UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

Pré-Processamento no Processo de Descoberta de Conhecimento em Banco de Dados

por

RITA DE CÁSSIA DAVID DAS NEVES

Dissertação submetida à avaliação,
como requisito parcial para a obtenção do
grau de Mestre em Ciência da Computação

Prof. Dr. Luis Otávio Campos Alvares
Orientador

Porto Alegre, abril de 2003.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Neves, Rita de Cássia David das
Pré-Processamento no Processo de Descoberta de Conhecimento em Banco de Dados / por Rita de Cássia David das Neves. – Porto Alegre: PPGC da UFRGS, 2003.
137 p. : il.

Dissertação (Mestrado) – Universidade Federal do Rio Grande do Sul. Instituto de Informática. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2003. Orientador: Alvares, Luis Otávio Campos.

1. Inteligência Artificial. 2. Descoberta de Conhecimento em Banco de Dados. 3. Mineração de Dados. 4. Pré-Processamento. I. Alvares, Luis Otávio Campos. II. Título.

A minha querida mãe, Rita do Carmo,
com muito amor.

Agradecimentos

A Deus pela minha existência e por todas as bençãos concedidas à mesma;

A minha mãe Rita do Carmo David das Neves, pela compreensão, dedicação, cuidados com a Lady, bem como pelo apoio sentimental e financeiro para o cumprimento de mais esta etapa acadêmica da minha vida;

Aos meus irmãos Moisés, Isac, Débora e Daniel pelo incentivo e crença na realização dos meus objetivos, ressaltando um agradecimento especial ao Moisés por todos os favores a mim prestados, enquanto estive aqui em Porto Alegre;

A minha amiga-irmã Claudinha pelo carinho e amparo nos momentos mais difíceis e também por acreditar e torcer pelo meu sucesso;

Ao meu orientador Prof. Dr. Luis Otávio Campos Alvares, por ser tão acessível, compreensivo, dedicado e sensato em sua orientação para a concretização deste trabalho;

Ao Prof. Dr. Paulo Martins Engel por ter me concedido a oportunidade de participar do Projeto "Desenvolvimento de Metodologia para Extração de Conhecimento de Bases de Dados da Saúde do Estado para Avaliação e Planejamento", assim como também por todo o auxílio e elucidação de dúvidas;

Aos funcionários e contratados da SES do RS em especial a Ana Cecília, Mariazinha, Luis Fernando Domingues e Sodenir pela receptividade e esclarecimentos de fundamental importância para o desenvolvimento dos experimentos deste trabalho;

À coordenação do MINTERCC pela efetivação e organização do andamento deste mestrado;

A todos os professores e funcionários do Instituto de Informática da UFRGS pelo suporte oferecido ao longo do curso deste mestrado;

Aos professores do departamento de informática da UFPA, em especial aos mestres Mariane, Alfredo e Arnaldo Prado por incentivarem meu ingresso na pós-graduação;

A PROPESP da UFPA pela liberação de passagens aéreas no ano de 2002;

Aos colegas do MINTERCC, em especial a Olinda, Miriam, Délcio e Iraçú por todas as dificuldades, ansiedades e preocupações compartilhadas;

Aos colegas gaúchos em especial a Anelise, Carine, João, Sílvio, Fabrício, Luis Henrique e Daniela por toda ajuda e companheirismo;

Às Irmãs da Residência Universitária pela acolhida e torcida para a conclusão deste mestrado durante os meses de minha vivência em Porto Alegre;

Aos meus amigos que sempre desejaram boa sorte e

Finalmente, a todos que direta ou indiretamente contribuiram para a realização deste trabalho.

Sumário

Lista de Abreviaturas

DCBD	Descoberta de Conhecimento em Banco de Dados
CRISP-DM	Cross – Industry Standard Process for Data Mining
KDD	Knowledge Discovery in Databases
SGBD	Sistema de Gerenciamento de Banco de Dados
AIH	Autorização de Internação Hospitalar
SES	Secretaria Estadual de Saúde
UFRGS	Universidade Federal do Rio Grande do Sul
UCS	Universidade de Caxias do Sul
RS	Rio Grande do Sul
I-MIN	Intension Mining
MD	Mineração de Dados
DTM	Decision Tree Method
B & B	Branch and Bound
MDLM	Minimum Description Length Method
MDLC	Minimum Description Length Criterion
POE + ACC	Probability of Error & Average Correlation Coefficient
ACC	Average Correlation Coefficient
LVF	Las Vegas Filter
LVI	Las Vegas Incremental
SBUD	Sequential Backward for Unsupervised Data
AMB & B	Approximate Monotonic Branch and Bound
BS	Beam Search
WSFG	Wrapper Sequential Forward Generation
WSBG	Wrapper Sequential Backward Generation
SBS-SLASH	Sequential Backward Selection-SLASH
BDS	Bi-Directional Search
PQSS	(p,q)Sequential Search
LOOCV	Leave-One-Out Cross Validation
RC	Relevance in Context
RMHC-PF1	Random Mutation Hill Climbing-Prototype and Feature Selection
GA	Genetic Algorithm
SA	Simulated Annealing
LVW	Las Vegas Wrapper
FSSEM	Feature Subset Selection and EM clustering
EM	Expectation-Maximization
ELSA	Evolutionary Local Search Algorithm
BBHFS	Boosting-Based Hybrid Feature Selection
SNM	Sorted Neighbourhood Method
DE – SNM	Duplication Elimination SNM
MVC	Missing Values Completion
RAR	Robust Association Rules
MDLP	Minimum Description Length Principle

Lista de Figuras

Lista de Tabelas

Resumo

A Descoberta de Conhecimento em Banco de Dados (DCBD) é uma nova área de pesquisa que envolve o processo de extração de conhecimento útil implícito em grandes bases de dados. Existem várias metodologias para a realização de um processo de DCBD cuja essência consiste basicamente nas fases de entendimento do domínio do problema, pré-processamento, mineração de dados e pós-processamento. Na literatura sobre o assunto existem muitos trabalhos a respeito de mineração de dados, porém pouco se encontra sobre o processo de pré-processamento.

Assim, o objetivo deste trabalho consiste no estudo do pré-processamento, já que é a fase que consome a maior parte do tempo e esforço de todo o processo de DCBD pois envolve operações de entendimento, seleção, limpeza e transformação de dados. Muitas vezes, essas operações precisam ser repetidas de modo a aprimorar a qualidade dos dados e, conseqüentemente, melhorar também a acurácia e eficiência do processo de mineração.

A estrutura do trabalho abrange cinco capítulos. Inicialmente, apresenta-se a introdução e motivação para trabalho, juntamente com os objetivos e a metodologia utilizada. No segundo capítulo são abordadas metodologias para o processo de DCBD destacando-se CRISP-DM e a proposta por Fayyad, Piatetsky-Shapiro e Smyth. No terceiro capítulo são apresentadas as sub-fases da fase de pré-processamento contemplando-se entendimento, seleção, limpeza e transformação de dados, bem como os principais métodos e técnicas relacionados às mesmas. Já no quarto capítulo são descritos os experimentos realizados sobre uma base de dados real. Finalmente, no quinto capítulo são apresentadas as considerações finais sobre pré-processamento no processo de DCBD, apontando as dificuldades encontradas na prática, contribuições do presente trabalho e pretensões da continuidade do mesmo.

Considera-se como principais contribuições deste trabalho a apresentação de métodos e técnicas de pré-processamento existentes, a comprovação da importância da interatividade com o especialista do domínio ao longo de todo o processo de DCBD, mas principalmente nas tomadas de decisões da fase de pré-processamento, bem como as sugestões de como realizar um pré-processamento sobre uma base de dados real.

Palavras – chaves: Descoberta de Conhecimento em Banco de Dados, Mineração de Dados, Pré-Processamento.

TITLE: PRE-PROCESSING IN KNOWLEDGE DISCOVERY IN DATABASES PROCESS

Abstract

Knowledge Discovery in Databases (KDD) is a new research area involving the process of implicit useful knowledge extraction in big databases. There are several methodologies for the accomplishment of a KDD process whose essence consists basically of understanding phases of the problem domain, pre-processing, data mining and post-processing. Literature in this topic contains many works on data mining, however, little is found about pre-processing process.

Thus, the aim of this work consists of studying pre-processing, this phase that undertakes the most of the time and effort of the overall KDD process because it involves operations of understanding, selecting, cleaning and transforming data. Sometimes these operations must be repeated in order to refine data quality and, consequently, improving accuracy and efficacy of the data mining process as well.

The present work is divided into five chapters. Initially there is the introduction and motivation, as well as objectives and methodology. The second chapter approaches methodologies for the KDD process, highlighting CRISP-DM and the one proposed by Fayyad, Piatetsky-Shapiro and Smyth. The third chapter presents pre-processing sub-phases comprehending data understanding, selection, cleaning and transformation, as well as the main methods and techniques related to them. Chapter four describes experiences carried out on a real database. Finally, the fifth chapter brings final considerations about pre-processing in KDD process, pointing problems found in practice, the contributions of the present work and plans of further studies.

The main contributions of this work are meant to be the presentation of existing pre-processing methods and techniques, confirming the importance of interactivity with the domain expert all over the KDD process, mainly during the decision-taking phase of pre-processing, and suggestions of how to accomplish pre-processing on a real database.

Keywords: Knowledge Discovery in Databases, Data Mining, Pre-processing.

1 Introdução

Os constantes avanços tecnológicos, permitem a facilidade encontrada nos últimos anos em armazenar grande quantidade de dados. Porém, mesmo com os recursos oferecidos pelos SGBDs para armazenar e recuperar informações de grandes bases de dados, ainda existe a dificuldade de interpretação e compreensão por parte das pessoas de um volume muito grande de dados.

Para tratar este problema, uma nova linha de pesquisa denominada Descoberta de Conhecimento em Banco de Dados (DCBD) ou *Knowledge Discovery in Databases (KDD)* vem a ser considerada comprometendo-se em extrair conhecimento útil implícito em grandes bases de dados a partir da utilização de técnicas de mineração de dados e ferramentas apropriadas para tal.

O surgimento da área de Descoberta de Conhecimento consta em torno de 1981, quando a *Technical University Berlin* desenvolveu o sistema METAXA, a partir do objetivo de adquirir regras de inferência para uma linguagem natural através da modelagem baseada em aprendizagem. Este sistema fazia parte do projeto LERNER, relacionado ao conceito de Aprendizagem de Máquina. Nesse projeto foi desenvolvida a metodologia *Balanced Cooperative Modeling* (Modelagem de Balanceamento Cooperativo), fundamental para a construção de um novo tipo de sistema que integrava a aquisição de conhecimento a diferentes algoritmos de aprendizagem (MORIK et al., 1993).

Sendo que, foram atribuídos vários nomes ao processo de encontrar padrões úteis em bases de dados, como Extração de Conhecimento, Descoberta de Informação, Mineração de Dados e Processamento do Padrão de Dados. Somente em 1989, o termo Descoberta de Conhecimento em Banco de Dados foi criado por Piatetsky-Shapiro durante o primeiro *workshop* de descoberta de conhecimento (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

Após este *workshop*, ocorreu a Primeira Conferência Internacional sobre Descoberta de Conhecimento em Banco de Dados e Mineração de Dados realizada em Montreal, Canadá, em 1995, em que a expressão DCBD foi definida como sendo todo o processo para obtenção de conhecimento em bases de dados, e Mineração de Dados correspondendo a uma das fases deste processo, na qual são aplicadas várias técnicas de inteligência artificial (FÉLIX, 1998). Desta forma, tais definições serão adotadas neste trabalho.

A Descoberta de Conhecimento em Banco de Dados apresenta caráter multidisciplinar, já que envolve a participação de outras áreas de pesquisa como Aprendizagem de Máquina, Estatística, Banco de Dados, Sistemas Inteligentes e Visualização de Dados, como visualizado na Figura 1.1.

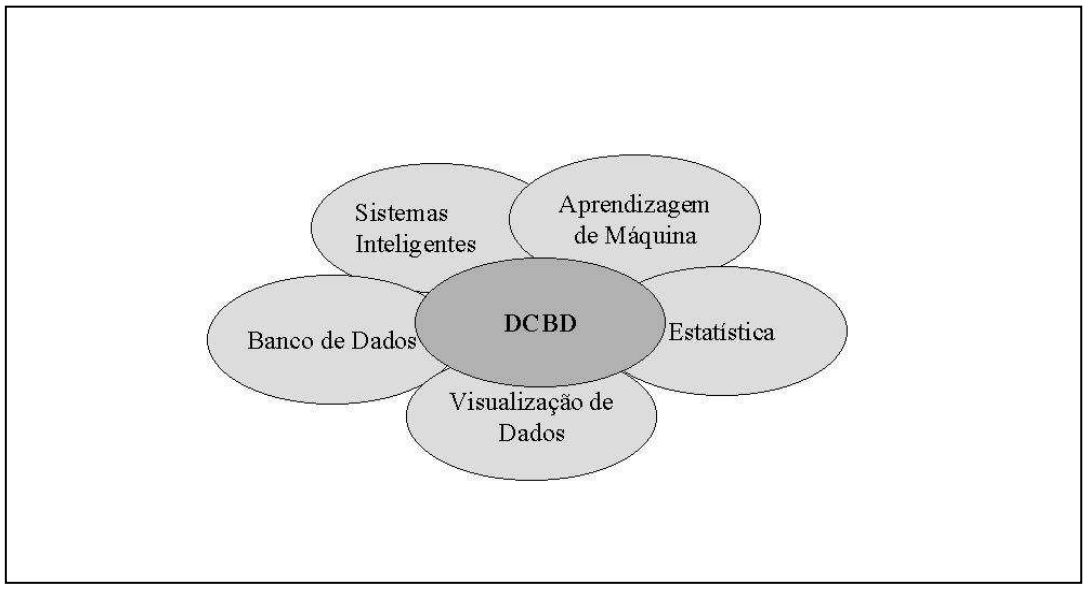

FIGURA 1.1 – Áreas de Apoio a DCBD
Fonte: FÉLIX, 1998. p. 27

Mediante a combinação dos diversos recursos provenientes destas diferentes áreas de apoio torna-se possível utilizar a DCBD em certas aplicações, tais como algumas das citadas por Fayyad; Piatetsky-Shapiro; Smyth (1996) e Noguez (2000):

a) Análise de riscos: que pode ser realizada através do agrupamento das características que identificam por exemplo, o cliente como um bom ou mau pagador em sistemas de prevenção de níveis de empréstimos arriscados em um banco.

b) *Marketing* direto: através da identificação dos clientes que devem ser incluídos em uma mala direta para se obter um retorno mais alto, o que pode ser feito através da análise da base de dados e da seleção exclusiva dos clientes que apresentam maior possibilidade de responder à mala direta.

c) Detecção de fraudes: descobrindo indicações de transações fraudulentas no meio comercial (seguros, bancos), na área da saúde (despesas e procedimentos de hospitais), etc.

d) Análise de tendências: considera o fator temporal para analisar por exemplo, a diferença entre o perfil do consumidor atual e o perfil do consumidor de antes. Podendo ser utilizada a mineração de dados para procurar padrões nas preferências dos clientes existentes, utilizando estes padrões para selecionar clientes posteriores, descobrindo-se assim clientes em potencial para um novo produto.

Já que o propósito do emprego da DCBD é obter conhecimento útil, tem-se que a fase de mineração de dados ao aplicar algoritmos específicos de aprendizagem indutiva sobre um conjunto de dados é capaz de extrair conhecimento a partir de instâncias, envolvendo repetidas aplicações iterativas dos métodos (HALMENSCHLAGER, 2002).

Segundo Ávila (apud HALMENSCHLAGER, 2002, f. 21), o aprendizado indutivo consiste na criação de um modelo, o que requer que durante o aprendizado, o sistema observe e reconheça similaridades entre instâncias correspondentes a uma determinada realidade, agrupando instâncias similares em classes e construindo regras

que forneçam o comportamento das instâncias de cada classe. Existem dois tipos de técnicas de aprendizado indutivo:

a) Aprendizado supervisionado: é aquele em que são fornecidas as classes e instâncias de cada classe ao sistema. Este terá então que descobrir a descrição da classe, ou seja, os atributos comuns nas instâncias de cada classe;

b) Aprendizado não-supervisionado: neste o próprio sistema é que deve descobrir a classe, a partir dos atributos comuns das instâncias.

O enfoque da mineração de dados a partir dos objetivos da aplicação pode ser dado segundo algum tipo de problema ou tarefa de mineração de dados, dentre os apresentados por Fayyad; Piatetsky-Shapiro; Smyth (1996) e John (1997), sejam eles:

a) Classificação: consiste no aprendizado supervisionado de uma função que classifica, ou seja, atribui um rótulo a uma instância dentre várias classes previamente estabelecidas. Os métodos de classificação podem ser utilizados para identificar instâncias de interesse em grandes bases de dados, sendo as árvores e regras de decisão, assim como também as redes neurais, técnicas que podem ser utilizadas para esta finalidade.

b) Associação: corresponde a padrões informativos na forma $X \rightarrow Y$, em que X e Y são conjuntos de itens. Assim, são pesquisadas todas as possíveis regras $X \rightarrow Y$, representando estruturas de valores dos atributos de uma instância. Esta regra indica que as transações da base de dados que contêm X tendem a conter também Y.

c) Regressão Linear: refere-se ao aprendizado de uma função que mapeia um item de dado para uma variável de valor real. Este tipo de problema de mineração de dados é semelhante à classificação, exceto pelo valor contínuo[1] do atributo de classe, ao invés do simbólico[2], como ocorre na classificação. Os métodos de regressão linear permitem a discriminação dos dados através da combinação dos atributos de entrada.

d) *Clustering* ou Agrupamento: considerado um aprendizado que identifica um conjunto finito de categorias ou agrupamentos para descrever os dados, ou seja, é uma classificação não-supervisionada. O objetivo consiste em particionar a base de dados em um determinado número de *clusters* (grupos), em que as instâncias de um *cluster* sejam similares. As categorias podem ser mutuamente exclusivas, hierárquicas ou ainda possuir características em comum. Como técnicas utilizadas para agrupar dados tem-se segmentação demográfica e redes neurais.

e) Sumarização: corresponde a métodos que encontram uma descrição compacta para um subconjunto de dados. Métodos mais sofisticados envolvem a derivação de regras de resumo e descobertas de relações funcionais entre atributos. As técnicas de sumarização são sempre aplicadas à análise exploratória de dados e à geração automática de relatórios.

f) Dependência: destina-se a encontrar um modelo que descreva dependências significativas entre variáveis. O modelo de dependência pode ser de nível estrutural ou de nível quantitativo. No nível estrutural, o modelo está sempre representado de uma forma gráfica e com atributos localmente dependentes em relação a outros, enquanto

[1] Ao longo deste trabalho os termos contínuo e numérico para tipo de dado de um atributo são tratados como sinônimos.
[2] O termo simbólico e discreto são tratados como sinônimos neste trabalho.

que, no nível quantitativo, o modelo especifica a força das dependências com alguma escala numérica. As redes de dependência probabilística utilizam independência condicional para especificar o aspecto estrutural do modelo e probabilidades, e adotam correlação para especificar a força das dependências.

g) Desvio ou Detecção de Desvio: caracteriza-se por detectar alterações significativas nos dados com relação a valores normativos medidos anteriormente. A partir de um conjunto de dependências, seqüências e/ ou descrições de conceitos, o algoritmo procura os elementos contidos no banco de dados que estão fora dos padrões, que são exceções às regras ou anômalos.

