UNIVERSIDADE FEDERAL DE PELOTAS
Bacharelado em Ciência da Computação
Instituto de Física e Matemática

Trabalho Acadêmico

Mineração de Dados para Descoberta de Padrões: Estudo Aplicado à Base de Dados da Delegacia Regional do Trabalho.

Douglas Gomes de Sousa

Pelotas, 2008

DOUGLAS GOMES DE SOUSA

MINERAÇÃO DE DADOS PARA DESCOBERTA DE PADRÕES: ESTUDO APLICADO À BASE DE DADOS DA DELEGACIA REGIONAL DO TRABALHO

Trabalho acadêmico apresentado ao Curso de Ciência da Computação da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Bacharelado de Ciência da Computação.

Orientadora: Prof[a]. MSc. Ana Marilza Pernas Fleischmann
Co-Orientador (es): Prof. MSc. Juliano Lucas Gonçalves
Prof. MSc. Anderson Priebe Ferrugem

Pelotas, 2008

Dados de catalogação na fonte:
Ubirajara Buddin Cruz – CRB-10/901
Biblioteca de Ciência & Tecnologia - UFPel

S729m Sousa, Douglas Gomes de

Mineração de dados para descoberta de padrões: estudo aplicado à base de dados da Delegacia Regional do Trabalho / Douglas Gomes de Sousa ; orientador Ana Marilza Pernas Fleischmann ; co-orientador Juliano Lucas Gonçalves e Anderson Priebe Ferrugem. – Pelotas, 2008. – 81f. : il. - Monografia (Conclusão de curso). Curso de Bacharelado em Ciência da Computação. Departamento de Informática. Instituto de Física e Matemática. Universidade Federal de Pelotas. Pelotas, 2008.

1.Informática. 2.Mineração de dados. 3.Agrupamento. 4.K-médias. 5.Mapas auto-organizáveis. I.Fleischman, Ana Marilza Pernas. II.Gonçalves, Juliano Lucas. III.Ferrugem, Anderson Priebe. IV.Título.

CDD: 005.75

Banca examinadora:

.....
Prof[a]. MSc. Ana Marilza Pernas Fleischmann - (DINFO/UFPel)

.....
Prof. MSc. Gil Carlos Rodrigues Medeiros - (DINFO/UFPel)

.....
Prof. MSc. Ricardo Matsumura de Araújo - (DINFO/UFPel)

AGRADECIMENTOS

À minha orientadora Ana Marilza e aos meus co-orientadores Juliano e Anderson pela ajuda dada ao meu trabalho.

Aos meus pais, pela criação, educação, oportunidades e por tudo que eu tive na vida graças a vocês.

Ao meu irmão, pela companhia nas horas de lazer, pelo apoio e pela ajuda oferecidos por todos os anos da minha vida.

Aos meus tios e primos, pelo apoio que me deram em todos os momentos.

A todos vocês, MUITO OBRIGADO!

RESUMO

SOUSA, Douglas Gomes de. **Mineração de Dados para Descoberta de Padrões: Estudo Aplicado à Base de Dados da Delegacia Regional do Trabalho**. 2008. 81f. Monografia (Bacharelado em Ciência da Computação). Universidade Federal de Pelotas, Pelotas.

Mineração de dados é um processo no qual são utilizadas técnicas específicas para encontrar padrões em um banco de dados. Tais padrões, por sua vez, podem revelar algum conhecimento útil escondido nestes dados. Este trabalho realiza o processo de mineração na base de dados da Delegacia Regional do Trabalho, que contém dados de trabalhadores gaúchos utilizados na feitura das carteiras de trabalho na década de 1930. No processo de mineração realizado por este trabalho foram utilizadas técnicas de agrupamento, que consistem na descoberta de grupos de dados que possuam características semelhantes. Os algoritmos de agrupamento utilizados foram o K-médias e os Mapas Auto-Organizáveis (SOMs). Tais algoritmos foram implementados em uma ferramenta que foi desenvolvida neste trabalho e utilizada em algumas das etapas da mineração. O processo de mineração foi realizado com cada um dos algoritmos, de modo a comparar os resultados obtidos por estes. De modo geral, o SOM apresentou um melhor resultado em relação ao K-médias, no sentido dos padrões extraídos serem mais compreensíveis e com maior capacidade de revelar alguma informação oculta dos dados.

Palavras-chave: Mineração de dados. Agrupamento. K-médias. Mapas Auto-Organizáveis.

ABSTRACT

SOUSA, Douglas Gomes de. **Mineração de Dados para Descoberta de Padrões: Estudo Aplicado à Base de Dados da Delegacia Regional do Trabalho**. 2008. 81f. Monografia (Bacharelado em Ciência da Computação). Universidade Federal de Pelotas, Pelotas.

Data Mining is a process where especific techniques are used to find patterns in a database. Such patterns, in turn, can reveal some useful knowlege hidden in these data. This work performs data mining process in the database of Delegacia Regional do Trabalho, which have data of workers from Rio Grande do Sul used for making worker identifications in the 1930 decade. In the data mining process performed by this work were used clustering techniques, which consists in discovering groups of data that have similar caracteristcs. The clustering algorithms used were the K-means and the Self-Organizing Maps (SOMs). Such algorithms were implemented in a tool developed in this work and used in some of the steps of data mining. The data mining process was performed with each one of the algorithms, in order to compare the results obtained by these. In general, SOM presented a better result than K-means, in the meaning of extracted patterns being more compreensible and with higher capability to reveal some hidden information from the data.

Keywords: Data Mining. Clustering. K-means. Self-Organizing Maps.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

BMU	*Best Matching Unit*
CRISP-DM	*Cross-Industry Standard Process for Data Mining*
CURE	*Clustering Using Representatives*
DRT	Delegacia Regional do Trabalho
DCBD	Descoberta de Conhecimento em Banco de Dados
EM	*Expectation Maximization*
KDD	*Knowledge Discovery in Databases*
NDH	Núcleo de Documentação Histórica
SOM	*Self-Organizing Map*
TIA	*Terrorism Information Awareness*

SUMÁRIO

1 INTRODUÇÃO

A tecnologia da computação e os sistemas de informação encontram-se cada vez mais presentes nas organizações, sejam estas organizações empresas, universidades, hospitais ou órgãos governamentais. Cada vez mais organizações utilizam sistemas computacionais que armazenam, gerenciam e interpretam dados, de modo a fornecer informação relevante para elas.

Com o advento da tecnologia da computação, ocorreu também um aumento substancial no volume de dados mantidos pelas organizações, tornando cada vez mais difícil extrair informação útil e compreensível desses dados por meio dos métodos operacionais tradicionais.

Como forma de contornar esta dificuldade surgiu então o conceito de mineração de dados. A mineração de dados é parte de um processo conhecido como DCBD (Descoberta de Conhecimento em Banco de Dados) que, como o próprio nome diz, tem como finalidade auxiliar a descoberta de conhecimento implícito em grandes volumes de dados.

Este trabalho realiza um estudo de caso sobre os dados mantidos pela Delegacia Regional do Trabalho (DRT), no qual são utilizadas técnicas de mineração de dados. O banco de dados da DRT foi construído a partir do acervo existente na mesma, que contém fichas com informações de trabalhadores gaúchos utilizadas na feitura das carteiras de trabalho entre as décadas de 1930 e 1960. Parte destas fichas foi digitada e armazenada no banco de dados.

1.1 Motivação

Segundo Han e Kamber (2006), a mineração de dados tem atraído grande atenção na indústria da informação e na sociedade como um todo nos últimos anos, graças à sua capacidade de extrair informação que antes era difícil ou mesmo impossível de se obter de uma vasta quantidade de dados. A informação e conhecimento obtidos por ela podem ser utilizados em uma variedade de aplicações, tais como análise de mercado, controle de produção e exploração científica.

Dado este potencial existente na mineração de dados, a mesma poderia ser utilizada em benefício dos pesquisadores do Núcleo de Documentação Histórica (NDH) da UFPel. Atualmente o NDH tem à disposição um sistema com o qual é possível armazenar, alterar e consultar os dados da DRT. Os pesquisadores do NDH utilizam este sistema para pesquisar a história dos trabalhadores gaúchos. As consultas são feitas através de técnicas simples da álgebra relacional, tais como seleções e junções, implementadas no sistema.

Entretanto, tais técnicas não são o suficiente para extrair toda a informação potencial existente nos dados. O volume de dados da DRT é grande e aumentará ainda mais à medida que mais dados forem inseridos, gerando uma grande quantidade de informação implícita (não visível com pesquisas simples). A DRT não dispunha de um sistema que realizasse o processo de mineração sobre a sua base de dados, capaz de extrair esta informação implícita, que poderia ser muito útil para entender a história dos trabalhadores gaúchos.

Existem diferentes classes de algoritmos de mineração, sendo que as principais, segundo Han e Kamber (2006), são o agrupamento, a classificação e a associação. Um algoritmo de agrupamento (classe especialmente estudada neste trabalho) parte de um conjunto de dados heterogêneo e constrói grupos de dados que possuam características semelhantes. As técnicas de agrupamento seriam as mais indicadas para os dados da DRT, pois através delas é possível agrupar registros de trabalhadores que possuam determinadas características em comum e assim descobrir padrões e encontrar possíveis relações entre estas características.

Desta forma, este trabalho procura melhorar o acesso a informações contidas nos dados da DRT, estudando uma forma de realizar o cruzamento destes através de técnicas de mineração de dados, mais apropriadas para um volume grande de dados. Ao final, foi desenvolvida uma ferramenta que aplicasse tais técnicas, sendo utilizada na tentativa de resgatar informações não visíveis apenas com pesquisas simples.

1.2 Objetivos

O foco deste trabalho encontra-se nos seguintes objetivos principais: o estudo das técnicas de mineração da classe de agrupamento, o desenvolvimento de uma ferramenta que implemente algumas das técnicas estudadas e a realização do

processo de mineração de dados, utilizando a ferramenta desenvolvida para encontrar relações nos dados da DRT e descobrir padrões.

Os objetivos específicos do trabalho são:

- Fazer um estudo geral sobre o conceito de mineração de dados e suas aplicações.
- Fazer um estudo sobre técnicas de mineração de dados específicas para a tarefa do agrupamento.
- Avaliar os dados da DRT, verificando se tais dados precisam passar por alguma limpeza ou alteração e quais informações relevantes podem ser extraídas pela mineração de dados.
- Escolher os algoritmos de agrupamento a serem utilizados na mineração.
- Desenvolver uma ferramenta, implementando os algoritmos escolhidos.
- Realizar o processo de mineração no conjunto de dados da DRT, utilizando a ferramenta desenvolvida, e avaliar os resultados.

1.3 Organização do Trabalho

No capítulo 2 serão apresentados o conceito de mineração de dados, suas aplicações, os diferentes tipos de algoritmos e uma visão geral do processo de descoberta de conhecimento em banco de dados (DCBD), o qual tem a mineração de dados como etapa principal.

No capítulo 3 são descritos conceitos relativos às técnicas de agrupamento, uma classe de algoritmos de mineração de dados. Será dada uma ênfase especial nas técnicas utilizadas no trabalho.

No capítulo 4, é dada uma explicação mais detalhada do problema da DRT abordado neste trabalho, explicando a origem dos dados nos quais a mineração de dados foi realizada.

No capítulo 5 é descrito todo o processo de mineração de dados realizado no trabalho, explicando o que foi feito em cada uma de suas etapas, incluindo a análise prévia dos dados e a seleção dos algoritmos a serem utilizados.

No capítulo 6, é apresentada a descrição da ferramenta desenvolvida, utilizada na mineração de dados, incluindo os materiais e métodos utilizados e a sua implementação. É explicado o funcionamento interno da ferramenta, além de sua interface.

No capítulo 7 são apresentados os resultados obtidos pelo processo de mineração e uma análise comparativa entre os resultados gerados pelos diferentes algoritmos utilizados.

Finalmente, no capítulo 8 são apresentadas as conclusões obtidas e sugestões para trabalhos futuros.

2 MINERAÇÃO DE DADOS

Na definição de Fayyad, Piatetsky-Shapiro e Smyth (1996), mineração de dados é a aplicação de algoritmos específicos para extrair padrões de um conjunto de dados. Um padrão, no ponto de vista deste processo, pode ser definido como um evento ou combinação de eventos que ocorrem freqüentemente em um banco de dados, onde cada evento é representado por um conjunto de dados (LUCAS, 2006).

A mineração de dados é parte de um processo conhecido como Descoberta de Conhecimento em Bancos de Dados (DBCD). O termo em inglês (*Knowledge Discovery in Databases* – KDD) foi criado em 1989 e é definido como o processo não trivial de identificação de padrões válidos, novos, potencialmente úteis e compreensíveis, embutidos nos dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Este processo é constituído de várias etapas, tais como a preparação dos dados, seleção dos dados, limpeza dos dados e a própria mineração de dados, além da interpretação apropriada dos resultados da mineração.

Entretanto, na indústria, na mídia e no meio de pesquisas em banco de dados, o termo “mineração de dados” tornou-se mais popular que o longo termo “descoberta de conhecimento em bancos de dados”. Desta forma, alguns autores, tais como Han e Kamber (2006) e Witten e Frank (2005), utilizam o termo “mineração de dados” para descrever todo o processo de extrair informação nova e potencialmente útil dos dados. Esta mesma definição será utilizada neste trabalho.

A mineração de dados surgiu para atender a uma necessidade crescente do mundo atual. Estima-se que a quantidade de informação no mundo dobre a cada 20 meses. O crescimento em tamanho e em quantidade dos bancos de dados é provavelmente ainda mais rápido (FRAWLEY; PIATETSKY-SHAPIRO; MATHEUS, 1992). Há uma distância cada vez maior entre a quantidade de dados gerados e a capacidade de compreender e interpretar estes dados, ou seja, à medida que o volume de dados aumenta, a proporção destes dados que as pessoas conseguem entender diminui. Há uma necessidade por ferramentas de análise inteligentes que consigam diminuir esta distância. Han e Kamber (2006) descrevem esta

necessidade, aliada à abundância de dados, como uma situação rica em dados, mas pobre em informação.

Os dados coletados em grandes repositórios de dados tornaram-se "túmulos de dados" – arquivos de dados que são raramente visitados. Conseqüentemente, decisões importantes são freqüentemente tomadas baseadas não nos dados à disposição, mas sim na intuição do tomador de decisão, simplesmente porque este não possui as ferramentas para extrair o conhecimento embutido nos dados (HAN; KAMBER, 2006).

Ferramentas de mineração de dados realizam análise nos dados e podem desmascarar importantes padrões nos dados, contribuindo fortemente às estratégias de negócios, bases de conhecimento, pesquisa científica e médica, entre outros.

2.1 Áreas de aplicação

A mineração de dados é utilizada para uma variedade de propósitos, tanto no setor público quanto no privado. De acordo com Seifert (2005), diversos tipos de indústrias como bancos, seguradoras, farmacêuticas e vendas de varejo, comumente utilizam a mineração de dados para reduzir custos, aprimorar a pesquisa e aumentar as vendas. Seifert (2005) ainda afirma que, no setor público, as aplicações de mineração de dados eram inicialmente utilizadas como um meio de detectar fraudes e desperdícios, mas cresceram a ponto de serem utilizadas também para medir e melhorar o desempenho de programas.

Frawley, Piatetsky-Shapiro e Matheus (1992) citam diversas áreas onde a mineração de dados é utilizada, tais como: medicina, finanças, agricultura, marketing, engenharia, física, química, militar, ciência espacial e publicidade. A seguir são apresentados exemplos de situações onde a mineração de dados foi aplicada.

O primeiro exemplo é o retratado por Witten e Frank (2005). Cientistas ambientais analisam imagens tiradas via satélite para detectar vazamento de óleo em determinadas regiões e, desta forma, alertar sobre desastres ecológicos iminentes e dissuadir despejos ilegais. Entretanto, detectar um vazamento é um processo manual caro que requer profissionais altamente treinados avaliando cada região da imagem, o que torna pouco viável a análise exaustiva das imagens. Desta forma, foi desenvolvido um sistema capaz de identificar, por meio da mineração de

dados, as imagens que possam potencialmente representar um vazamento de óleo na região. Assim, o processo de análise manual ficaria restrito somente às imagens consideradas suspeitas pelo sistema.

O segundo exemplo é o retratado por Seifert (2005). Depois dos ataques terroristas de 11 de setembro de 2001, iniciou-se nos Estados Unidos o projeto de um sistema, denominado TIA (*Terrorism Information Awareness*), para auxiliar na detecção de grupos terroristas. O sistema TIA analisaria dados de civis americanos, tais como aluguéis de veículos, fichas criminais e compras de passagens aéreas, de forma encontrar padrões que pudessem sugerir alguma atividade terrorista. Mas, apesar da garantia de privacidade da informação obtida, o projeto sofreu grande oposição por parte da comunidade defensora dos direitos civis e acabou sendo cancelado em 2003.

O último exemplo é o estudo de caso realizado por Krogel (2000). O objetivo era utilizar a mineração de dados no banco de dados de uma seguradora para prever quais clientes estavam potencialmente interessados em uma apólice de seguro para *motor homes*, além de descrever os clientes – atuais ou potenciais – e tentar explicar porque estes clientes compram uma apólice para tais veículos. Como resultado da mineração, descobriu-se que os clientes mais propensos a comprar tal apólice são aqueles que tendem a pagar por seguros de valor elevado para carros (normalmente utilizados para mover *motor homes*) e/ou seguros contra incêndios (provavelmente porque *motor homes* são mais propensas a estes) e que vivem em áreas onde predominam pessoas com um elevado nível social e educacional (o que normalmente implica num maior poder aquisitivo e um maior acesso ao luxo e segurança).

2.2 Mineração de dados e aprendizagem de máquina

De acordo com Witten e Frank (2005) e Holsheimer e Siebes (1994), o funcionamento dos algoritmos de mineração de dados é fortemente baseado no conceito de aprendizagem de máquina. É através dos algoritmos de aprendizagem que são extraídos padrões que podem representar informações úteis acerca dos dados.

Holsheimer e Siebes (1994) relacionam o conceito de aprendizagem de máquina com o conceito de aprendizagem indutiva. Sistemas cognitivos, tais como

os seres humanos, tentam entender seu ambiente usando uma simplificação deste ambiente – chamado de modelo. A criação de tal modelo é chamada de aprendizagem indutiva. Durante a fase de aprendizagem, o sistema cognitivo observa seu ambiente e reconhece similaridades entre os objetos e eventos neste ambiente. Ele agrupa objetos similares em classes e constrói regras que prevêem o comportamento dos habitantes de uma classe.

Para Holsheimer e Siebes (1994), duas técnicas de aprendizagem são de interesse especial. Na aprendizagem supervisionada, um professor externo define classes e fornece ao sistema cognitivo exemplos de cada classe. O sistema deve descobrir propriedades nos exemplos para cada classe – a descrição da classe. Esta técnica é também conhecida como aprendizagem através de exemplos. Uma classe, junto com sua descrição, forma uma regra de classificação 'se <descrição> então <classe>' que pode ser usada para prever a classe de objetos não vistos previamente. Na aprendizagem não-supervisionada não existem classes previamente definidas. O sistema deve descobrir as classes por conta própria, baseando-se apenas em propriedades comuns entre os objetos. Assim, esta técnica também é conhecida como aprendizagem através da observação e descoberta.

O estudo e a modelagem computacional destes processos é o assunto de uma área de pesquisa chamada aprendizagem de máquina. Um sistema de aprendizagem de máquina não interage diretamente com o ambiente, mas utiliza observações codificadas, geralmente armazenadas num conjunto – chamado de conjunto de treinamento. Quando se utiliza um banco de dados como um conjunto de treinamento, o processo de aprendizagem é chamado de mineração de dados. A aprendizagem é realizada sobre os dados com o objetivo de encontrar descrições estruturais que ajudem o usuário a compreender e fazer previsões em cima destes dados.

