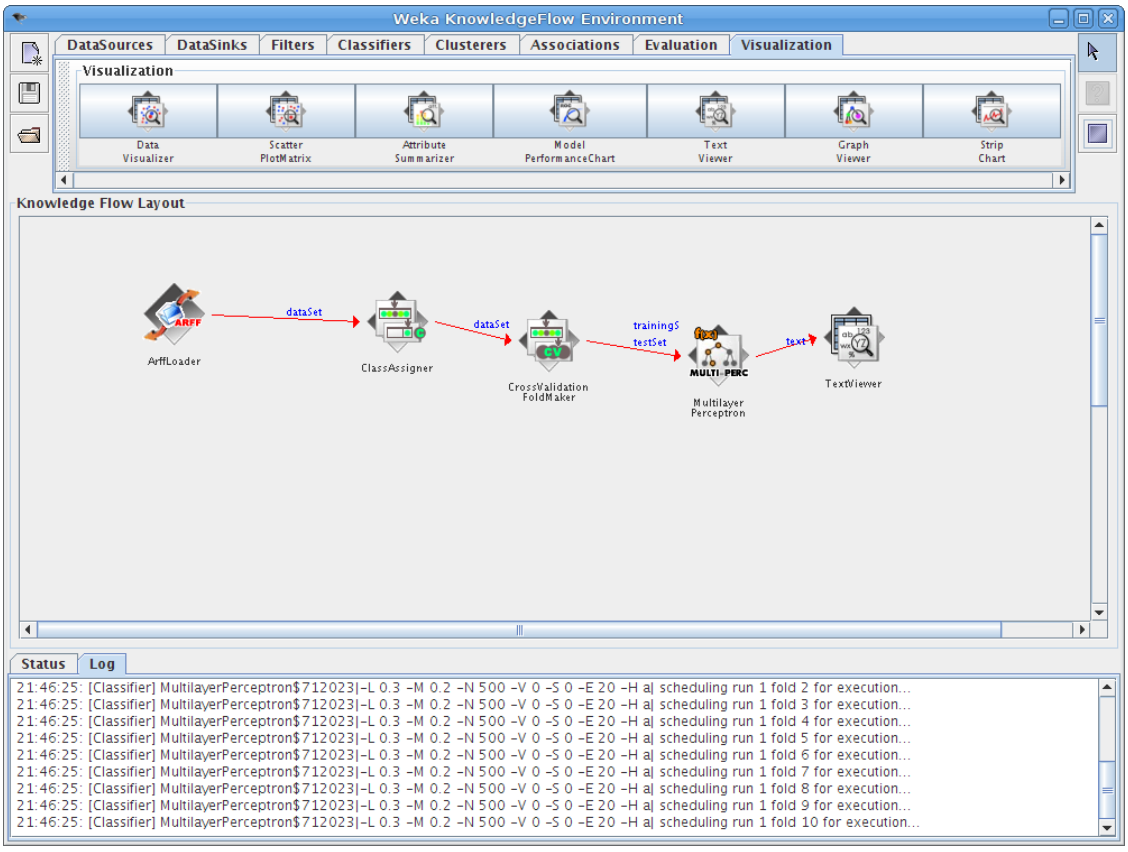

Figura 2.9. Interface *Knowledge Flow* do software Weka.