Assim, embora existam várias técnicas e ferramentas de mineração de dados que podem ser utilizadas é importante considerar a adoção de uma metodologia para a realização do processo de DCBD.

Conforme Neves (2001), existem várias metodologias propostas por diversos autores, sendo que as mais conhecidas são: CRISP-DM de Chapman et al. (1999) e a proposta por Fayyad; Piatetsky-Shapiro; Smyth (1996).

Como a essência destas metodologias é quase a mesma, então de um modo geral considera-se neste trabalho que as fases do processo de DCBD consistem em: Entendimento do Domínio do Problema, Pré-Processamento, Mineração de Dados e Pós-Processamento.

Ressaltando-se que, a ênfase deste trabalho é dada sobre a fase de Pré-Processamento considerada de fundamental importância para a descoberta de conhecimento útil, esta fase é explorada em termos de entendimento, seleção, limpeza e transformação de dados, a nível teórico e prático, correspondendo este último a experimentos realizados em processos de Descoberta de Conhecimento em Banco de Dados sobre a base de dados de AIHs (Autorização de Internação Hospitalar) da SES (Secretaria Estadual de Saúde) do Rio Grande do Sul.

1.1 Motivação

O processo de DCBD é dito ser não-trivial por requerer uma série de considerações relacionadas principalmente com a preparação das bases de dados.

Conforme Fayyad; Piatetsky-Shapiro; Smyth (1996) e Feldens (1997) existem alguns desafios da DCBD referentes às bases de dados tais como:

a) Volume da base de dados: as bases de dados com centenas de campos e tabelas ocupam muito espaço de armazenamento, o que pode resultar numa variedade enorme de padrões, combinações e hipóteses;

b) Alta dimensionalidade da base de dados: esta é medida pelo grande número de atributos de uma base de dados, o que maximiza de forma explosiva o tamanho do espaço de busca e também a probabilidade do algoritmo encontrar padrões falsos. Para resolver este problema podem ser utilizados métodos para reduzir efetivamente a dimensionalidade além de se poder também adotar prioridades para identificar atributos irrelevantes;

c) Bases de dados redundantes: tanto a redundância quanto a estrutura hierárquica dos atributos e as relações entre os mesmos encontradas no banco de dados não podem ser consideradas conhecimento pelo algoritmo de extração;

d) Dados inconsistentes: além de atributos com valores ausentes ou errôneos no banco de dados, alguns atributos importantes para o processo de descoberta podem não estar presentes no banco de dados. Uma solução seria utilizar estratégias estatísticas sofisticadas para identificar atributos ocultos e utilizar amostras maiores de dados, diminuindo assim a inconsistência;

e) Dados irregulares: diferentes bases de dados são utilizadas em várias partes da organização, e conseqüentemente, os dados operacionais podem ter diferentes domínios para definir uma mesma informação além de variarem em relação a qualidade. Uma das soluções para este problema seria uma análise efetiva de qual a melhor base de dados para selecionar os dados, ou então utilizar um *data warehouse*, o qual integra em um único repositório, os dados operacionais necessários ao processo decisório e que são encontrados em diferentes e heterogêneas bases de dados, apresentando então um ambiente estável e integrado dos dados;

f) Dados constantemente alterados: a natureza dinâmica dos dados permite que eles sejam constantemente alterados, conduzindo a conclusões temporárias, pois os atributos analisados podem ter sido removidos ou modificados. Uma possível solução seria utilizar métodos para atualizar os padrões ou utilizar apenas os padrões que sofreram alterações;

g) Privacidade dos dados: a essência investigatória dos sistemas de descoberta de conhecimento podem revelar por exemplo, informações particulares de indivíduos, as quais não devem ser utilizadas de forma indevida ou não autorizada. Assim, a privacidade deve ser mantida através da troca ou exclusão de atributos de identificação, descobrindo assim, os padrões sem invadir a privacidade dos indivíduos explícita nos dados.

Diante dos desafios acima citados, entende-se que os problemas relacionados às bases de dados ocorrem porque estas não foram geradas visando a descoberta de conhecimento e sim a propósitos distintos.

Desta forma, verifica-se que a qualidade dos dados para a descoberta de um conhecimento útil sobre os mesmos é fundamental. Assim, para o processo de DCBD a ocorrência de alguns problemas em grandes bases de dados tais como o número excessivo de atributos e tabelas, atributos com valores ausentes ou errôneos e falta de padronização dos dados, bem como a necessidade de apresentar os dados em uma forma apropriada para a técnica de mineração de dados escolhida, requer toda uma preparação ou pré-processamento, fase esta que segundo Manilla (apud KLEMETTINEN, 1999, p. 82) consome cerca de 80% do esforço total de todo o processo de DCBD, merecendo assim uma atenção especial.

Em termos das metodologias existentes para o processo de DCBD, a fase de pré-processamento geralmente é pouco detalhada assim como também a predominância das referências bibliográficas na área de DCBD recai sobre técnicas de mineração de dados enquanto que pouco se encontra a respeito de pré-processamento, exceto algoritmos de discretização e seleção de atributos e instâncias.

Isto tudo motivou o enfoque da pesquisa e experimentação deste trabalho sobre a fase de pré-processamento no sentido de apresentar o que existe a respeito e como esta poderia ser realizada sobre uma base de dados real.

1.2 Objetivos

a) Objetivo Geral

Estudo do pré-processamento como parte fundamental do processo de DCBD.

b) Objetivos Específicos

- Estudar metodologias para o processo de DCBD;

- Estudar métodos e técnicas específicas de pré-processamento;

- Concluir com alguns experimentos realizando o processo de DCBD na base de dados de AIH (Autorização de Internação Hospitalar) da SES do RS.

1.3 Metodologia

Realizou-se uma pesquisa bibliográfica sobre as metodologias de DCBD existentes selecionando-se as mais conhecidas como a proposta por Fayyad; Piatetsky-Shapiro; Smyth (1996) e CRISP-DM de Chapman et al. (1999) assim como também foram pesquisadas referências sobre as técnicas específicas de pré-processamento.

Para os experimentos foi utilizada a base de dados de AIHs da SES do RS referente ao ano de 2000, cujo acesso foi possível através do projeto "Desenvolvimento de Metodologia para Extração de Conhecimento de Bases de Dados de Saúde do Estado para Avaliação e Planejamento" realizado como colaboração entre pesquisadores da UFRGS e da UCS, e a SES.

1.4 Organização do Trabalho

Este trabalho apresenta-se dividido em cinco capítulos. No segundo capítulo são abordadas metodologias para o processo de DCBD destacando-se CRISP-DM de Chapman et al. (1999) e a proposta por Fayyad; Piatetsky-Shapiro; Smyth (1996). No terceiro capítulo são apresentadas as subfases da fase de pré-processamento contemplando-se entendimento, seleção, limpeza e transformação de dados, bem como as principais técnicas relacionadas as mesmas. Já no quarto capítulo são descritos os experimentos realizados sobre a base de dados de AIHs da SES do RS. Finalmente, no quinto capítulo são apresentadas as considerações finais sobre pré-processamento no

processo de DCBD, apontando as dificuldades encontradas na prática, contribuições do presente trabalho e pretensões da continuidade do mesmo.

Assim, espera-se que seja despertada a atenção em relação a importância que deve ser dada a fase de pré-processamento no processo de DCBD e que as experiências relatadas possam contribuir para futuras consultas ou tentativas de novos pré-processamentos, bem como todo o levantamento feito a respeito das várias técnicas de pré-processamento estimule e sirva como embasamento para o desenvolvimento de ferramentas de pré-processamento para descoberta de conhecimento em banco de dados oferecendo suporte a todas as subfases do mesmo e permitindo a interatividade por parte do usuário.

2 Metodologias para o Processo de Descoberta de Conhecimento em Banco de Dados

A Descoberta de Conhecimento em Banco de Dados (DCBD) utilizada em vários tipos de aplicações tem o apoio de técnicas e ferramentas de grande utilidade para a sua realização, porém a adoção de uma metodologia também é de fundamental importância para se tentar estabelecer um planejamento e organização da execução do processo de DCBD podendo ainda servir de referência para a realização de novos processos de DCBD.

Existem várias metodologias de DCBD propostas por diversos autores, como pode ser verificado em Neves (2001), onde são apresentadas algumas metodologias encontradas na literatura tais como: CRISP-DM de Chapman et al. (1999); a de Fayyad; Piatetsky-Shapiro; Smyth (1996); I-MIN de Gupta et al. (2000); a de Bruha (2000); a de Klemettinen (1999); a de Williams; Huang (1996); a de Lee; Kerschberg (1998); a de Brachman; Anand (1996); as utilizadas por Curotto (2000), e por Jha; Hui (1998).

Sendo que, neste trabalho serão apresentadas de forma sucinta as metodologias CRISP-DM e a proposta por Fayyad; Piatetsky-Shapiro; Smyth (1996) por serem as mais conhecidas na área de DCBD, ressaltando-se que no levantamento bibliográfico realizado CRISP-DM mostrou-se como a metodologia mais detalhada permitindo ser feita uma maior descrição da mesma.

Em ambas metodologias são consideradas algumas expressões que segundo John (1997) referem-se as duas partes envolvidas no processo de descoberta de conhecimento em banco de dados, sejam elas: o analista de mineração de dados e o especialista do domínio do problema (usuário).

Geralmente, o especialista do domínio conhece muito bem o problema a ser resolvido sabendo explicá-lo ao analista de mineração de dados que por sua vez detém o conhecimento das técnicas ou métodos necessários para a descoberta de conhecimento. Desta forma, é estabelecida a interação entre o especialista e o analista, a qual é essencial ao bom andamento do processo. A seguir, são apresentadas as referidas metodologias.

2.1 A Metodologia CRISP-DM

A metodologia CRISP-DM (*CRoss - Industry Standard Process for Data Mining*) corresponde a um modelo de processo parcialmente fundado em julho de 1997 pela Comissão Européia, de acordo com o programa ESPRIT, incluindo como membros: NCR, o principal fornecedor no mundo de soluções de *data warehouse*; ISL - *Integral Solutions Limited,* desenvolvedores do sistema de mineração de dados Clementine, tornou-se parte de SPSS em dezembro de 1998; Daimler - Chrysler (*European industrial power center*); e OHRA, uma grande companhia de seguros. A conclusão deste modelo ocorreu em 1999.

O CRISP-DM não se restringe a uma ferramenta ou tecnologia específica devido ao próprio objetivo de sua criação que era o estabelecimento de uma metodologia padrão que pudesse conduzir a realização do processo de descoberta de conhecimento em banco de dados (DCBD).

A metodologia CRISP-DM consiste em 6 (seis) fases contendo cada uma algumas tarefas particulares. Destacando-se que, não existe uma seqüência rigorosa quanto a realização destas fases e tarefas, pois sempre que necessário é possível voltar ou avançar entre diferentes fases. Sendo importante a participação do especialista do domínio do problema durante todo o processo.

Como apresentado na Figura 2.1, as fases de CRISP-DM são descritas a seguir:

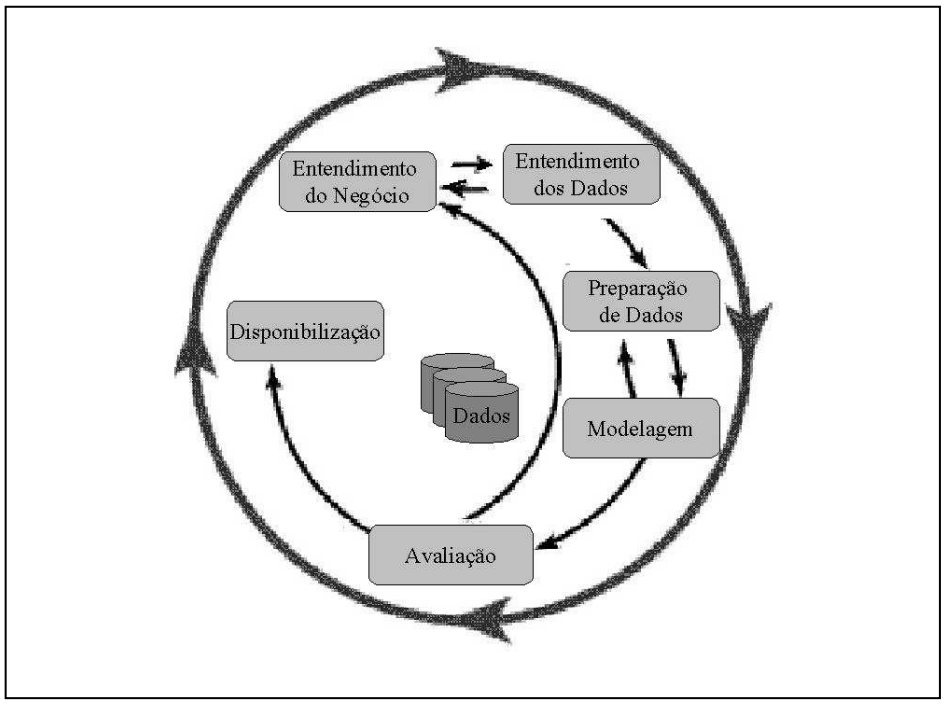

FIGURA 2.1 – Fases do Modelo de Referência CRISP-DM
Fonte: CHAPMAN et al., 1999. p. 06

1) Entendimento do Negócio

Esta fase é tratada com esta denominação por Chapman et al. (1999), porém neste trabalho ela é referenciada como entendimento do domínio do problema. Segundo Chapman et al. (1999), consiste em identificar os objetivos do usuário sob o ponto de vista do negócio sendo este conhecimento convertido dentro de uma definição de problema de mineração de dados e em um plano inicial projetado para alcançar estes objetivos.

As tarefas relacionadas a esta fase são as seguintes, conforme Chapman et al. (1999):

a) Determinar objetivos do negócio: para tanto, é preciso identificar as pessoas de maior relevância na organização que estejam diretamente relacionadas ao domínio do problema em questão; a área do problema (*marketing*, atendimento ao cliente, etc); os grupos interessados no resultado do projeto, e principalmente o objetivo crucial do especialista do domínio do problema em termos de negócio.

b) Avaliar Situação: refere-se a verificação da disponibilidade de recursos, e da existência de limitações, terminologia, etc.

Quanto aos recursos estes podem ser: pessoal (especialistas do domínio do problema, especialistas de dados, suporte técnico, equipe de mineração de dados), dados, hardware, software (ferramentas de mineração de dados e outros softwares importantes). Em termos de limitações considera-se principalmente as referentes a questões legais, escala de tempo, acessibilidade técnica do dado (sistemas operacionais, formato de arquivo ou banco de dados) e do conhecimento relevante além das limitações de orçamento. E, deve-se procurar familiarizar-se com a terminologia do domínio do problema junto aos especialistas do mesmo.

c) Determinar Objetivos de Mineração de Dados (MD): as questões do domínio do problema devem ser traduzidas para objetivos de mineração de dados e deve ser especificado o tipo de problema de MD (classificação, *clustering*, etc). Além disso, é necessário especificar critérios para avaliação do modelo (acurácia do modelo, desempenho e complexidade).

d) Produzir Plano de Projeto: um plano de projeto pretendido deve ser descrito visando atingir os objetivos de MD e conseqüentemente alcançar os objetivos do domínio do problema. Neste plano deve constar um antecipado conjunto de fases a serem concretizadas e possíveis repetições de algumas durante o processo de descoberta de conhecimento em banco de dados (DCBD), incluindo a seleção inicial de ferramentas e técnicas. Este plano de projeto deve ser um documento dinâmico, onde as atualizações e revisões realizadas no final de cada fase deverão ser registradas no mesmo.

2) Entendimento dos Dados

A partir da coleta inicial dos dados é preciso explorá-los, verificando suas propriedades e qualidade de modo a disponibilizar em relatórios qualquer informação e/ou resultados das análises realizadas sobre os mesmos.

Considera-se então as seguintes tarefas relativas a esta fase:

a) Coletar Dados Iniciais: refere-se a obtenção dos dados dentro do projeto que foram listados nos recursos do mesmo conduzindo assim a uma preparação de dados inicial. Caso os dados tenham sido obtidos de múltiplas fontes cabe a integração dos mesmos.

Assim é preciso planejar quais informações são necessárias e se as mesmas estão disponíveis; especificar o critério de seleção (por exemplo, quais atributos são necessários para especificar os objetivos de MD? Quais atributos são considerados irrelevantes? Quantos atributos podem ser manipulados com as técnicas escolhidas?) além de selecionar tabelas (arquivos) e dados dentro destas.

b) Descrever Dados: um levantamento sobre as propriedades gerais dos dados adquiridos deve ser feito com conseqüente relato sobre o mesmo. Então, é interessante verificar por exemplo:

- O número de instâncias e atributos em cada tabela;

- Os tipos e faixas de valores de atributos;

- As correlações de atributos;

- O significado de cada atributo e sua importância em termos do domínio do problema;

- Estatísticas básicas para certos atributos (distribuições, média, máximo, mínimo, desvio padrão, etc);

- A relevância do atributo para o objetivo específico de mineração de dados (MD) e a opinião do especialista do domínio sobre a mesma,

- As relações entre chaves.

c) Explorar Dados: para exploração dos dados, análises sobre propriedades de atributos de interesse podem ser feitas como por exemplo, estatísticas básicas e identificação das características das mesmas; além de formulação de hipóteses e transformação das mesmas em um ou mais objetivos de MD se possível, tornando-os mais precisos.

d) Verificar Qualidade de Dados: a qualidade dos dados deve ser examinada considerando-se por exemplo:

- A existência e freqüência de erros nos dados;

- Identificação de atributos com valores ausentes,

- Apresentação de atributos com diferentes valores que tenham significados similares.

3) Preparação de Dados

A preparação de dados é necessária para produzir um conjunto de dados que se apresente no formato adequado para ser submetido a MD. Ressaltando-se que deverá ser feita uma descrição deste conjunto de dados para auxiliar na execução da fase seguinte.

Destacando-se como principais tarefas da fase de preparação de dados, as descritas a seguir:

a) Selecionar Dados: consiste em decidir sobre os dados que serão utilizados na MD, considerando a seleção de atributos e instâncias das tabelas de acordo com a relevância dos mesmos para os objetivos de MD, qualidade e limitações técnicas como volume e/ou tipo de dados. Durante a seleção, os dados a serem incluídos/ excluídos e as razões para estas decisões deverão ser documentados.

b) Limpar Dados: a limpeza de dados refere-se ao aumento da qualidade dos dados para o nível requerido pelas técnicas de MD selecionadas.