2.3 Etapas do processo de Mineração de Dados

Mineração de dados é o processo de descoberta de conhecimento interessante a partir de grandes quantidades de dados armazenados em banco de dados, *data warehouses*, ou em quaisquer outros repositórios de informações (HAN; KAMBER, 2006). Este processo envolve uma série de etapas cuja principal etapa é a aplicação do algoritmo de mineração. Entretanto, a execução das outras etapas é

essencial para que os padrões extraídos pelo algoritmo de mineração sejam úteis e compreensíveis. O processo pode envolver interação significativa com o usuário e pode conter iterações (*loops*) entre duas etapas (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

A definição das etapas de mineração de dados varia de acordo com o autor. Fayyad, Piatetsky-Shapiro e Smyth (1996) trazem uma abordagem mais completa, descrevendo o processo de descoberta de conhecimento através de nove etapas:

- definição do objetivo da descoberta de conhecimento;
- seleção dos dados;
- limpeza e pré-processamento dos dados;
- redução dos dados;
- casamento do objetivo com um método de mineração de dados particular;
- escolha do algoritmo de mineração de dados;
- aplicação do algoritmo de mineração de dados;
- interpretação dos padrões minerados;
- utilização do conhecimento descoberto.

Esta mesma definição das etapas é adotada por Chung e Gray (1999). Cabe salientar que nesta abordagem a mineração de dados é apenas uma etapa do processo de descoberta de conhecimento.

Han e Kamber (2006) e Rezende et al. (2003) trazem abordagens mais simplificadas, focadas apenas no processamento e na apresentação dos dados. Nestas abordagens pressupõe-se que o usuário já tenha o conhecimento prévio do domínio da aplicação, traçado o objetivo da mineração de dados e feita a escolha do algoritmo de mineração de dados. Rezende et al. (2003) definem o processo de mineração de dados como sendo constituído de três grandes etapas: pré-processamento, extração dos padrões e pós-processamento. A etapa de pré-processamento é precedida pela identificação do problema e a etapa de pós-processamento é sucedida pela utilização do conhecimento.

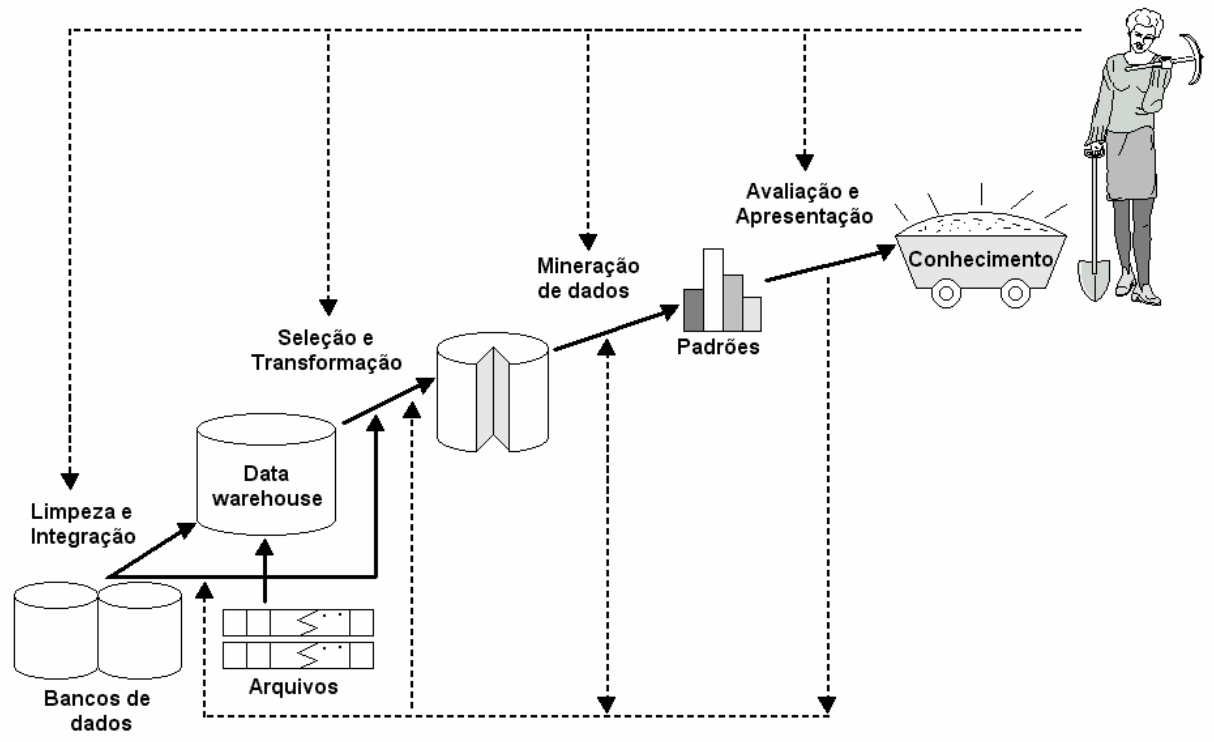

Figura 1: Etapas da mineração de dados
Fonte: Adaptação feita a partir de HAN; KAMBER, 2006.

Na definição de Han e Kamber (2006), o processo de mineração de dados é constituído de sete etapas:

- limpeza dos dados;
- integração dos dados;
- seleção dos dados;
- transformação dos dados;
- aplicação do algoritmo de mineração de dados;
- avaliação dos padrões;
- representação do conhecimento.

A Fig. 1 ilustra este processo. Esta é a definição utilizada no trabalho e será descrita de forma mais detalhada nas sub-seções seguintes.

Por fim, existem abordagens do processo de mineração voltadas para aplicações específicas. Chapman et. al. (2000) definiram um modelo, denominado CRISP-DM 1.0 (*Cross-Industry Standard Process for Data Mining)*, voltado a profissionais de negócios que utilizam banco de dados empresariais. Este modelo, que visa auxiliar a tomada de decisões, é composto por seis etapas: entendimento do negócio, entendimento dos dados, preparação dos dados, modelagem, avaliação do modelo e publicação.

2.3.1 Limpeza dos dados

Os dados contidos em um banco de dados possuem, na maioria das vezes, uma série de erros que podem prejudicar os resultados da mineração se não forem adequadamente corrigidos. Os dados contidos tendem a estar incompletos, ruidosos e inconsistentes. A etapa de limpeza dos dados procura tratar ou pelo menos contornar estes problemas. Para cada defeito existente nos dados existem diferentes soluções, mostradas a seguir.

Valores faltantes: ocorre quando determinadas tuplas do banco de dados não possuem um valor em um determinado atributo. O valor é normalmente representado por *null*, um espaço em branco ou um valor numérico fora do intervalo permitido. As possíveis soluções, apresentadas por Han e Kamber (2006), incluem: ignorar a tupla, não a utilizando na mineração, preencher manualmente o valor que está faltando, utilizar uma constante global (como “Desconhecido”) para preencher automaticamente os valores faltantes, utilizar a média do atributo calculada pelas demais tuplas e, finalmente, utilizar o valor mais provável para preencher o atributo – este valor pode ser determinado por árvores de decisão, regressão, formalismo Bayesiano e outras ferramentas de inferência. Witten e Frank (2005) salientam, porém, que um valor faltante nem sempre implica em um erro nos dados. Por exemplo, em um banco de dados contendo dados de pacientes, alguns valores de um determinado paciente podem estar ausentes simplesmente porque o médico julgou desnecessário medi-los, uma vez que apenas os valores já medidos eram suficientes para fornecer o diagnóstico completo daquele paciente.

Dados ruidosos: ruído é um erro ou variância aleatória em uma variável medida. Conforme Han e Kamber (2006), dados ruidosos geralmente se apresentam como valores discrepantes, isto é, valores que se distanciam muito da média dos valores de um determinado atributo. Para eliminar o ruído, pode-se utilizar técnicas para suavizar os dados, tais como *binning* ou regressão linear. Podem-se ainda utilizar técnicas de agrupamento para detectar valores discrepantes. Conforme será explicado no capítulo 3, técnicas de agrupamento dividem o conjunto de dados em grupos, de modo que objetos do mesmo grupo sejam semelhantes entre si. Intuitivamente, valores que se distanciam consideravelmente dos grupos formados podem ser considerados discrepantes. Witten e Frank (2005) também sugerem o

uso de histogramas e outras visualizações gráficas para encontrar valores discrepantes.

Dados inconsistentes: podem ser fruto de erros de digitação ou valores escritos de formas diferentes que possuem o mesmo significado como, por exemplo, “Estados Unidos” e “Estados Unidos da América”. Estas inconsistências acabam erroneamente gerando novos valores possíveis para um atributo, quando na verdade são apenas valores já existentes representados de uma forma diferente. Segundo Witten e Frank (2005) a solução neste caso é ter conhecimento do domínio para detectar e corrigir manualmente as inconsistências.

2.3.2 Integração dos dados

Muitas vezes os dados a serem utilizados no processo de mineração estão espalhados em diferentes repositórios de dados. Neste caso, deve-se realizar a integração dos dados, isto é, o processo de combinar dados vindos de várias fontes diferentes em um único e coerente repositório de dados, como um *data warehouse*. Estas fontes podem incluir diferentes bancos de dados, cubos de dados ou arquivos.

2.3.3 Seleção dos dados

A etapa de seleção dos dados consiste em recuperar os dados relevantes para a realização da análise. A relevância ou não de determinados dados vai depender do objetivo previamente traçado que se deseja atingir com a mineração. Nesta etapa, é necessário que o analista tenha o conhecimento do domínio compreendido pelos dados para determinar quais destes dados poderão gerar padrões interessantes e quais devem ser descartados. Selecionar apenas os dados relevantes garante um melhor resultado da mineração de dados, tornando o processo mais rápido e os padrões encontrados mais fáceis de serem interpretados.

2.3.4 Transformação dos dados

Na etapa que antecede a aplicação do algoritmo de mineração de dados, os dados são transformados e consolidados em um formato apropriado para a mineração. A transformação de dados, segundo Han e Kamber (2006), pode envolver as operações apresentadas a seguir.

Suavização: utilizada para remover ruídos dos dados. Tais técnicas incluem *binning*, regressão e agrupamento. A suavização pode ser realizada na etapa de transformação ou, conforme explicado anteriormente, na etapa de limpeza.

Agregação: onde os dados são sumarizados ou agregados. Por exemplo, os dados sobre vendas diárias podem ser agregados de modo a computar quantidades totais de vendas mensais ou anuais.

Generalização: onde dados “primitivos” ou de baixo nível são substituídos por conceitos de alto nível através do uso de hierarquias conceituais. Por exemplo, atributos categóricos, como rua, podem ser generalizados para conceitos de mais alto nível, como cidade ou estado.

Normalização: um atributo é normalizado escalando seus valores de modo que fiquem num pequeno intervalo especificado, como 0,0 a 1,0. Normalização é particularmente útil para algoritmos de classificação envolvendo redes neurais ou envolvendo medidas de distância, como agrupamento. Neste último caso, a normalização permite que atributos com intervalos grandes, como salário anual, e pequenos, como idade, tenham o mesmo peso no cálculo da distância, além de permitir o cálculo da distância entre objetos com atributos de tipos mistos (numérico e categórico).

Construção de atributos: onde novos atributos são construídos e adicionados a partir de um dado conjunto de atributos para auxiliar o processo de mineração. Por exemplo, construir o atributo “área” a partir dos atributos “largura” e “altura”.

2.3.5 Aplicação do algoritmo de mineração de dados

A aplicação do algoritmo de mineração é a etapa principal do processo, todas as demais etapas são realizadas em função desta. É a mineração de dados propriamente dita. Alguns autores, como Fayyad, Piatetsky-Shapiro e Smyth (1996), limitam o termo “mineração de dados” a esta etapa somente. O algoritmo de mineração recebe como entrada os dados pré-processados (pelas etapas anteriores) e gera como saída padrões que refletem o comportamento dos dados recebidos. A forma como os padrões são gerados e o significado destes variam para cada algoritmo de mineração de dados. A escolha do algoritmo que apresentará os padrões mais apropriados vai depender do tipo de conhecimento que se deseja obter com a mineração. Uma visão geral dos diferentes tipos de padrões é apresentada na seção 2.4.

Segundo Han e Kamber (2006) os algoritmos de mineração podem ser utilizados para dois propósitos principais: a descrição e a previsão. A tarefa de descrição caracteriza as propriedades gerais dos dados em um banco de dados, apresentando ao usuário uma descrição legível destes dados. A tarefa de previsão realiza inferência nos dados atuais para prever outras variáveis e valores desconhecidos.

2.3.6 Avaliação dos padrões

Nem todos os padrões gerados pela mineração são válidos ou úteis. De fato, apenas uma parcela destes padrões é interessante. Na etapa de avaliação dos padrões, são realizadas medidas de interesse para identificar os padrões que realmente apresentam alguma utilidade. Estes padrões úteis representam o referido “conhecimento”. Tais medidas de interesse podem ser objetivas ou subjetivas.

Medidas objetivas são baseadas na estrutura dos padrões descobertos e utilizam funções matemáticas para avaliar a credibilidade destes padrões e, às vezes, filtrar os padrões que não satisfazem determinadas condições. Cada medida objetiva é específica para um determinado tipo de padrão. Medidas subjetivas são baseadas no conhecimento prévio do usuário e nos resultados que ele espera encontrar. Padrões que eram previamente desconhecidos pelo analista ou que confirmam uma hipótese previamente formulada que ele deseja validar são considerados úteis por este.

2.3.7 Representação do conhecimento

Uma vez que os padrões foram encontrados e possivelmente filtrados pela etapa anterior, o último passo é a apresentação do conhecimento obtido em uma forma legível ao usuário. Nesta etapa, são utilizadas técnicas de visualização e representação do conhecimento. Estas técnicas incluem construção de regras, árvores de decisão, representação de grupos, tabelas, entre outras.

2.4 Classificação dos algoritmos de mineração de dados

Os algoritmos de mineração de dados podem ser classificados de diversas maneiras. Segundo Han e Kamber (2006) os algoritmos podem ser classificados quanto ao tipo de banco de dados minerado (relacional, orientado a objetos ou *data*

warehouse), tipo de técnica utilizada (métodos estatísticos, visualização, redes neurais, etc.), aplicação adaptada (finanças, telecomunicações, marketing, análise de DNA, entre outros) e ao tipo de padrão encontrado (classificação, associação e agrupamento). Esta última forma de classificação dos algoritmos foi adotada como referência por este trabalho, uma vez que o estudo realizado pelo mesmo é focado em um tipo específico de padrão. Quanto ao tipo de padrão encontrado, os algoritmos de mineração de dados podem ser classificados como se segue.

Classificação: este algoritmo consiste em encontrar descrições de classes – previamente definidas – a partir dos dados que lhes são fornecidos. Na primeira fase, a fase de treinamento, o algoritmo recebe dados de objetos que pertencem a uma determinada classe (ou seja, já classificados) e, baseado nas características presentes nos objetos de uma mesma classe, constrói regras que descrevem esta classe. Estas regras são denominadas regras de classificação e são escritas na forma “se <descrição> então <classe>”. Uma vez construídas estas regras, elas podem ser utilizadas para determinar a classe dos objetos que não foram classificados. Esta constitui a segunda fase, a fase de classificação. Devido à necessidade das classes serem previamente definidas, a fase de treinamento é conhecida como aprendizagem supervisionada, conforme explicado na seção 2.2. O primeiro exemplo apresentado na seção 2.1 consiste de um problema de classificação. Nele, as imagens eram classificadas pelo sistema como “possível vazamento” ou “sem vazamento”.

Associação: constrói regras, denominadas regras de associação, que descrevem o comportamento de determinados atributos, associando um valor de um atributo a um valor de outro atributo. Diferente das regras de classificação, que prevêem o comportamento de um único atributo categórico (a classe), as regras de associação podem prever o comportamento de quaisquer atributos presentes nos objetos. Exemplos de regras de associação são as regras do tipo “se <atributo A> = X então <atributo B> = Y” ou “se <atributo A> ≤ X então <atributo C> = Z”.

Agrupamento: é utilizado quando não há uma definição prévia das classes a serem descritas. Este algoritmo constrói um número finito de grupos e distribui os objetos do banco de dados em tais grupos baseando-se na similaridade entre estes objetos, de modo que objetos semelhantes fiquem no mesmo grupo e objetos distintos fiquem em grupos diferentes. Cada grupo de objetos pode ser visto como uma classe recém descoberta. Pelo fato das classes não serem previamente

definidas, este algoritmo realiza aprendizagem não-supervisionada. O segundo exemplo exposto na seção 2.1, no qual desejava-se utilizar o sistema TIA para encontrar padrões de crimes, consiste de um exemplo de agrupamento.

Este trabalho realiza um estudo de caso no qual tenta-se extrair padrões de dados que não possuem classes definidas ou atributos a serem associados, sendo assim necessário utilizar algoritmos de agrupamento. Desta forma, foi dada uma atenção maior a esta classe, que será abordada de forma mais detalhada no capítulo seguinte.

3 AGRUPAMENTO

Conforme explicado no capítulo anterior, o agrupamento é uma das classes de algoritmos de mineração de dados quanto ao tipo de padrão encontrado. O estudo de caso realizado neste trabalho baseou-se na utilização de algoritmos desta categoria.

Em certas ocasiões, deseja-se realizar a mineração de dados em objetos que não possuem um rótulo (atributo) especificando sua classe. Em geral, isto ocorre simplesmente porque as classes não são inicialmente conhecidas. O agrupamento pode ser utilizado para encontrar classes e gerar rótulos para elas. Conforme explicado anteriormente, o agrupamento é uma forma de aprendizagem não-supervisionada, pois não requer a definição prévia das classes para que o algoritmo seja capaz descrever as similaridades e diferenças presentes nos objetos.

Os algoritmos de agrupamento operam sobre um conjunto de dados heterogêneo, construindo um número finito de grupos e distribuindo os dados nestes grupos, de modo que os objetos dos dados que possuam características semelhantes fiquem em um mesmo grupo e objetos distintos fiquem em grupos diferentes. O objetivo, portanto, é maximizar a similaridade intragrupal e minimizar a similaridade intergrupal (QUINTALES, 2007). Cada grupo formado constitui uma classe recém descoberta.

Por ser considerado uma técnica de mineração de dados o agrupamento é geralmente executado na quinta etapa do processo de mineração de dados descrito no capítulo anterior, isto é, na etapa onde o algoritmo de mineração é utilizado para extrair padrões dos dados. Entretanto, o agrupamento também pode ser utilizado nas etapas de pré-processamento dos dados, de forma a auxiliar a mineração realizada por outros tipos de técnicas. Em um exemplo descrito por Witten e Frank (2005), o agrupamento pode ser utilizado para auxiliar um algoritmo de classificação, que utiliza objetos classificados no seu treinamento. Às vezes os objetos classificados são escassos, o que impede um treinamento eficiente. Utiliza-se, então, o agrupamento para determinar as classes mais prováveis dos objetos que não foram classificados e então utilizar estes objetos no treinamento. Outro exemplo

é o citado na sub-seção 2.3.1, onde o agrupamento é utilizado para detectar valores discrepantes.

3.1 Categorização dos principais métodos de agrupamento

Uma grande variedade de algoritmos de agrupamento existe na literatura. Dividir esta variedade em categorias não é um processo trivial, pois estas categorias podem se sobrepor, de modo que um método pode conter características de duas ou mais delas. Assim como na abordagem das etapas de mineração de dados e na classificação dos algoritmos de mineração, não há uma definição universal para as categorias dos algoritmos de agrupamento, que são abordadas de uma forma diferente por cada autor.