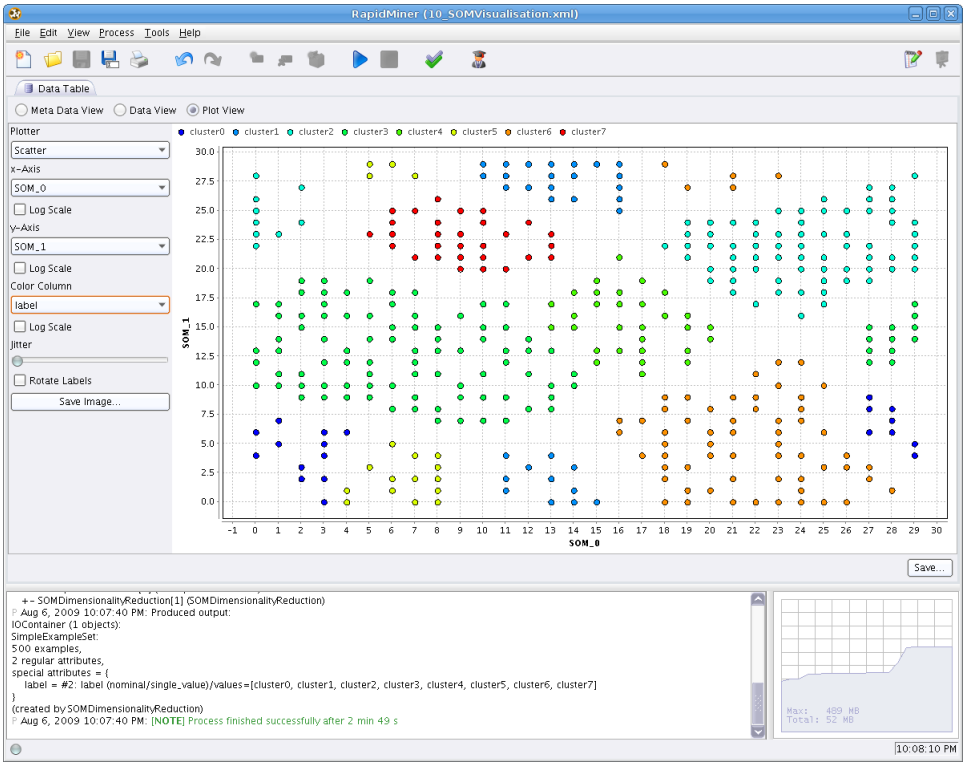

Figura 2.10. Software RapidMiner (Mapas Auto-Organizáveis de Kohonen).

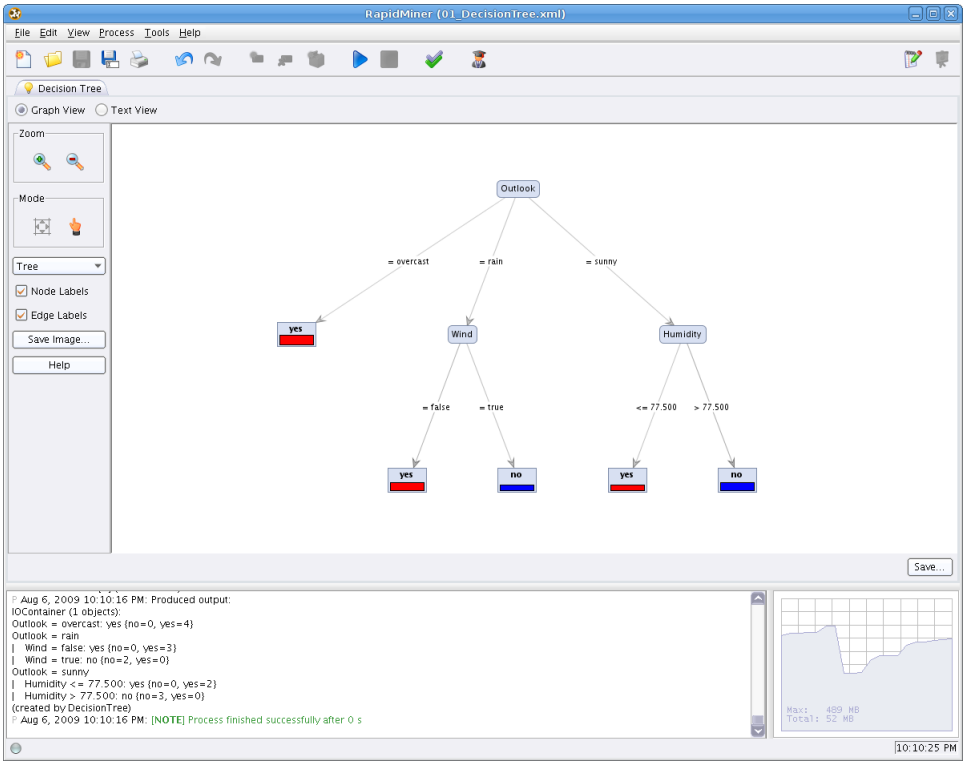

Figura 2.11. Software RapidMiner (Árvore de Decisão).

Referências

[1] Zeinab Abbassi and Vahab S. Mirrokni. A recommender system based on local random walks and spectral methods. In Haizheng Zhang, Myra Spiliopoulou, Bamshad Mobasher, C. Lee Giles, Andrew McCallum, Olfa Nasraoui, Jaideep Srivastava, and John Yen, editors, *Advances in Web Mining and Web Usage Analysis – 9th International Workshop on Knowledge Discovery on the Web, WebKDD 2007 and 1st International Workshop on Social Networks Analysis, SNA-KDD 2007, San Jose, CA, USA, August 12-15, 2007, Revised Papers (LNAI 5439)*, pages 139–153, 2009.

[2] Syed Toufeeq Ahmed, Srinivas Vadrevu, and Hasan Davulcu. *Advances in Web Intelligence and Data Mining (Studies in Computational Intelligence 23)*, chapter DataRover: An Automated System for Extracting Product Information from Online Catalogs, pages 1–10. Springer, 2006.

[3] Alexandre Donizeti Alves. Visualização de relacionamentos entre pesquisadores bolsistas do CNPq. Relatório final da disciplina CAP-359, Programa de Pós-Graduação em Computação Aplicada, Instituto Nacional de Pesquisas Espaciais, 2008.