Sendo que, as decisões e ações tomadas para tratar os problemas de qualidade de dados (por exemplo, corrigir, remover ou ignorar dados errôneos) verificados durante a fase de entendimento do dado devem ser descritas, assim como também relatadas todas as transformações do dado para o propósito de limpeza e seus conseqüentes impactos na análise de resultados.

c) Construir Dados: a construção de dados pode ser realizada através de algumas operações de preparação de dados de caráter construtivo como por exemplo, a produção de atributos derivados ou transformação de valores de atributos existentes.

d) Integrar Dados: consiste em combinar múltiplas tabelas ou outras fontes de informação para criar novas instâncias ou valores. A combinação é apresentada em uma nova tabela combinando atributos das tabelas fontes, devendo-se levar em consideração a possibilidade e facilidade da integração assim como também reconsiderar os critérios de seleção de dados diante de experiências de integração de dados, isto é, pode-se desejar incluir/ excluir outros conjuntos de dados.

e) Formatar Dados: as transformações de formatação de dados referem-se a modificações sintáticas realizadas nos mesmos sem alterar seu significado, mas que poderiam ser requeridas pela ferramenta de MD como exemplo, tem-se a reordenação de atributos, já que algumas ferramentas apresentam requisitos referentes a ordem de atributos tais como, o primeiro atributo ser uma chave para cada instância ou o último atributo ser destinado ao resultado que o modelo deverá prover.

4) Modelagem

O termo modelagem utilizado por Chapman et al. (1999), neste trabalho é entendido como a fase de MD do processo de DCBD.

Segundo Chapman et al. (1999), nesta fase devem ser consideradas as seguintes tarefas:

a) Selecionar a técnica de modelagem: a técnica de MD a ser utilizada deve ser escolhida neste momento de acordo com o suporte que é oferecido pela ferramenta de MD selecionada, já que muitas técnicas de MD requerem apresentações específicas da qualidade ou formato do dado como por exemplo, considerar que um atributo de classe deve ser discreto.

Estas apresentações requeridas pela técnica de MD devem ser comparadas às suposições que se encontram no relatório de descrição do dado, assegurando que os requisitos da técnica sejam mantidos, sendo possível retornar a fase de preparação de dados se necessário.

b) Gerar Projeto de Teste: de acordo com Chapman et al. (1999), antes de realmente ser construído um modelo é preciso gerar um mecanismo para testar a qualidade e a validade do modelo como por exemplo, nas tarefas de mineração de dados (MD) supervisionadas, como a classificação, geralmente são usadas taxas de erros como medidas de qualidade em modelos de MD. Assim, o projeto de teste especificará que o conjunto de dados seja dividido em conjunto de treinamento e conjunto de teste, sendo que o modelo é construído no conjunto de treinamento, e a sua qualidade é estimada no conjunto de teste.

c) Construir Modelo: para realizar esta tarefa a ferramenta de MD deve ser executada utilizando o conjunto de dados preparado para criar um ou mais modelos. Sendo importante também o relato dos parâmetros e seus respectivos valores escolhidos na utilização da ferramenta, das razões para a escolha destes valores; a descrição do modelo resultante avaliando sua acurácia esperada, robustez e possibilidade de falhas sendo relatado ainda qualquer dificuldade encontrada em relação a interpretação do (s) modelo (s).

d) Avaliar Modelo: o analista de mineração de dados (MD) deve interpretar o (s) modelo (s) gerados de acordo com o seu domínio de conhecimento, os critérios de sucesso de MD definidos e o projeto de teste desejado e juntamente com o (s) especialista (s) do domínio do problema deve avaliar estes resultados de MD sob a perspectiva do negócio.

Assim, para a avaliação do modelo o analista de MD pode por exemplo, testar o resultado de acordo com a estratégia de teste (treinamento e teste, validação cruzada, etc); comparar a avaliação de resultados e interpretação; selecionar melhores modelos; interpretar resultados em relação ao domínio do problema, comparar o modelo com a base de conhecimento dada para verificar se a informação descoberta é nova e útil.

Além disso, as definições de parâmetros poderão ser revisadas e estes assim redefinidos para uma próxima execução da tarefa de construir modelo, já que iterações do modelo construído e avaliado devem ser realizadas para que seja garantida a descoberta do (s) melhor (es) modelo (s).

5) Avaliação

A fase de avaliação envolve a necessidade de interpretar e avaliar os resultados de MD em relação aos objetivos do usuário; revisar o processo de MD para identificar problemas e ações imprevistas e as respectivas sugestões sobre as mesmas, assim como também descrever as próximas ações a serem realizadas. Deste modo, para Chapman et al. (1999), as tarefas relacionadas a esta fase são:

a) Avaliar Resultados: os resultados de MD devem ser interpretados e avaliados em relação aos objetivos do usuário e aqueles que corresponderem poderão ser considerados como modelos aprovados.

b) Revisar o Processo: mesmo que aparentemente o modelo resultante corresponda aos objetivos do usuário é preciso fazer uma revisão detalhada do processo de MD de forma a identificar falhas e ações alternativas que não tenham sido previstas e apresentar sugestões sobre as mesmas ou sobre atividades que deveriam ser repetidas.

c) Determinar próximos passos: o responsável pelo processo de DCBD deve decidir se segue para a próxima fase, a disponibilização, ou se inicia iterações, ou ainda se inicia novos projetos de MD, sempre levando em consideração a análise de recursos e orçamentos restantes que influenciam as tomadas de decisões. Assim, deve ser produzida uma lista de ações possíveis juntamente com as razões para qualquer tipo de escolha e a descrição das decisões relacionadas.

6) Disponibilização

Nesta fase, existe a preocupação com a estratégia a ser utilizada para a apresentação dos resultados de MD e manutenção dos mesmos, além da produção de um relatório final contendo a descrição de toda a experiência ao longo do processo de DCBD.

As tarefas relacionadas a esta fase são as seguintes:

a) Planejar Disponibilização: significa decidir a estratégia para a disponibilização do resultado de MD no ambiente do domínio do problema levando em consideração possíveis dificuldades que possam ser encontradas quando disponibilizados os resultados de MD.

b) Planejar Monitoramento e Manutenção: para a utilização dos resultados de MD diariamente no ambiente do domínio do problema é importante considerar por exemplo, os aspectos dinâmicos do mesmo, as situações em que estes resultados de MD não devem ser utilizados, sendo assim preciso elaborar um plano de monitoramento e manutenção destes resultados de MD, incluindo os passos necessários e como realizá-los.

c) Produzir Relatório Final: o responsável pelo processo de DCBD e sua equipe deverão preparar um relatório final contendo um sumário do projeto e suas experiências ou poderão realizar uma apresentação final dos resultados de MD.

d) Revisar o Projeto: consiste principalmente em avaliar pontos positivos e negativos do projeto, identificando o que foi bem feito e o que precisa ser melhorado; problemas e sugestões para seleção de técnicas de MD mais apropriadas em situações semelhantes; verificar se os usuários estão satisfeitos com os resultados de MD, ou se precisam de um suporte adicional, sendo todo este levantamento registrado como uma documentação da experiência.

2.2 Metodologia proposta por Fayyad; Piatetsky-Shapiro; Smyth

Segundo Fayyad; Piatetsky-Shapiro; Smyth (1996), a definição da Descoberta de Conhecimento em Banco de Dados (DCBD) é dada como o processo não-trivial de identificação de padrões válidos, novos, potencialmente úteis e compreensíveis implícitos nos dados.

Em relação às características dos padrões mencionados, tem-se que:

- A validade refere-se a determinado grau de certeza que os padrões descobertos devem assumir;

- A novidade deve-se ao fato de não serem de conhecimento prévio;

- A expressão “potencialmente úteis” está relacionada à necessidade dos padrões conduzirem a algumas ações realmente úteis;

- A compreensibilidade corresponde ao entendimento do padrão obtido de modo a facilitar a compreensão do que não é evidenciado nos dados.

Além disso, o processo de DCBD é interativo e iterativo envolvendo 9 (nove) fases como apresentado na Figura 2.2 a seguir.

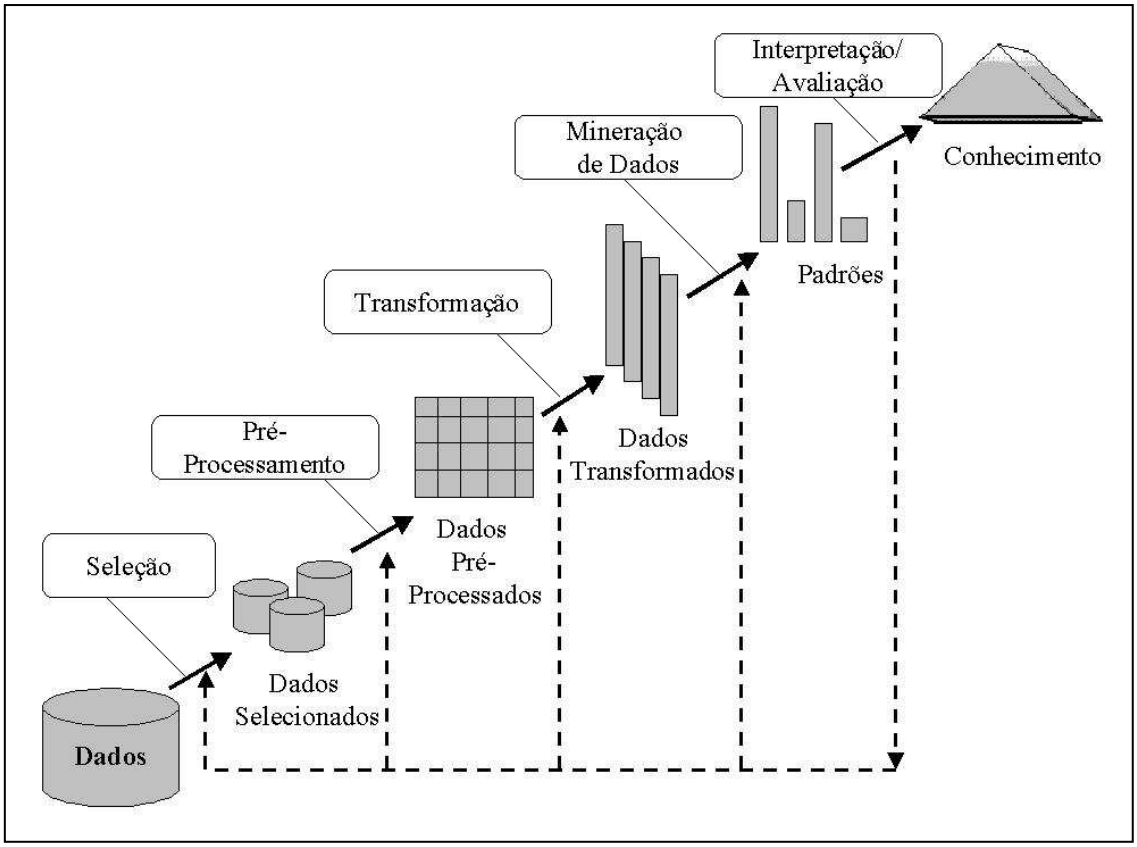

FIGURA 2.2 – Uma Visão Geral das Fases Inclusas no Processo DCBD
Fonte: FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996. p. 10

Estas fases consistem em:

1) Desenvolvimento de um entendimento do domínio da aplicação

Implica no levantamento de um conhecimento prévio relevante e dos objetivos do usuário.

2) Criação de um conjunto de dados alvo

Significa selecionar um conjunto de dados ou concentrar-se em um subconjunto de atributos ou instâncias sobre os quais a descoberta será efetuada.

3) Limpeza de Dados e Pré-Processamento

Consiste em operações básicas tais como remoção de dados errôneos e manipulação de atributos com valores ausentes.

4) Redução e Projeção de Dados

Envolve métodos de transformação para reduzir o número efetivo de atributos relevantes para a representação da dependência de dado em relação ao tipo de problema de MD.

5) Escolha do tipo de problema de MD

Nesta fase, é decidido se o objetivo do processo de DCBD é classificação, regressão, *clustering*, etc.

6) Escolha do (s) algoritmo (s) de MD

Consiste na seleção de quais métodos ou técnicas podem ser apropriados, tais como árvores de decisão, regras de decisão, redes neurais, etc.

7) Mineração de dados

Significa a efetiva aplicação dos algoritmos de MD selecionados anteriormente de modo a encontrar padrões interessantes nos dados.

8) Interpretação de padrões obtidos

Refere-se a análise dos resultados obtidos da MD permitindo verificar a necessidade de retornar a qualquer fase anterior, promovendo iterações.

9) Consolidação do conhecimento descoberto

Nesta fase, o conhecimento descoberto pode ser incorporado ao desempenho de um sistema, ou documentado e relatado às partes interessadas.

Apesar da diferença quanto ao número de fases e detalhamento das mesmas assumido em cada uma das metodologias acima apresentadas, percebe-se que tanto a metodologia CRISP-DM de Chapman et al. (1999) quanto a proposta por Fayyad; Piatetsky-Shapiro; Smyth (1996) tentam orientar o analista de mineração de dados no planejamento e execução de suas ações visando descobrir conhecimento útil a partir de grandes bases de dados. Sendo que, para atingir este objetivo é importante considerar a possibilidade de retornar e avançar entre fases repetindo certas operações para se obter um melhor resultado (configurando como o caráter iterativo do processo), assim como também contar com a contribuição do especialista do domínio do problema em todas as fases do processo (correspondendo ao caráter interativo) de Descoberta de Conhecimento em Banco de Dados.

3 Técnicas e Métodos de Pré-Processamento

Tendo-se por base as metodologias de Descoberta de Conhecimento em Banco de Dados (DCBD) existentes, considera-se neste trabalho que o processo DCBD, como visualizado na Figura 3.1, sob uma visão geral, consiste basicamente em 4 (quatro) fases: Entendimento do Domínio do Problema, Pré-Processamento, Mineração de Dados, e Pós-Processamento. Sendo que, não existe nenhum rigor quanto a seqüência em que estas fases devem ocorrer porém, é de fundamental importância a presença do especialista do domínio do problema durante todo o processo.

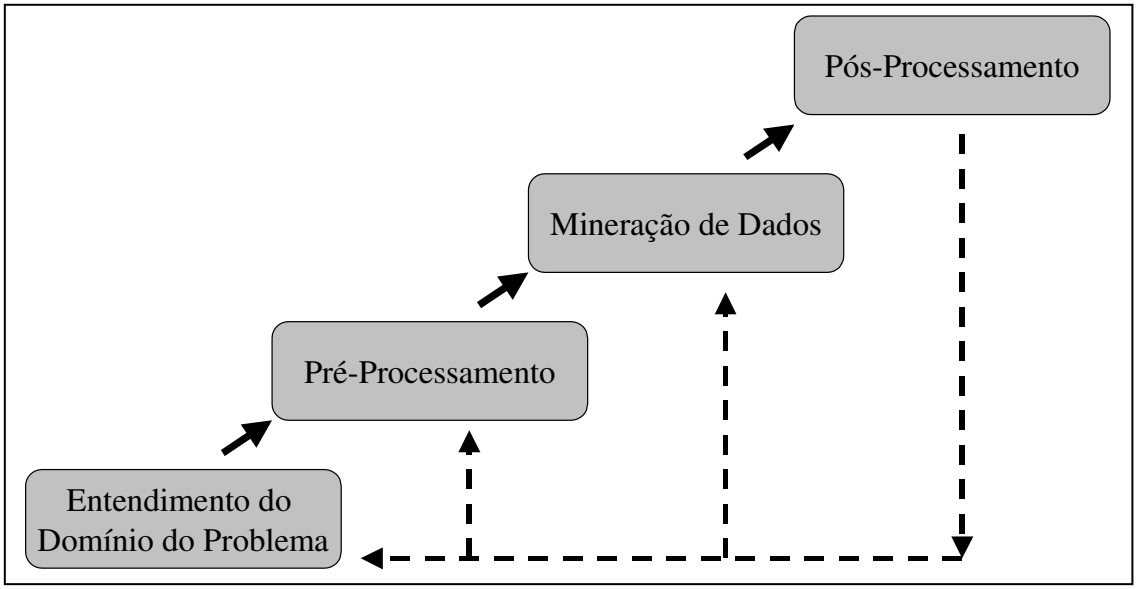

FIGURA 3.1 – Uma Visão Geral das Fases do Processo DCBD

A fase Entendimento do Domínio do Problema envolve entrevistas com o especialista do domínio do problema (usuário) a fim de identificar os objetivos do mesmo, levantamento dos recursos disponíveis (pessoal, dados, hardware, software, financeiro, e outros), verificação da existência de qualquer conhecimento prévio que possa contribuir ao alcance dos objetivos, tradução dos objetivos do especialista em objetivos de mineração de dados (MD) pelo analista de MD, definição do tipo de problema de MD (classificação, *clustering*, etc) a ser adotado bem como da técnica de MD (árvores de decisão, regras de decisão, redes neurais, etc) a ser utilizada.

A fase seguinte, Pré-Processamento, engloba uma análise inicial dos dados para se ter sólidas definições dos mesmos (tais como, estrutura das tabelas, valores potenciais dos atributos, sistema fonte original, formatos e tipos de dados), além de toda e qualquer operação necessária para a escolha dos dados relevantes aos objetivos do usuário, limpeza e transformação dos mesmos para tornar possível a MD a ser feita pela técnica escolhida. Corresponde então às fases de entendimento dos dados e preparação de dados da metodologia CRISP – DM, e às fases de criação de um conjunto de dados alvo, limpeza de dados e pré-processamento, redução e projeção de dados da metodologia de Fayyad; Piatetsky-Shapiro; Smyth (1996).

Já a fase de Mineração de Dados pode ser definida pela efetiva aplicação da técnica de MD através de algum algoritmo de aprendizagem.

Finalmente, a fase de Pós-Processamento consiste na interpretação e avaliação dos resultados de MD, que se forem reconhecidos como conhecimento útil deverão ser disponibilizados a empresa/ instituição solicitante.

Considerando-se a breve apresentação feita sobre as fases do processo DCBD, neste trabalho será enfatizada a fase de pré-processamento que consome a maior parte do tempo de realização do processo DCBD, na tentativa de se preparar os dados antes de submetê-los à mineração.

Isto deve-se ao fato que dados do mundo real tendem a ser errôneos, incompletos e inconsistentes, e técnicas de pré-processamento podem aprimorar a qualidade do dado conseqüentemente melhorando também a acurácia e eficiência do processo de mineração (HAN; KAMBER, 2001).

Conforme o apresentado na Figura 3.2, a fase de Pré-Processamento é composta pelas seguintes subfases: entendimento, seleção, limpeza e transformação de dados.

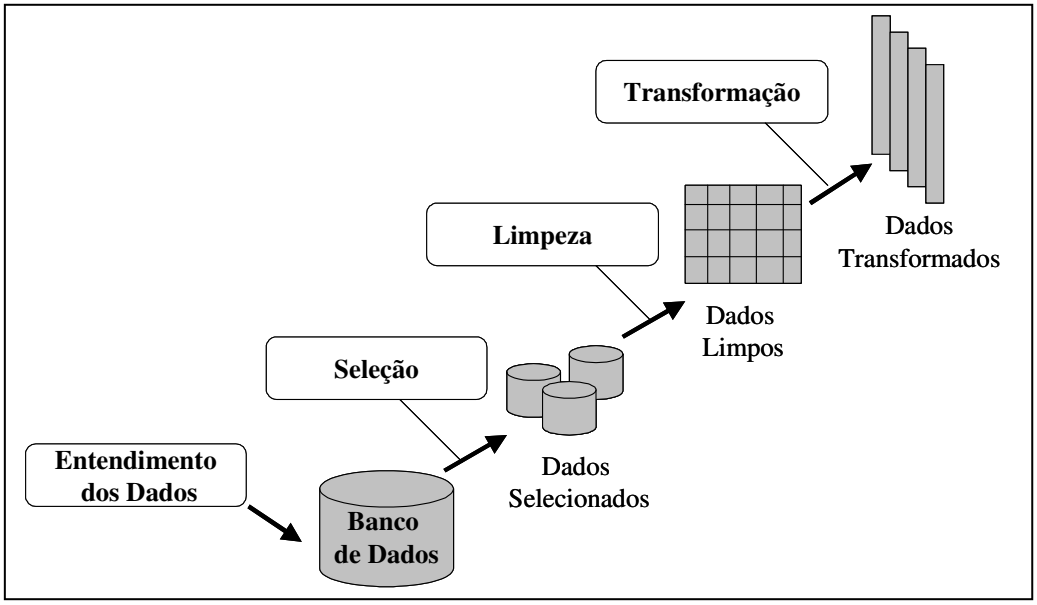

FIGURA 3.2 – Subfases de Pré-Processamento

Ressaltando-se que, assim como no processo geral de DCBD, não existe uma seqüência obrigatória quanto à ocorrência das subfases de pré-processamento, pois dependendo da situação pode-se por exemplo, preferir realizar a limpeza antes de um determinado tipo de seleção.