Quintales (2007) divide os algoritmos em duas categorias apenas: particionais e hierárquicos. Berkhin (2002) define oito categorias de algoritmos: hierárquicos, particionais, baseados em densidade, baseados em grades, baseados em co-ocorrências de dados categóricos, agrupamento baseado em restrições, algoritmos baseados em redes neurais, e algoritmos para dados de muitas dimensões. Por fim, Han e Kamber (2006) definem sete categorias, seis delas semelhantes às de Berkhin, porém não existindo uma categoria específica para redes neurais.

A definição de Berkhin (2002) é mais completa das citadas acima, pois além de abranger todas as técnicas descritas por Quintales (2007) e Han e Kamber (2006), traz algumas não citadas nas outras referências, como as baseadas em co-ocorrência de dados categóricos. Além disso, é a única definição a possuir uma classe específica para algoritmos baseados em redes neurais – que foram utilizados neste trabalho. Desta forma, optou-se por adotar a definição de Berkhin (2002), que será descrita a seguir.

3.1.1 Algoritmos de particionamento

Dado um banco de dados com n objetos, um método de particionamento constrói k partições, onde cada partição representa um grupo e $k \leq n$. Ou seja, o método classifica os dados em k grupos de modo que cada grupo tenha pelo menos um objeto. O valor k é normalmente definido pelo usuário antes da execução do algoritmo (QUINTALES, 2007).

Logo no início, os k grupos são construídos e cada objeto é alocado em um grupo. Depois disso, o algoritmo inicia um processo de relocação iterativa, onde os objetos são movidos de um grupo para outro, de modo a tornar os grupos os mais homogêneos possíveis. Para realizar esta relocação, utilizam-se métodos heurísticos, tais como o k-médias, onde cada grupo é representado pela média dos valores de seus objetos, ou o k-*medoids*, onde cada grupo é representado pelo objeto mais próximo de seu centro.

Nas técnicas acima citadas, cada objeto pertence exclusivamente a um grupo, isto é, um objeto não pode fazer parte de dois ou mais grupos ao mesmo tempo. Por outro lado, existem técnicas de agrupamento probabilístico onde cada objeto pertence a cada um dos k grupos com uma determinada probabilidade. Exemplo de método de agrupamento probabilístico é o algoritmo EM (*Expectation Maximization*), que aproxima cada grupo a um modelo de distribuição probabilístico (DEMPSTER; LAIRD; RUBIN, 1977).

3.1.2 Algoritmos hierárquicos

Um método hierárquico cria uma decomposição hierárquica dos objetos do banco de dados, agrupando estes objetos em uma árvore de grupos, também conhecida como dendrograma. Métodos hierárquicos podem ser aglomerativos ou divisivos, dependendo de como a decomposição hierárquica é formada.

Na abordagem aglomerativa, também chamada de abordagem *bottom-up*, cada objeto inicialmente forma um grupo separado. Então os objetos semelhantes são sucessivamente fundidos até que todos os grupos sejam unidos em um único grupo (o nível mais alto da hierarquia), ou até que uma condição de parada seja satisfeita (esta condição normalmente é o número k de grupos solicitados). O oposto corre na abordagem divisiva, também chamada de abordagem *top-down*, onde todos os objetos inicialmente formam um único grupo, que é sucessivamente dividido em grupos menores até que cada grupo contenha um único objeto ou até atingir um critério de parada.

Uma importante característica dos métodos hierárquicos é que, uma vez que uma divisão ou junção é feita, ela nunca é desfeita. Diferente do que ocorre no particionamento, onde os grupos são iterativamente reestruturados de forma a otimizar o resultado do agrupamento, no método hierárquico os grupos são rígidos e não são alterados. Por um lado, isto resulta num menor custo computacional, pois o

algoritmo não consome tempo revisando os grupos formados. Por outro, a decomposição hierárquica geralmente resulta em um agrupamento de qualidade inferior ao do particionamento (BERKHIN, 2002). Exemplos de algoritmos hierárquicos são o CURE (Clustering Using Representatives) (GUHA; RASTOGI; SHIM, 1998) e o Chameleon (KARYPIS; HAN; KUMAR, 1999).

3.1.3 Algoritmos baseados em densidade

A maior parte dos algoritmos de particionamento é baseada na distância entre os objetos. Tais algoritmos conseguem apenas encontrar grupos de formato esférico. Para descobrir grupos de formato arbitrário, foram desenvolvidos métodos de agrupamento baseados na noção de densidade. A idéia geral destes métodos é fazer com que cada grupo formado (inicialmente pequeno) continue crescendo desde que a densidade (número de objetos) na sua “vizinhança” atinja um valor mínimo. Isto é, para cada objeto presente no grupo, a área ao seu redor precisa conter pelo menos um número determinado de objetos. Este número é definido pelo usuário, assim como o raio que determina a área de vizinhança dos objetos.

Exemplos de algoritmos baseados em densidade são o DBSCAN (ESTER et al., 1996) e o DENCLUE (HINNEBURG; KEIM, 1998). Além de descobrir grupos de forma arbitrária, estes algoritmos têm a capacidade de filtrar valores discrepantes. Apesar destas vantagens, Berkhin (2002) salienta que os resultados gerados por estes algoritmos apresentam problemas na hora de serem interpretados.

3.1.4 Algoritmos baseados em grades

Algoritmos baseados em grades quantizam o espaço dos objetos em um número finito de células que formam uma estrutura de grade. Todas as operações do agrupamento são realizadas nesta grade. Ao contrário da maioria dos algoritmos de agrupamento, os algoritmos baseados em grade possuem um custo de processamento que não depende do número de objetos dos dados, depende apenas do número de células em cada dimensão do espaço quantizado. Um número maior de células gera grupos de maior qualidade, mas gera um custo maior de processamento.

Um exemplo típico de método baseado em grade é o algoritmo STING (WANG; YANG; MUNTZ, 1997). Outro exemplo é o algoritmo WaveCluster, que

combina funcionalidades dos métodos baseados em grade e baseados em densidade (SHEIKHOLESLAMI; CHATTERJEE; ZHANG, 1998).

3.1.5 Algoritmos baseados em co-ocorrências de dados categóricos

Alguns bancos de dados possuem dados categóricos que são apresentados na forma de transações. Uma transação é um conjunto finito de elementos, chamados de itens, vindos de um universo de itens comum. Cada transação é representada em um formato ponto-por-atributo, enumerando todos os itens j, e associando a cada transação um conjunto de atributos binários que indicam se um determinado item j pertence a uma transação ou não. Tal representação é bastante esparsa (com muitos valores iguais a zero) e duas transações possuem poucos itens em comum, além de terem muitas dimensões.

Para lidar com estas características, algoritmos baseados na idéia de co-ocorrência de dados categóricos foram elaborados. Exemplos destes algoritmos são o ROCK (GUHA; RASTOGI; SHIM, 1999), que também é um algoritmo de agrupamento hierárquico, o SNN (ERTOZ; STEINBACH; KUMAR, 2002) e o CACTUS (GANTI; GEHRKE; RAMAKRISHNAN, 1999).

3.1.6 Agrupamento baseado em restrições

Determinados algoritmos de agrupamento podem incorporar, em seu processo, restrições especificadas pelo usuário ou orientadas à aplicação. Uma restrição expressa a expectativa do usuário ou descreve propriedades dos resultados desejados do agrupamento e fornece um meio eficiente de comunicação com o processo de agrupamento. Em outros termos, o agrupamento baseado em restrições encontra grupos que satisfaçam preferências especificadas pelo usuário.

Exemplos de restrições incluem o número esperado de grupos, o tamanho mínimo ou máximo do grupo, pesos para diferentes objetos ou dimensões, atributos escolhidos, função de distância a ser utilizada, etc. Uma importante aplicação baseada em restrições é o agrupamento de dados com presença de obstáculos como, por exemplo, agrupar estabelecimentos de uma cidade separados por um rio ou uma montanha.

3.1.7 Algoritmos baseados em redes neurais

Uma rede neural é um conjunto de unidades de entrada e saída conectadas, onde cada conexão possui um peso associado a ela. Redes neurais possuem propriedades que as tornam populares para o agrupamento (HAN; KAMBER, 2006): são arquiteturas de processamento inerentemente paralelo e distribuído, aprendem ajustando os pesos de suas conexões de modo a se “encaixarem” melhor nos dados e conseguem representar grupos usando apenas fatores quantitativos.

O exemplo mais conhecido de rede neural para agrupamento são os mapas auto-organizáveis (*Self-Organizing Maps* - SOMs). Uma descrição mais detalhada desta rede é feita na seção 3.4.

3.1.8 Algoritmos para dados de muitas dimensões

O agrupamento apresenta problemas quando os dados possuem muitos atributos ou dimensões (mais de 10 dimensões ou até centenas em alguns casos). Quando a dimensionalidade aumenta, geralmente apenas um pequeno número de dimensões é relevante para certos grupos. Os dados presentes nas dimensões irrelevantes podem produzir ruídos e prejudicar a qualidade dos grupos descobertos. Para contornar este problema utilizam-se técnicas de transformação, que sumarizam os dados criando combinações lineares dos atributos, e técnicas de seleção de atributos, que removem os atributos considerados irrelevantes ou redundantes.

Entre os algoritmos utilizados para lidar com dados de muitas dimensões estão o CLIQUE (AGRAWAL et al., 1998) e o PROCLUS (AGGARWAL et al., 1999), que são baseados em descobertas de sub-espaços, e o pCluster (WANG et al., 2002), que é baseado na freqüência dos padrões.

3.2 Calculando similaridades entre objetos

Conforme explicado anteriormente, o agrupamento é normalmente realizado com base na semelhança entre os objetos do banco de dados. O grau de similaridade entre dois objetos irá determinar se estes objetos serão colocados ou não em um mesmo grupo. Utilizam-se, então, medidas de distância para determinar o quão semelhante um objeto é de outro. Quanto menor a distância, maior o grau de similaridade entre os objetos. De acordo com Teknomo (2006b), uma medida de

distância deve satisfazer pelo menos as três primeiras dentre as quatro condições listadas abaixo, para todo x, y, e z. Se a medida de distância satisfizer todas as quatro condições, ela é chamada de métrica.

d(x,y) ≥ 0;

d(x,y) = 0 se e somente se x = y;

d(x,y) = *d(y,x)*, simetria;

d(x,y) ≤ *d(x,z)* + *d(y,z)*, desigualdade triangular;

Uma série de medidas de distância é apresentada por Teknomo (2006b). Existem medidas diferenciadas para atributos numéricos (valores quantitativos) e atributos categóricos (valores qualitativos). Dentre elas, as medidas mais comuns são apresentadas a seguir. Para todas elas, considera-se n o número de atributos presentes em cada objeto e x_i um atributo do objeto X.

3.2.1 Medindo a distância entre objetos com atributos numéricos

Uma medida normalmente utilizada para calcular a distância entre objetos com atributos numéricos é a distância Euclidiana (1). Esta distância, medida entre dois objetos X e Y, é apresentada na seguinte fórmula:

$$d(X,Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (1)$$

Uma fórmula genérica para medir a distância numérica é a fórmula de Minkowski (2). Nela, é utilizado um parâmetro λ. Para λ=1 a fórmula equivale à distância de Manhatan e para λ=2 a fórmula equivale à distância Euclidiana.

$$d(X,Y) = \sqrt[\lambda]{\sum_{i=1}^{n} |x_i - y_i|^{\lambda}} \quad (2)$$

3.2.2 Medindo a distância entre objetos com atributos categóricos

A distância entre dois objetos com atributos categóricos pode ser calculada pela fórmula da distância de Hamming (3). Embora esta medida tenha sido originalmente desenvolvida para objetos com atributos binários (com apenas dois valores possíveis), Lourenço et al. (2004) afirmam que ela pode ser utilizada para atributos categóricos com três ou mais valores possíveis. Esta medida pode ser

definida pelo número total de atributos que possuem valores diferentes entre os objetos. A fórmula da distância é dada por

$$d(X,Y) = \sum_{i=1}^{n} \delta(x_i, y_i) \tag{3}$$

onde

$$\delta(x_i, y_i) = \begin{cases} 0, & se \quad x_i = y_i \\ 1, & se \quad x_i \neq y_i \end{cases} \tag{4}$$

3.2.3 Medindo a distância entre objetos com atributos mistos

Quando se trabalha com bancos de dados, na maioria das vezes são encontrados objetos com tipos mistos, que possuem tanto valores numéricos quanto categóricos. A solução neste caso, segundo Teknomo (2006b), é calcular a distância separadamente para atributos numéricos e categóricos (utilizando as fórmulas apresentadas anteriormente) e então somar as distâncias calculadas. Entretanto, para que esta solução possa ser utilizada, é necessário normalizar os valores numéricos, em um intervalo de 0 a 1, para que tenham o mesmo peso dos valores categóricos no cálculo da distância.

Uma das formas de normalização é a normalização MinMax. Para cada valor x_i do atributo i, a versão normalizada x'_i é dada pela formula (5), onde max_i é o maior valor existente do atributo i e min_i é o menor valor existente do atributo i.

$$x'_i = \frac{x_i - \min_i}{\max_i - \min_i} \tag{5}$$

Existem ainda outras formas de normalização, tais como a *z-score*, que utiliza a média e o desvio padrão do atributo, e a normalização decimal, que desloca o ponto decimal do valor de modo que este fique entre 0 e 1.

3.3 O algoritmo K-médias

O algoritmo K- médias é um método de particionamento, apresentado por MacQueen (1967), que distribui n objetos em k grupos, k ≤ n, de modo que cada objeto fique o mais próximo possível do centróide de seu grupo. No K-médias, o centróide de um grupo é representado pela média dos valores dos objetos que pertencem ao grupo.

O algoritmo funciona da maneira como se segue. Inicialmente, k objetos (k definido pelo usuário) são selecionados aleatoriamente dentre os n objetos disponíveis. Cada um dos k objetos selecionados será o centróide inicial de um grupo. Depois, cada um dos objetos restantes é atribuído ao grupo cujo centróide é o mais próximo deste objeto. Então, o novo centróide de cada grupo é calculado e os objetos são reagrupados de acordo com estes novos centróides. O processo segue de forma iterativa até que os grupos fiquem estáveis, isto é, até que todos os objetos permaneçam em seus grupos durante uma iteração. O procedimento é sumarizado na Fig. 2.

Entrada:
k – o número desejado de clusters
D – um conjunto de dados contendo n objetos
Saída:
Um conjunto de k grupos

Método:
(1) Escolha aleatoriamente k objetos de D como os centróides iniciais
(2) **repita**
(3) (re)atribua cada objeto ao grupo cujo centróide seja o mais próximo
(4) atualize o centróide de cada grupo, isto é, calcule a média dos valores dos objetos de cada grupo
(5) **até** não haver mais alterações nos grupos.

Figura 2: O algoritmo de particionamento K-médias

O critério utilizado para avaliar a qualidade dos grupos formados pelo algoritmo é a soma das distâncias de cada objeto ao centróide de seu respectivo grupo (LEI; HE; LI, 2006). Quanto menor o resultado da soma, maior a qualidade dos grupos. A soma SD é dada pela fórmula (6) abaixo, onde X_i é um dos n objetos, C_j é um dos k grupos e Q_j é o centróide de C_j.

$$SD = \sum_{j=1}^{k} \sum_{X_i \in C_j} d\left(X_i, Q_j\right) \tag{6}$$

O fato dos grupos serem reestruturados a cada iteração permite que o K-médias forme grupos de boa qualidade, principalmente se comparado aos algoritmos hierárquicos. Isto, aliado à sua simplicidade, torna o K-médias um algoritmo bastante popular (BERKHIN, 2002). Entre as desvantagens do algoritmo está o fato do resultado do agrupamento ser muito sensível aos centróides definidos aleatoriamente no início do algoritmo e o fato do cálculo da média ser bastante afetado por valores discrepantes (TEKNOMO, 2006a).

3.3.1 Uma variação do K-médias para dados categóricos

O algoritmo K-médias descrito anteriormente é restrito a objetos que possuam apenas valores numéricos, uma vez que não é possível calcular a média entre valores categóricos. Uma alternativa a ser utilizada é o algoritmo K-modas, onde o centróide de cada grupo é representado pela moda, isto é, pela categoria mais freqüentemente encontrada no grupo. Lei, He e Li (2006) afirmam, entretanto, que este algoritmo é bastante instável, uma vez que a moda de um grupo geralmente não é única.

Assim, uma nova versão do algoritmo K-médias para dados categóricos foi proposta por Lei, He e Li (2006). Nesta versão, o centróide de cada grupo é representado pela freqüência com que cada categoria ocorre no grupo. O funcionamento do algoritmo, de maneira geral, é o mesmo do K-médias, diferenciando-se apenas na formação dos centróides e no cálculo da distância.

Dado um grupo C = {X_1,..., X_p} de objetos categóricos com X_i = ($x_{i,1}$,...,$x_{i,m}$). Denote o conjunto formado pelos valores categóricos $x_{1,j}$,...,$x_{p,j}$ por D_j. Por exemplo, o conjunto formado pelos valores a,b,a,c é {a,b,c}. Então o centróide de C é definido por Q = (q_1,...,q_m) com

$$q_j = \{(c_j, f_{C\,j}) \mid c_j \in D_j\} \tag{7}$$

onde f_{Cj} é a freqüência relativa da categoria c_j em C, isto é, $f_{Cj} = n_{Cj} / p$, onde n_{Cj} é o número de objetos em C contendo a categoria c_j no *j*-ésimo atributo.

Devido à modificação proposta na formação dos centróides para objetos categóricos, uma nova maneira de calcular a distância entre um objeto categórico e o centróide de um grupo foi definida. Seja C = {X_1,..., X_p} um grupo de objetos categóricos, X = (x,...,x_m) um objeto categórico e Q = (q_1,...,q_m) o centróide do grupo C. A distância entre X e Q é dada por

$$d(X,Q) = \sum_{j=1}^{m} \sum_{c_j \in D_j} f_{C_j} \cdot \delta(x_j, c_j) \tag{8}$$

Assim, a distância d(X,Q) é calculada com base nas freqüências relativas dos valores categóricos no grupo e na função δ, mostrada em (4), entre os valores categóricos.

3.4 Mapas auto-organizáveis

Os mapas auto-organizáveis (*Self-Organizing Maps* - SOMs), também conhecidos como Mapas de Kohonen, são uma técnica de visualização de dados criada por Teuvo Kohonen que consiste em representar dados multidimensionais em um espaço de poucas dimensões (geralmente uma ou duas) através do uso de redes neurais (KOHONEN, 2001). Diferente de outros tipos de redes neurais, tais como o Back Propagation, os SOMs realizam aprendizagem não-supervisionada. Necessitam apenas dos dados de entrada, não há necessidade do usuário informar os valores que as saídas da rede neural devem ter (CHESNUT, 2004).

A estrutura de um SOM é composta por um número finito e pré-determinado de nodos, dispostos de formas variadas, como um quadrado (Fig. 3a) ou um favo de mel (Fig. 3b). Cada nodo é conectado a um conjunto de vetores de entrada, que constitui o conjunto de treinamento do algoritmo.