[4] Sarabjot Singh Anand and Bamshad Mobasher. Contextual recommendation. In Bettina Berendt, Andreas Hotho, Dunja Mladenič, and Giovanni Semeraro, editors, *From Web to Social Web: Discovering and Deploying User and Content Profiles – Workshop on Web Mining, WebMine 2006, Berlin, Germany, September 18, 2006, Revised Selected and Invited Papers (LNAI 4737)*, pages 142–160, 2007.

[5] Sarabjot Singh Anand, Maurice Mulvenna, and Karine Chevalier. On the deployment of web usage mining. In Bettina Berendt, Andreas Hotho, Dunja Mladenic, Maarten van Someren, Myra Spiliopoulou, and Gerd Stumme, editors, *Web Mining: From Web to Semantic Web – First European Web Mining Forum, EWMF 2003, Cavtat-Dubrovnik, Croatia, September 22, 2003, Invited and Selected Revised Papers (LNAI 3209)*, pages 23–42, 2004.

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

[7] Ricardo Baeza-Yates and Barbara Poblete. A website mining model centered on user queries. In Markus Ackermann, Bettina Berendt, Marko Grobelnik, Andreas Hotho, Dunja Mladenič, Giovanni Semeraro, Myra Spiliopoulou, Gerd Stumme, Vojtěch Svátek, and Maarten van Someren, editors, *Semantics, Web and Mining – Joint International Workshops, EWMF 2005 and KDO 2005, Porto, Portugal, October 3 and 7, 2005, Revised Selected Papers (LNAI 4289)*, pages 1–17, 2006.

[8] Ricardo Baeza-Yates, Álvaro Pereira, and Nivio Ziviani. Understanding content reuse on the web: Static and dynamic analyses. In Olfa Nasraoui, Myra Spiliopoulou, Jaideep Srivastava, Bamshad Mobasher, and Brij Masand, editors, *Advances in Web Mining and Web Usage Analysis – 8th International Workshop on Knowledge*

Discovery on the Web, WebKDD 2006, Philadelphia, PA, USA, August 20, 2006, Revised Papers (LNAI 4811), pages 227–246, 2007.

[9] Steffan Baron and Myra Spiliopoulou. *The Adaptive Web – Methods and Strategies of Web Personalization (LNCS 4321)*, chapter Monitoring the Evolution of Web Usage Patterns, pages 181–200. Springer, 2007.

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

[11] Bettina Berendt. Detail and context in web usage mining: Coarsening and visualizing sequences. In Ron Kohavi, Brij M. Masand, Myra Spiliopoulou, and Jaideep Srivastava, editors, *WEBKDD 2001 – Mining Web Log Data Across All Customers Touch Points, Third International Workshop, San Francisco, CA, USA, August 26, 2001, Revised Papers (LNAI 2356)*, pages 1–24, 2002.

[12] Bettina Berendt, Andreas Hotho, Dunja Mladenic, Maarten van Someren, Myra Spiliopoulou, and Gerd Stumme. A roadmap for web mining: From web to semantic web. In Bettina Berendt, Andreas Hotho, Dunja Mladenic, Maarten van Someren, Myra Spiliopoulou, and Gerd Stumme, editors, *Web Mining: From Web to Semantic Web – First European Web Mining Forum, EWMF 2003, Cavtat-Dubrovnik, Croatia, September 22, 2003, Invited and Selected Revised Papers (LNAI 3209)*, pages 1–22, 2004.

[13] Bettina Berendt, Bamshad Mobasher, Miki Nakagawa, and Myra Spiliopoulou. The impact of site structure and user environment on session reconstruction in web usage analysis. In Osmar R. Zaïane, Jaideep Srivastava, Myra Spiliopoulou, and Brij Masand, editors, *WEBKDD 2002 – Mining Web Data for Discovering Usage Patterns and Profiles, 4th International Workshop, Edmonton, Canada, July 23, 2002, Revised Papers (LNAI 2703)*, pages 159–179, 2003.

[14] André Bergholz. Coping with sparsity in a recommender system. In Osmar R. Zaïane, Jaideep Srivastava, Myra Spiliopoulou, and Brij Masand, editors, *WEBKDD 2002 – Mining Web Data for Discovering Usage Patterns and Profiles, 4th International Workshop, Edmonton, Canada, July 23, 2002, Revised Papers (LNAI 2703)*, pages 86–99, 2003.

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

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

[17] Smriti Bhagat, Graham Cormode, and Irina Rozenbaum. Applying link-based classigication to label blogs. In Haizheng Zhang, Myra Spiliopoulou, Bamshad Mobasher, C. Lee Giles, Andrew McCallum, Olfa Nasraoui, Jaideep Srivastava, and John Yen, editors, *Advances in Web Mining and Web Usage Analysis – 9th International Workshop on Knowledge Discovery on the Web, WebKDD 2007 and 1st International Workshop on Social Networks Analysis, SNA-KDD 2007, San Jose, CA, USA, August 12-15, 2007, Revised Papers (LNAI 5439)*, pages 97–117, 2009.