O entendimento dos dados seguindo as orientações de CRISP – DM consiste em analisar os dados fornecidos pelos especialistas, entendendo do que se tratam as tabelas envolvidas, o significado, relevância, formato, tamanho e tipo de dado dos atributos; identificando os atributos chaves; realizando levantamentos estatísticos e verificando a qualidade dos dados.

A seleção de dados envolve a escolha da (s) tabela (s), atributos e instâncias da (s) mesma (s) em relação aos objetivos do usuário, considerando-se ainda que, na necessidade de se manipular informações de várias tabelas cabe a integração das mesmas de modo a obter-se um conjunto único de instâncias sobre o qual será dada a continuidade do pré-processamento e/ ou do processo DCBD.

A limpeza de dados refere-se a garantia da qualidade dos dados que pode ser obtida através de algumas operações tais como: padronização de dados, tratamento de valores ausentes, eliminação de dados errôneos e de duplicatas.

Quanto à transformação de dados esta corresponde a operações que tornem a apresentação dos dados apropriada a técnica de mineração de dados a ser utilizada, assim encontram-se descritas operações do tipo normalização de dados, conversões de valores simbólicos para valores numéricos, discretização e composição de atributos.

Algumas das preocupações em pré-processamento, tais como seleção/ integração de tabelas e limpeza de dados são também consideradas em *Data Warehouse*, já que para este último, os dados são adquiridos de diversas fontes, internas ou externas à empresa, contendo informações estruturadas ou não, devendo ainda ser submetidos a um processo de limpeza e garantia de consistência das informações, objetivando formar uma fonte única de dados.

A seguir são apresentadas as subfases do pré-processamento bem como algumas técnicas para a realização de suas operações.

3.1 Entendimento dos Dados

Antes de ser iniciada qualquer operação de seleção, limpeza e transformação de dados é de fundamental importância que se tenha o conhecimento ou entendimento dos dados, pois isto orienta as tomadas de decisões do que se investigar e o que será necessário ser realizado em termos de preparação dos dados para a mineração.

De acordo com a metodologia CRISP – DM é preciso conhecer as tabelas envolvidas, bem como os atributos e o número de instâncias das mesmas.

No que se refere ao conhecimento das tabelas, é importante verificar principalmente o tipo de base de dados e o conteúdo que é tratado em cada tabela.

Em termos de atributos cabe a análise da quantidade destes em cada tabela, do significado de cada um, além do tipo de dado, tamanho, formato, valores assumidos e relevância para os objetivos de mineração de dados.

Já o levantamento do número de instâncias de cada tabela é importante para se ter noção da quantidade de dados que se dispõe e caso esta seja muito grande pode-se planejar de acordo com os objetivos de mineração de dados uma futura seleção de instâncias.

Destaca-se também nesta subfase a realização de estatísticas básicas sobre os dados como média, mínimo, máximo e desvio padrão para atributos numéricos

(contínuos), assim como também pode ser feito o levantamento da proporção de cada valor distinto dos atributos simbólicos (discretos).

A qualidade dos dados é outro fator que precisa ser considerado, pois é preciso verificar a existência de dados errôneos, de valores ausentes de atributos, a presença de duplicatas e a falta de padronização dos dados, todos problemas comuns de ocorrerem em bases de dados reais e que podem ser questionados aos especialistas do domínio.

Para todas estas investigações a serem feitas nesta subfase de entendimento dos dados podem ser utilizadas algumas ferramentas próprias para isto, como por exemplo Statistica™ ou alguns recursos oferecidos por outros *softwares* como Excel, Access e Database Desktop.

3.2 Seleção de Dados

Diante da quantidade de dados disponível no contexto do domínio do problema é preciso selecionar as tabelas, atributos e instâncias das mesmas que estejam mais relacionados aos objetivos do usuário a serem alcançados permitindo assim a geração de um novo conjunto de dados único e conciso que poderá ser submetido às demais subfases de pré-processamento bem como ao restante do processo de descoberta de conhecimento em banco de dados (DCBD).

3.2.1 Seleção/ Integração de Tabelas

Após a identificação das fontes de dados existentes, de acordo com o entendimento do domínio do problema é preciso selecionar as tabelas realmente relevantes em relação aos objetivos a serem alcançados.

Caso seja verificada a necessidade de manipular informações de várias tabelas de estruturas iguais e/ ou diferentes é aconselhável integrar estas tabelas de modo a obter um conjunto de dados único que será a base para o restante do processo de DCBD.

Em aplicações reais uma forma de promover a integração de tabelas é através do *data warehouse* cuja definição por Inmon (1992 apud CABENA et al., 1998, p. 19) é dada como: "... é uma coleção de dados orientada a assunto, integrada, variável no tempo, e não-volátil de suporte a tomada de decisões".

Conforme Witten; Frank (2000), é de grande utilidade a presença de um *data warehouse* como precursor à mineração de dados, e caso não esteja disponível, muitos dos passos envolvidos na criação do mesmo terão que ser tomados para preparar os dados para a mineração.

Entretanto mesmo um *data warehouse* poderá não conter todos os dados necessários, como por exemplo dados demográficos, sendo preciso buscar fora da organização dados relevantes ao problema de mineração de dados. Algumas vezes chamados dados de *overlay*, estes devem ser limpos e integrados aos demais dados já selecionados (WITTEN; FRANK, 2000).

3.2.2 Seleção de Atributos

No conjunto de dados selecionado é comum existirem atributos irrelevantes aos objetivos de mineração de dados (MD) pretendidos, assim é importante selecionar um determinado subconjunto de atributos a serem manipulados durante a MD.

Embora também possam ser utilizados todos os atributos, a maioria dos autores consultados defende a seleção de atributos relevantes e eliminação dos que são totalmente desnecessários, na tentativa de diminuir a complexidade do problema e alcançar um desempenho ótimo de aprendizagem.

Complementarmente pode-se afirmar que a seleção de atributos desempenha um papel importante na seleção e preparação de dados para a mineração de dados, na medida em que remove dados errôneos, redundantes e irrelevantes e reduz a dimensionalidade do espaço de atributos (LIU; YU, 2002). Entende-se que, a dimensionalidade refere-se aos atributos utilizados para descrever um conjunto de instâncias (RIAÑO, 1997).

No caso de uma excessiva dimensionalidade, existem várias razões que justificam a redução do número de atributos conforme Koller; Sahami (1996 apud RIAÑO, 1997, p. 54-55):

a) Requisitos de tempo e espaço: o custo tempo-espaço dos algoritmos de indução está diretamente relacionado ao número de atributos considerado.

b) Simplicidade: a redução do número de atributos é refletida na criação de estruturas menores que permitem um melhor entendimento do domínio se a taxa de erro não aumenta significativamente.

c) Relevância: alguns atributos podem ser inúteis e portanto removidos.

d) Redundância: alguns atributos podem refletir informação que já está embutida em outros atributos.

e) Acurácia: pré-processar o conjunto de atributos pode aumentar a acurácia do modelo resultante.

Assim os efeitos imediatos da seleção de atributos para a aplicação são: a execução mais rápida do algoritmo de mineração de dados e a melhoria da qualidade do dado, que conseqüentemente conduz a um melhor desempenho da mineração de dados e aumento da compreensibilidade dos resultados de mineração (LIU; YU, 2002) .

A seleção de atributos pode ser realizada “manualmente”, isto é, pela escolha direta feita pelo usuário ou através de algoritmos.

a) Seleção Manual: o usuário pode auxiliar o analista de MD na escolha dos atributos que possam contribuir ao alcance dos objetivos de MD, assim como também o analista de MD baseado em sua experiência e no entendimento do domínio do problema pode selecionar os atributos que considera relevantes.

b) Seleção por Algoritmos: dependendo da disponibilidade da informação de classe no dado, os métodos de seleção de atributos existentes podem ser considerados como pertencentes a abordagem de seleção de atributos supervisionada ou não-supervisionada (LIU; YU, 2002).

De acordo com Dash; Liu (2000), dentre os métodos encontrados na literatura, a maioria está relacionada ao problema de classificação supervisionada enquanto pouco trabalho tem sido feito em termos de classificação não-supervisionada ou *clustering*. Neste trabalho são apresentados alguns métodos pertencentes a estas respectivas abordagens.

Conforme Liu; Yu (2002), um método típico de seleção de atributo compreende quatro passos básicos, como visualizado na Figura 3.3, que inicialmente foram projetados para seleção de atributos supervisionada, mas que com o desenvolvimento dos trabalhos realizados sobre a não-supervisionada, foram mantidos com o acréscimo de considerar um novo critério de avaliação a partir de modelos filtro e *wrapper* dentro do passo de avaliação do subconjunto. A seguir tem-se a descrição dos referidos passos por (LIU; YU, 2002).

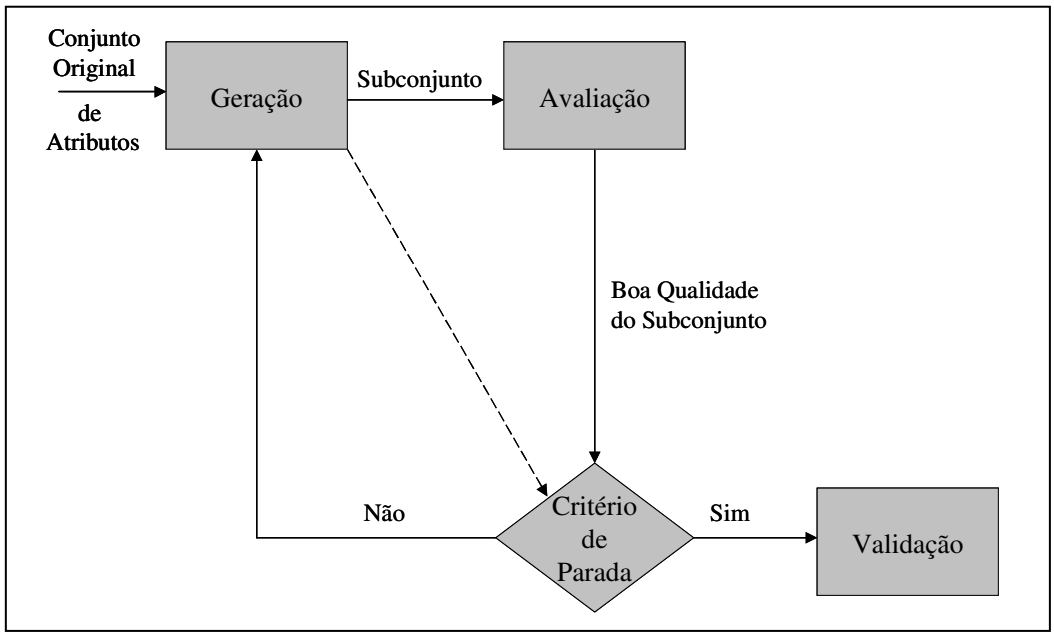

FIGURA 3.3 – Procedimentos Gerais de Seleção de Atributos
Fonte: LIU; YU, 2002. p. 02

3.2.2.1 Procedimento de geração ou de busca

Destinado a gerar subconjuntos de atributos para avaliação, este procedimento define a direção da busca que pode ser geração *forward*, geração *backward* ou geração aleatória.

A direção de geração *forward* consiste em expandir um subconjunto de atributos a partir de um conjunto vazio, sendo que os atributos são adicionados um por vez, e em cada iteração, o melhor atributo entre os não selecionados é escolhido baseado no critério de avaliação. O subconjunto cresce até ser alcançado um conjunto completo de atributos originais ou o critério de parada ser satisfeito.

Já a direção oposta, que é a de geração *backward,* inicia a busca a partir de um conjunto completo de atributos. Em cada iteração o atributo menos importante é removido baseado no critério de avaliação, conseqüentemente o subconjunto é reduzido até existir somente um atributo no conjunto ou o critério de parada ser satisfeito.

A direção de geração aleatória refere-se a busca iniciada a partir de um subconjunto aleatoriamente selecionado, adicionando e removendo atributo de maneira aleatória.

Segundo Liu; Yu (2002), além do aspecto de direção da busca, descrito anteriormente, tem-se ainda a questão da estratégia de busca que pode ser classificada em:

a) Busca Completa: segundo Schlimmer (1993 apud LIU; YU, 2002, p. 05), a busca completa – isto é, todo subconjunto ótimo é encontrado – não necessariamente significa que deve ser exaustiva – isto é, todo subconjunto tem que ser avaliado. Caso o critério de avaliação possua alguma propriedade (por exemplo, monotonicidade[3]), é possível encontrar um subconjunto ótimo sem avaliar todos os 2^N subconjuntos. De qualquer modo o espaço de busca é ainda na ordem de $O(2^N)$.

b) Busca Heurística: esta estratégia de busca evita ser completa, mas ao mesmo tempo arrisca perder subconjuntos ótimos. O tipo de busca é em profundidade (*depth-first*) e a complexidade do espaço poderia ser $O(N^2)$ ou menos, sendo que existem algumas exceções como verificado nos algoritmos Relief e DTM (LIU; YU, 2002). Os algoritmos de busca heurística são considerados muito simples para implementar e muito rápidos na produção de resultados porque o espaço de busca só é quadrático em termos do número de atributos.

c) Busca Não-Determinística: adotando-se esta estratégia, a busca pelo próximo conjunto é realizada de forma aleatória. E, embora o espaço de busca seja ainda $O(2^N)$, esta estratégia busca por um número de subconjuntos menor que 2^N através da definição de um número máximo de iterações possíveis.

3.2.2.2 Função de avaliação

Uma função de avaliação é responsável por medir a boa qualidade de um subconjunto produzido por algum procedimento de geração, sendo este valor comparado com o do subconjunto anteriormente considerado melhor. Caso o atual seja melhor, então o mesmo substituirá o anterior.

O subconjunto ótimo está sempre relacionado a um determinado critério de avaliação, isto é, um subconjunto ótimo escolhido por algum critério de avaliação pode não ser o mesmo quando utilizando outro critério de avaliação. Considerando-se a dependência do critério de avaliação em relação ao algoritmo de aprendizagem aplicado sobre o subconjunto de atributos selecionado, tem-se que os critérios de avaliação podem ser divididos em dois grupos:

a) Critérios Independentes

Comumente utilizados pelos métodos de seleção filtro (*filter*), tentam avaliar a boa qualidade de um atributo ou subconjunto de atributos sem o envolvimento de um algoritmo de aprendizagem no processo. Como exemplos de tais critérios tem-se medidas de distância, de informação, de dependência e de consistência.

[3] Monotonicidade refere-se ao fato que um subconjunto de atributos não deve ser melhor que qualquer conjunto maior que contenha o subconjunto (DASH; LIU, 1997).

- Medidas de Distância: também conhecidas por medidas de separabilidade, divergência ou discriminação. Considerando um problema de duas classes, um atributo X é preferido a outro atributo Y se X induz uma maior diferença entre as probabilidades condicionais de duas-classes que Y; se a diferença é zero, então X e Y não se apresentam como distintos. Como exemplo tem-se a medida de distância euclidiana.

- Medidas de Informação: determinam o ganho de informação de um atributo. O ganho de informação é entendido como a diferença entre a incerteza anterior e a incerteza posterior utilizando X. O atributo X é preferido a um atributo Y se o ganho de informação do atributo X é maior que o de Y, como exemplo tem-se a medida de entropia.

- Medidas de Dependência: também conhecidas por medidas de correlação, qualificam a habilidade para predizer o valor de um atributo a partir do valor de outro. O coeficiente é uma medida de dependência clássica e pode ser utilizado para encontrar a correlação entre um atributo e uma classe. Caso a correlação de um atributo X com a classe C seja maior que a correlação do atributo Y com C, então o atributo X é preferido a Y. Todas as funções de avaliação baseadas em medidas de dependência podem ser divididas em medidas de distância e de informação.

- Medidas de Consistência: estas são medidas mais recentes e diferem das demais medidas devido a sua total confiança no conjunto de treinamento e pelo uso do *bias*[4] *Min-Features* na seleção de um subconjunto de atributos. O *bias Min-Features* prefere hipóteses consistentes definidas sobre um conjunto mínimo de atributos. Estas medidas encontram o subconjunto de tamanho mínimo que satisfaça uma taxa de inconsistência aceitável, que é freqüentemente definida pelo usuário.

b) Critérios Dependentes

Utilizados pelos métodos de seleção *wrapper*, tentam avaliar a boa qualidade de um atributo ou subconjunto de atributos pela avaliação do desempenho do algoritmo de aprendizagem aplicado sobre o subconjunto selecionado, ou seja, é a mesma medida de desempenho do algoritmo de aprendizagem aplicado. Como exemplos de tais critérios tem-se as medidas para classificação e as medidas para *clustering.*

- Medidas para Classificação: têm o objetivo principal de maximizar a acurácia preditiva (utilizando geralmente como critério de avaliação a acurácia e a taxa de erro de um classificador).

- Medidas para *Clustering*: estimam a qualidade dos resultados de *clustering*.

Em relação a avaliação dos subconjuntos de atributos alternativos é preciso selecionar alguma estratégia para tal, que pode ser escolhida dentre métodos filtro (*filter)* e *wrapper.*

a) Seleção Filtro

Os métodos de filtro tentam encontrar o conjunto de atributos relevantes antes do processo de aprendizagem indutiva através da seleção de alguns atributos e exclusão

[4] *Bias* de um modo geral significa tendenciosidade (CAMARÃO, 1994). Corresponde a qualquer preferência de uma hipótese (descrição de conceito) sobre outra, além da simples consistência com as instâncias (RUSSEL; NORVIG, 1995 apud BARANAUSKAS, 2001, p. 23)

de outros irrelevantes e/ ou redundantes. Assim o algoritmo de indução recebe como entrada o conjunto de dados contendo apenas os atributos selecionados pelo filtro.

Embora exista a possibilidade de aplicar filtros dividindo o conjunto de dados em subconjunto de treinamento e teste, geralmente todo o conjunto de dados é submetido ao filtro (BARANAUSKAS, 2001).

Para a seleção filtro são consideradas as características gerais do conjunto de dados, o que caracteriza tais métodos como independentes do algoritmo de indução (BLUM; LANGLEY, 1997).