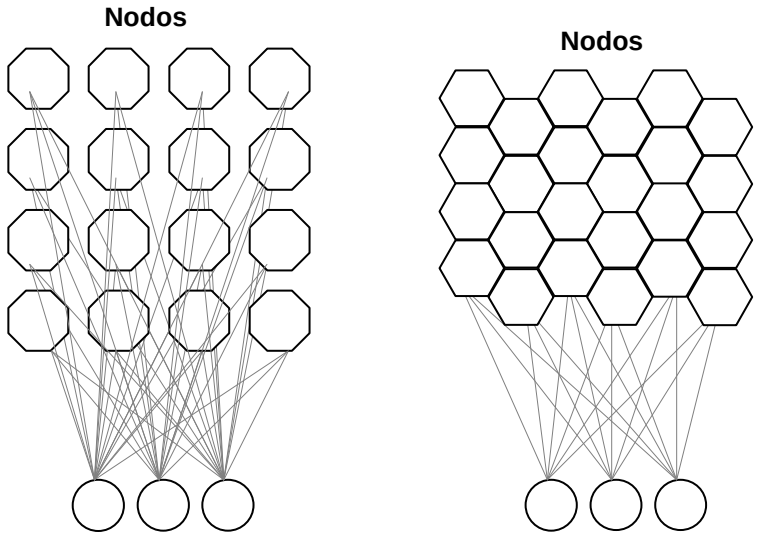

Figura 3: Estrutura de um SOM

Cada nodo possui uma posição topológica específica (uma coordenada x, y no mapa) e contém um vetor de pesos com o mesmo número de dimensões dos vetores de entrada. Assim, cada vetor de entrada é representado na forma $X = (x_1,...,x_n)$ e cada nodo contém um vetor de pesos $W = (w_1,...,w_n)$, sendo n o número de dimensões.

Durante o treinamento, os pesos dos nodos são ajustados de modo a se assemelharem aos vetores de entrada. Este ajuste ocorre de forma iterativa,

formando áreas no mapa que refletem as características dos vetores de entrada. Assim, ao final do treinamento, o SOM constitui um mapeamento dos valores dos dados de entrada. O algoritmo de treinamento do SOM é dado pela seqüência de passos mostrada na Fig. 4, que serão descritos de forma detalhada a seguir.

(1) Inicialize os pesos dos nodos do mapa
(2) **Para** t **de** 1 **até** número_de_iterações **faça**
(3) Selecione aleatoriamente um vetor de entrada do conjunto de treinamento
(4) Verifique qual nodo possui a maior semelhança com o vetor de entrada escolhido. Este nodo será a BMU (Best Matching Unit) do vetor de entrada.
(5) Ajuste os pesos da BMU de modo que fiquem mais próximos aos valores do vetor de entrada. Ajuste também os pesos dos nodos vizinhos (posicionados próximos à BMU no mapa). Quanto mais próximo for o nodo vizinho da BMU, maior será o ajuste.
(6) t ← t + 1

Figura 4: Algoritmo de treinamento do SOM

O primeiro passo é inicializar os pesos dos nodos do mapa. Tipicamente, uma rede neural trabalha com valores normalizados, assim um valor real entre 0 e 1 é atribuído aos pesos de cada nodo. Estes valores podem ser atribuídos de forma aleatória ou de uma forma pré-determinada, como a distribuição gradiente (GERMANO, 1999).

Então se inicia um processo iterativo, onde a cada iteração é escolhido aleatoriamente um vetor de entrada do conjunto de treinamento e a BMU (*Best Matching Unit*) deste vetor é determinada. Para tanto, é calculado o grau de similaridade do vetor de entrada com cada nodo do mapa, utilizando funções de distância entre seus valores. O nodo que tiver a maior semelhança (menor distância resultante) será a BMU.

Uma vez encontrada a BMU, o próximo passo é determinar quais nodos estão na área de vizinhança da BMU no mapa. É uma área circular e, portanto, determinada por um raio. Conforme AI-Junkie (2004), o raio de vizinhança, que diminui a cada iteração, é dado pela função

$$\sigma(t) = \sigma^0 \exp\left(-\frac{t}{\lambda}\right) \tag{9}$$

onde σ^0 é o raio de vizinhança em t=0, λ é uma constante de tempo e t é a iteração atual. O raio inicial σ^0 é determinado pela fórmula abaixo, sendo w e h respectivamente a largura e a altura do mapa (em número de nodos).

$$\sigma^0 = \frac{MAX(w,h)}{2} \tag{10}$$

A constante λ é dependente de σ^0 e do número de iterações r que o algoritmo irá executar, conforme a fórmula

$$\lambda = \frac{r}{\log(\sigma^0)} \tag{11}$$

Todos os nodos presentes na área de vizinhança da BMU, incluindo a própria BMU, têm seus pesos ajustados conforme a função abaixo

$$w_i(t+1) = w_i(t) + \Theta(t)L(t)(w_i(t) - x_i(t)) \quad 1 \le i \le n \tag{12}$$

onde n é o número de pesos, w_i é o i-ésimo peso do nodo a ser ajustado, x_i é o i-ésimo valor do vetor de entrada, t é a iteração atual, L(t) é a taxa de aprendizagem e $\Theta(t)$ é uma função que representa a influência da distância do nodo a ser ajustado até a BMU. Basicamente, quanto mais distante for o nodo da BMU, menor será o valor de $\Theta(t)$. A taxa de aprendizagem L(t), assim como o raio de vizinhança, diminui a cada iteração e é dada por

$$L(t) = L_0 \exp\left(-\frac{t}{\lambda}\right) \tag{13}$$

L_0 é a taxa inicial, definida pelo usuário. O valor de L_0 adotado varia para cada autor da bibliografia. AI-Junkie (2004) adotou L_0 como sendo 0,1. Para Chesnut (2004), L_0 vale 0,9. Finalmente, a função $\Theta(t)$ é dada por

$$\Theta(t) = \exp\left(-\frac{d^2}{2\sigma^2(t)}\right) \tag{14}$$

sendo d a distância do nodo até a BMU e σ o raio de vizinhança calculado pela equação (9). Assim, o ajuste dos pesos marca o fim da iteração.

No algoritmo abordado nesta seção, um vetor de entrada é selecionado aleatoriamente do conjunto de treinamento e mapeado no SOM, a cada iteração. Existe uma variação do algoritmo, descrita por Kohonen (2001), em que numa mesma iteração todos os vetores do conjunto de treinamento são selecionados e mapeados, um de cada vez, determinando suas BMUs e ajustando os pesos dos nodos. As fórmulas e procedimentos utilizados são os mesmos descritos no algoritmo anterior.

Algumas medidas são utilizadas para avaliar a qualidade do mapeamento do SOM. Segundo Vesanto (1997), as mais comuns são a precisão da quantização e a preservação da topologia. A primeira é estimada pela distância média entre os vetores de entrada e suas BMUs correspondentes. A segunda é estimada pelo

número de vetores de entrada cuja BMU e o segundo nodo mais semelhante não estão adjacentes no mapa.

3.4.1 Estendendo o SOM para dados categóricos

O algoritmo de treinamento do SOM, em sua forma convencional, trabalha apenas com valores numéricos, o que acaba limitando o seu uso na mineração de dados, que muitas vezes envolve dados categóricos. De forma a contornar esta limitação, foi proposta por Hsu (2006) uma nova versão do algoritmo, capaz de realizar o treinamento com tais dados. Este algoritmo implementa uma nova maneira de calcular a similaridade entre objetos, baseada em hierarquias de distância. Uma hierarquia de distância é uma árvore composta por nodos contendo valores categóricos, onde os nodos de nível mais alto (mais próximos da raiz) representam valores genéricos enquanto os de nível mais baixo (mais próximos das folhas) representam valores mais específicos. A Fig. 5a ilustra um exemplo de árvore cujos nodos representam nomes e tipos de bebida. Pode-se notar que cada nodo intermediário (não folha) desta árvore possui um valor que generaliza os valores presentes em seus nós filhos. No exemplo, “Refrigerante” generaliza os valores “Coca-Cola” e “Pepsi” enquanto “Café” generaliza os valores “Mocca” e “Nescafé”. Finalmente, o valor “Qualquer” presente na raiz da árvore generaliza todos os valores existentes.

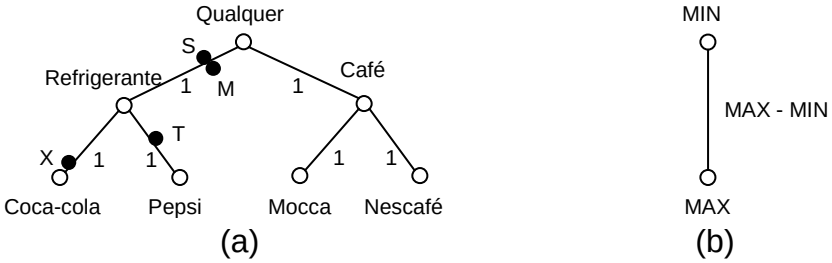

Figura 5: Exemplos de hierarquias de distância
FONTE: Adaptação feita a partir de HSU, 2006.

Cada aresta da árvore possui um peso que representa uma distância. A distância entre duas folhas da árvore é a soma dos pesos das arestas existentes no caminho entre estas folhas. No exemplo da Fig. 5a, todas as arestas possuem um peso igual a 1. Assim, a distância entre as folhas “Coca-cola” e “Pepsi” é igual a 2, enquanto a distância entre “Coca-cola” e “Nescafé” é igual a 4.

É possível representar matematicamente qualquer ponto posicionado em uma aresta ou nodo da árvore. Um ponto X localizado na árvore consiste de duas

partes: uma âncora, dada por N_x, e um valor real positivo denominado deslocamento, dado por d_x, onde a âncora é um nó folha e o deslocamento representa a distância entre X e a raiz da árvore. Assim, o ponto X é representado pelo par ordenado (N_x, d_x). Por exemplo, na Fig. 5a, assume-se M = (Pepsi, 0,3), indicando que M está no caminho entre “Pepsi” e a raiz da árvore e está a 0,3 de distância da raiz. Assim, a âncora de M, dada por N_M, vale “Pepsi” e o deslocamento de M, dado por d_M, vale 0,3.

Um ponto X é o antepassado do ponto Y se X estiver no caminho entre Y e a raiz da árvore. O menor antepassado comum (*least common ancestor*) de dois pontos X e Y, denotado por LCA(X,Y), é definido como o ponto localizado no nodo mais distante da raiz que é antepassado de X e de Y. O menor ponto comum (*least common point*) entre X e Y, denotado por LCP(X,Y), é definido como um dos três casos: 1) qualquer um dos pontos (X ou Y) caso eles sejam equivalentes, isto é, ocupem exatamente a mesma posição na árvore; 2) Y se Y é antepassado de X; 3) LCA(X,Y), caso nenhuma das condições acima seja satisfeita.

No exemplo da Fig. 5a, os pontos S e M são equivalentes e antepassados dos pontos X e T. LCA(X,T) é o ponto localizado no nodo “Refrigerante”. LCP(M,S) pode ser tanto o M quanto o S, uma vez que são equivalentes. LCP(M, X) = M uma vez que M é antepassado de X. LCP(X,T) = LCA(X,T), uma vez que X e T não são equivalentes e nenhum é antepassado do outro.

A distância entre dois pontos quaisquer localizados na árvore pode ser generalizada pela fórmula (15) a seguir. Sejam X = (N_X, d_X) e Y = (N_Y, d_Y) dois pontos da árvore, a distância entre X e Y é definida por

$$|X - Y| = d_X + d_Y - 2d_{LCP(X,Y)} \quad (15)$$

onde $d_{LCP(X,Y)}$ é distância entre a raiz e o menor ponto comum de X e Y.

Por exemplo, na Fig. 5a, assume-se X = (Coca-cola, 2), M = (Pepsi, 0,3), S = (Coca-cola, 0,3) e T = (Pepsi, 1,3). A distância entre S e M é nula, pois S e M são equivalentes. A distância entre T e M é (1,3 + 0,3 – 2 x 0,3) = 1. Uma vez que LCP(X,T) é o nodo “Refrigerante” e $d_{LCP(X,Y)}$ = 1, a distância entre X e T é (2 + 1,3 – 2 x 1) = 1,3.

Para que este conceito de hierarquias de distância, tratado no estudo de Hsu (2006), possa ser utilizado no algoritmo do SOM, é necessário mapear os valores existentes nos nodos do SOM e nos vetores de entrada em pontos na árvore. Para

cada atributo categórico existente nos dados é construída uma árvore contendo todos os valores possíveis deste atributo. Tais valores são colocados nas folhas da árvore (e somente nelas). Nos nodos intermediários (não folha), são atribuídas generalizações dos valores das folhas. Estas generalizações existem apenas na árvore, não fazem parte dos dados originais. A árvore da Fig. 5a foi construída a partir de um atributo denominado “Bebida favorita” de um banco de dados contendo informações sobre hábitos alimentares de estudantes. Cada valor categórico x existente nos vetores de entrada é mapeado em um ponto (N, d) situado numa folha da árvore, de modo que N = x e d é a distância entre a folha e a raiz.

Para um atributo numérico é construída uma árvore (Fig. 5b) contendo apenas uma aresta e dois nodos, sendo estes nodos a raiz e uma folha, representadas respectivamente por MAX (o maior valor existente no atributo) e MIN (o menor valor existente). O peso da aresta é igual ao intervalo entre o maior e o menor valor, isto é, MAX – MIN. Assim, cada valor numérico é mapeado em um ponto (MAX, d) na árvore, onde MAX é a âncora e d é a distância entre o ponto e valor MIN.

Um exemplo prático de mapeamento é apresentado a seguir. Supõe-se um conjunto de dados contendo dois atributos, um categórico especificando a bebida favorita (e utilizando a árvore da Fig. 5a) e um numérico especificando a quantidade diária de bebida ingerida, normalizada em um intervalo de 0.0 a 1.0. A árvore do atributo de quantidade terá então a raiz com valor 0 e uma folha com valor 1. Serão mapeados os vetores de entrada V_1 = (Pepsi, 0,4) e V_2 = (Nescafé, 0,2). O atributo “Bebida favorita” dos vetores V_1 e V_2 será mapeado respectivamente nos pontos (Pepsi, 2) e (Nescafé, 2) da árvore de bebidas, enquanto o atributo “Quantidade” de V_1 e V_2 será mapeado respectivamente nos pontos (1, 0,4) e (1, 0,2) da árvore de quantidades.

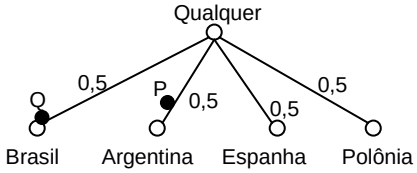

Figura 6: Hierarquia de distância com 2 níveis para simular a distância de Hamming

Segundo Hsu (2006), O conceito de hierarquias de distância é capaz de simular várias outras formas tradicionais de calcular a distância entre dois objetos como, por exemplo, a distância de Hamming. Esta distância, explicada da sub-seção

3.2.2, utiliza a função δ, mostrada na fórmula (4), que retorna 0 se dois valores categóricos são idênticos e retorna 1 caso contrário. Tal função pode ser implementada em uma árvore de dois níveis, como a da Fig. 6, com arestas de pesos iguais a 0,5 e um nó raiz como menor antepassado comum de todos os nós folha. A distância entre duas folhas desta árvore será 0 se elas forem iguais e será 1 caso contrário.

Conforme já explicado durante a descrição do algoritmo de Kohonen, os nodos do SOM são inicializados com valores aleatórios no início da execução do algoritmo. Esta mesma inicialização pode ser feita com as hierarquias de distância, bastando mapear o valor categórico de cada nodo em um ponto aleatório na árvore. Outro procedimento também explicado é o ajuste dos pesos do nodo do SOM de modo a se aproximarem dos valores de um vetor de entrada. Nesta versão do SOM baseada em hierarquias de distância, o ajuste é feito movendo-se o ponto que representa o valor do nodo em direção ao ponto que representa o valor do vetor. A quantidade do deslocamento é igual à distância entre os pontos, da fórmula (15), multiplicada pelo produto das funções L(t) e Θ(t), calculadas pelas fórmulas (13) e (14) respectivamente. Dada a Fig. 6, onde P = (Argentina, 0,4) é um valor de um nodo e Q = (Brasil, 0,5) é um valor de um vetor de entrada. Supõe-se que, em um dado momento, o ponto P seja movido em direção a Q. A seguir são apresentadas algumas situações que podem ocorrer neste deslocamento.

Na primeira situação, a quantidade do deslocamento é igual a 0.3. Movendo P em direção a Q nesta quantidade, o ponto P resultante será P = (Argentina, 0,1), tornando-o mais próximo da raiz, mas mantendo o mesmo valor da âncora.

Na segunda situação, a quantidade do deslocamento é igual a 0.6. Neste caso, o deslocamento de P irá cruzar o menor antepassado comum de P e Q. Como conseqüência, P terá sua âncora alterada para o mesmo valor da âncora de Q. Assim, o novo valor de P será P = (Brasil, 0,2).

Para hierarquias com mais de dois níveis, existem outras situações possíveis, abordadas por Hsu (2006). Entretanto, utilizou-se neste trabalho apenas hierarquias de distância de dois níveis, semelhantes à da Fig. 6. Desta forma, as demais situações estão fora do escopo do trabalho.

4 DEFINIÇÃO DO PROBLEMA

A DRT (Delegacia Regional do Trabalho) do Estado do Rio Grande do Sul possui em seu acervo inúmeras “fichas de qualificação profissional”, conhecidas como “fichas-espelho”, geradas entre 1933 e 1968, com os dados necessários para a feitura das carteiras de trabalho dos trabalhadores gaúchos. Este acervo encontra-se atualmente sob a guarda do Núcleo de Documentação Histórica (NDH) da Universidade Federal de Pelotas, sob responsabilidade da Profª. Beatriz Ana Loner. O acervo da DRT é composto por 1053 caixas do tipo "arquivo morto", mais 53 caixas de papel grande, para os registros em forma de livros. O total é de 627.213 fichas, com dados e fotos individuais.

Este acervo é uma fonte bastante rica, contendo vários dados individuais dos trabalhadores, tais como local de moradia, nível de instrução, profissão e local de origem, além de dados individuais antropométricos (altura, cor, peso, sinais peculiares, cor dos olhos e do cabelo, etc.) e dados culturais e sociais como filiação, estado civil, número de filhos, grau de instrução e outros. Digitados e interpretados, tais dados podem traçar um perfil físico e antropológico do trabalhador gaúcho depois de 1930 e suas modificações ao longo do tempo, servindo também de importante registro sobre empresas gaúchas, algumas delas desaparecidas, sobre o salário real regional por profissão e sua evolução ao longo do tempo.

Desta forma, tornou-se evidente a necessidade de utilização de um sistema computacional para que os dados do acervo da DRT fossem mantidos e explorados de forma mais adequada, por meio digital. Esse sistema seria o responsável por toda gerência, manutenção e consulta dos dados, devendo assim ser efetivo, claro e intuitivo, para livre uso do pessoal integrante do Núcleo de Documentação Histórica da UFPel. Com este sistema, os pesquisadores do núcleo poderiam facilmente acessar os registros dos trabalhadores e realizar cruzamentos com os dados ali contidos de forma a analisar, por exemplo, a evolução da qualificação profissional entre distintas gerações de uma mesma família, a entrada da mulher no mercado de trabalho, as ocupações com maior utilização de mão de obra feminina, entre outras estatísticas históricas. Acrescente-se que esta é uma fonte praticamente inédita e

que não se conhecem estudos em outros estados que tenham se utilizado de fontes assemelhadas, o que faz com que o estado do Rio Grande do Sul seja talvez o primeiro a estabelecer um padrão físico e real para o trabalhador brasileiro no século XX.