[18] Amit Bose, Kalyan Beemanapalli, Jaideep Srivastava, and Sigal Sahar. Incorporating concept hierarchies into usage mining based recommendations. In Olfa Nasraoui, Myra Spiliopoulou, Jaideep Srivastava, Bamshad Mobasher, and Brij Masand, editors, *Advances in Web Mining and Web Usage Analysis – 8th International Workshop on Knowledge Discovery on the Web, WebKDD 2006, Philadelphia, PA, USA, August 20, 2006, Revised Papers (LNAI 4811)*, pages 110–126, 2007.

[19] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

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

[21] Lijun Chen and Guozhu Dong. Succint and informative cluster descriptions for document repositories. In Jeffrey Xu Yu, Masaru Kitsuregawa, and Hong Va Leong, editors, *Advances in Web-Age Information Management, 7th International Conference, WAIM 2006, Hong Kong, China, June 17-19, 2006, Proceedings (LNCS 4016)*, pages 109–121, 2006.

[22] Qingfeng Chen, Yi-Ping Phoebe Chen, and Chengqi Zhang. Detecting inconsistency in biological molecular databases using ontologies. *Data Mining and Knowledge Discovery*, 15(2):275–296, 2007.

[23] Ed H. Chi, Adam Rosien, and Jeffrey Heer. Lumberjack: Intelligent discovery and analysis of web user traffic composition. In Osmar R. Zaïane, Jaideep Srivastava, Myra Spiliopoulou, and Brij Masand, editors, *WEBKDD 2002 – Mining Web Data for Discovering Usage Patterns and Profiles, 4th International Workshop, Edmonton, Canada, July 23, 2002, Revised Papers (LNAI 2703)*, pages 1–16, 2003.

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

[25] Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1:5–32, 1999.

[26] Germán Creamer, Ryan Rowe, Shlomo Hershkop, and Salvatore J. Stolfo. Segmentation and automated social hierarchy detection through email network analysis. In Haizheng Zhang, Myra Spiliopoulou, Bamshad Mobasher, C. Lee Giles, Andrew McCallum, Olfa Nasraoui, Jaideep Srivastava, and John Yen, editors, *Advances in Web Mining and Web Usage Analysis – 9th International Workshop on Knowledge Discovery on the Web, WebKDD 2007 and 1st International Workshop on Social Networks Analysis, SNA-KDD 2007, San Jose, CA, USA, August 12-15, 2007, Revised Papers (LNAI 5439)*, pages 40–58, 2009.

[27] Germán Creamer and Sal Stolfo. A link mining algorithm for earnings forecast and trading. *Data Mining and Knowledge Discovery*, 18(3):419–445, 2009.

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

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

[30] 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.

[31] Kathleen T. Durant and Michael D. Smith. Predicting the political sentiment of web log posts using supervised machine learning techniques coupled with feature selection. In Olfa Nasraoui, Myra Spiliopoulou, Jaideep Srivastava, Bamshad Mobasher, and Brij Masand, editors, *Advances in Web Mining and Web Usage Analysis – 8th International Workshop on Knowledge Discovery on the Web, WebKDD 2006, Philadelphia, PA, USA, August 20, 2006, Revised Papers (LNAI 4811)*, pages 187–206, 2007.

[32] Nuno F. Escudeiro and Alípio M. Jorge. Semi-automatic creation and maintenance of web resources with webTopic. In Markus Ackermann, Bettina Berendt, Marko Grobelnik, Andreas Hotho, Dunja Mladenič, Giovanni Semeraro, Myra Spiliopoulou, Gerd Stumme, Vojtěch Svátek, and Maarten van Someren, editors, *Semantics, Web and Mining – Joint International Workshops, EWMF 2005 and KDO 2005, Porto, Portugal, October 3 and 7, 2005, Revised Selected Papers (LNAI 4289)*, pages 82–102, 2006.

[33] F. Esposito, G. Semeraro, S. Ferilli, M. Degemmis, N. Di Mauro, T.M.A. Basile, and P. Lops. Evaluation and validation of two approaches to user profiling. In Bettina Berendt, Andreas Hotho, Dunja Mladenic, Maarten van Someren, Myra Spiliopoulou, and Gerd Stumme, editors, *Web Mining: From Web to Semantic Web – First European Web Mining Forum, EWMF 2003, Cavtat-Dubrovnik, Croatia, September 22, 2003, Invited and Selected Revised Papers (LNAI 3209)*, pages 130–147, 2004.

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

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

[36] Enrique Frías-Martínez and Vijay Karamcheti. A customizable behavior model for temporal prediction of web user sequences. In Osmar R. Zaïane, Jaideep Srivastava, Myra Spiliopoulou, and Brij Masand, editors, *WEBKDD 2002 – Mining Web Data for Discovering Usage Patterns and Profiles, 4th International Workshop, Edmonton, Canada, July 23, 2002, Revised Papers (LNAI 2703)*, pages 66–85, 2003.