Por um lado isto resulta em uma desvantagem dos métodos de filtro como apresentado em John; Kohavi; Pfieger (1994), já que eles ignoram totalmente os efeitos do subconjunto de atributos selecionado no desempenho do algoritmo de indução, pois para determinar um subconjunto útil de atributos o algoritmo de seleção de subconjuntos deve considerar o *bias* do algoritmo de indução de modo a selecionar um subconjunto com alta acurácia preditiva em dados não vistos.

b) Seleção *Wrapper*

Este tipo de seleção utiliza o algoritmo de indução como uma caixa preta, ou seja, o próprio indutor é executado utilizando um determinado subconjunto de atributos como candidato a ser o subconjunto escolhido, após esta indução é verificada a acurácia dos classificadores induzidos para avaliar o subconjunto de atributos em questão. Este processo é repetido para cada subconjunto candidato até que algum critério de parada seja satisfeito.

Deste modo, na seleção *wrapper* o classificador é a própria função de avaliação, ou seja, o classificador seleciona os atributos e posteriormente os utiliza na predição de rótulos de classes de instâncias não vistas, o que torna o nível de acurácia muito alto embora seja também computacionalmente dispendioso.

Segundo Blum; Langley (1997), a questão da boa estimativa de acurácia deve-se ao fato que é utilizado o mesmo algoritmo de indução para selecionar os atributos e posteriormente induzir sobre o subconjunto de atributos já selecionado. Enquanto que, o custo computacional pode ser muito alto em razão do algoritmo de indução ser solicitado para cada conjunto de atributos considerado, o que não é muito apropriado no caso da manipulação de um grande número de atributos.

Conforme Langley (1994 apud DASH; LIU, 1997, p. 05), os métodos de seleção de atributos são dispostos nestes dois grupos apresentados acima, filtro e *wrapper*, considerando-se a dependência de tais métodos em relação ao algoritmo de indução que finalmente utilizará o subconjunto selecionado. Complementarmente, em Liu; Yu (2002) são apresentados algoritmos pertencentes a um terceiro grupo de métodos que utilizam uma seleção híbrida, ou seja, combinam vantagens dos modelos filtro e *wrapper*.

O Anexo 1 apresenta uma descrição destes métodos dentro de suas respectivas abordagens supervisionada e não-supervisionada. E na Tabela 3.1 é exibido um resumo das características de tais métodos.

TABELA 3.1 – Comparação entre Métodos de Seleção de Atributos (continua)

Método	**Abordagem**	**Estratégia de Avaliação**	**Estratégia de Busca**	**Critério de Avaliação**	**Tipo de Dados**	**Manipula Dados Errôneos?**	**Tipo de Problema de MD**	**Técnica de MD**	**Número de Classes**	**Gera um Subconj. Ótimo?**	**Manipula Conjunto de Dados Grande?**
FOCUS	Supervisionada	Filtro	Completa	Medidas de consistência	Simbólicos	Não	Classificação	Árvore de decisão	Mínimo duas	Sim	Não
De Schlimmer	Supervisionada	Filtro	Completa	Medidas de consistência	Simbólicos (booleanos)	Não	Classificação	Árvore de decisão	Mínimo duas	Sim	Não
MIFES-1	Supervisionada	Filtro	Completa	Medidas de consistência	Simbólicos (booleanos)	Não	Classificação	Árvore de decisão	Mínimo duas	Sim	Não
RELIEF	Supervisionada	Filtro	Heurística	Medidas de distância	Contínuos e simbólicos	Sim	Classificação	*naive-bayes*	Somente duas	Não	Sim
B & B	Supervisionada	Filtro	Completa	Medidas de distância	Contínuos e simbólicos	-	Classificação	-	Mínimo duas	Sim	-
BFF	Supervisionada	Filtro	Completa	Medidas de distância	Contínuos e simbólicos	-	Classificação	Árvore de decisão	Mínimo duas	Sim	-
De Bobrowski	Supervisionada	Filtro	Completa	Medidas de distância	Contínuos e simbólicos	-	Classificação	-	Mínimo duas	Sim	-
SFG	Supervisionada	Filtro	Heurística	Medidas de informação	-	-	Classificação	-	-	-	-
DTM	Supervisionada	Filtro	Heurística	Medidas de informação	Contínuos e simbólicos	-	Classificação	Baseada em instância	Mínimo duas	Não	Sim

TABELA 3.1 – Comparação entre Métodos de Seleção de Atributos (continuação)

Método	**Abordagem**	**Estratégia de Avaliação**	**Estratégia de Busca**	**Critério de Avaliação**	**Tipo de Dados**	**Manipula Dados Errôneos?**	**Tipo de Problema de MD**	**Técnica de MD**	**Número de Classes**	**Gera um Subconj. Ótimo?**	**Manipula Conjunto de Dados Grande?**
De Koller; Sahami	Supervisionada	Filtro	Heurística	Medidas de informação	Simbólicos	-	Classificação	Árvore de decisão, *naive-bayes*	Mínimo duas	Não	Sim
MDLM	Supervisionada	Filtro	Completa	Medidas de informação	Contínuos e simbólicos	-	Classificação	-	Mínimo duas	Não	-
POE + ACC	Supervisionada	Filtro	Heurística	Medidas de dependência	Contínuos e simbólicos	-	Classificação	-	Mínimo duas	Não	-
PRESET	Supervisionada	Filtro	Heurística	Medidas de dependência	Contínuos e simbólicos	-	Classificação	-	Mínimo duas	Não	Sim
LVF	Supervisionada	Filtro	Não-Determínistica	Medidas de consistência	Simbólicos	Sim	Classificação	Árvore de decisão	Mínimo duas	Sim	Sim
SBUD	Não-Supervisionada	Filtro	Heurística	Medidas de informação	-	-	*clustering*	-	-	-	-
De Mitra	Não-Supervisionada	Filtro	Heurística	Medidas de distância	-	-	*clustering*	-	-	-	-
De Ichino; Sklansky	Supervisionada	*wrapper*	Completa	Medidas para classificação	-	-	Classificação	-	-	-	-
AMB & B	Supervisionada	*wrapper*	Completa	Medidas para classificação	-	-	Classificação	-	-	-	-
BS	Supervisionada	*wrapper*	Completa	Medidas para classificação	-	-	-	-	-	-	-
WSFG	Supervisionada	*wrapper*	Heurística	Medidas para classificação	-	-	Classificação	-	-	-	-

TABELA 3.1 – Comparação entre Métodos de Seleção de Atributos (continuação)

Método	**Abordagem**	**Estratégia de Avaliação**	**Estratégia de Busca**	**Critério de Avaliação**	**Tipo de Dados**	**Manipula Dados Errôneos?**	**Tipo de Problema de MD**	**Técnica de MD**	**Número de Classes**	**Gera um Subconj. Ótimo?**	**Manipula Conjunto de Dados Grande?**
WSBG	Supervisionada	*wrapper*	Heurística	Medidas para classificação	-	-	Classificação	-	-	-	-
SBS-SLASH	Supervisionada	*wrapper*	Heurística	Medidas para classificação	-	-	Classificação	Árvore de decisão	-	-	-
BDS	Supervisionada	*wrapper*	Heurística	Medidas para classificação	-	-	-	-	-	-	-
PQSS	Supervisionada	*wrapper*	Heurística	Medidas para classificação	-	-	-	-	-	-	-
Schemata	Supervisionada	*wrapper*	Heurística	Medidas para classificação	-	-	-	-	-	-	-
RC	Supervisionada	*wrapper*	Heurística	Medidas para classificação	Contínuos, simbólicos	Sim	-	-	Mínimo duas	-	-
De Queiroz; Gelsema	Supervisionada	*wrapper*	Heurística	Medidas para classificação	-	-	Classificação	*Naive-bayes*	-	-	-
De Devaney; Ram	Não-supervisionada	*wrapper*	Heurística	Medidas para *clustering*	Contínuos, Simbólicos	-	*Clustering*	*Clustering* conceitual	Mínimo duas	-	-
FSSEM	Não-supervisionada	*wrapper*	Heurística	Medidas para *clustering*	-	-	-	-	-	-	-
ELSA	Não-supervisionada	*wrapper*	Heurística	Medidas para *clustering*	Contínuos	Sim	*Clustering*	k-medias	-	-	-

TABELA 3.1 – Comparação entre Métodos de Seleção de Atributos (continuação)

Método	**Abordagem**	**Estratégia de Avaliação**	**Estratégia de Busca**	**Critério de Avaliação**	**Tipo de Dados**	**Manipula Dados Errôneos?**	**Tipo de Problema de MD**	**Técnica de MD**	**Número de Classes**	**Gera um Subconj. Ótimo?**	**Manipula Conjunto de Dados Grande?**
BBHFS	Supervisionada	Híbrida	Heurística	Medidas de informação + medidas para classificação	-	-	Classificação	Naive-bayes, árvore de decisão, K-NN	-	-	-
De Xing et al.	Supervisionada	Híbrida	Heurística	Medidas de informação + medidas para classificação	-	-	-	-	-	-	-
De Dash; Liu	Não-Supervisionada	Híbrida	Heurística	Medidas de informação + medidas para *clustering*	-	-	*Clustering*	k-médias	-	-	-

- não informado

3.2.2.3 Critério de parada

Existe a necessidade do estabelecimento de um critério de parada para que o processo de seleção de atributo não seja executado exaustivamente ou infinitamente através do espaço de subconjuntos. Considerando-se que os procedimentos de geração e as funções de avaliação podem influenciar na escolha do critério de parada, tem-se que:

a) Critério de parada baseado no procedimento de geração é identificado como tal, se um número pré-definido de atributos é selecionado ou se um número pré-definido de iterações é alcançado.

b) Critério de parada baseado na função de avaliação é identificado como tal, se a adição (ou remoção) de qualquer atributo não produz um subconjunto melhor ou se um subconjunto ótimo é obtido de acordo com alguma função de avaliação.

Assim o *loop* continua até algum critério de parada ser satisfeito.

3.2.2.4 Procedimento de validação

Apesar deste procedimento não fazer parte do processo de seleção em si, um método de seleção de atributos deve ser validado, o que pode ser conseguido comparando os resultados com outros anteriormente estabelecidos ou com os resultados de métodos de seleção de atributos utilizando conjuntos de dados artificiais, reais ou ambos.

3.2.3 Seleção de Instâncias

Idealmente, é desejável considerar todas as instâncias juntas para um melhor desempenho de aprendizagem, porém quando o conjunto de dados é muito grande, nem todos os dados podem ser "carregados" na memória (SYED; LIU; SUNG, 1999). Além do problema da capacidade de memória, existe ainda a questão da relevância de instâncias, que segundo Riaño (1997), assim como ocorre a dimensionalidade excessiva ou super-dimensionalidade tratando-se do número de atributos, em um conjunto de dados, também ocorre a cardinalidade excessiva ou super-cardinalidade.

Entendendo-se que, a cardinalidade corresponde a visualização do conjunto de dados como uma lista de instâncias, exemplos ou observações. A cardinalidade excessiva refere-se então a situação em que o número de instâncias é maior que o número de instâncias realmente necessárias ao propósito de aprendizagem.

Deste modo, recomenda-se realizar uma seleção de instâncias sobre um conjunto de dados devido a duas situações naturais, conforme Riaño (1997):

- As pessoas que manipulam conjuntos de dados são atentas aos atributos utilizados em um certo domínio, mas não têm ciência de todas as instâncias introduzidas, especialmente para grandes conjuntos de dados.

- No caso em que os dados são obtidos a partir de algum processo automático (por exemplo, por informação de sensor), a probabilidade de introduzir informação redundante é maior que pela introdução de dados controlados por humanos.

Portanto, tem-se que a seleção de instâncias, assim como a de atributos, pode ser realizada "manualmente" através da escolha direta pelo especialista do domínio ou através de algoritmos.

a) Seleção Manual: neste trabalho é considerado que podem ser selecionadas algumas instâncias de um conjunto de instâncias conforme os objetivos de mineração de dados a serem alcançados e especificações do usuário. Assim, pode-se escolher instâncias de um determinado período de tempo ou que apresentem um determinado valor ou conjunto de valores para qualquer outro atributo considerado relevante.

b) Seleção por Algoritmos: muita pesquisa tem sido desenvolvida sobre algoritmos que selecionam um subconjunto de atributos reduzido para facilitar o processo de aprendizagem, porém pouca referência é encontrada a respeito de redução do número de instâncias o que justifica o fato de serem descritos neste trabalho menos métodos relativos a seleção de instâncias que os de seleção de atributos.

Segundo Riaño (1997), em termos de aprendizagem supervisionada, os algoritmos destinados a seleção de instâncias podem ser classificados dentro de quatro categorias de estratégia de busca: métodos exaustivos, métodos heurísticos, métodos de convergência e métodos não-determinísticos ou probabilísticos.

Os métodos exaustivos caracterizam-se por tentar selecionar um subconjunto de instâncias através de uma busca realizada sobre o espaço de instâncias disponíveis, o que acarreta grande consumo de espaço de memória e tempo computacional.

Já os métodos heurísticos atuam de acordo com um critério de decisão pré-definido entre as instâncias disponíveis, de modo que em cada passo, a instância que maximiza cada critério de decisão é considerada como pertencente ao subconjunto final.

Os métodos de convergência associam um peso a cada instância que é apropriadamente alterado através de um processo iterativo. Ao final do processo, somente aquelas instâncias com um peso acima de um determinado limiar são consideradas como pertencentes ao subconjunto final.

Quanto aos métodos não-determinísticos ou probabilísticos, estes têm por objetivo obter o conjunto de instâncias relevantes como um processo aleatório que repetidamente conduz à solução. Em cada iteração, um número aleatório é selecionado, então uma quantidade limitada de conjuntos contendo este número de instâncias é construída, e o que apresentar maior acurácia é mantido. O processo é repetido um número pré-definido de vezes, e finalmente o melhor conjunto de instâncias obtido é dado.

No Anexo 2 são descritos alguns métodos de seleção de instâncias cujas características são apresentadas de forma sucinta na Tabela 3.2.

TABELA 3.2 – Comparação entre Métodos de Seleção de Instâncias

Método	Abordagem	Estratégia de Busca	Estratégia de Avaliação
IFOCUS	Supervisionada	Exaustiva	Filtro
Forward	Supervisionada	Heurística	Filtro
ISET	Supervisionada	Heurística	Filtro
IRET	Supervisionada	Convergência	Filtro
IPRO1	Supervisionada	Não-determinística	Filtro
IPRO2	Supervisionada	Não-determinística	Filtro
Windowing	Supervisionada	Não-determinística	*Wrapper*

3.3 Limpeza de Dados

A limpeza de dados visa garantir a qualidade dos mesmos, a qual constitui um fator muito importante a ser mantido em aplicações de mineração de dados.

Segundo Galhardas et al. (2000-a), os problemas de qualidade de dados podem ser encontrados em dados de uma única fonte, como arquivos e banco de dados, e especialmente quando tratando de dados de múltiplas fontes como os que serão integrados em *data warehouses.*

Tratando-se de dados de uma única fonte tem-se problemas a nível de esquema e a nível de instância de dados. Os problemas a nível de esquema[5] freqüentemente ocorrem devido a um projeto de má qualidade de esquema ou falta de restrições de integridade (em modelos relacionais e orientados a objeto) ou devido a limitações no modelo de dados[6] (por exemplo, manipulando arquivos). Os problemas a nível de instância referem-se a erros e inconsistências no conteúdo do dado atual, isto é muito freqüente e geralmente deve-se a ocorrência de valores ausentes, abreviações e transposições de palavras, referências erradas, duplicação de instâncias, etc.

Em termos de dados de múltiplas fontes, os problemas freqüentemente aparecem devido a modelos de dados e projetos de esquemas heterogêneos. Assim, caso nenhuma notação padrão seja imposta existe a possibilidade de que dados de diferentes fontes possam ser diferentemente representados, permitindo por exemplo, que um atributo apresente informação sob vários formatos. Tais inconsistências geradas durante a combinação de dados de múltiplas fontes devem ser resolvidas.

Visando garantir a qualidade dos dados é preciso aplicar operações de limpeza de dados de modo que estes sejam apresentados em uma forma apropriada aos algoritmos de mineração de dados.

As principais operações de limpeza que podem ser utilizadas são descritas a seguir.

[5] Esquema de dados é a apresentação do modelo de dados (HEUSER, 2000, p. 05).
[6] Modelo de dados é a descrição formal da estrutura de um banco de dados (HEUSER, 2000, p. 05).

3.3.1 Eliminação de Dados Errôneos

Neste trabalho, considera-se como dados errôneos as instâncias que contém valores de atributos comprometidos por problemas causados durante a importação de tabelas, queda de tensão, ou ainda valores que excedem uma determinada faixa, por exemplo um valor altíssimo para o atributo idade de uma pessoa.

Os valores errôneos nos dados podem ocorrer conforme o citado abaixo:

- Falhas nos meios utilizados para obtenção de dados devido a erros humanos ou do computador cometidos na entrada de dados;

- Erros na transmissão de dados também podem ocorrer devido a limitações tecnológicas, tais como tamanho de *buffer* limitado para coordenar transferência e consumo de dados sincronizados.

Algumas técnicas de detecção e eliminação de dados errôneos que podem ser adotadas são as seguintes, conforme Han; Kamber (2001):

a) Criação de Intervalos (*Binning*): os dados para um determinado atributo que contenha valores errôneos são primeiramente classificados e então distribuídos dentro de um número de "*buckets*" ou intervalos (*bins*). Dentre as técnicas de criação de intervalos (*binning*) destacam-se limpeza por média de intervalos (*smoothing by bin medians*), que consiste em substituir cada valor do intervalo pela média do mesmo, e a limpeza por limites de intervalo *(smoothing by bin boundaries)*, que considera os valores máximo e mínimo de um determinado intervalo como os limites do mesmo, substituindo cada valor do intervalo pelo valor do limite mais próximo. Estas técnicas podem ser visualizadas através de um exemplo, descrito na Figura 3.4.

Dados classificados pelo preço (em dólares): 4, 8, 15, 21, 21, 24, 25, 28, 34

Partição dentro de intervalos:
Intervalo 1: 4, 8, 15
Intervalo 2: 21, 21, 24
Intervalo 3: 25, 28, 34

Limpeza por média de intervalo (*smoothing by bin medians*):
Intervalo 1: 9, 9, 9
Intervalo 2: 22, 22, 22
Intervalo 3: 29,29,29

Limpeza por limites de intervalo (*smoothing by bin boundaries*):
Intervalo 1: 4, 4, 15
Intervalo 2: 21, 21, 24
Intervalo 3: 25, 25, 34

FIGURA 3.4 – Métodos de Criação de Intervalos para Limpeza de Dados
Fonte: HAN; KAMBER, 2001. p. 110

b) *Clustering*: valores similares são organizados dentro de grupos, ou *clusters*, e os valores fora do conjunto de *clusters* podem ser considerados errôneos.

c) Inspeção humana combinada ao computador: os valores excedentes de uma determinada faixa podem ser colocados em uma lista e o especialista do domínio pode então examiná-los para identificar o que realmente é errôneo e deve ser excluído.

d) Regressão: os dados podem ser limpos por ajustar o dado a uma função, tal como através de regressão. Existem dois tipos de regressão que podem ser utilizadas: a regressão linear e regressão linear múltipla. Na regressão linear, o objetivo é encontrar a melhor linha para ajustar dois atributos de modo que um atributo pode ser usado para predizer o outro. Assim, os dados são modelados para ajustar uma linha diretamente. Por exemplo, um atributo aleatório Y (chamado atributo de resposta), pode ser modelado como uma função linear de outro atributo aleatório X (chamado atributo preditor) através da equação:

$$Y = \alpha + \beta X$$

Onde a variância[7] de Y é uma constante e os coeficientes α e β (chamados coeficientes de regressão) especificam a interseção e declividade da linha em relação a Y, respectivamente. Estes coeficientes podem ser resolvidos pelo método de mínimos quadrados, que minimiza o erro entre o dado atual e o estimado da linha. Dados os pontos de dados da forma (x_1, y_1), (x_2, y_2), ..., (x_s,y_s), então os coeficientes de regressão podem ser estimados usando este método com as seguintes equações (HAN; KAMBER, 2001):

$$\beta = \frac{\sum_{i=1}^{s}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{s}(x_i - \bar{x})^2}$$

$$\alpha = \bar{y} - \beta \bar{x}$$

onde $\bar{x}$ é a média de $x_1, x_2, ..., x_s$, e $\bar{y}$ é a média de $y_1, y_2, ..., y_s$.