A necessidade de observação destas características mostrou que a simples compra de um sistema já pronto não seria o suficiente, pois seriam necessárias diversas modificações no sistema pronto para que se adequasse a todas as exigências. Além disso, novas exigências poderão surgir na medida em que novos cruzamentos dos dados são necessários para gerar novo conhecimento, o que implica no uso de um sistema aberto, que possa ser livremente alterado.

De acordo com estas questões analisadas, constatou-se que a melhor estratégia seria a criação de um sistema voltado exclusivamente para armazenamento das informações do acervo da DRT, desenvolvido de forma a preencher todas as exigências existentes. Este sistema se basearia em um banco de dados operacional, ideal para armazenar o grande volume de dados do acervo (SILBERSCHATZ; KORTH; SUDARSHAN, 1997).

Assim, deu-se início ao Projeto NDH, realizado por pesquisadores e bolsistas dos cursos de Ciência da Computação e História. Durante a primeira fase do projeto, foi realizado o desenvolvimento do sistema. Ainda nesta fase, foi realizada a digitação dos dados referentes à década de 30, contabilizando cerca de 20.000 fichas.

O sistema consiste de um módulo para inserção de dados na base de dados, além de um módulo para alteração e pesquisa de dados no banco de dados criado, permitindo futuras alterações necessárias na forma com que os dados são apresentados aos pesquisadores. A pesquisa e o cruzamento dos dados feitos por este sistema são realizados através de técnicas simples da álgebra relacional, tais como seleções e junções, por meio da linguagem SQL.

Entretanto, tais técnicas não são o suficiente para extrair toda a informação potencial existente nos dados (BRAUNER, 2003). O volume de dados da DRT é grande e aumentará ainda mais à medida que mais dados forem inseridos, tornando cada vez mais difícil a compreensão total destes dados e de enxergar a toda a informação contida neles. Assim, o sistema estaria fadado à situação descrita por Han e Kamber (2006): rica em dados, mas pobre em informação.

Desta forma, estudou-se a possibilidade de realizar o processo de mineração de dados sobre a base de dados da DRT, capaz de extrair esta informação implícita, que poderia ser muito útil para entender a história dos trabalhadores gaúchos.

Assim, deu-se início ao desenvolvimento deste trabalho, uma tentativa de melhorar o acesso aos dados da DRT através da construção de uma ferramenta que implementasse técnicas de mineração de dados e da utilização desta ferramenta em um elaborado processo de mineração a fim de descobrir padrões e encontrar relações nos dados da DRT. Além disso, planejou-se disponibilizar a ferramenta desenvolvida aos pesquisadores do NDH, de modo que estes pudessem utilizá-la na tentativa de encontrar padrões interessantes.

No capítulo seguinte, é explicado todo o processo acima mencionado, desde a análise dos dados e seleção do algoritmo até a execução de cada uma das etapas da mineração.

5 O PROCESSO DE MINERAÇÃO NOS DADOS DA DRT

Para tentar obter conhecimento do banco de dados da DRT, realizou-se um processo de mineração de dados seguindo a seqüência de etapas descrita na seção 2.3. Antes de dar início à mineração, foi realizada uma análise dos dados da DRT a fim de compreender o domínio de sua aplicação, determinar qual classe de algoritmos de mineração de dados (classificação, associação ou agrupamento) seria utilizada, verificar quais campos contidos nestes dados poderiam resultar em padrões úteis e verificar se os valores deveriam passar por alguma limpeza ou alteração.

A partir do estudo das diferentes classes de algoritmos de mineração, constatou-se que o problema da DRT se enquadra perfeitamente no problema do agrupamento, pois os registros dos trabalhadores não possuem uma classe definida - como ocorre no problema da classificação - e não existem campos que possam ser associados ao valor de outros campos - como ocorre no problema da associação. Assim, o agrupamento seria utilizado para construir grupos de registros que possuam determinadas características em comum.

O próximo passo foi observar a estrutura das fichas armazenadas. Cada ficha é composta por 54 campos dentre os quais estão: identificador da ficha (chave primária), número da carteira de trabalho, nome do trabalhador, sexo, altura, cor da pele, filiação, data de nascimento, cidade natal, país de nascimento, grau de instrução, estado civil, residência, profissão, salário, nome do sindicato, nome do estabelecimento (empresa), espécie e cidade do estabelecimento, número de beneficiários, cidade e ano onde a carteira foi solicitada e anotações presentes na ficha.

Estas fichas estão ainda disponíveis em dois modelos: o antigo, utilizado antes de 1944, e o novo, utilizado depois de 1944. Alguns campos, como o nome do sindicato, existem apenas no modelo antigo e outros, como o salário, existem apenas no novo. No momento em que este trabalho foi desenvolvido apenas as fichas do modelo antigo haviam sido digitadas no banco de dados.

Evidentemente, nem todos estes campos mostraram-se potencialmente úteis para a mineração. É pouco provável, por exemplo, obter algum conhecimento a respeito de toda a história dos trabalhadores relacionando o número de sua carteira de trabalho ou sua filiação. Desta forma, foi feita uma pré-seleção de 13 campos que poderiam apresentar alguma utilidade: sexo, altura, cor da pele, ano de nascimento, país de nascimento, estado civil, profissão, sindicato, grau de instrução, nome do estabelecimento, espécie e cidade do estabelecimento e ano de solicitação da carteira. As possíveis relações entre estes campos são explicadas na seção 5.3.

Outro aspecto observado foi a grande quantidade de erros presentes nestes dados. Ausência de nomes ou nomes escritos de forma errada, nomes escritos de forma diferente, mas que possuem o mesmo significado e valores absurdos para datas e medidas de altura são alguns exemplos. Parte destes problemas, segundo os pesquisadores do NDH, deve-se ao fato de muitas fichas não estarem em perfeito estado de conservação. Algumas estavam rasgadas e outras estavam com o texto apagado. Todos estes problemas foram contornados na medida do possível durante a etapa de limpeza dos dados.

Com a análise dos dados feita, o último passo antes de dar início ao processo de mineração foi decidir qual algoritmo de mineração seria utilizado. A partir da análise dos algoritmos de agrupamento disponíveis, um que se mostrou bastante interessante foram os mapas auto-organizáveis (SOMs), graças à sua capacidade de representar dados de três ou mais dimensões em um espaço de apenas duas dimensões. Assim como a maioria dos bancos de dados, o sistema da DRT possui várias dimensões (cada dimensão representando um campo), então esta característica dos SOMs é importante para que os dados da DRT possam ser graficamente representados.

Entretanto, Germano (1999) salienta que esta funcionalidade tem como preço um alto custo computacional. Assim, resolveu-se adotar, além dos SOMs, uma alternativa que pudesse apresentar menor custo computacional, como o algoritmo K-médias. Este algoritmo possui ainda a vantagem de gerar grupos bem definidos, ao contrário dos SOMs, onde muitas vezes não se sabe a qual grupo um determinado objeto pertence. Assim, foi realizado o processo de mineração utilizando ambos os algoritmos de forma a comparar seus resultados e avaliar a relação entre custo (processamento) e “benefício” (qualidade da informação) destes algoritmos. Devido à existência de muitos valores categóricos presentes nos dados, foram utilizadas

variações dos SOMs e K-médias, propostas por Hsu (2006) e Lei, He e Li. (2006) respectivamente, que permitem o uso de tais valores.

Para realizar as etapas de seleção dos dados, transformação dos dados, extração de padrões (onde o algoritmo de mineração é aplicado) e representação do conhecimento foi construída uma ferramenta de mineração cujo funcionamento é explicado no capítulo 6. A solução de descartar fichas cujos valores estejam faltando, utilizada na etapa de limpeza, também foi implementada nesta ferramenta.

Nas seções seguintes é explicado o que foi feito em cada uma das etapas da mineração de dados.

5.1 Limpeza dos dados

Conforme explicado na sub-seção 2.3.1, os principais problemas que podem existir nos dados são os valores faltantes, o ruído e os valores inconsistentes. Nos dados da DRT foram encontrados todos eles. Para contornar o problema dos valores faltantes, optou-se por descartar as fichas que possuíam valores nulos em um de seus campos que tenham sido utilizados na mineração. O descarte é feito automaticamente pela ferramenta de mineração desenvolvida. Tais fichas representam uma pequena parcela sobre o número total de fichas (menos de 5%), então elas puderam ser descartadas sem prejudicar o resultado da mineração.

O ruído era representado por alguns valores numéricos absurdos, como a altura de uma pessoa acima de 5 metros ou um ano de nascimento antes de 1800, provavelmente frutos de erros de digitação. Entretanto, estes defeitos estavam presentes em um número muito pequeno de fichas (menos de 30 fichas dentre um total de 21134), assim não houve necessidade de tentar estimar estes valores. Bastou-se atribuir um valor nulo aos campos com tais defeitos e tratar estes valores como faltantes.

Finalmente, foram analisados os valores inconsistentes, causados por nomes redundantes, isto é, nomes com o mesmo significado, mas escritos de uma forma diferente. Por exemplo, valores referentes à espécie do estabelecimento que podiam significar uma fábrica de tecidos eram “fab. tecidos”, “fab. de tecidos”, “f. tecidos” ou ainda “Fábrica de Tecidos”. Este era o problema que mais afetava o banco de dados e que consumiu mais tempo para ser solucionado, já que cada uma das redundâncias teve que ser manualmente analisada e corrigida. Esta correção

conseguiu reduzir em torno de 30% a quantidade de valores diferentes para profissões, nomes de estabelecimentos, espécies de estabelecimentos e nomes de sindicatos.

5.2 Integração dos dados

Integração é o processo de reunir dados vindos de diferentes fontes (arquivos, bancos de dados, etc.) e integrá-los em uma única base de dados. No caso da DRT, porém, todas as fichas estão armazenadas em uma única fonte. Assim, os dados da DRT já se encontram integrados, não havendo necessidade de realizar nenhum procedimento nesta etapa.

5.3 Seleção dos dados

Na etapa de seleção é feita a escolha dos dados considerados relevantes para a mineração. Realizou-se, então, uma análise para verificar quais campos existentes nas fichas poderiam possuir uma relação interessante entre eles. Desta forma, foram feitas algumas escolhas de campos (dentre os 13 pré-selecionados), descritas abaixo.

a) Profissão – Ano de solicitação da carteira – Sexo: o ano onde a carteira foi solicitada indica o ano onde as condições daquele trabalhador foram registradas. Assim, uma relação entre estes campos poderia revelar quais profissões predominavam em um determinado ano e qual era a profissão predominante de cada sexo.
b) Profissão – Grau de Instrução – Cidade do Estabelecimento: analisar as profissões predominantes nas diferentes cidades e nos diferentes níveis de escolaridade.
c) Sexo – Estado Civil – “É estrangeiro?”: conforme será explicado no capítulo 6, “É estrangeiro” é uma cláusula implementada na ferramenta que atribui “Não” às fichas cujo campo de nacionalidade apresenta o valor “Brasil” e “Sim” caso contrário. Esta relação poderia comparar a proporção de solteiros e casados entre os sexos e comparar esta mesma proporção entre brasileiros e estrangeiros.

d) Nacionalidade – Ano de solicitação da carteira – Altura: comparar as alturas de trabalhadores vindos de diferentes países e como a altura da população variou no decorrer dos anos.
e) Nome do estabelecimento – “Possui Sindicato?”: esta relação utiliza uma outra cláusula implementada na ferramenta denominada “Possui Sindicato?”. O objetivo desta relação é analisar quais empresas possuíam trabalhadores com sindicato e quais não possuíam.

Assim, a etapa de seleção dos dados e – conseqüentemente – as etapas posteriores foram realizadas várias vezes, escolhendo a cada vez um conjunto pequeno e diferente de campos e extraindo padrões que possam revelar uma relação entre estes campos.

5.4 Transformação dos dados

Na etapa de transformação dos dados, algumas operações como suavização, agregação, generalização, normalização e construção de atributos são realizadas nos dados. Na ferramenta de mineração desenvolvida neste trabalho foram implementadas as operações de normalização e construção de atributos. Assim, a etapa de transformação é automatizada pela ferramenta. Basicamente a construção de atributos é utilizada para gerar o campo “Idade” a partir dos campos “Ano de nascimento” e “Ano de solicitação da carteira”.

5.5 Aplicação do algoritmo de mineração de dados

Esta é a etapa onde as fichas, com seus valores já transformados e com os campos selecionados, são utilizadas como dados de entrada dos algoritmos K-médias e SOM, implementados na ferramenta, resultando na geração de padrões. Detalhes de como os dados são utilizados e da interface destes algoritmos encontram-se no capítulo 6.

Conforme explicado na sub-seção 2.3.5, os algoritmos de mineração podem ser utilizados para a descrição ou para a previsão. Neste trabalho, o K-médias e o SOM foram utilizados para a tarefa da descrição, pois se desejava encontrar padrões que caracterizassem de modo geral os dados dos trabalhadores.

5.6 Avaliação dos padrões

Nesta etapa, são utilizadas medidas de interesse para avaliar o quão úteis são os padrões gerados pelos algoritmos. As medidas utilizadas no trabalho foram apenas subjetivas, baseadas no conhecimento e nas expectativas do usuário, pois não houve a utilização de filtros ou funções matemáticas para avaliar os padrões. Para cada conjunto de campos que se tentou relacionar, foi feita uma comparação entre os padrões encontrados pelo SOM e pelo K-médias de modo a verificar qual dos algoritmos apresentou um melhor resultado, no sentido dos padrões serem interessantes e compreensíveis. Os resultados obtidos e suas comparações são apresentados no capítulo 7.

5.7 Representação do conhecimento

Uma vez extraídos os padrões, eles devem ser representados em uma forma visual para que possam ser observados pelo usuário. Tal representação é feita por meio da interface fornecida pela ferramenta. Detalhes de como os padrões são visualizados estão no capítulo 6. Pela definição de Han e Kamber (2006), esta é a última etapa da mineração de dados. Porém, neste trabalho, a representação teve de ser realizada antes da avaliação dos padrões (etapa anterior), pois é necessário que o usuário possa visualizar estes padrões antes de compará-los. Assim, utilizou-se o recurso de iteração (*loop*) entre as duas últimas etapas, de modo que fosse possível voltar à penúltima etapa (avaliação) após a execução da última (representação).

6 FERRAMENTA DESENVOLVIDA

Para realizar a tarefa de mineração nos dados da DRT, foi desenvolvida uma ferramenta que implementa o processo de extração de padrões por meio do algoritmo K-médias e dos mapas auto-organizáveis. Além de realizar a extração e a representação de padrões, a ferramenta ainda automatiza a etapa de transformação dos dados e auxilia o usuário na etapa de seleção dos dados.

A ferramenta foi elaborada especialmente para os dados da DRT, conectando-se diretamente ao banco de dados dos trabalhadores durante sua execução e fornecendo uma interface clara de modo a permitir que os pesquisadores do NDH, mesmo não possuindo conhecimentos de computação, possam utilizar a ferramenta. Na seção 6.3, é apresentado todo o funcionamento desta interface.

6.1 Linguagem e bibliotecas

A ferramenta de mineração foi escrita na linguagem Java, que é a mesma na qual havia sido desenvolvido o sistema de alteração e inserção dos dados do Projeto NDH. Além da vantagem de ser uma linguagem portável (qualquer computador que tenha uma máquina virtual Java instalada poderá utilizar a ferramenta), o Java tem como um de seus ambientes de desenvolvimento o NetBeans, um ambiente gratuito e poderoso que facilita a criação de interfaces gráficas.

O dados da DRT estão armazenados em um servidor MySQL. Assim, foi utilizada uma biblioteca de MySQL para Java que permite ao programa conectar-se ao servidor e extrair estes dados para utilizá-los na mineração.

6.2 Funcionamento interno

A ferramenta desenvolvida é constituída de 16 classes, 6 delas para interface e as demais apenas para processamento de dados. Algumas classes de

interface, como a “FrameSelecao” (a ser descrita mais adiante), também realizam processamento de dados. As classes utilizadas pela interface herdam características de uma classe Java denominada “JFrame”, que permite a construção de janelas com botões, caixas de texto, rótulos, etc. Na Fig. 7 é mostrado um diagrama UML com as classes existentes na ferramenta.

A primeira classe a ser executada no programa é a classe “FramePrincipal”, que apresenta uma janela de *login* pela qual o usuário deve autenticar sua conexão com o banco de dados MySQL. Uma vez autenticada, a conexão é então estabelecida e referenciada por um objeto da classe “Connection”, presente na biblioteca do MySQL. Este objeto é utilizado toda vez que se deseja acessar o banco de dados. Ele é criado na classe “FramePrincipal” e passado para as demais classes da interface.

Uma vez estabelecida a conexão com o banco de dados e criado o objeto “Connection”, a classe “FrameSelecao” é instanciada. Esta classe apresenta uma interface ao usuário para que este possa escolher quais campos do banco de dados serão utilizados na mineração, qual algoritmo (SOM ou K-médias) será executado, e ainda se o usuário deseja realizar uma nova mineração – e gerar um novo mapa – ou apenas visualizar um mapa gerado por um processo de mineração anteriormente realizado. Na seção 6.3 é explicado com mais detalhe esta interface, quais campos estão disponíveis para seleção e quais restrições o usuário pode impor nos dados.

Caso o usuário tenha optado por realizar uma nova mineração, este deverá ainda fornecer ao programa o nome do mapa que será gerado como resultado da mineração. Inicia-se então um processo de preparação dos dados para que estes possam ser utilizados nos algoritmos de mineração. Este processo de preparação, realizado pela mesma classe “FrameSelecao”, depende das escolhas feitas pelo usuário, uma vez que somente os dados desejados serão recuperados do servidor MySQL e a maneira como cada dado será tratado depende do seu tipo (numérico ou categórico).