[37] Andreas Geyer-Schulz and Michael Hahsler. Comparing two recommender algorithms with the help of recommendations by peers. In Osmar R. Zaïane, Jaideep Srivastava, Myra Spiliopoulou, and Brij Masand, editors, *WEBKDD 2002 – Mining Web Data for Discovering Usage Patterns and Profiles, 4th International Workshop, Edmonton, Canada, July 23, 2002, Revised Papers (LNAI 2703)*, pages 137–158, 2003.

[38] Andreas Geyer-Schulz, Michael Hahsler, and Maximillian Jahn. A customer purchase incidence model applied to recommender services. In Ron Kohavi, Brij M. Masand, Myra Spiliopoulou, and Jaideep Srivastava, editors, *WEBKDD 2001 – Mining Web Log Data Across All Customers Touch Points, Third International Workshop, San Francisco, CA, USA, August 26, 2001, Revised Papers (LNAI 2356)*, pages 25–47, 2002.

[39] Wojciech Gryc, Mary Helander, Rick Lawrence, Yan Liu, Claudia Perlich, Chandan Reddy, and Saharon Rosset. Looking for great ideas: Analyzing the innovation jam. In Haizheng Zhang, Myra Spiliopoulou, Bamshad Mobasher, C. Lee Giles, Andrew McCallum, Olfa Nasraoui, Jaideep Srivastava, and John Yen, editors, *Advances in Web Mining and Web Usage Analysis – 9th International Workshop on Knowledge Discovery on the Web, WebKDD 2007 and 1st International Workshop on Social Networks Analysis, SNA-KDD 2007, San Jose, CA, USA, August 12-15, 2007, Revised Papers (LNAI 5439)*, pages 21–39, 2009.

[40] Birgit Hay, Geert Wets, and Koen Vanhoof. Web usage mining by means of multidimensional sequence alignment methods. In Osmar R. Zaïane, Jaideep Srivastava, Myra Spiliopoulou, and Brij Masand, editors, *WEBKDD 2002 – Mining Web Data for Discovering Usage Patterns and Profiles, 4th International Workshop, Edmonton, Canada, July 23, 2002, Revised Papers (LNAI 2703)*, pages 50–65, 2003.

[41] Conor Hayes, Paolo Avesani, and Uldis Bojars. Predicting the political sentiment of web log posts using supervised machine learning techniques coupled with feature selection. In Bettina Berendt, Andreas Hotho, Dunja Mladenič, and Giovanni Semeraro, editors, *From Web to Social Web: Discovering and Deploying User and Content Profiles – Workshop on Web Mining, WebMine 2006, Berlin, Germany, September 18, 2006, Revised Selected and Invited Papers (LNAI 4737)*, pages 1–20, 2007.

[42] Petteri Hintsanen and Hannu Toivonen. Finding reliable subgraphs from large probabilistic graphs. *Data Mining and Knowledge Discovery*, 17(1):3–23, 2008.

[43] Joshua Zhexue Huang, Michael Ng, Wai-Ki Ching, Joe Ng, and David Cheung. A cube model and cluster analysis for web access sessions. In Ron Kohavi, Brij M. Masand, Myra Spiliopoulou, and Jaideep Srivastava, editors, *WEBKDD 2001 – Mining Web Log Data Across All Customers Touch Points, Third International Workshop, San Francisco, CA, USA, August 26, 2001, Revised Papers (LNAI 2356)*, pages 48–67, 2002.

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

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

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

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

[48] Peter Kiefer, Klaus Stein, and Christoph Schlieder. Visibility analysis on the web using co-visibilities and semantic networks. In Markus Ackermann, Bettina Berendt, Marko Grobelnik, Andreas Hotho, Dunja Mladenič, Giovanni Semeraro, Myra Spiliopoulou, Gerd Stumme, Vojtěch Svátek, and Maarten van Someren, editors, *Semantics, Web and Mining – Joint International Workshops, EWMF 2005 and KDO 2005, Porto, Portugal, October 3 and 7, 2005, Revised Selected Papers (LNAI 4289)*, pages 34–50, 2006.

[49] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46:668–677, 1999.

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

[51] Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. In *Proceedings of the 14th International Conference on Machine Learning ICML97*, pages 170–178. Morgan Kaufmann, 1997.

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

[53] Jun Z. Li, Devin M. Absher, Hua Tang, Audrey M. Southwick, Amanda M. Casto, Sohini Ramachandran, Howard M. Cann, Gregory S. Barsh, Marcus Feldman, Luigi L. Cavalli-Sforza, and Richard M. Myers. Worldwide human relationships inferred from genome-wide patterns of variation. *Science*, 319(5866):1100–1104, 2008.