Já a regressão linear múltipla é uma extensão da regressão linear, onde mais de dois atributos são envolvidos e os dados são ajustados a uma superfície multidimensional, isto é, um atributo de resposta Y é modelado como uma função linear de um vetor de atributos multidimensional. Um exemplo de um modelo de regressão linear múltipla baseado em dois atributos ou variáveis preditores, X_1 e X_2, é:

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2$$

Igualmente pode ser aplicado o método de mínimos quadrados para obter os valores de α, β_1 e β_2.

[7] Variância: Define-se a variância como sendo a medida que se obtém somando os quadrados dos desvios das instâncias da amostra, relativamente à sua média, e dividindo pelo número de instâncias da amostra menos um.

3.3.2 Padronização de Dados

Segundo Hsu et al. (2000), o problema de dados não padronizados em banco de dados deve-se ao fato de que com as rápidas atualizações de *software* e *hardware* dentro de um curto período de tempo ocorrem também significativas mudanças de formatação. Assim, com tais mudanças de formato são geradas inconsistências como por exemplo, formatos de datas tais como, dd/mm/aa ou mm/dd/aa e uso de abreviações do tipo, "um" versus "1".

No sistema de limpeza de dados implementado por Hsu et al. (2000) é possível tratar estes problemas de padronização, pois ele permite ao usuário definir mapeamentos entre atributos de formatos diferentes, o esquema usado é codificado e os atributos podem ser mantidos limpos no banco de dados. Com esta especificação, um esquema de formato padronizado final é gerado. E, baseado neste esquema de formato padronizado cada um dos arquivos de banco de dados é transformado dentro deste formato padronizado.

De acordo com Félix (1998), em casos por exemplo, em que o sexo das pessoas seja especificado através de valores M ou F, 0 ou 1, Mas ou Fem, é recomendável transformar sempre que for possível, em valores numéricos, no exemplo dado poderia ser padronizado para 0 e 1. Isto facilitaria o tratamento do dado pelo algoritmo de aprendizagem, no caso de ser uma exigência do mesmo trabalhar com números, embora para o usuário seja mais intuitivo utilizar Mas ou Fem, M ou F.

3.3.3 Eliminação de Duplicatas

A eliminação de duplicatas ou de instâncias duplicadas é considerada uma tarefa muito importante na fase de limpeza de dados, já que é possível encontrar em banco de dados algumas entidades do mundo real representadas por várias instâncias muitas das vezes por não serem sintaticamente equivalentes. Algumas razões para tal ocorrência podem ser citadas abaixo:

a) Valores incorretos ou ausentes devido a erros de entrada de dados;

b) Formatos diferentes de valores de entrada causando inconsistências nas convenções de nomes;

c) Informação incompleta devido o dado não ter sido obtido ou não estar disponível;

d) Informações incorretas ou não atualizadas.

Para remover instâncias duplicadas de um banco de dados é preciso comparar instâncias a partir dos atributos correspondentes para determinar o grau de similaridade dos mesmos tendo por base seus valores sintáticos, como exemplo pode ser citado um algoritmo específico de domínio chamado algoritmo Smith-Waterman que compara DNA e seqüências proteicas.

Existem *softwares*, tal como o Access, que disponibilizam recursos para detecção e eliminação de duplicatas. Além disso, podem ser citados alguns métodos destinados à tais propósitos, conforme Lee et al. (1999) e Hernandez; Stolfo (1995):

1) Método Padrão de Bitton e DeWitt (1995 apud LEE et al., 1999, p. 03)

Consiste em classificar o banco de dados e verificar se instâncias próximas são idênticas. Sendo que, a forma mais confiável para detectar duplicatas próximas é comparar cada instância com outra instância no banco de dados, porém este é um processo muito lento que requer N(N – 1)/ 2 comparações de instância onde N é o número de instâncias no banco de dados.

2) Método SNM (*Sorted Neighbourhood Method*) de Hernandez; Stolfo (1995 apud LEE et al., 1999, p. 03)

Este detecta duplicatas próximas a partir da classificação do banco de dados por uma chave escolhida específica da aplicação para apresentar instâncias "potencialmente equivalentes" dentro de uma vizinhança restrita. Devendo ser ressaltado que não existe nenhuma regra a ser seguida quanto a forma como esta chave deve ser criada, podendo a mesma ser uma seqüência de um subconjunto de atributos ou *substrings* dentro dos atributos com potência discriminante suficiente na identificação apropriada de candidatos para equivalência.

Após a definição da chave as comparações de pares de instâncias próximas são feitas em uma janela de tamanho fixo onde são apresentadas as instâncias classificadas do banco de dados. Considerando-se que o tamanho da janela seja w instâncias, então cada nova instância adicionada na janela é comparada com o w – 1 instância para encontrar instâncias iguais. O método SNM é mais rápido já que requer somente wN comparações, porém sua eficiência depende da qualidade das chaves escolhidas (LEE et al., 1999).

3) Método DE – SNM (*Duplication Elimination SNM*) de Hernandez (1995 apud LEE et al., 1999, p. 03-04)

O método DE – SNM melhora os resultados de SNM devido a primeiramente classificar as instâncias por uma chave escolhida e então dividir as instâncias classificadas em duas listas: uma lista de duplicatas e outra lista de não-duplicatas. A lista de duplicatas contém todas as instâncias com chaves duplicadas, as demais instâncias são repassadas para a lista de não-duplicatas. Então uma pequena "varredura" na janela é feita sobre a lista de duplicatas com o objetivo de encontrar as listas de instâncias equivalentes ou não equivalentes. A lista de instâncias não-equivalentes é combinada com a lista original de não-duplicatas e uma segunda "varredura" na janela é realizada, porém o problema que pode ser encontrado utilizando o SNM ainda pode ocorrer com o DE – SNM.

4) Método proposto por Lee et al. (1999)

Este método agrega primeiramente a padronização dos dados e formação de grupos significativos de valores *strings* dentro dos atributos com posterior classificação destes grupos, sendo estes aspectos configurados como um diferencial do método de Lee et al. (1999) em relação aos métodos anteriormente citados. Em seguida, as instâncias devem ser classificadas para futura comparação.

Para a comparação de instâncias, Lee et al. (1999) propõem um método denominado *field weightage* que indica a importância relativa de um atributo para computar o grau de similaridade entre duas instâncias. O *field weightage* é fornecido pelo especialista do domínio e a soma de todos os *field weightages* deve ser igual a 1.

Assim, por exemplo, se o especialista quer eliminar instâncias duplicadas baseado nos atributos nome e endereço então ele deve associar um *weightage* de 0.5 para cada um dos dois atributos e 0 para os outros atributos da instância.

Ressalta-se que o processo de verificação da similaridade entre duas instâncias inicia comparando-se os grupos classificados dos atributos correspondentes. Os grupos são comparados usando equivalência exata de *string*, equivalência de erro simples, equivalência de abreviação e equivalência de prefixo. A partir da comparação dos grupos de atributo, a similaridade entre o atributo inteiro é verificada, posteriormente a similaridade de instância pode ser computada a partir da similaridade de atributos e do *weightage* de atributos.

Finalmente, instâncias equivalentes, consideradas fonte parcial de informação, são combinadas para obter uma instância com informação mais completa.

5) Método *Multi-Pass Neighborhood* de Hernandez; Stolfo (1995):

Utiliza-se este método para detectar o número máximo de instâncias duplicadas exatas ou aproximadas (devido a erros ou a falta de padronização de atributo) em menos tempo.

Após ter concatenado todas as instâncias a serem limpas dentro de um arquivo simples, este método consiste na repetição dos seguintes passos: primeiro, escolher uma chave (consistindo de um ou vários atributos, ou *substrings* dentro de atributos) e classificar as instâncias pela mesma. Segundo, comparar estas instâncias que estão dentro de uma janela pequena e de tamanho fixo.

O critério para comparação de instâncias de modo a encontrar duplicatas aproximadas é definido através de um conjunto de regras de equivalência codificadas em uma linguagem de programação (por exemplo, C). Sendo que, cada execução destes dois passos descritos anteriormente (cada vez com uma chave diferente) gera um conjunto de pares de instâncias equivalentes. Um agrupamento por transitividade (se a instância I1 é duplicata da instância I2 e a instância I2 é uma duplicata da instância I3, então por transitividade I1 é uma duplicata de I3) é feito sobre estes pares de instâncias, produzindo uma união de todos os pares gerados por todas as execuções independentes, mais todos os outros pares que podem ser inferidos por transitividade de igualdade.

Como citado em HSU et al. (2000), 70% das instâncias duplicadas são detectadas e removidas. Isto porque na prática é muito difícil detectar todas as duplicatas em um conjunto de dados.

3.3.4 Tratamento de Valores Ausentes

A ocorrência de valores ausentes deve-se geralmente a erros humanos, não disponibilidade da informação no momento de entrada de dados ou porque o atributo não é de preenchimento obrigatório.

Para tratar a ausência de valores de atributos tem-se as seguintes alternativas:

1) Excluir Instâncias

Diante de um conjunto de dados com uma quantidade muito grande de instâncias é possível excluir aquelas que para um ou mais atributos apresentem valores ausentes, não representando uma perda de informação muito significativa. Porém, se o conjunto de dados for muito pequeno esta eliminação de instâncias pode comprometer o resultado final.

2) Completar os Valores Ausentes

Existem várias formas de se complementar valores ausentes de atributos tais como as descritas a seguir, conforme Han; Kamber (2001) e Ragel (1998):

2.1) Complemento Manual: esta alternativa consome muito tempo tornando-se inviável quando se trata de um conjunto de dados muito grande com muitos valores ausentes.

2.2) Complemento com uma constante global: Considerar o valor ausente como um valor regular adicional para um determinado atributo, adotando algum valor neutro para preenchê-lo como o termo “desconhecido” ou “nenhum”.

2.3) Complementar com o valor mais provável: que pode ser realizado utilizando regressão, ferramentas baseadas em inferências usando formalismo bayesiano ou indução de árvores de decisão.

2.4) Método Majoritário de Kononenko; Roscar (1984 apud LOBO; NUMAO, 1999, p. 502): que considera o valor mais freqüente do atributo como um bom candidato para complementar o valor ausente. Sendo que, este método foi aprimorado no sentido de utilizar o valor mais freqüente do atributo para a classe da instância que apresenta valor ausente.

2.5) Complementar com o valor médio do atributo: a complementação ocorre considerando o conjunto de dados inteiro, ou todas as instâncias pertencentes a mesma classe que a instância que apresenta valor ausente para determinado atributo.

2.6) Método MVC (*Missing Values Completion*) proposto por Ragel (1998) consiste em dois passos:

a) Geração de todas as regras de associação através do algoritmo RAR (*Robust Association Rules*) que permite manipular banco de dados com múltiplos valores ausentes. O objetivo de RAR é descobrir regras em conjuntos de dados válidos, ou seja, em subconjuntos do conjunto de dados maior que não apresentem valores ausentes.

b) Complemento de atributos com valores ausentes, utilizando a conclusão obtida das regras geradas para um atributo de valor ausente. Sendo que, deve ser considerado o seguinte: caso todas as regras obtidas indiquem a mesma conclusão, então esta deve ser usada, senão se várias regras concluem valores diferentes deve ser

utilizada a medida de confiança e o número de regras que concluem um mesmo valor para ajudar na escolha do valor a ser adotado.

O método MVC permite que um atributo após ter tido seu valor descoberto e apresentando alta confiança seja utilizado na descoberta do valor ausente de outro atributo e que o especialista do domínio interfira no processo de complementação, adicionando novas regras, removendo ou modificando as já descobertas.

3.4 Transformação de Dados

Geralmente os algoritmos utilizados na mineração de dados (MD) requerem que os dados se apresentem em um formato apropriado, existindo assim a necessidade da aplicação de operações de transformação destes dados na fase de pré-processamento. Dentre elas, podem ser citadas:

3.4.1 Normalização de Dados

Segundo Han; Kamber (2001), a normalização consiste em converter os valores de atributos para faixas de –1 a 1 ou de 0 a 1, sendo de grande utilidade para algoritmos de classificação como redes neurais ou baseados em distância tais como, *nearest-neighbor* e de *clustering.*

Existem vários métodos de normalização de dados, conforme Han; Kamber (2001) e Witten; Frank (2000):

a) Normalização min-max: que realiza uma transformação linear do dado original, considerando $\min_A$ e $\max_A$ os valores mínimo e máximo de um atributo A, e que um valor v de A é mapeado para *v'* na faixa [novo_$\min_A$, novo_$\max_A$]

$$v' = \frac{v - \min_A}{\max_A - \min_A}\left(\text{novo\_max}_A - \text{novo\_min}_A\right) + novo\_\min_A$$

b) Nomalização z-score: que permite a conversão dos valores de um atributo A baseada na média e desvio padrão deste atributo, sendo um valor *v* de A normalizado para *v'* através de

$$v' = \frac{v - \overline{A}}{\sigma_A}$$

onde $\overline{A}$ e σ_A correspondem respectivamente, a média e desvio padrão do atributo A.

Este método é considerado muito útil quando os valores mínimo e máximo do atributo são desconhecidos.

c) Normalização por escalamento decimal: nesta o ponto decimal dos valores de A é movido. Sendo que o número de pontos decimal movido depende do valor máximo absoluto de A, e um valor *v* de A é normalizado para *v'* da seguinte forma:

$$v' = \frac{v}{10^{j}}$$

onde *j* é o menor inteiro tal que $\text{Max}(|v'|) < 1$.

d) Normalização através da divisão de todos os valores do atributo pelo valor máximo encontrado no mesmo: considerando-se um atributo A, seus valores podem ser normalizados através de

$$v' = \frac{v}{\max_A}$$

3.4.2 Conversões de Valores Simbólicos para Numéricos

A manipulação de tipos de atributos diferentes, numéricos e simbólicos, depende da capacidade e necessidade da técnica de mineração utilizada, já que algumas técnicas como árvores de decisão, podem manipular valores simbólicos enquanto outras tais como redes neurais, podem manipular somente uma representação numérica de um valor simbólico.

Devido ao fato de todas as técnicas de mineração de dados poderem manipular dados numéricos, mas algumas não poderem manipular dados simbólicos, torna-se preciso aplicar algum método de transformação de valores simbólicos em uma representação numérica apropriada (PYLE, 1999).

Dentre estes métodos é possível aplicar o entendimento do analista de mineração de dados bem como o do especialista do domínio do problema para a determinação de uma boa representação, ou utilizar técnicas automatizadas que podem ser utilizadas para associar uma representação numérica a valores simbólicos, embora não exista uma garantia de que estas encontrem uma boa representação, pois a obtida depende da forma em que os valores simbólicos são submetidos a preparação e da informação contida no conjunto de dados.

Considerando-se os métodos utilizados a partir do entendimento do domínio do problema tem-se:

1) Remapeamento 1 de n

Neste uma representação binária é atribuída a cada valor simbólico, sendo definido como "1" quando na presença de um determinado valor simbólico relevante e "0" caso contrário. Considerando-se "n" valores simbólicos distintos de um atributo, somente um valor é definido como "1" dos n possíveis originando assim a denominação 1 de n.

2) Remapeamento m de n

Neste também se utiliza uma representação binária, sendo que considerando-se "n" valores simbólicos distintos de um atributo, m (mais de 1) destes são definidos como "1".

Assim por exemplo, tendo-se uma lista de muitos valores simbólicos para um atributo, ao invés de se utilizar o remapeamento 1 de n, pode ser mais apropriado formar grupos destes valores considerando a similaridade de características de acordo com a situação, e aplicar o remapeamento m de n aos nomes dos grupos.

Segundo Pyle (1999), o remapeamento pode ser útil nas seguintes situações:

a) A densidade da informação a ser remapeada é alta;

b) A dimensionalidade da representação do conhecimento aprendido é somente modestamente expandida pelo remapeamento;

c) Um raciocínio lógico pode ser empregado para o remapeamento e

d) Quando é importante para a representação do conhecimento aprendido que não exista ordenação entre os valores.

Já as técnicas automatizadas tentam encontrar uma ordenação apropriada para valores simbólicos, considerando-se ainda que, o inter-relacionamento entre os valores simbólicos e o conjunto de dados como um todo é que permite uma numeração adequada.

Conforme Pyle (1999), entre quaisquer dois atributos numéricos ou simbólicos existe algum grau de relacionamento, tanto que no contexto estatístico os atributos podem ser mais ou menos independentes de cada um, caso sejam totalmente independentes não existe relacionamento entre eles.

No caso de existir pelo menos um atributo numérico no conjunto de dados, este é utilizado para determinar uma ordem e distância para os valores de atributos simbólicos. Assim, um espaço de estado – um gráfico em que cada atributo tratado corresponde a um eixo – pode ser utilizado para "plotar" todos os valores de dois atributos numéricos normalizados e verificar a disposição dos valores simbólicos de um terceiro atributo relacionado.

Em seguida, deve ser encontrada a posição central para os valores simbólicos que pode ser obtida através da média calculada dos valores simbólicos para cada atributo numérico. As posições centrais para cada valor do atributo simbólico são então localizadas no espaço de estado e os pontos ligados formando por exemplo, algo aproximadamente parecido a uma linha reta onde serão definidos os extremos como 0 e 1 associados aos valores simbólicos correspondentes, assim como também os valores entre estes extremos são proporcionalmente associados.

Já na ausência de atributos numéricos é preciso encontrar alguma ordenação lógica presente no relacionamento entre os valores simbólicos. Sendo para isto necessário identificar como os valores simbólicos de atributos diferentes se relacionam, o que pode ser conseguido através de uma tabela de freqüência de combinação ou de distribuição de combinação. Assim, sabendo-se as várias freqüências para cada

combinação de valores simbólicos de atributos distintos pode-se fazer uma associação dos mesmos a valores normalizados dentro de uma faixa, tal como de 0 a 1.

3.4.3 Discretização de Atributos

A discretização consiste em transformar valores contínuos de atributos em valores simbólicos (discretos). Assim como também equivale a formação de categorias de valores que já sejam discretos mas que se apresentam em grande número.