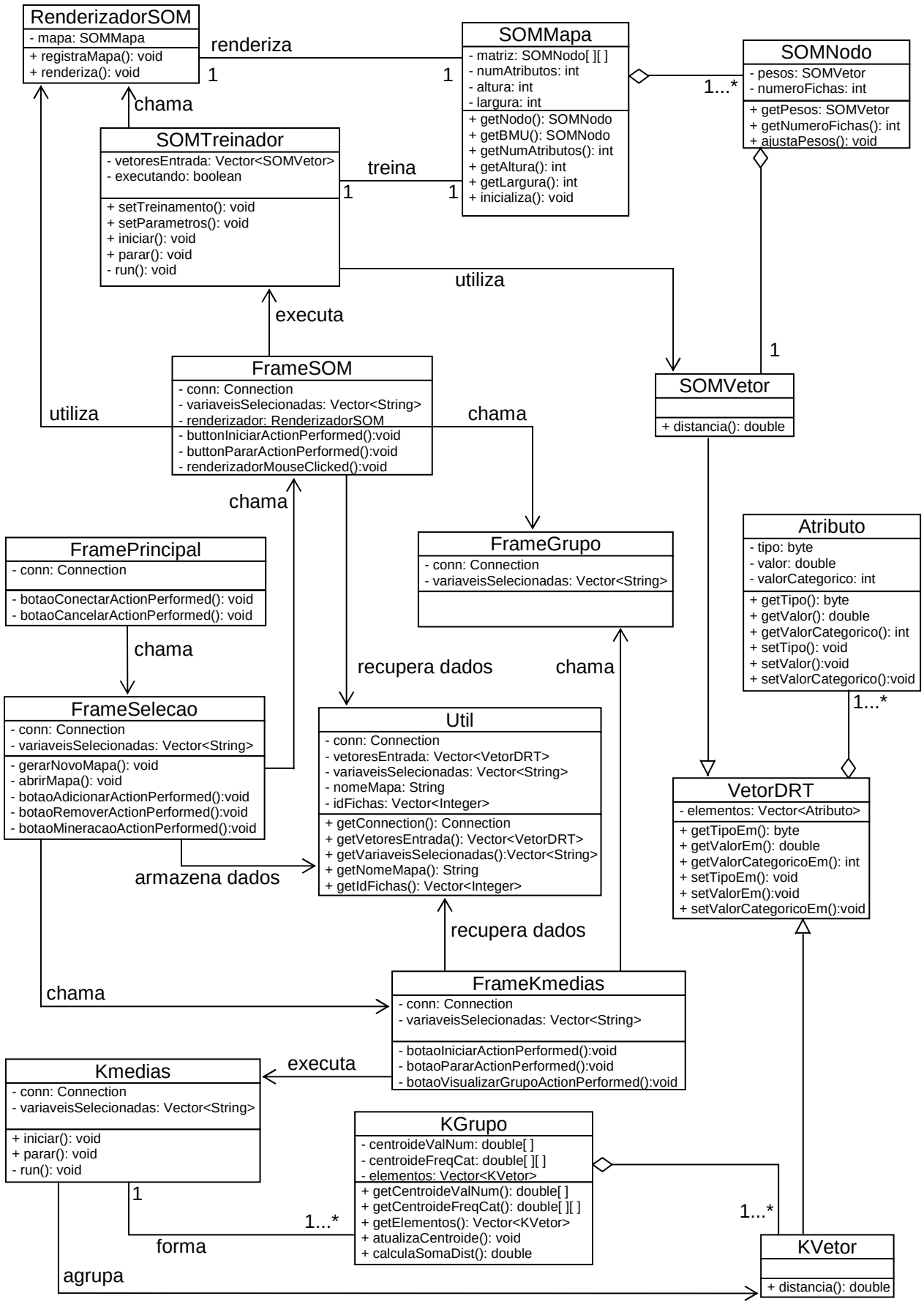

Figura 7: Diagrama de classe da ferramenta

O primeiro passo a ser executado no processo de preparação é uma consulta SQL do tipo “select”, utilizada para obter tuplas de uma tabela do banco de dados. Neste tipo de consulta, pode-se especificar quais condições cada tupla deve satisfazer e quais campos destas tuplas serão recuperados. O banco de dados da DRT é composto por uma única tabela e cada tupla desta tabela representa uma ficha de um trabalhador[1]. Assim, a consulta “select” é realizada para obter as fichas da DRT e recuperar, de cada ficha, os campos selecionados pelo usuário. A fim de gerar melhores resultados na mineração, foi imposta uma condição para obter somente as fichas que possuem um valor definido (não nulo) nos campos selecionados, realizando assim o descarte de fichas com valores nulos, explicado na seção 5.1. As fichas ainda podem ser descartadas de acordo com outras restrições impostas pelo usuário, conforme será explicado na seção 6.3.

Todas as fichas que foram recuperadas (e não foram descartadas) são então armazenadas num objeto da classe “ResultSet”, da biblioteca MySQL. Este objeto contém os dados dos campos selecionados de cada ficha. O próximo passo é realizar a transformação destes dados de acordo com seu tipo. Para dados do tipo numérico é feita a normalização MinMax, da fórmula (5), de modo que fiquem num intervalo entre 0,0 e 1,0. Para dados do tipo categórico (representados na forma de *strings*), é construído um mapa de categorias. Este mapa é um vetor que armazena todos os valores distintos encontrados em um campo e atribui a cada valor um número inteiro. Assim, se um determinado campo possui 10 valores categóricos distintos, estes valores serão armazenados no mapa e numerados de 0 a 9. Então o valor categórico de cada ficha é substituído pelo número correspondente. Isto faz com que os algoritmos SOM e K-médias trabalhem com números inteiros em vez de *strings*, tornando as operações de comparação – utilizadas para dados categóricos – mais rápidas de serem executadas. A Fig. 8 ilustra um exemplo de transformação das fichas.

[1] Na época em que o banco de dados da DRT foi projetado, optou-se por esta solução (de colocar todo o conteúdo das fichas em uma única tabela) depois de um estudo sobre os dados contidos nas fichas e desenvolvimento de uma modelagem mais detalhada para o banco. Dada a heterogeneidade dos dados (fato que se comprovou durante a mineração), a criação de mais tabelas, com apenas um ou dois campos cada, levaria a um excesso de junções e criação de chaves desnecessárias, o que levaria a economia de espaço, mas perda de desempenho. Como o tamanho máximo do banco era conhecido, optou-se pela eficiência e não pelo espaço.

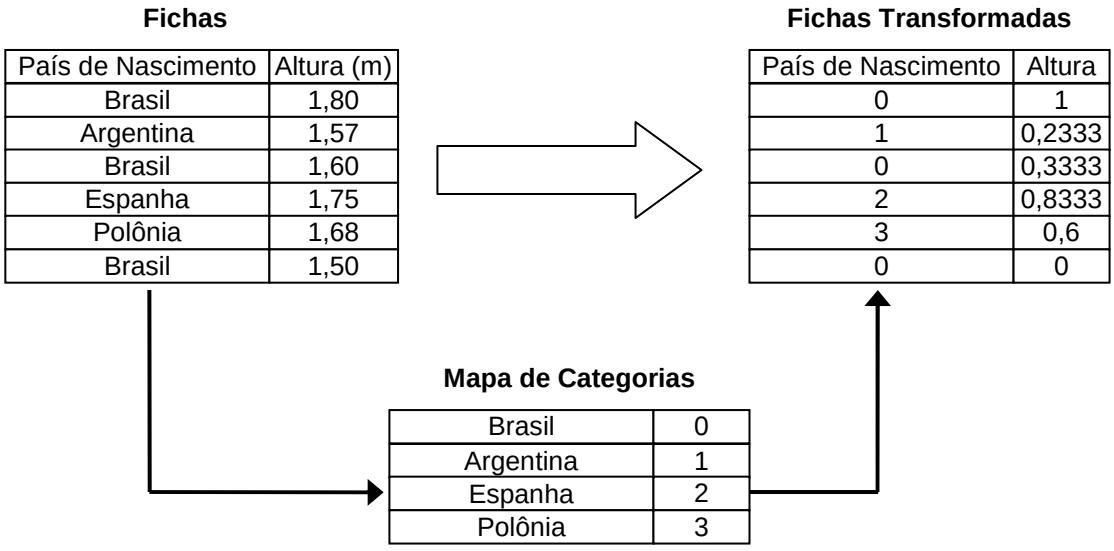

Figura 8: Exemplo de transformação das fichas.
Na normalização deste exemplo, considera-se a altura máxima como sendo 1,80m e a altura mínima como sendo 1,50m.

O último passo do processo de preparação é armazenar as fichas, com seus dados transformados, em vetores de entrada. Estes vetores serão utilizados como conjunto de treinamento (caso seja executado o método SOM) ou como objetos a serem agrupados (caso seja executado o método K-médias). Cada ficha é armazenada em um objeto da classe “VetorDRT”, cujos elementos – que representam os campos da ficha – são objetos da classe “Atributo“. Cada atributo é composto por uma variável que representa seu tipo (numérico ou categórico), uma variável de ponto flutuante para armazenar um valor numérico normalizado e uma variável inteira para armazenar um valor categórico. No caso do método SOM, em que os dados são mapeados em pontos nas hierarquias de distância, a variável inteira é utilizada para armazenar o valor da âncora, enquanto a variável de ponto flutuante é utilizada para armazenar o valor do deslocamento.

Terminada a preparação dos dados, será executada a classe “FrameSOM” ou “FrameKMeans”, dependendo de qual algoritmo foi escolhido pelo usuário. Nas sub-seções seguintes será explicado o processamento destes algoritmos.

6.2.1 Processamento do método SOM

Após a realização da preparação dos dados, é iniciada a execução das classes referentes ao método SOM, caso este tenha sido escolhido. Os vetores da classe “VetorDRT” são convertidos em objetos da classe “SOMVetor”. Esta classe é

herdeira da classe “VetorDRT” e possui a mesma estrutura de dados, com a adição de operações específicas utilizadas no algoritmo de treinamento do SOM, como a função que calcula a distância entre o vetor e um nodo do mapa. Todos os dados que serão úteis ao algoritmo, tais como os vetores de entrada, nome e tipo dos campos selecionados e o objeto “Connection”, são armazenados em um objeto da classe “Util”.

Neste ponto, é iniciada a execução da classe “FrameSOM”, que recebe o objeto “Util” como parâmetro e deste objeto extrai os dados armazenados. Esta classe apresenta a interface do SOM, com botões para iniciar e parar a execução do algoritmo e um painel onde os nodos do SOM são projetados na tela. A projeção é feita através da classe “RenderizadorSOM”. Ao receber o comando do usuário para iniciar a execução do algoritmo, o programa invoca a classe “SOMTreinador” que recebe como parâmetros (fornecidos pelo usuário) o número de iterações e a taxa de aprendizagem inicial. É nesta classe que está implementado o algoritmo de treinamento do SOM apresentado na seção 3.4 deste trabalho. Foi implementada a variação do algoritmo que utiliza todos os vetores de entrada em cada iteração. Optou-se por utilizar esta forma de seleção dos vetores, pois a mesma garante que os dados de todas as fichas sejam utilizados e tenham igual influência no treinamento do SOM. As funções de cálculo de distância e ajuste dos pesos dos nodos são baseadas nas hierarquias de distância, abordadas na sub-seção 3.4.1. Para atributos categóricos foram utilizadas hierarquias de dois níveis, semelhantes à da Fig. 6 ilustrada nesta mesma sub-seção.

O mapa a ser treinado possui uma forma quadrada, conforme mostrado na Fig. 3a da seção 3.4, e é um objeto da classe “SOMMapa”, que contém uma matriz 20 x 20 de objetos da classe “SOMNodo”. O “SOMMapa” possui ainda um método para encontrar a BMU de um vetor de entrada e um método para inicializar os valores de todos os nodos. Cada objeto “SOMNodo”, por sua vez, é composto por um vetor de pesos da classe “SOMVetor”, um método para ajustar o valor de seus pesos (conforme o esquema de deslocamento de pontos, explicado em 3.4.1) e uma variável que conta o número de fichas inseridas no nodo. Conforme já explicado, o conjunto de treinamento utilizado no algoritmo é o mesmo conjunto de vetores de entrada (cada vetor corresponde a uma ficha da DRT) gerados pelo processo de preparação dos dados da classe “FrameSelecao” e convertidos em objetos da classe “SOMVetor”.

Fornecidos os parâmetros e o conjunto de treinamento, o algoritmo do SOM é então executado utilizando as classes acima citadas. Depois de concluído o treinamento do mapa, o algoritmo começa a inserir cada ficha do conjunto de entrada no nodo que corresponde à sua BMU. Esta inserção, realizada pela classe “SOMTreinador”, é feita através de comandos SQL do tipo “create table” e “insert”, com os quais é gerada uma tabela no banco de dados onde são armazenados em cada tupla os identificadores da ficha e do nodo onde ela foi inserida. Esta tabela recebe o nome do mapa fornecido pelo usuário no início do programa. O usuário pode visualizar quais fichas foram inseridas em um determinado nodo, através da classe “FrameGrupo”, que realiza consultas nesta tabela e apresenta o resultado em uma janela. Tais consultas utilizam também os dados da tabela da DRT, de modo a recuperar os dados das fichas na sua forma original, antes de serem transformados.

A tabela com os dados dos nodos e das fichas inseridas neles é mantida no banco de dados mesmo depois do encerramento do programa, de modo que o usuário possa consultar o resultado da mineração novamente utilizando a opção de revisitar um mapa gerado, disponível na interface da classe “FrameSelecao”. Após a inserção das fichas, a classe “RenderizadorSOM” é invocada para pintar os nodos do mapa em um tom de cinza de acordo com o número de fichas inseridas neles. Quanto mais fichas forem inseridas, mais escuro é o tom de cinza. Finalmente, é calculada a precisão da quantização, mostrada na seção 3.4, cujo valor é apresentado ao usuário por meio da interface.

6.2.2 Processamento do método K-médias

Caso o usuário tenha optado pelo método K-médias, a execução das classes relativas a este método é iniciada. Os vetores de entrada são convertidos em objetos da classe “KVetor” que, assim como o “SOMVetor”, herda a estrutura de dados do objeto “VetorDRT”, mas contém uma função exclusiva, utilizada no algoritmo K-médias e apresentada na fórmula (8), que calcula a distância entre um objeto e um centróide. Os vetores de entrada, o objeto “Connection” e o nome dos campos selecionados são armazenados em um objeto da classe “Util” que, conforme explicado em 6.2.1, serve para transportar dados essenciais para as classes que executam os algoritmos.

O programa inicia a execução da classe “FrameKmedias”, que recebe o objeto “Util” como parâmetro e extrai os dados contidos nele. A classe

“FrameKmedias” apresenta uma interface ao usuário para que este possa fornecer o número de grupos a serem formados e executar comandos para iniciar e parar a execução do algoritmo. Uma vez dado o comando para iniciar a execução, a classe “KMedias” é instanciada. Esta classe implementa o algoritmo apresentado na seção 3.3 deste trabalho e o método que o executa recebe como parâmetro o número de grupos a serem formados. Cada grupo formado pelo algoritmo é um objeto da classe “KGrupo”, que armazena e calcula os dados do centróide e guarda informações de todos os objetos que pertencem a aquele grupo. Conforme explicado anteriormente, os objetos a serem agrupados são os vetores de entrada gerados pelo processo de preparação dos dados da classe “FrameSelecao” e convertidos em objetos da classe “KVetor”.

Assim, é iniciada a execução do algoritmo K-médias, utilizando as classes acima citadas. Depois de terminada a execução, é construída uma tabela no banco de dados onde cada tupla é composta pelo identificador da ficha e do grupo onde ela foi inserida. Esta construção é realizada pela classe “KMedias”. Assim como a tabela criada pelo método SOM, esta tabela recebe o nome do mapa fornecido pelo usuário no início do programa e é mantida no banco de dados depois do encerramento do programa, permitindo que o usuário visualize os grupos formados sempre que desejar, através da opção de abrir um mapa existente. Esta tabela também é consultada pela classe “FrameGrupo”, que apresenta uma janela listando as fichas que foram inseridas em um determinado grupo. Finalmente, é calculado o valor do somatório das diferenças entre cada ficha e o centróide de seu respectivo grupo, da fórmula (6), e apresentado ao usuário por meio da interface.

6.3 Funcionamento da interface

Nesta seção, é explicado como ocorre a interação do usuário com a ferramenta. Logo no início da execução, é apresentada uma janela de *login* onde o usuário deve autenticar sua conexão com o servidor MySQL (Fig. 9). Uma vez estabelecida a conexão, é mostrada a próxima janela.

Figura 9: Janela de *login*

Figura 10: Janela de opções e seleção dos campos

Na segunda janela (Fig. 10) é apresentada uma série de opções ao usuário para controlar o processo de mineração de dados. Primeiramente, tem-se o painel "Mapeamento", no qual o usuário tem a opção de gerar um novo mapa - e assim realizar uma nova mineração - ou abrir um mapa anteriormente gerado. Os mapas gerados pela mineração de dados são armazenados no próprio servidor do banco de dados. Desta forma, o usuário poderá revê-los através da opção de abrir um mapa. O nome do mapa a ser gerado deve ser especificado no campo "Nome".

Ao optar por gerar um novo mapa, o usuário poderá escolher no painel "Algoritmo de mineração" qual método (SOM ou K-médias) será utilizado na mineração.

O próximo - e maior - passo é escolher no painel "Seleção de variáveis" quais dos campos presentes nas fichas dos trabalhadores serão utilizados na mineração. Aqui é onde a etapa de seleção dos dados é executada. Os campos disponíveis para escolha são aqueles que foram pré-selecionados na análise feita antes do início da mineração. Assim, somente os campos potencialmente úteis estão disponíveis: sexo, altura, cor, ano de nascimento, país de nascimento, estado civil, profissão, sindicato, grau de instrução, estabelecimento, espécie do estabelecimento, cidade do estabelecimento e ano de solicitação da carteira.

No caso de campos categóricos, como o nome do estabelecimento, o usuário pode ainda definir qual o número mínimo de fichas que devem possuir um determinado valor para que este seja considerado. Por exemplo, empresas (estabelecimentos) que tenham apenas um ou dois trabalhadores registrados no banco de dados podem ser vistas como irrelevantes pelo usuário. Assim, se o número mínimo de fichas for ajustado para 10, fichas com nomes de empresas que tenham menos de 10 trabalhadores serão descartadas.

Além dos campos mencionados acima, o programa ainda oferece as cláusulas "É estrangeiro?" e "Possui sindicato?", que podem interessar a determinados usuários. Por exemplo, a cláusula "Possui sindicato?" atribui "Sim" às fichas que possuam algum valor no campo sindicato e "Não" às fichas que possuam o campo do sindicato igual a nulo (neste caso elas não serão descartadas). Finalmente, o programa oferece para escolha o campo derivado "Idade" que é obtido subtraindo-se o ano de solicitação da carteira pelo ano de nascimento do trabalhador.

A partir do estudo dos diferentes tipos de agrupamento mostrados no capítulo 3, pode-se observar que esta funcionalidade de seleção dos campos segue alguns princípios do agrupamento baseado em restrições - pois permite ao usuário impor restrições aos dados que serão utilizados - e do agrupamento para dados de muitas dimensões - pois o descarte dos campos considerados irrelevantes reduz a dimensionalidade (número total de atributos) dos dados, permitindo um melhor resultado na mineração.

Depois de terminada a seleção dos campos desejados, o usuário poderá clicar no botão "Iniciar Mineração de Dados". Após o clique, o programa realiza um processo de preparação dos dados, onde os mesmos são recuperados do servidor, transformados e então armazenados em vetores de entrada, conforme explicado na seção 6.2. Após a realização do processo de preparação dos dados, é apresentada uma interface referente ao método de mineração utilizado. A interface de cada método será explicada nas sub-seções seguintes.

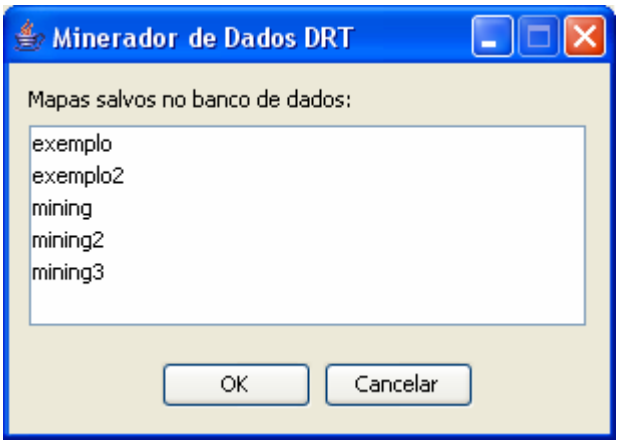

Figura 11: Janela dos mapas armazenados

Caso o usuário tenha optado por abrir um mapa já pronto, aparecerá uma janela onde estão listados os nomes dos mapas armazenados (Fig. 11). Para selecionar o mapa desejado, basta clicar em seu nome e em seguida clicar em "OK".

6.3.1 Interface do SOM

Caso tenha sido escolhido o método SOM, será mostrada uma janela como a da Fig. 12. Na parte superior desta janela encontram-se alguns campos que podem ser utilizados para ajustar os parâmetros do algoritmo - a taxa de aprendizagem inicial e o número de iterações. Estes parâmetros têm como valor padrão 0,1 e 10 respectivamente, com os quais é possível realizar um bom mapeamento. Assim, não há necessidade do usuário ajustar estes parâmetros caso não saiba o que eles significam.