[54] Erik Linstead, Sushil Bajracharya, Trung Ngo, Paul Rigor, Cristina Lopes, and Pierre Baldi. Sourcerer: mining and searching internet-scale software repositories. *Data Mining and Knowledge Discovery*, 18(2):300–336, 2009.

[55] Bing Liu and Philip S. Yu. *The Top Ten Algorithms in Data Mining*, chapter PageRank, pages 73–100. Taylor & Francis, 2009.

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

[57] Lin Lu, Margaret Dunham, and Yu Meng. Mining significant usage patterns from clickstream data. In Olfa Nasraoui, Osmar Zaïane, Myra Spiliopoulou, Bamshad Mobasher, Brij Masand, and Philip S. Yu, editors, *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 (LNAI 4198)*, pages 1–17, 2006.

[58] Mehrdad Mahdavi and Hassan Abolhassani. Harmony K-means algorithm for document clustering. *Data Mining and Knowledge Discovery*, 18(3):370–391, 2009.

[59] Alex Markov, Mark Last, and Abraham Kandel. Fast categorization of web documents represented by graphs. In Olfa Nasraoui, Myra Spiliopoulou, Jaideep Srivastava, Bamshad Mobasher, and Brij Masand, editors, *Advances in Web Mining and Web Usage Analysis – 8th International Workshop on Knowledge Discovery on the Web, WebKDD 2006, Philadelphia, PA, USA, August 20, 2006, Revised Papers (LNAI 4811)*, pages 57–61, 2007.

[60] Florent Masseglia, Pascal Poncelet, M. Teisseire, and Alice Marascu. Web usage mining: extracting unexpected periods from web logs. *Data Mining and Knowledge Discovery*, 16(1):39–65, 2008.

[61] Andrew McCallum, Ronald Rosenfeld, Tom M. Mitchell, and Andrew Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 359–367, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[62] Dieter Merkl. Exploration of document collections with self-organizing maps: A novel approach to similarity representation. In Jan Komorowski and Jan Zytkow, editors, *Principles of Data Mining and Knowledge Discovery – First European Symposium, PKDD '97, Trondheim, Norway, June 24-27, 1997, Proceedings (LNAI 1263)*, pages 101–111, 1997.

[63] Dunja Mladenić and Marko Grobelnik. Mapping documents onto web page ontology. In Bettina Berendt, Andreas Hotho, Dunja Mladenic, Maarten van Someren, Myra Spiliopoulou, and Gerd Stumme, editors, *Web Mining: From Web to Semantic Web – First European Web Mining Forum, EWMF 2003, Cavtat-Dubrovnik, Croatia, September 22, 2003, Invited and Selected Revised Papers (LNAI 3209)*, pages 77–96, 2004.

[64] Bamshad Mobasher. *The Adaptive Web – Methods and Strategies of Web Personalization (LNCS 4321)*, chapter Data Mining for Web Personalization, pages 90–135. Springer, 2007.

[65] Bamshad Mobasher, Robin Burke, Chad Williams, and Runa Bhaumik. Analysis and detection of segment-focused attacks against collaborative recommendation. In Olfa Nasraoui, Osmar Zaïane, Myra Spiliopoulou, Bamshad Mobasher, Brij Masand, and Philip S. Yu, editors, *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 (LNAI 4198)*, pages 96–118, 2006.

[66] Bamshad Mobasher, Xin Jin, and Yanzan Zhou. Semantically enhanced collaborative filtering on the web. In Bettina Berendt, Andreas Hotho, Dunja Mladenic, Maarten van Someren, Myra Spiliopoulou, and Gerd Stumme, editors, *Web Mining: From Web to Semantic Web – First European Web Mining Forum, EWMF*

2003, Cavtat-Dubrovnik, Croatia, September 22, 2003, Invited and Selected Revised Papers (LNAI 3209), pages 57–76, 2004.

[67] Shigeru Oyanagi, Kazuto Kubota, and Akihiko Nakase. Mining WWW access sequence by matrix clustering. In Osmar R. Zaïane, Jaideep Srivastava, Myra Spiliopoulou, and Brij Masand, editors, *WEBKDD 2002 – Mining Web Data for Discovering Usage Patterns and Profiles, 4th International Workshop, Edmonton, Canada, July 23, 2002, Revised Papers (LNAI 2703)*, pages 119–136, 2003.

[68] Apostolos N. Papadopoulos, Apostolos Lyritsis, and Yannis Manolopoulos. Skygraph: an algorithm for important subgraph discovery in relational graphs. *Data Mining and Knowledge Discovery*, 17(1):57–76, 2008.

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

[70] Petra Perner. *Data Mining on Multimedia Data*, volume 2558. 2002.

[71] Gregory Piatetsky-Shapiro. Data mining and knowledge discovery 1996 to 2005: overcoming the hype and moving from "university" to "business" and "analytics". *Data Mining and Knowledge Discovery*, 15(1):99–105, 2007.