Dentre os principais motivos para discretizar atributos considera-se os seguintes:

a) Atributos discretos são mais fáceis para entender, utilizar e explicar;

b) Algumas técnicas de mineração de dados tais como árvores e regras de decisão quando utilizando atributos discretos são geralmente mais concisas, permitindo serem mais minuciosamente examinadas, comparadas, usadas e reusadas;

c) A discretização é útil para transformar o dado de entrada em um formato requerido (RIAÑO, 1997);

d) A discretização também é útil para realizar uma aprendizagem mais rápida (RIAÑO, 1997).

Tratando-se de atributos contínuos a discretização é geralmente realizada antes da aprendizagem (RIAÑO, 1997).

Segundo Hussain et al. (1999), um processo típico de discretização de atributos contínuos envolve quatro etapas, sejam elas:

a) Classificação do Atributo

Os valores contínuos do atributo a ser discretizado são classificados em ordem decrescente ou crescente. Para tornar mais rápido o processo de discretização é importante selecionar um algoritmo de classificação eficiente.

b) Seleção de *cut-points*/intervalos adjacentes

Entende-se por *cut-point* um valor real dentro da faixa de valores contínuos que divide a faixa dentro de dois intervalos, um intervalo é menor que ou igual ao *cut-point* e o outro intervalo é maior que o *cut-point*.

Após a classificação, é preciso encontrar o melhor *cut-point* para dividir a faixa de valores contínuos ou o melhor par de intervalos adjacentes para combinar. Para esta escolha, uma função de avaliação comum é determinar a correlação de uma divisão ou uma combinação com o atributo de classe, tais como medidas de entropia e estatísticas.

c) Divisão/ Combinação

Para a divisão é preciso avaliar *cut-points* e escolher o melhor deles para dividir a faixa de valores contínuos dentro de duas partições. A discretização continua em cada parte (incrementado por 1) até o critério de parada ser satisfeito. Similarmente, para combinação, intervalos adjacentes são avaliados para encontrar o melhor par de

intervalos para combinar em cada iteração. A discretização continua com o número reduzido (decrementado de 1) de intervalos até o critério de parada ser satisfeito.

d) Parada do processo

Um critério de parada especifica quando o processo de discretização deve parar. Para isso pode-se utilizar um critério de parada muito simples como fixar o número de intervalos no início, ou um mais complexo como uma função de avaliação.

De um modo geral, os métodos de discretização existentes podem ser classificados sob diferentes abordagens descritas a seguir por Hussain et al. (1999):

a) Supervisionado ou Não-Supervisionado

Os métodos de discretização supervisionados consideram a informação do atributo de classe ao contrário dos não-supervisionados.

Os métodos de discretização não-supervisionados consideram as faixas contínuas serem divididas dentro de subfaixas pela largura (faixa de valores) ou pela freqüência (número de instâncias em cada intervalo) especificada pelo usuário.

Na literatura encontra-se mais métodos de discretização supervisionados que não-supervisionados talvez em função da discretização ser geralmente associada com a tarefa de classificação (HUSSAIN et al., 1999).

b) Dinâmico ou Estático

Um método dinâmico discretiza valores contínuos quando um classificador está sendo construído, tal como no C4.5. Enquanto na abordagem estática a discretização é feita antes da tarefa de classificação.

c) Local ou Global

Um método local discretiza em um subconjunto de instâncias, enquanto um método de discretização global usa o conjunto inteiro de dados para discretizar. Assim, um método local é freqüentemente associado com um método de discretização dinâmico em que somente uma parte do conjunto de dados é usada para discretização.

d) *Top-Down* ou *Bottom-up*

Os métodos *top-down* iniciam com uma lista vazia de *cut-points* e adicionam novos a lista pelos intervalos divididos como o progresso da discretização. Enquanto os métodos *bottom-up* iniciam com a lista completa de todos os valores contínuos do atributo como *cut-points* e removem alguns deles pela combinação de intervalos combinados como o progresso da discretização.

e) Direto ou Incremental

Os métodos diretos dividem a faixa em vários intervalos simultaneamente (usando *equal-width* e *equal-frequency*) requerendo uma entrada adicional do especialista do domínio para determinar o número de intervalos. Enquanto os métodos incrementais iniciam com uma simples discretização e passam por um processo de melhoramento, requerendo um critério adicional para saber quando a discretização deve terminar.

No Anexo 3 são descritos alguns métodos de discretização de atributos contínuos e na Tabela 3.3, são apresentadas as principais características de tais métodos, conforme Hussain et al. (1999).

TABELA 3.3 – Métodos de Discretização Representativa em Múltiplas Dimensões

Métodos	**Global / Local**	**Supervisionado/ Não-Supervisionado**	**Direto/ Incremental**	**Divisão/ Combinação**	**Estático/ Dinâmico**
Equal-width	Global	Não-Supervisionado	Direto	Divisão	Estático
Equal-frequency	Global	Não-Supervisionado	Direto	Divisão	Estático
1R	Global	Supervisionado	Direto	Divisão	Estático
D2	Local	Supervisionado	Incremental	Divisão	Estático
MDLP	Local	Supervisionado	Incremental	Divisão	Estático
Mantaras	Local	Supervisionado	Incremental	Divisão	Estático
ID3	Local	Supervisionado	Incremental	Divisão	Dinâmico
Zeta	Global	Supervisionado	Direto	Divisão	Estático
Adaptive Quantizer	Global	Supervisionado	Direto	Divisão	Estático
ChiMerge	Global	Supervisionado	Incremental	Combinação	Estático
Chi2	Global	Supervisionado	Incremental	Combinação	Estático
ConMerge	Global	Supervisionado	Incremental	Combinação	Estático

Fonte: HUSSAIN et al., 1999. p. 17

3.4.4 Composição de Atributos

Conforme Zheng (1996), o processo de composição ou combinação de atributos é conhecido por Indução Construtiva, cujo objetivo consiste em construir um pequeno conjunto de novos atributos a partir de atributos originais tal que os resultados construídos usando os novos atributos apresentem maior acurácia e concisão que aqueles criados diretamente usando os atributos originais.

Para realizar a indução construtiva é preciso decidir quais atributos originais serão combinados assim como também quais operadores construtivos serão utilizados, tais como operadores de conjunção, disjunção e/ ou negação.

Segundo Baranauskas (2001), existe um exemplo clássico de indução construtiva que é o dos robôs amigos e inimigos. Na Tabela 3.4 é apresentado o conjunto de instâncias sobre robôs com os atributos originais, sendo na Figura 3.5 exibida a árvore de decisão resultante por aplicação do C4.5 para esse conjunto de instâncias.

TABELA 3.4 – Exemplos de robôs amigos e inimigos

Exemplos	Cabeça	Corpo	Sorri	Segura	Classe
T1	Triangular	Triangular	Sim	Balão	Amigo
T2	Quadrada	Quadrado	Sim	Balão	Amigo
T3	Redonda	Redondo	Sim	Bandeira	Amigo
T4	Quadrada	Triangular	Não	Espada	Inimigo
T5	Triangular	Redondo	Sim	Espada	Inimigo
T6	Redonda	Quadrado	Não	Bandeira	Inimigo

Fonte: BARANAUSKAS, 2001. p. 99

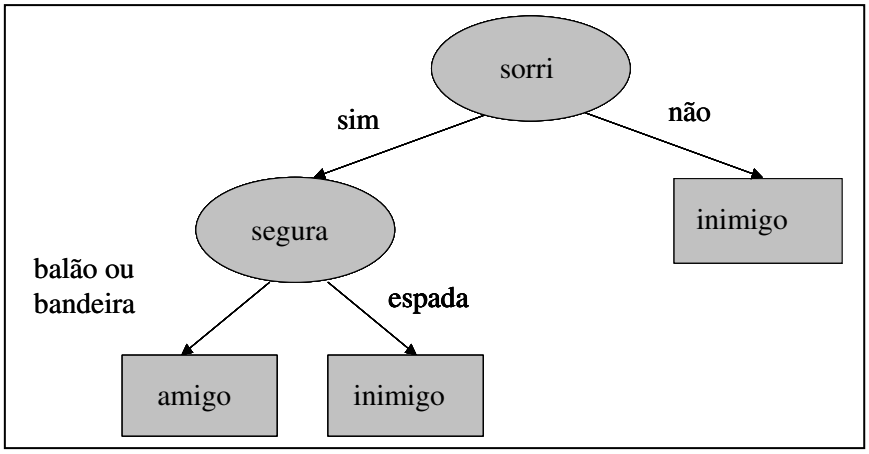

FIGURA 3.5 – Árvore de decisão para o exemplo de robôs amigos e inimigos
Fonte: BARANAUSKAS, 2001. p. 99 com adaptações

No exemplo dos robôs amigos e inimigos, aplicando a indução construtiva pode-se construir um novo atributo chamado mesma-forma a partir da verificação dos valores dos atributos originais cabeça e corpo. Caso o robô apresente cabeça e corpo com a mesma forma, o novo atributo mesma - forma assume o valor "sim", senão assume o valor "não" como exibido na Tabela 3.5.

TABELA 3.5 – Exemplos de robôs amigos e inimigos depois da construção do atributo mesma-forma

Exemplos	Cabeça	Corpo	Sorri	Segura	Mesma-forma	Classe
T1	Triangular	Triangular	Sim	Balão	Sim	Amigo
T2	Quadrada	Quadrado	Sim	Balão	Sim	Amigo
T3	Redonda	Redondo	Sim	Bandeira	Sim	Amigo
T4	Quadrada	Triangular	Não	Espada	Não	Inimigo
T5	Triangular	Redondo	Sim	Espada	Não	Inimigo
T6	Redonda	Quadrado	Não	Bandeira	Não	Inimigo

Fonte: BARANAUSKAS, 2001. p. 99

Assim, a nova árvore de decisão gerada a partir do indutor C4.5 sobre o conjunto de instâncias com o novo atributo mostra um resultado mais simples como visualizado na Figura 3.6.

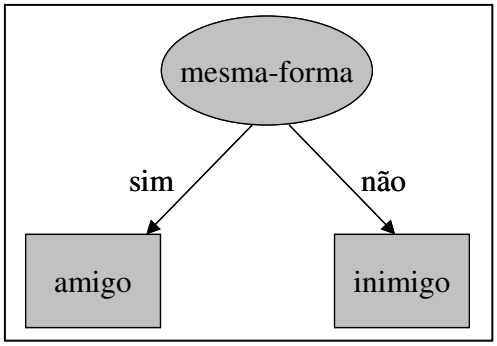

FIGURA 3.6 – Árvore de decisão para o exemplo de robôs amigos e inimigos após a construção do atributo mesma-forma
Fonte: BARANAUSKAS, 2001. p. 99

Segundo Baranauskas (2001), a composição de atributos pode ser guiada pelo conhecimento do especialista do domínio ou pelo próprio algoritmo de aprendizagem. Neste último caso tem-se as seguintes abordagens: indução construtiva guiada pelos dados, indução construtiva guiada por hipóteses, além da indução construtiva multi-estratégia.

A indução construtiva guiada pelos dados consiste na construção de novos atributos através da aplicação de vários operadores matemáticos e lógicos aos atributos originais, como exemplo tem-se o algoritmo AQ17-PRE de Bloedorn; Michalski (1991 apud RIAÑO, 1997, p. 89).

Já a indução construtiva guiada por hipóteses refere-se a construção de novos atributos em iterações onde cada iteração envolve uma etapa de aprendizagem e outra etapa que constrói novos atributos baseado nas hipóteses da etapa de aprendizagem. Como exemplos podem ser citados os algoritmos DUCE de Muggleton (1987 apud RIAÑO, 1997, p. 90), CIGOL de Muggleton; Buntine (1988 apud RIAÑO, 1997), FRINGE de Pagallo (1989 apud RIAÑO, 1997) e RINCON de Wogulis; Langley (1989 apud RIAÑO, 1997).

Finalmente, a indução construtiva multi-estratégia pode combinar quaisquer das abordagens anteriormente descritas para resolver o problema de indução construtiva. São considerados exemplos desta abordagem os seguintes algoritmos: AQ17-MCI de Bloedorn; Wnek; Michalski (1993 apud RIAÑO, 1997), CN2-MCI de Kramer (1994 apud RIAÑO, 1997) e CIPF de Pfahringer (1994 apud RIAÑO, 1997).

Enfatizando-se a indução construtiva guiada pelo conhecimento (*knowledge-driven construtive induction*) temos que a mesma baseia-se no conhecimento do domínio fornecido pelo especialista deste, sendo utilizado para construir um novo conjunto de atributos para o conjunto de instâncias.

Uma metodologia geral para aplicação da indução construtiva guiada pelo conhecimento é proposta por Baranauskas (2001) e consiste em 3 (três) etapas:

a) A partir da sugestão pelo especialista dos L atributos construídos por composição, estes são incluídos no conjunto de dados original T, sendo que é feita uma inclusão por vez. Assim, são formados L conjuntos de instâncias com os novos atributos, referenciados por {T1', T2', ..., TL'}, onde cada um contém m (número de atributos originais) + 1 atributos onde Ti' {X1, X2, ..., Xm, Xi'} e $1 \leq i \leq L$.

b) Cada conjunto de instâncias com o novo atributo {T1', T2', ..., TL'} é submetido a algum indutor. Caso o novo atributo Xi' não apareça no resultado de indução gerado, significa que este atributo não é relevante, podendo ser desconsiderado na próxima etapa. Deste modo, somente o subconjunto dos conjuntos de instâncias com novos atributos formados na primeira etapa e que satisfazem a condição da segunda etapa serão tratados na terceira etapa.

c) Finalmente, é realizada a estimativa de erro aplicando-se uma validação cruzada *stratified* de r-partições tanto no conjunto de dados original T quanto em cada um dos conjuntos de instâncias com novos atributos. Este tipo de validação cruzada foi adotado para que a proporção de instâncias de cada classe em cada partição fosse a mesma que o conjunto de dados original. Assim, somente aqueles conjuntos que apresentam uma diferença absoluta em desvios padrões são selecionados para investigação futura, já que o novo atributo além de aparecer no resultado induzido também melhora sua precisão.

4 Pré-Processamento em uma Base de Dados Real: base de dados de AIHs da SES do RS

Neste capítulo são apresentados alguns experimentos realizados sobre uma base de dados real, a base de dados de Autorização de Internações Hospitalares (AIHs) fornecidas pela Secretaria Estadual de Saúde (SES) do Rio Grande do Sul (RS).

Ressalta-se que é dada ênfase à fase de pré-processamento e que esta foi realizada utilizando-se os recursos disponíveis no ambiente de pesquisa. A seguir são apresentados os experimentos.

4.1 Experimento 1

Neste experimento tem-se por objetivo identificar as características comuns das AIHs dos três procedimentos de maior freqüência no RS em relação ao custo das mesmas. Para tanto o problema de mineração de dados será considerado como de classificação.

4.1.1 Entendimento do Domínio do Problema

A Secretaria Estadual de Saúde do Rio Grande do Sul (SES/RS) é responsável pela administração das informações do Sistema Único de Saúde (SUS), controlando várias bases de dados de saúde, como as de internações hospitalares, mortalidade, vacinação, laboratoriais, procedimentos de alta complexidade, etc.

A partir de várias entrevistas realizadas com os especialistas do domínio sendo considerados como tais auditores, analista de sistema, programadores e funcionários do setor administrativo, percebeu-se que existe um interesse muito grande da SES em tentar detectar impropriedades nas cobranças de atendimentos ambulatoriais e hospitalares assim como também, obter informações para tomada de decisões gerenciais.

Apesar de já existirem algumas regras de bloqueio de AIHs que apresentem alguma impropriedade tais como: septicemia (infecção generalizada), cuidados prolongados, politraumatizados, cirurgias múltiplas, transplante, AVC agudo e homônimos/ duplicidade adotadas pelos auditores sabe-se que estas ainda não são suficientes para bloquear todas as AIHs com impropriedades, pois a quantidade de dados é muito grande e novas regras ou acréscimos de considerações nas já existentes são desejadas, o que torna necessária a utilização de novas técnicas como por exemplo, a mineração de dados visando atingir estes objetivos.

A partir do projeto "Desenvolvimento de Metodologia para Extração de Conhecimento de Bases de Dados de Saúde do Estado para Avaliação e Planejamento" desenvolvido por pesquisadores da UFRGS, UCS e a SES, foi possível ter acesso à base de dados de AIHs (Autorização de Internação Hospitalar) do ano de 2000 que registram

as internações, procedimentos e diagnósticos realizados em hospitais dos municípios do Rio Grande do Sul através do Sistema Único de Saúde naquele ano.

Mensalmente as AIHs são apresentadas à SES para que os hospitais, profissionais e serviços auxiliares de diagnose e terapia - SADT recebam seus honorários pela prestação de serviços ao SUS. A SES aplica alguns critérios de bloqueio sobre estas AIHs e depois envia ao Ministério da Saúde as bloqueadas e as demais apresentadas. O Ministério da Saúde realiza uma nova filtragem sobre estas AIHs para verificar quais AIHs deverão ser pagas. Caso uma AIH não seja paga no mês de apresentação, o hospital tem o direito de reapresentá-la posteriormente.

Vários hospitais contratam serviços terceirizados de empresas chamadas *bureaux* que convertem os dados de AIHs em um sistema e o entregam na SES. Estas empresas recebem um percentual da fatura do hospital. Os hospitais que não estão associados a *bureaux* utilizam um programa disponibilizado pelo SUS para gerar os disquetes com as AIHs. E a SES a partir de um sistema chamado SGAIH em Clipper gerencia e integra estes movimentos de diferentes municípios gerando a base mensal do Estado. Estas AIHs apresentadas à SES correspondem a metade do movimento do Estado, pois existem 11 municípios que enviam seus movimentos diretamente para o Ministério da Saúde, por exemplo Porto Alegre.

Assim, considerando-se a explicitação do contexto do problema bem como das necessidades dos especialistas do domínio e a ciência de alguns relatórios do ano de 2000 da SES, tal como o dos 100 procedimentos de maior freqüência no RS, definiu-se juntamente com os especialistas o objetivo que encaminharia o processo de descoberta de conhecimento na base de dados de AIHs, sendo este apresentado a seguir.

4.1.1.1 Objetivo

Identificar as características comuns das AIHs dos três procedimentos de maior freqüência no RS em relação ao custo (baixo, médio, alto).

4.1.1.2 Tipo do Problema/ Técnica/ Ferramenta de Mineração de Dados (MD)

Diante do objetivo formado decidiu-se considerar como tipo de problema de mineração de dados a classificação, em que o atributo de classe é o custo. Quanto a técnica de MD a ser adotada optou-se por utilizar as regras de classificação geradas pela ferramenta See5. Estas escolhas foram feitas devido a disponibilidade e acessibilidade da ferramenta no ambiente de pesquisa.

Na próxima seção é descrito o pré-processamento realizado sobre as AIHs apresentadas à SES no ano de 2000.

4.1.2 Pré - Processamento

Ressalta-se que, as operações realizadas nesta fase são descritas nesta seção na ordem cronológica de realização. Isto deve-se ao caráter iterativo do processo de descoberta de conhecimento em banco de dados (DCBD) e principalmente da fase de pré-processamento, em que muitas das vezes é preciso avançar ou retornar entre fases ou subfases do processo, fato que é registrado em todas as bibliografias de metodologias de DCBD. A seguir tem-se o relato da fase em questão.