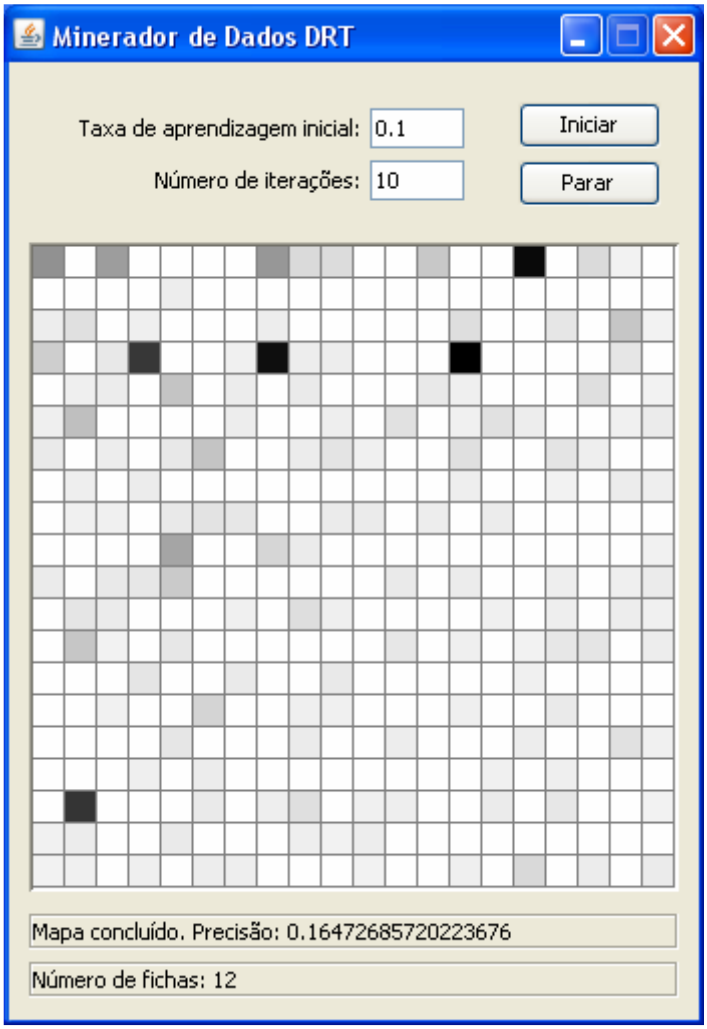

Figura 12: Janela do algoritmo SOM

Uma vez definidos os parâmetros, o usuário poderá dar início ao treinamento clicando no botão "Iniciar". O treinamento poderá ser abortado a qualquer momento clicando-se no botão "Parar".

No centro da janela está localizado um grande painel onde os nodos do SOM estão projetados na forma de pequenos quadrados. Nesta projeção, cada nodo é pintado de um tom de cinza baseado no número de fichas de trabalhadores inseridas naquele nodo. Quanto maior for o número de fichas, mais escuro será o tom de cinza do nodo.

Logo abaixo do painel de representação dos nodos são encontrados dois rótulos. O primeiro informa o status da execução do treinamento. Assim, o usuário saberá que o mapa está completamente treinado quando aparecer a mensagem "Mapa concluído". Neste mesmo rótulo é mostrada a precisão da quantização, uma medida de qualidade do mapa gerado. O segundo rótulo informa o número de fichas inseridas no nodo que está sendo apontado pelo cursor do mouse.

Finalmente, o usuário poderá clicar em um dos nodos e visualizar todas as fichas que foram inseridas naquele nodo. A visualização é feita por meio da janela representada na Fig. 13.

Figura 13: Janela de listagem das fichas

Todas as propriedades do mapa (topologia, número de acertos, fichas inseridas) são salvas automaticamente no banco de dados. Assim, o usuário poderá rever estas propriedades revisitando o mapa com a opção "Abrir um mapa existente" disponível no programa.

6.3.2 Interface do K-médias

A interface do algoritmo K-médias é apresentada na janela da Fig. 14. Na parte superior há um campo onde é definido o número de grupos que o usuário deseja formar. Após definir este número, o usuário poderá clicar no botão "Iniciar" para dar início à formação dos grupos e no botão "Parar" caso deseje interromper o processo.

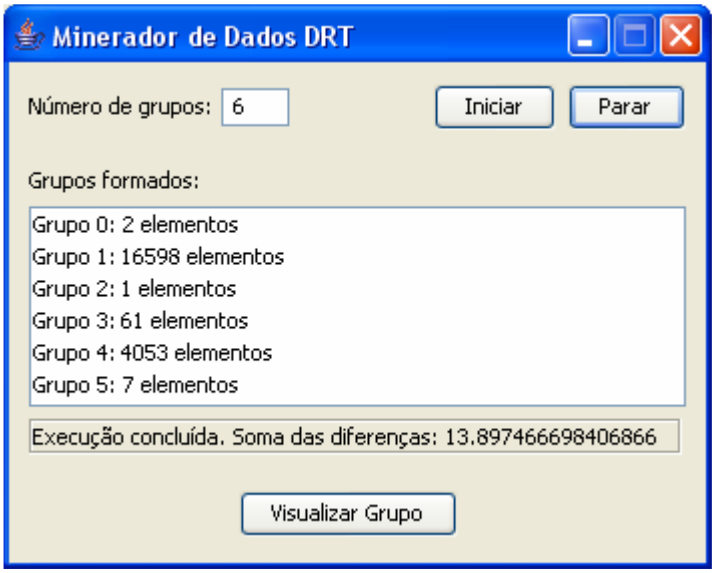

Figura 14: Janela do algoritmo K-médias

Assim que a execução do algoritmo terminar, os grupos formados serão mostrados no painel localizado no centro da janela. Cada grupo é representado por um índice e pela quantidade de fichas presentes naquele grupo. Estas fichas podem ser visualizadas clicando-se no botão "Visualizar grupo", sendo mostrada uma janela como na Fig. 13 (a mesma utilizada para ver as fichas inseridas nos nodos do SOM).

Finalmente, há um rótulo abaixo do painel que informa o status da execução do algoritmo, com a mensagem "Execução concluída" indicando que a formação dos grupos terminou. Neste mesmo rótulo aparece o valor do somatório da diferenças entre cada objeto e o centróide de seu respectivo grupo.

Se os grupos formados não se mostrarem úteis ao usuário, este poderá clicar no botão "Iniciar" e tentar um novo agrupamento. Neste caso, os grupo anteriormente formados serão descartados. Ao sair do programa, o último agrupamento realizado será mantido no banco de dados e poderá ser revisto com a opção "Abrir um mapa existente" disponível no programa.

7 RESULTADOS OBTIDOS

Depois de desenvolvida a ferramenta (descrita no capítulo 6) e realizado o processo de mineração (explicado no capítulo 5) em diferentes conjuntos de dados, foram obtidos alguns resultados a serem mostrados neste capítulo. Para cada seleção de campos, descrita na seção 5.3, foi realizada a extração de padrões através dos métodos SOM e K-médias. A fase de testes realizada neste trabalho consistiu em comparar, para cada conjunto de dados, os resultados obtidos por estes métodos.

Em todos os testes realizados pelo SOM, adotou-se a taxa de aprendizagem inicial como sendo 0,1 e o número de iterações como sendo 10. No método K-médias, o número de grupos a serem formados variou para cada teste. De modo geral, para agrupamentos envolvendo campos com muitos valores categóricos diferentes foi adotado um número grande de grupos, de forma a tornar os grupos mais homogêneos possíveis. Para testes envolvendo apenas valores categóricos com pouca variação, foi adotado um número menor de grupos.

As variáveis categóricas “Profissão” e “Estabelecimento” apresentavam uma quantidade muito elevada de nomes distintos (em torno de 1300 e 4500 respectivamente), muitos deles presentes em apenas uma ou duas fichas. Para estes campos, foi adotado um filtro para que somente os valores presentes em 10 ou mais fichas fossem minerados de forma a diminuir os tempo de execução dos algoritmos e tornar os resultados mais compreensíveis.

Como esperado, o K-médias foi executado em um tempo menor que o SOM. Porém, esta diferença foi pouco significativa para um número grande de grupos (em torno de 50). Além disso, os grupos formados pelo K-Médias não possuem uma representação gráfica – é feita apenas uma lista dos mesmos, conforme mostrado na Fig. 14. Como conseqüência, os resultados obtidos pelo K-médias foram mais difíceis de interpretar que os do SOM.

7.1 Primeiro teste

O primeiro teste foi realizado com os dados dos campos “Profissão”, “Ano de solicitação da carteira” e “Sexo” presentes na fichas da DRT. Ao executar o método SOM com esta seleção de campos, foi obtido o mapeamento ilustrado na Fig. 15 abaixo.

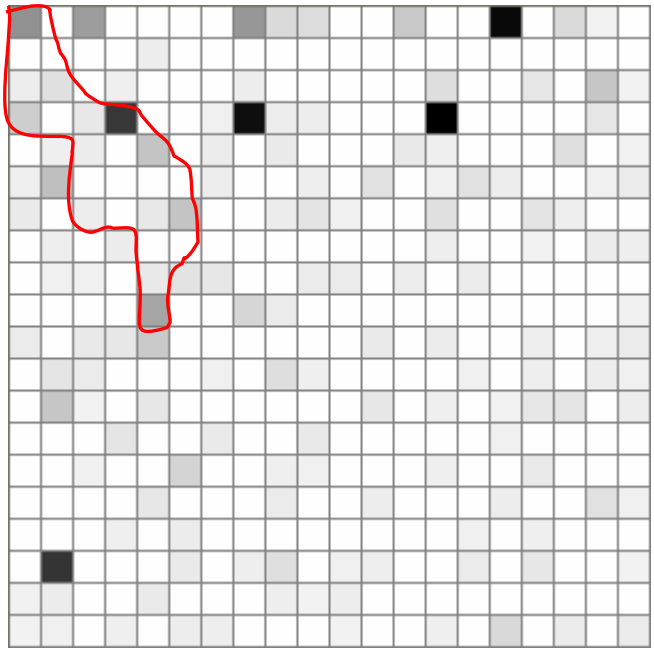

Figura 15: SOM resultante da primeira seleção de campos

Na área da figura destacada pelo contorno vermelho estão as fichas de trabalhadores do sexo feminino. Fora desta área estão os trabalhadores do sexo masculino. De modo geral, cada nodo do SOM agrupou fichas de trabalhadores de uma determinada profissão. As profissões menos abrangentes (que possuíam poucos trabalhadores) ficaram agrupadas em um mesmo nodo. Os nodos mais escuros (e, portanto, com mais fichas) representados no mapa foram os que agruparam mais de uma profissão. O campo “Profissão” possui uma grande quantidade de valores diferentes (mesmo utilizando o filtro explicado anteriormente) o que fez com que cada profissão ficasse em apenas um nodo do mapa ou nem mesmo tivesse um nodo próprio.

Na área das fichas do sexo feminino foram encontradas, em maioria, fichas com a profissão de servente, costureira, cozinheira, fiandeira e doméstica, entre outras. Estas seriam as profissões predominantes do sexo feminino na década de 1930. Percebe-se também que o número de mulheres com carteira de trabalho era um tanto menor que o de homens. Não foi possível obter qualquer conhecimento a

respeito dos valores referentes ao ano de solicitação da carteira. Estes valores nem mesmo tiveram influência na formação do mapa. A possível causa da irrelevância de tais valores seria o fato do banco de dados, em seu estado atual, possuir apenas fichas digitadas na década de 1930. Desta forma, as fichas apresentavam pouca diferença em relação ao ano de solicitação.

Esta mesma relação foi então testada com o método K-médias. Devido à grande quantidade de profissões diferentes, optou-se por executar o algoritmo de modo a formar 50 grupos. Como resultado, os grupos formados tiveram um comportamento semelhante aos nodos do SOM, isto é, em cada grupo formado haviam fichas de uma determinada profissão e as profissões menos abrangentes ficaram em um mesmo grupo. Entretanto, ao contrário do que ocorre no SOM, os grupos do K-médias são apenas listados, não são posicionados em um mapa de forma a facilitar sua compreensão. É mais difícil, por exemplo, procurar apenas por fichas de trabalhadores do sexo feminino, pois os grupos contendo tais fichas não são posicionados próximos, como ocorreu no SOM.

7.2 Segundo teste

Para o segundo teste, foram escolhidos os campos “Profissão”, “Grau de Instrução” e “Cidade do Estabelecimento”. O mapa resultante do método SOM é mostrado na Fig. 16 abaixo.

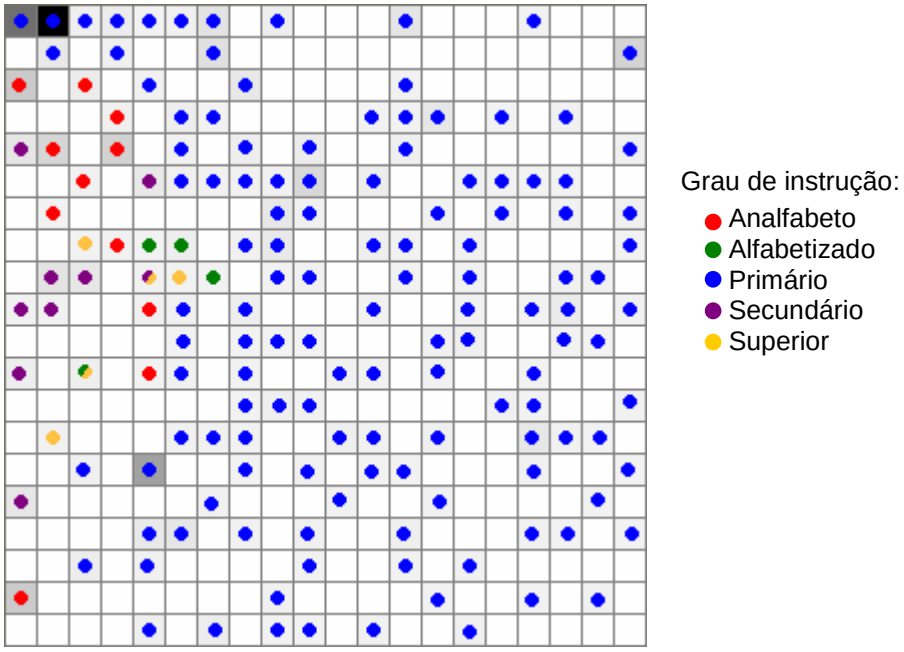

Figura 16: SOM resultante da segunda seleção de campos

A fim de facilitar a demonstração do resultado, foram desenhados pontos coloridos nos nodos do mapa, através de um programa de edição de imagens, de acordo com o valor do campo “Grau de Instrução” presente nas fichas de cada nodo. Como se pode observar, a maioria das fichas são de trabalhadores com nível de educação primário. Nas fichas de trabalhadores com ensino superior foram encontradas, em sua maioria, as profissões de jornalista, engenheiro agrônomo, ferroviário e comerciante. De modo geral, cada nodo agrupou trabalhadores de uma determinada profissão e cidade. Porém, não houve um agrupamento que revelasse com clareza quais profissões predominavam em cada cidade, pois, da mesma forma que nas profissões, havia muitos nomes diferentes de cidades, o que gerou um agrupamento fragmentado destes valores. Assim como ocorreu no teste anterior, profissões pouco abrangentes foram agrupadas em um mesmo nodo. O mesmo ocorreu com os nomes de cidades.

Para testar esta mesma seleção de campos com o método K-médias, optou-se novamente por gerar 50 grupos. Em cada grupo gerado como resultado, as fichas possuíam semelhança em um determinado atributo. Por exemplo, alguns grupos possuíam fichas com uma determinada profissão em comum, em outros havia fichas com a mesma cidade. Outros ainda tinham fichas com dois campos em comum. Grupos com fichas que possuíam a mesma profissão e grau de instrução foram o tipo mais encontrado. Da mesma forma que ocorreu com os sexos do primeiro teste, o fato dos grupos serem apenas listados (não posicionados), dificultou a procura por fichas com um determinado grau de instrução. Com o K-médias também não houve uma relação clara entre profissões e cidades.

7.3 Terceiro teste

No terceiro teste foram utilizados os campos “Sexo” e “Estado Civil” e a cláusula “É estrangeiro?”. São variáveis categóricas que apresentam poucos valores diferentes: dois (“m” e “f”) para o sexo, quatro (“solteiro”, “casado”, “separado” e “viúvo”) para o estado civil e dois (“Sim” e ”Não”) para a cláusula “É estrangeiro?”. A execução do algoritmo do SOM resultou no mapa da Fig. 17 abaixo.

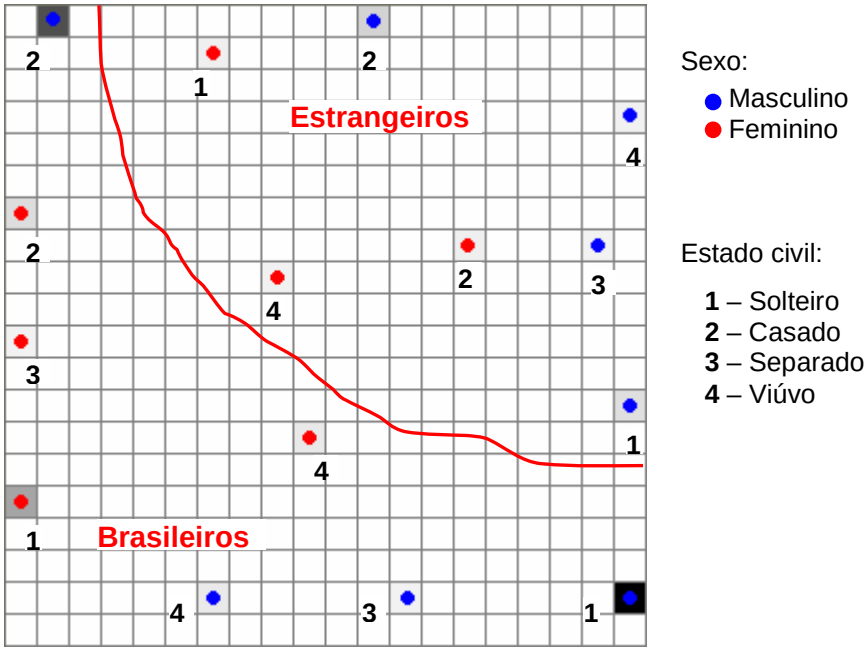

Figura 17: SOM resultante da terceira seleção de campos

Assim como nos testes anteriores, cada nodo pintado do mapa resultante agrupou fichas com uma determinada combinação de valores. Todavia, o número de nodos pintados no mapa foi um tanto menor (um total de 15), uma vez que foram utilizadas variáveis bastante simples. Esta seleção de variáveis resultou em um agrupamento bem definido, com fichas de brasileiros distribuídas na parte inferior esquerda do mapa e as fichas de estrangeiros na parte superior direita. O nodo que armazenou o maior número de fichas foi o nodo contendo brasileiros solteiros do sexo masculino, localizado no canto inferior direito do mapa, seguido do nodo com brasileiros casados no sexo masculino, no canto superior esquerdo. O terceiro nodo com mais fichas foi o com brasileiros solteiros do sexo feminino, no canto inferior esquerdo.

No método K-médias, optou-se por gerar 15 grupos, uma vez que os campos utilizados neste teste apresentavam poucos valores diferentes. Os grupos gerados apresentaram exatamente o mesmo comportamento dos 15 nodos pintados gerados pelo método SOM, ou seja, cada grupo armazenou fichas com uma determinada combinação de valores (homens solteiros brasileiros em um grupo, homens casados brasileiros em outro grupo e assim por diante). Para esta seleção de variáveis, tanto o SOM quanto o K-médias realizaram um bom agrupamento.