[72] Dimitrios Pierrakos, Georgios Paliouras, Christos Papatheodorou, Vangelis Karkaletsis, and Marios Dikaiakos. Web community directories: A new approach to web personalization. In Bettina Berendt, Andreas Hotho, Dunja Mladenic, Maarten van Someren, Myra Spiliopoulou, and Gerd Stumme, editors, *Web Mining: From Web to Semantic Web – First European Web Mining Forum, EWMF 2003, Cavtat-Dubrovnik, Croatia, September 22, 2003, Invited and Selected Revised Papers (LNAI 3209)*, pages 113–129, 2004.

[73] Katharina Probst, Rayid Ghani, Marko Krema, Andy Fano, and Yan Liu. Extracting and using attribute-value pairs from product descriptions on the web. In Bettina Berendt, Andreas Hotho, Dunja Mladenič, and Giovanni Semeraro, editors, *From Web to Social Web: Discovering and Deploying User and Content Profiles – Workshop on Web Mining, WebMine 2006, Berlin, Germany, September 18, 2006, Revised Selected and Invited Papers (LNAI 4737)*, pages 41–60, 2007.

[74] John R. Punin, Mukkai S. Krishnamoorthy, and Mohammed J. Zaki. LOGML: Log markup language for web usage mining. In Ron Kohavi, Brij M. Masand, Myra Spiliopoulou, and Jaideep Srivastava, editors, *WEBKDD 2001 – Mining Web Log Data Across All Customers Touch Points, Third International Workshop, San Francisco, CA, USA, August 26, 2001, Revised Papers (LNAI 2356)*, pages 88–112, 2002.

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

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

[77] Hyoung rae Kim and Philip K. Chan. Personalized search results with user interest hierarchies learnt from bookmarks. In Olfa Nasraoui, Osmar Zaïane, Myra Spiliopoulou, Bamshad Mobasher, Brij Masand, and Philip S. Yu, editors, *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 (LNAI 4198)*, pages 158–176, 2006.

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

[79] 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.

[80] 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.

[81] Rafael Santos. *Webmedia 2008 - XIV Simpósio Brasileiro de Sistemas Multimídia e Web*, chapter Conceitos de Mineração de Dados Multimídia, pages 98–144. Sociedade Brasileira de Computação, 2008.

[82] 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.

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

[84] Adam Schenker, Horst Bunke, Mark Last, and Abraham Kandel. *Graph-Theoretic Techniques for Web Content Mining*. World Scientific, 2005.

[85] Nicu Sebe, Yuncai Liu, Yueting Zhuang, and Thomas S. Huang, editors. *Multimedia Content Analysis and Mining – International Workshop, MCAM 2007*, volume 4577, 2007.

[86] Giovanni Semeraro, Pierpaolo Basile, Marco de Gemmis, and Pasquale Lops. Discovering user profiles from semantically indexed scientific papers. In Bettina Berendt, Andreas Hotho, Dunja Mladenič, and Giovanni Semeraro, editors, *From Web to Social Web: Discovering and Deploying User and Content Profiles – Workshop on Web Mining, WebMine 2006, Berlin, Germany, September 18, 2006, Revised Selected and Invited Papers (LNAI 4737)*, pages 61–81, 2007.

[87] Harshit S. Shah, Neeraj R. Joshi, Ashish Sureka, and Peter R. Wurman. Mining ebay: Bidding strategies and shill detection. In Osmar R. Zaïane, Jaideep Srivastava, Myra Spiliopoulou, and Brij Masand, editors, *WEBKDD 2002 – Mining Web Data for Discovering Usage Patterns and Profiles, 4th International Workshop, Edmonton, Canada, July 23, 2002, Revised Papers (LNAI 2703)*, pages 17–34, 2003.

[88] Cyrus Shahabi and Farnoush Banaei-Kashani. A framework for efficient and anonymous web usage mining based on client-side tracking. In Ron Kohavi, Brij M. Masand, Myra Spiliopoulou, and Jaideep Srivastava, editors, *WEBKDD 2001 – Mining Web Log Data Across All Customers Touch Points, Third International Workshop, San Francisco, CA, USA, August 26, 2001, Revised Papers (LNAI 2356)*, pages 113–144, 2002.

[89] Myra Spiliopoulou, Bamshad Mobasher, Bettina Berendt, and Miki Nakagawa. A framework for the evaluation of session reconstruction heuristics in web-usage analysis. *INFORMS Journal on Computing (Special Issue on* Mining Web-based Data for E-Business Applications*)*, 15(2):171–190, 2003.

[90] Pang-Ning Tan and Vipin Kumar. Mining indirect associations in web data. In Ron Kohavi, Brij M. Masand, Myra Spiliopoulou, and Jaideep Srivastava, editors, *WEBKDD 2001 – Mining Web Log Data Across All Customers Touch Points, Third International Workshop, San Francisco, CA, USA, August 26, 2001, Revised Papers (LNAI 2356)*, pages 145–166, 2002.