7.4 Quarto teste

Os campos selecionados no quarto teste foram: “Nacionalidade”, “Ano de solicitação da carteira” e “Altura”. A Fig. 18 abaixo ilustra o SOM resultante.

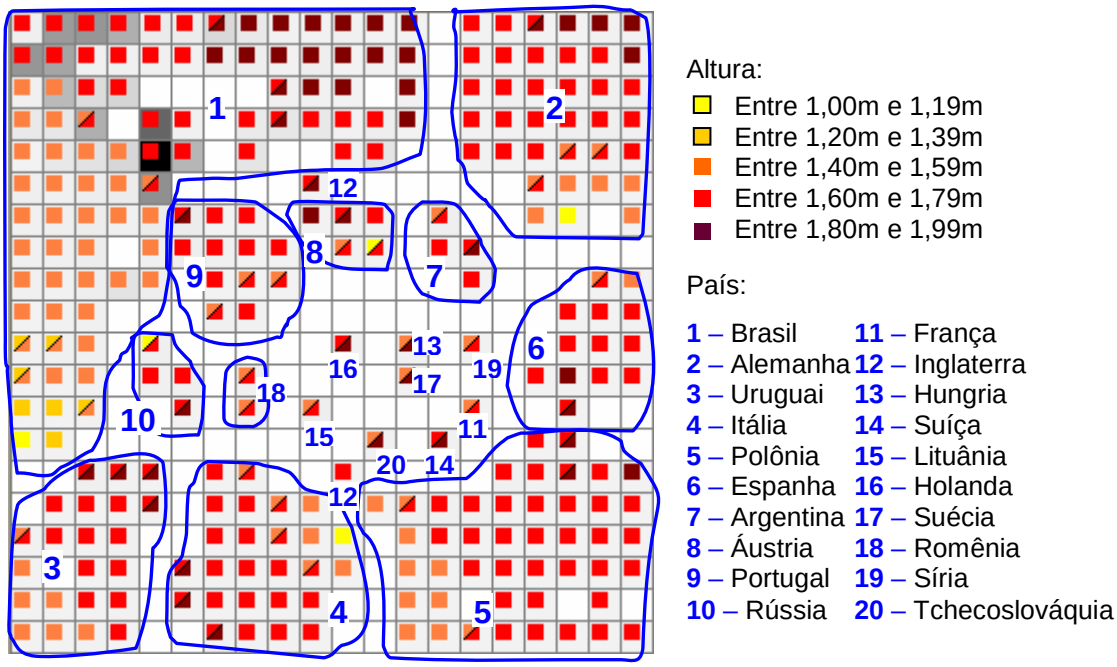

Figura 18: SOM resultante da quarta seleção de campos

As regiões contornadas na figura delimitam os dados dos trabalhadores de cada país. Utilizando um programa de edição de imagens foi desenhado em cada nodo um pequeno quadrado cuja cor indica o intervalo dos valores da altura das fichas encontradas naquele nodo. Por exemplo, o quadrado vermelho indica que as fichas existentes no nodo possuem o valor da altura entre 1,60m e 1,79m. Na região com os dados dos trabalhadores brasileiros, percebe-se uma proporção semelhante entre as áreas de cor laranja, vermelha e marrom. Por outro lado, nas regiões da maioria dos outros países predomina a cor vermelha. É possível concluir que entre os imigrantes a altura apresenta uma menor variação, estando na maioria das vezes no intervalo entre 1,60m e 1,79m, enquanto que entre os brasileiros há uma maior variação da altura, com muitos trabalhadores no intervalo entre 1,40m e 1,59m ou no intervalo entre 1,80m e 1,99m. Assim como ocorreu no primeiro teste (da seção 7.1), os valores do ano de solicitação da carteira não foram relevantes no resultado, uma vez que as fichas apresentavam pouca diferença nestes valores.

Na execução do K-médias foram gerados 20 grupos. Em alguns grupos foram encontradas apenas fichas de um mesmo país. Em outros, havia dois nomes de países diferentes. Um caso especial foram as fichas de trabalhadores brasileiros,

que apareceram em 5 grupos. Estes grupos se diferenciavam basicamente pelo intervalo dos valores da altura encontrados neles: em um dos grupos a altura ficava entre 1,60m e 1,70m, em outro, ficava entre 1,71m e 1,85m e assim por diante. Entretanto, com este agrupamento ficou mais difícil obter algum conhecimento a respeito da altura dos trabalhadores, como foi obtido com o SOM. Novamente, este apresentou um resultado melhor que o K-médias, graças à sua capacidade de representar graficamente o agrupamento.

7.5 Quinto teste

Para o último teste foram selecionados o campo “Estabelecimento” e a cláusula “Possui Sindicato?”. Os dados destas variáveis foram agrupados pelo SOM conforme mostrado na Fig. 19.

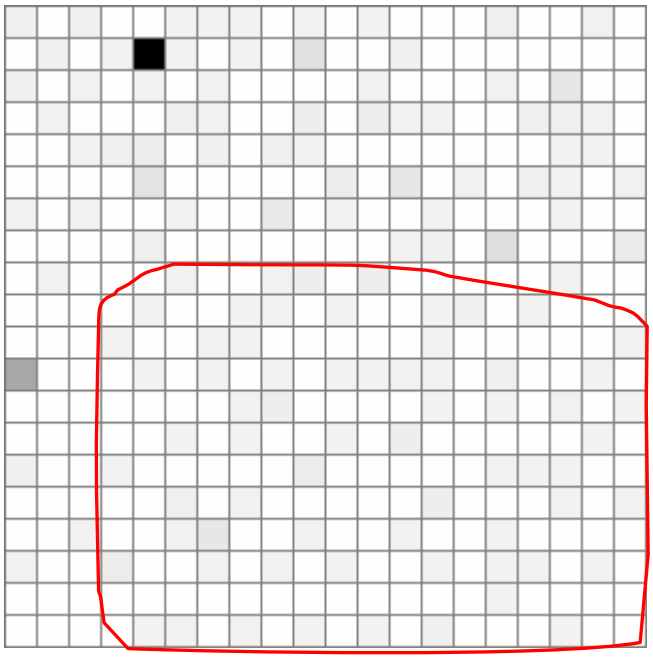

Figura 19: SOM resultante da quinta seleção de campos

Esta última relação é bastante simples, mostrando quais empresas possuem trabalhadores com sindicato e quais não possuem. Os dados de trabalhadores com sindicato estão dentro da área contornada e cada nodo agrupou trabalhadores de uma determinada empresa. As empresas com menos trabalhadores foram agrupadas no nodo preto localizado na parte superior do mapa. Algumas empresas apareceram em apenas uma das regiões (com ou sem sindicato), enquanto outras apareceram em ambas regiões. Por exemplo, uma empresa denominada “A. J. Renner” possuía trabalhadores com e sem sindicato,

enquanto nas Prefeituras existiam apenas trabalhadores sem sindicato e o Banco Nacional do Comércio continha apenas empregados com sindicato. De fato, nomes de bancos eram mais fáceis de serem encontrados na região de trabalhadores com sindicato.

No teste realizado com o método K-médias, foram formados 50 grupos. De forma semelhante aos nodos do SOM, cada grupo agregava fichas com um determinado nome de empresa. Assim como no resultado obtido pelo SOM, foi possível verificar quais empresas empregavam pessoas com sindicato e quais não empregavam.

7.6 Comparação dos resultados

Em cada um dos testes realizados foram obtidos os valores da precisão da quantização e da soma das diferenças, calculados respectivamente pelo algoritmo do SOM e pelo K-médias. Quanto menor o valor retornado por estes cálculos, maior a qualidade do agrupamento, no sentido de maior semelhança entre dados de um mesmo nodo (no caso do SOM) ou grupo (no caso do K-médias). Na tab. 1 abaixo são apresentados os valores obtidos em cada teste. Cabe ressaltar que nesta tabela é feita uma comparação apenas entre os testes, não é feita uma comparação entre o SOM e o K-médias em um mesmo teste.

Tabela 1: Comparativos das medidas de qualidade

Teste	Precisão (SOM)	Soma das diferenças (K-médias)
1	0,0961456161278477	6513,13812763979
2	0,16472685720223676	12920,1194535933
3	0,00001842599003872	0,0
4	0,00175491678149062	203,170684187289
5	0,12465998569804827	7316,77682067178

Tanto para SOM quanto para o K-médias, o agrupamento de maior qualidade foi o obtido no terceiro teste, enquanto o de pior qualidade foi o do segundo teste. De modo geral, os algoritmos apresentaram um resultado melhor quando foram utilizadas variáveis categóricas com poucos valores diferentes, como sexo e estado civil, ou variáveis numéricas, como altura. A existência de campos categóricos com muitos valores, como “Profissão” e “Cidade do estabelecimento”, resultou em um agrupamento de menor qualidade.

8 CONCLUSÃO

A principal contribuição deste trabalho foi fornecer um exemplo prático do uso da mineração de dados, podendo servir como referência para possíveis trabalhos futuros realizados na área da mineração que, segundo Han e Kamber (2006), é uma área ainda jovem, e ainda há muita pesquisa a ser realizada na mesma.

O trabalho consistiu no desenvolvimento de uma ferramenta que implementa técnicas de agrupamento e no uso desta ferramenta em um processo de mineração de dados de forma a extrair padrões do acervo da DRT.

Os resultados obtidos demonstraram a importância de um meio eficiente de visualizar os padrões extraídos em um processo de mineração. O método K-médias carece de uma forma melhor de visualização dos dados, fazendo com que seja difícil obter conhecimento a partir dos grupos formados por este método. Os mapas auto-organizáveis (SOMs) mostraram-se melhores neste quesito, fornecendo um meio de visualizar dados de três ou mais dimensões em um espaço de duas dimensões. A sua capacidade de visualização compensa o seu custo de processamento.

Este trabalho também demonstrou, tanto para o K-médias quanto para o SOM, que a existência de variáveis categóricas com muitos valores diferentes dificulta o agrupamento, comprometendo a qualidade dos grupos formados e dificultando a interpretação destes grupos por parte do usuário. No banco de dados da DRT, por exemplo, havia milhares de nomes de empresas, milhares de profissões e dezenas de nomes de cidades.

A principal função dos algoritmos de agrupamento é organizar um conjunto de dados de modo que dados semelhantes fiquem em um mesmo grupo ou posicionados próximos uns dos outros. Os testes realizados com variáveis mais simples – sem os problemas descritos no parágrafo anterior – demonstraram que os algoritmos implementados na ferramenta conseguem de fato realizar o referido agrupamento. Assim, a ferramenta desenvolvida funcionou de forma satisfatória. A mesma ficará à disposição dos pesquisadores do NDH, de modo que estes possam dar seguimento à tentativa de extrair conhecimento útil dos dados da DRT.

8.1 Trabalhos futuros

Como sugestão de trabalho futuro, destaca-se a busca por novos algoritmos de mineração que possam apresentar resultados melhores que os algoritmos utilizados neste trabalho. Conforme demonstrado no capítulo 7, algumas imagens resultantes do método SOM foram editadas de forma a melhorar a apresentação dos resultados. Esta edição poderia ser realizada de forma automática pela ferramenta, baseada nos valores encontrados nas fichas de cada nodo, o que facilitaria a visualização dos dados.

Finalmente, o processo de mineração poderá ser realizado novamente nos dados da DRT quando mais fichas forem inseridas no banco de dados. No momento em que este trabalho foi feito, apenas as fichas da década de 1930 estavam presentes. Quando as fichas das décadas de 1940, 1950 e 1960 forem digitadas, o banco da DRT terá uma quantidade maior de informação a ser extraída através da mineração de dados.

REFERÊNCIAS

AGGARWAL, C. C.; PROCOPIUC, C.; WOLF, J.; YU, P. S.; PARK, J.-S. Fast algorithms for projected clustering. In: Proc. 1999 ACM-SIGMOD Int. Conf. Management of Data, Philadelphia. **Anais do...** Philadelphia: 1999. p.61-72.

AGRAWAL, R.; GEHRKE, J.; GUNOPULOS, D.; RAGHAVAN, P. Automatic subspace clustering of high dimensional data for data mining applications. In: Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data, Seattle. **Anais do...** Seattle: 1998. p.94-105.

AI-JUNKIE - Kohonen's Self Organizing Feature Maps. 2004. Disponível em: <http://www.ai-junkie.com/ann/som/som1.html> Acesso em: 25 jul. 2008.

BERKHIN, Pavel. **Survey of clustering data mining techniques**. Technical report, Accrue Software, San Jose, CA, 2002.

BRAUNER, Daniela F. **O Processo de Descoberta de Conhecimento em Banco de Dados:** Um Estudo de Caso Sobre os Dados da UFPel. 2003. Monografia de Conclusão de Curso (Bacharelado em Ciência da Computação) - Instituto de Física e Matemática, Universidade Federal de Pelotas, Pelotas.

CHAPMAN, Pete; CLINTON, J.; KERBER, R.; KHABAZA, T.; REINARTZ, T.; SHARER, C.; WIRTH, R. CRIPS-DM 1.0 Step-by-step data mining guide. 2000. Disponível em: <http://www.crisp-dm.org> Acesso em: 7 ago. 2008.

CHESNUT, Casey. Self Organizing Map AI for Pictures. 2004. Disponível em: <http://www.generation5.org/content/2004/aiSomPic.asp> Acesso em: 27 jul. 2008.

CHUNG, H. Michael; GRAY, Paul. **Special Section: Data Minig**. Journal of Management Information Systems, v.16, n.1, p.11-16, 1999.

DEMPSTER, A.; LAIRD, N.; RUBIN, D. **Maximum likelihood from incomplete data via the EM algorithm**. J. Royal Statistical Society, n.39, p.1-38, 1977.

ERTOZ, L.; STEINBACH, M.; KUMAR, V. **Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data**, Technical Report, 2002.

ESTER, M.; KRIEGEL, H.-P; SANDER, J.; XU, X. A density-based algorithm for discovering clusters in large spatial databases. In: Proc. 1996 Int. Conf. Knowledge Discovery and Data Mining, Portland. **Anais do...** Portland: 1996. p.226-231.

FAYYAD, Usama M.; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. **From Data Mining to Knowledge Discovery**: An Overview. Advances. In Knowledge Discovery and Data Mining. Menlo Park: AAAI Press: 1996. p.11-34.

FRAWLEY, W. J.; PIATETSKY-SHAPIRO, G.; MATHEUS, C. J. **Knowledge discovery in databases: An overview**. AI Magazine v.13, n.3, p.57-70, 1992.

GANTI, V.; GEHRKE, J. E.; RAMAKRISHNAN, R. CACTUS - clustering categorical data using summaries. In: Proc. 1999 Int. Conf. Knowledge Discovery and Data Mining, San Diego. **Anais do...** San Diego: 1999. p.73-83.

GERMANO, Tom. Self-Organizing Maps. 1999. Disponível em: <http://davis.wpi.edu/~matt/courses/soms/index.html> Acesso em: 30 jul. 2008.

GUHA, S.; RASTOGI, R.; SHIM, K. Cure: An efficient clustering algorithm for large databases. In: Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data, Seattle. **Anais do...** Seattle: 1998. p.73-84.

GUHA, S.; RASTOGI, R.; SHIM, K. ROCK: A robust clustering algorithm for categorical attributes. In: Proc. 1999 Int. Conf. Data Engineering, Sydney. **Anais do...** Sydney : 1999. p.512-521.

HAN, J.; KAMBER, M. **Data Mining: Concepts and Techniques**. 2 ed. Morgan Kaufmann, 2006. 772p.

HINNEBURG, A.; KEIM D. A. An efficient approach to clustering in large multimedia databases with noise. In: Proc. 1998 Int. Conf. Knowledge Discovery and Data Mining, New York. **Anais do...** New York: 1998. p.58–65.

HOLSHEIMER, Marcel; SIEBES, Arno. **The Search for Knowlege in Databases**. Technical report CS-R9406, CWI, Países baixos, 1994.

HSU, Chung-Chian. Generalizing **Self-Organizing Map for Categorical Data**. IEEE Transactions on Neural Networks, v.17, n.2, p.294-304, 2006.

KARYPIS, G.; HAN, E.-H.; KUMAR, V. **CHAMELEON: A hierarchical clustering algorithm using dynamic modeling**. COMPUTER, n.32, p.68-75, 1999.

KOHONEN, T. **Self-Organizing Maps**. 3 ed. Springer. 2001. 260p.

KROGEL, Mark-André. A Data Mining Case Study. 2000. Disponível em: <http://citeseer.ist.psu.edu/459861.html> Acesso em: 11 ago. 2008.

LEI, Ming; HE, Pilian; LI, Zhichao. **An Improved K-means Algorithm for Clustering Categorical Data**. Journal of Communication and Computer v.13, n.3, p. 20-24, ISSN 1548-7709, EUA, 2006.

LOURENÇO, Fernando; LOBO, Victor; BAÇÃO, Fernando. Binary-based similarity measures for categorical data and their application in self-organizing maps. In: JOCLAD 2004 - XI Jornadas de Classificação e Análise de Dados. **Anais do...** Lisbon: 2004.

LUCAS, Joel P. **Mineração de Dados Apoiada pela Descoberta de Subgrupos Através do Pós-Processamento de Regras de Associação**. 2006. Monografia de Conclusão de Curso (Bacharelado em Ciência da Computação) - Instituto de Física e Matemática, Universidade Federal de Pelotas, Pelotas.

MACQUEEN, J. Some methods for classification and analysis of multivariate observations. In: Proc. 5th Berkeley Symp. Math. Statist. Prob., 1967. **Anais do...** 1967. p.281-297.

QUINTALES, Luis A. M. **Métodos de Agrupamiento en Minería de Dados**. 2007. 41 slides, p&b.

REZENDE, S. O.; PUGLIESE, J. B.; MELANDA, E. A.; PAULA, M. F. Mineração de Dados. In: **Sistemas Inteligentes: Fundamentos e Aplicações**. v.1. Barueri: Editora Manole, 2003. p. 307-335.

SEIFERT, Jeffrey W. **Data Mining: an overview**. Congressional Research Service (CRS) Report RL31798. Washington, EUA, 2005.

SHEIKHOLESLAMI, G.; CHATTERJEE, S.; ZHANG, A. WaveCluster: A multi-resolution clustering approach for very large spatial databases. In: Proc. 1998 Int. Conf. Very Large Data Bases, New York. **Anais do...** New York: 1998. p.428-439.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Database System Concepts**. McGrawHill, 1997.

TEKNOMO, Kardi. K-Means Clustering Tutorial. 2006a. Disponível em: <http://people.revoledu.com/kardi/tutorial/kMean/> Acesso em: 7 set. 2008.

TEKNOMO, Kardi. Similarity Measurement. 2006b. Disponível em: <http://people.revoledu.com/kardi/tutorial/Similarity/> Acesso em: 6 set. 2008.

VESANTO, Juha. **Data Mining Techniques Based on the Self-Organizing Map**. 1997. M.S. Thesis (Science in Engineering) - Helsinki University of Technology, Finland.

WANG, H.; WANG, W.; YANG, J.; YU, P. S. Clustering by pattern similarity in large data sets. In Proc. 2002 ACM-SIGMOD Int. Conf. Management of Data, Madison. **Anais do...** Madison: 2002. p.418-427.

WANG, W.; YANG, J.; MUNTZ, R. STING: A statistical information grid approach to spatial data mining. In: Proc. 1997 Int. Conf. Very Large Data Bases, Athens. **Anais do...** Athens: 1997. p.186-195.

WITTEN, Ian H.; FRANK, Eibe. **Data Mining: Practical Machine Learning Tools and Techniques**. 2 ed. Morgan Kaufmann, 2005. 558p.