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

[92] Hervé Utard and Johannes Fürnkranz. Link-local features for hypertext classification. In Markus Ackermann, Bettina Berendt, Marko Grobelnik, Andreas Hotho, Dunja Mladenič, Giovanni Semeraro, Myra Spiliopoulou, Gerd Stumme, Vojtěch Svátek, and Maarten van Someren, editors, *Semantics, Web and Mining – Joint International Workshops, EWMF 2005 and KDO 2005, Porto, Portugal, October 3 and 7, 2005, Revised Selected Papers (LNAI 4289)*, pages 51–64, 2006.

[93] Junze Wang, Yijun Mo, Benxiong Huang, Jie Wen, and Li He. Web search results clustering based on a novel suffix tree structure. In Chunming Rong, Martin Gilje Jaatun, Frode Eika Sandnes, Laurence T. Yang, and Jianhua Ma, editors, *Autonomic and Trusted Computing – 5th International Conference, ATC 2008, Oslo, Norway, June 23-25, 2008, Proceedings (LNCS 5060)*, pages 540–554, 2008.

[94] Chad A. Williams, Bamshad Mobasher, Robin Burke, and Runa Bhaumik. Detecting profile injection attacks in collaborative filtering: A classification-based approach. In Olfa Nasraoui, Myra Spiliopoulou, Jaideep Srivastava, Bamshad Mobasher, and Brij Masand, editors, *Advances in Web Mining and Web Usage Analysis – 8th International Workshop on Knowledge Discovery on the Web, WebKDD 2006, Philadelphia, PA, USA, August 20, 2006, Revised Papers (LNAI 4811)*, pages 167–186, 2007.

[95] Ian H. Witten and Eibe Frank. *Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations, 2nd edition*. Morgan Kaufmann Publishers, 2005.

[96] Wilson Wong, Wei Liu, and Mohammed Bennamoun. Tree-traversing ant algorithm for term clustering based on featureless similarities. *Data Mining and Knowledge Discovery*, 15(3):349–381, 2007.

[97] Li Yang and Adnan Rahi. Dynamic clustering of web search results. In Vipin Kumar, Marina L. Gavrilova, Chih Jeng Kenneth Tan, and Pierre L'Ecuyer, editors, *Computational Science and Its Applications – ICCSA 2003, International Conference, Montreal, Canada, May 18-21, 2003, Proceedings, Part I (LNCS 2667)*, pages 153–159, 2003.

[98] Ju-In Youn, He-Jue Eun, and Yong-Sung Kim. Fuzzy clustering for documents based on optimization of classifier using the genetic algorithm. In Osvaldo Gervasi, Marina L. Gavrilova, Vipin Kumar, Antonio Laganà, Heow Pueh Lee, Youngsong Mun, David Taniar, and Chih Jeng Kenneth Tan, editors, *Computational Science and Its Applications – ICCSA 2005, International Conference, Singapore, May 9-12, 2005, Proceedings, Part II (LNCS 3481)*, pages 10–20, 2005.

[99] Alexander Ypma and Tom Heskes. Automatic categorization of web pages and user clustering with mixtures of hidden markov models. In Osmar R. Zaïane, Jaideep Srivastava, Myra Spiliopoulou, and Brij Masand, editors, *WEBKDD 2002 – Mining Web Data for Discovering Usage Patterns and Profiles, 4th International Workshop, Edmonton, Canada, July 23, 2002, Revised Papers (LNAI 2703)*, pages 35–49, 2003.

[100] Osmar R. Zaïane, Jiyang Chen, and Randy Goebel. Mining research communities in bibliographical data. In Haizheng Zhang, Myra Spiliopoulou, Bamshad Mobasher, C. Lee Giles, Andrew McCallum, Olfa Nasraoui, Jaideep Srivastava, and John Yen, editors, *Advances in Web Mining and Web Usage Analysis – 9th International Workshop on Knowledge Discovery on the Web, WebKDD 2007 and 1st International Workshop on Social Networks Analysis, SNA-KDD 2007, San Jose, CA, USA, August 12-15, 2007, Revised Papers (LNAI 5439)*, pages 59–76, 2009.

[101] 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, 2003.

[102] Qiaoming Zhu, Junhui Li, Guodong Zhou, Peifeng Li, and Peide Qian. A novel hierarchical document clustering algorithm based on a kNN connection graph. In Yuji Matsumoto, Richard Sproat, Kam-Fai Wong, and Min Zhang, editors, *Computer Processing of Oriental Languages – Beyond the Orient: The Research Challenges Ahead, 21st International Conference, ICCPOL 2006, Singapore, December 17-19, 2006, Proceedings (LNAI 4285)*, pages 10–20, 2006.