Tratando-se do entendimento dos dados foram verificadas as principais informações referentes às tabelas de AIHs do ano de 2000, tais como: nome da tabela, descrição, número de instâncias e número de atributos, conforme visualizado nas Tabelas 4.1 e 4.2.

Especificamente tratando-se de atributos foram identificados quantos são relevantes, irrelevantes, com valores ausentes, o tipo de dado, tamanho, formato e valores assumidos.

TABELA 4.1 – Descrição de Tabelas fornecidas pela SES/RS

Nome da Tabela	**Descrição**	**Número de Instâncias**	**Número de Atributos**
DSMS010.dbf	Movimento de AIH	(*)	75
DSMS160.dbf	Valores da AIH	(*)	24
DSMS020.dbf	Procedimentos especiais	(*)	9
DSMS040.dbf	Movimento dos hospitais	392	25
DAIH050.dbf	Tabela de procedimentos	4.899	15
DAIH150.dbf	Tabela CID (Classificação Internacional de Doenças)	14.196	6
DAIH210.dbf	Tabela de municípios	5.508	3
Controle.dbf	Tabela de AIHs bloqueadas	41.483	18
Leitos.xls	Tabela de leitos de hospitais	379	5
Biros.xls	Tabela de *bureaux*	256	2

(*) número de instâncias mensais exibidos na Tabela 4.2.

TABELA 4.2 – Número Mensal e Total de Instâncias

Mês	DSMS010.dbf	DSMS160.dbf	DSMS020.dbf
Janeiro	48.749	48.749	8.447
Fevereiro	47.545	47.545	8.914
Março	49.855	49.855	9.601
Abril	52.794	52.794	10.509
Maio	49.858	49.858	10.224
Junho	48.042	48.042	9.913
Julho	48.813	48.813	10.675
Agosto	44.747	44.747	8.874
Setembro	45.889	45.889	9.507
Outubro	46.428	46.428	19.164
Novembro	46.855	46.855	9.664
Dezembro	44.776	44.776	8.884
Total	574.351	574.351	124.376

Como as principais tabelas que contém as informações gerais do paciente, do hospital, procedimento realizado, diagnóstico e valores da AIH são a DSMS010, DSMS160, DSMS040, DAIH150, Biros e Leitos então serão apresentados os seus respectivos atributos com algumas observações realizadas em uma análise inicial dos dados.

a) Quanto a tabela de Movimento de AIHs (DSMS010)

- Apresenta 3 atributos com valores ausentes e que podem ser considerados irrelevantes.
- Apresenta 32 atributos com valores preenchidos com zero, sendo considerados irrelevantes.
- Apresenta 22 atributos com valores considerados irrelevantes.
- Apresenta 18 atributos que podem ser considerados relevantes, cuja descrição dos mesmos é verificada na Tabela 4.3.

TABELA 4.3 – Descrição de Atributos da Tabela de Movimento de AIHs (continua)

Atributo	Descrição	Tipo de Dado	Tamanho	Formato
DCIH	Documento de Cobrança de Internação Hospitalar	Caracter	8	Todos os caracteres são numéricos
APRES	Data de apresentação da AIH	Caracter	6	Dois caracteres para mês e quatro para o ano
ESPEC	Especialidade da AIH	Caracter	2	Todos os caracteres são numéricos
CGC	CGC do hospital	Caracter	14	Todos os caracteres são numéricos
IDENT	Identificação da AIH	Caracter	1	Um caracter numérico
DT_NASC	Data de nascimento	Data		dd/mm/aaaa

TABELA 4.3 – Descrição de Atributos da Tabela de Movimento de AIHs (continuação)

Atributo	Descrição	Tipo de Dado	Tamanho	Formato
IDADE	Idade do paciente	Caracter	4	Todos os caracteres são numéricos
SEXO	Sexo do paciente	Caracter	1	Um caracter numérico
N_AIH	Número da AIH	Caracter	10	Todos os caracteres são numéricos
PROC_SOL	Código do procedimento solicitado	Caracter	8	Todos os caracteres são numéricos
CAR_INT	Caráter da internação	Caracter	2	Todos os caracteres são numéricos
PROC_REA	Código do procedimento realizado	Caracter	8	Todos os caracteres são numéricos
DT_INT	Data da internação do paciente	Data		dd/mm/aaaa
DT_SAI	Data de alta do paciente	Data		dd/mm/aaaa
DIAG_PRI	Código do diagnóstico principal segundo a CID	Caracter	4	Primeiro caracter é letra e os demais são numéricos
DIAG_SEC	Código do diagnóstico secundário segundo a CID	Caracter	4	Primeiro caracter é letra e os demais são numéricos
MOT_COB	Motivo da cobrança	Caracter	2	Todos os caracteres são numéricos
NAT	Natureza da relação do hospital com o SUS	Caracter	2	Todos os caracteres são numéricos

A seguir nas Tabelas 4.4, 4.5, 4.6, 4.7 e 4.8 tem-se a descrição dos valores assumidos por alguns dos atributos listados acima.

TABELA 4.4 – Descrição dos Valores de Caráter de Internação

CAR_INT	DESCRIÇÃO
01	Eletiva
03	Urgência/ Emergência, com AIH emitida antes da internação
04	Internação em AIH de alta complexidade fora do Estado
05	Urgência/ Emergência, com AIH emitida após a internação
06	Acidente no local de trabalho ou a serviço da empresa
07	Acidente no trajeto entre a residência e o trabalho
08	Outros tipos de acidentes de trânsito
09	Outros tipos de lesões e envenenamento
20	Urgência/ Emergência em unidade de referência
27	Urgência/ Emergência – Acidente trajeto residência trabalho
29	Urgência/ Emergência lesão/ envenenamento por agente químico/ fisico

TABELA 4.5– Descrição dos Valores de Identificação da AIH

IDENT	DESCRIÇÃO
1	AIH normal
3	AIH de continuação
5	AIH de longa permanência e FTP (Fora de Possibilidade Terapêutica)

TABELA 4.6 – Descrição dos Valores de Especialidade da AIH

ESPEC	DESCRIÇÃO
01	Cirurgia Geral
02	Obstetrícia
03	Clínica Médica
04	Crônico e FPT (Fora de Possibilidade Terapêutica)
05	Psiquiatria
06	Tisiologia (Tuberculose)
07	Pediatria
08	Reabilitação
09	Psiquiatria – hospital/dia

TABELA 4.7 – Descrição dos Valores de Natureza do Hospital

NAT	DESCRIÇÃO
10	Hospital próprio
20	Hospital contratado
30	Hospital federal
31	Hospital federal com verba própria
40	Hospital estadual
50	Hospital municipal
60	Hospital filantrópico
61	Hospital filantrópico isento pela IN 01/97 SRF
63	Hospital filantrópico parcialmente isento de imposto de renda
70	Hospital universitário de ensino
90	Hospital universitário com FIDEPS[8]
91	Hospital universitário isento de FIDEPS
92	Hospital universitário parcialmente isento de FIDEPS

TABELA 4.8 – Descrição dos Valores de Sexo

SEXO	DESCRIÇÃO
0	Ignorado
1	Masculino
3	Feminino

[8] FIDEPS significa Fundo de Incentivo ao Ensino e à Pesquisa

b) Quanto a tabela de Valores de AIHs (DSMS160)

- Apresenta 8 atributos já presentes na DSMS010, são eles: APRES, CGC, DCIH, N_AIH, IDENT, DT_SAI, PROC_REA e PRONT (número do prontuário, tipo de dado: caracter, tamanho: 7, formato: todos os caracteres numéricos).
- Apresenta 2 atributos com todos os valores ausentes, sendo considerados irrelevantes.
- Apresenta 1 atributo com todos os valores iguais a zero, sendo considerado irrelevante.
- Apresenta 2 atributos considerados irrelevantes.
- Apresenta 11 atributos considerados relevantes, cuja descrição dos mesmos é verificada na Tabela 4.9.

TABELA 4.9 – Descrição dos Atributos da Tabela de Valores de AIHs

Atributo	Descrição	Tipo de Dado	Tamanho	Formato
VALSH	Valor de serviços hospitalares	Número	12	Com duas casas decimais
VALSP	Valor de serviços profissionais	Número	12	Com duas casas decimais
VALSADT	Valor de Serviços Auxiliares de Diagnose/ Terapia	Número	12	Com duas casas decimais
VALOPM	Valor pago por permanência a maior	Número	12	Com duas casas decimais
VALSANG	Valor de sangue	Número	12	Com duas casas decimais
VALRN	Valor de recém-nato	Número	12	Com duas casas decimais
VALUTI	Valor de UTI	Número	12	Com duas casas decimais
VALACO	Valor de diárias de acompanhante	Número	12	Com duas casas decimais
VALUTINN	Valor de UTI neo-natal	Número	12	Com duas casas decimais
VALTRANSP	Valor de transplante	Número	12	Com duas casas decimais
VALNEURO	Valor de neurologia	Número	12	Com duas casas decimais

c) Quanto a tabela de Movimento dos Hospitais (DSMS040)

- Apresenta 15 atributos com todos os valores ausentes e que são considerados irrelevantes.
- Apresenta 4 atributos com alguns valores ausentes e que são considerados irrelevantes.
- Apresenta 2 atributos considerados irrelevantes.
- Apresenta 4 atributos considerados relevantes, sendo que destes 3 já existem na tabela DSMS010 porém com nomes diferentes, sejam eles: CGC_HOSP,

NAT_HOSP, MUN_HOSP. E o outro é NOM_HOSP (Nome do hospital, tipo de dado: caracter, tamanho: 60, formato: letras).

d) Quanto a tabela de Classificação Internacional de Doenças (DAIH150)

- Apresenta 3 atributos considerados irrelevantes.
- Apresenta 3 atributos considerados relevantes. A seguir é feita a descrição destes na Tabela 4.10.

TABELA 4.10 – Descrição dos Atributos da Tabela de Classificação Internacional de Doenças

Atributo	Descrição	Tipo de Dado	Tamanho	Formato
CID10	Código da Classificação Internacional de Doenças	Caracter	4	Primeiro caracter é letra e os demais caracteres são numéricos
DESCR	Descrição da doença	Caracter	50	Letras
RESTRSEXO	Código do sexo para o qual este procedimento é de aplicação exclusiva (1: restrito a homens, 3: restrito a mulheres, 5: não restrito)	Caracter	1	Um caracter numérico

e) Quanto a tabela de *Bureaux* (Biros)

Apresenta os seguintes atributos visualizados na Tabela 4.11.

TABELA 4.11 – Descrição dos Atributos da Tabela de *Bureaux*

Atributo	Descrição	Tipo de Dado	Tamanho	Formato
CGC	CGC do hospital relacionado com o *bureau*	Texto	14	Todos os caracteres são numéricos
BIRO	Nome do *bureau*	Texto	50	Letras

f) Quanto a tabela de Leitos do Hospital (Leitos)

Apresenta os atributos descritos na Tabela 4.12.

TABELA 4.12 – Descrição dos Atributos da Tabela de Leitos do Hospital

Atributo	Descrição	Tipo de Dado	Tamanho	Formato
CRS	Coordenadoria Regional de Saúde	Número	2	Sem casas decimais
MUNICIPIO	Município do hospital	Texto	30	Letras
RAZAO	Nome do hospital	Texto	60	Letras
CGC_HOSP	CGC do hospital	Texto	14	Todos os caracteres são numéricos
QTE_LEITOS	Número de leitos	Número	4	Sem casas decimais

Dentre as tabelas selecionadas citam-se as referentes ao movimento de AIHs (DSMS010) juntamente com as suas respectivas tabelas de valores de AIHs (DSMS160) de cada mês, tendo estas sido importadas para um banco de dados criado no Access 97 nomeado bdJuntas. Além destas, foram também selecionadas as tabelas de movimento dos hospitais (DSMS040), a de *bureaux* (Biros) e a de leitos de hospitais (Leitos), sendo estas importadas para um outro banco de dados criado no Access 97 nomeado bdHospitais.

Antes de realizar as integrações das tabelas foi preciso realizar algumas operações de limpeza sobre os dados. Assim, primeiramente investigou-se sobre os atributos comuns às duas tabelas e a padronização do formato dos valores dos mesmos, já que seriam possíveis candidatos à chave primária das mesmas.

Verificou-se que, somente no mês de janeiro tanto na tabela de movimento de AIHs quanto na de valores da AIH, o atributo APRES (data de apresentação da AIH) apresentava erros no valor da data do tipo “020000”, enquanto a maioria das instâncias apresentava o valor “022000”. Como posteriormente teriam que ser investigadas duplicatas e nos demais meses este atributo não seria relevante para realizar as integrações das tabelas, já que teriam que ser mês-a-mês sendo o valor do atributo APRES o mesmo para todas as instâncias do seu respectivo mês, optou-se por tratar por último as AIHs do mês de janeiro.

Segue-se assim com a investigação da presença de duplicatas nas instâncias das tabelas de movimento de AIHs e de valores de AIHs, mês-a-mês, através do recurso assistente de consulta/ localizar duplicatas do Access 97. Para tanto, já que as tabelas não apresentam chaves definidas, foi preciso escolher atributos que pudessem funcionar como chaves, definindo-se os seguintes: CGC (CGC do hospital), N_AIH (número da AIH), DCIH (Documento para Cobrança de Internação Hospitalar), IDENT (identificação da AIH) e DT_SAI (data de alta do paciente).

Para a determinação destes atributos como chaves foram realizadas várias consultas de localizar duplicatas com todos os atributos comuns nas tabelas em questão de modo a encontrar a combinação necessária para realizar as integrações mês-a-mês sem dúvidas, conseqüentemente esta tarefa consumiu bastante tempo, pois inicialmente tentou-se utilizar um único atributo, o N_AIH, que aparentemente seria o mais indicado. Entretanto, o atributo N_AIH não é suficiente para a identificação única de uma AIH,

já que foram observadas várias instâncias com o mesmo N_AIH, porém de hospitais diferentes.

Tendo sido isto questionado a um dos especialistas obteve-se a confirmação de que o N_AIH deve ser combinado com o CGC do hospital para identificar uma AIH já que é possível ocorrerem erros por exemplo, de digitação no momento de registro do N_AIH. Entretanto, mesmo com estes dois atributos combinados ainda eram encontradas AIHs duplicadas.

Como os especialistas não conseguiram esclarecer este problema, foram combinados os dois atributos N_AIH e CGC com outros atributos também comuns às duas tabelas, sempre acrescentando-se um atributo por vez à combinação anterior de atributos.

Então, verificou-se que o DCIH, cujo valor indica o conjunto de AIHs por especialidade era importante para a integração das tabelas, apesar dos especialistas o considerarem um atributo irrelevante. E, os atributos IDENT e DT_SAI estão extremamente relacionados, já que uma AIH pode ser de IDENT = 1 (normal) ou de IDENT = 5 (longa permanência, geralmente casos de psiquiatria ou FPT – Fora de Possibilidade Terapêutica). E um paciente de AIH de IDENT = 5 pode ser internado mais de uma vez mantendo o número de IDENT, porém sendo percebidos as diferentes datas de alta (DT_SAI).

Desta forma, comprovou-se que utilizando a combinação dos atributos CGC, N_AIH, DCIH, IDENT e DT_SAI não foi detectada nenhuma duplicata nos meses de fevereiro a dezembro.

Quanto às AIHs do mês de janeiro aplicou-se a combinação de atributos descrita no parágrafo anterior para tentar detectar duplicatas e de fato existiam AIHs duplicadas onde o atributo APRES (data de apresentação da AIH) apresentava erros no valor da data do tipo “020000”, enquanto a maioria das instâncias apresentava o valor “022000”.

Assim as AIHs das tabelas de movimento de AIHs (DSMS010) e de valores de AIHs (DSMS160) do mês de janeiro, em particular, foram integradas considerando o atributo APRES, além dos 5 atributos descritos anteriormente. Aplicou-se a seguinte consulta em SQL para a integração utilizando o recurso criar tabela do Access:

```
Select *
From dsms010mes1 INNER JOIN dsms160mes1
On dsms010mes1.CGC = dsms160mes1.CGC AND dsms010mes1.N_AIH =
dsms160mes1.N_AIH AND dsms010mes1.DCIH = dsms160mes1.DCIH AND
dsms010mes1.IDENT = dsms160mes1.IDENT AND dsms010mes1.DT_SAI =
dsms160mes1.DT_SAI AND dsms010mes1.APRES = dsms160mes1.APRES;
```

Já com todas as informações gerais das AIHs de janeiro foi possível mostrar ao especialista do domínio a presença destas duplicatas, que não eram do seu conhecimento mas que poderiam ser eliminadas.

Diante disto, foi aplicada uma padronização de dados através da correção de 4.795 instâncias do mês de janeiro cujas datas de apresentação continham valor “020000” passando a assumir o valor “022000”. A correção foi feita através da consulta em SQL

```
update dsms010mes1
set apres = '022000'
where apres = '020000';
```

A partir daí foram eliminadas 44 instâncias duplicadas encontradas no mês de janeiro. Esta eliminação foi realizada utilizando os recursos do Access de criação de uma nova tabela a partir da cópia da estrutura daquela que apresenta duplicatas, definindo-se uma chave primária que no caso é considerada composta pelos atributos: CGC + N_AIH + DCIH + IDENT + DT_SAI + APRES que apresentavam valores duplicados. E posterior acréscimo na nova tabela de somente instâncias exclusivas com todos os atributos da tabela original.

Dando continuidade a operação de integração das AIHs dos demais meses, estas foram ligadas mês-a-mês considerando a combinação dos atributos: CGC + N_AIH + DCIH + IDENT + DT_SAI como chave através da seguinte consulta SQL utilizando o recurso criar tabela do Access:

```
Select *
From dsms010 INNER JOIN dsms160
On dsms010.CGC = dsms160.CGC AND dsms010.N_AIH = dsms160.N_AIH AND
dsms010.DCIH = dsms160.DCIH AND dsms010.IDENT = dsms160.IDENT AND
dsms010.DT_SAI = dsms160.DT_SAI ;
```

Como a estrutura das tabelas de movimento de AIHs (DSMS010) e de valores de AIHs (DSMS160) já ligadas mês-a-mês era a mesma então era preciso juntá-las verticalmente para compor a base do ano todo, o que foi conseguido através da utilização do aplicativo APPENDA disponibilizado pelo SUS. Assim a nova tabela denominada JunAno.dbf passa a conter 574.307 instâncias correspondentes as AIHs de todo o ano de 2000.

Realizando uma análise dos valores distintos dos atributos de JunAno através de consultas realizadas no Access, verificou-se a presença de valores errôneos nos atributos IDADE (idade) e DT_NASC (data de nascimento), como visualizado na Tabela 4.13:

TABELA 4.13 – Valores Errôneos Encontrados

IDADE	DT_NASC
3104	26/07/0954
3105	10/01/0951
3108	06/12/0912
3321	29/06/1680
3353	23/03/1648
3602	07/08/1399

O atributo IDADE tem o valor preenchido automaticamente a partir de um programa criado pelo departamento de informática do SUS, que considera horas, dias, meses e ano. Assim pelos valores que temos acima as idades dos pacientes seriam 104, 105, 108, 321, 353, 602 anos, respectivamente. Então foi realizada uma padronização de dados corrigindo-se as três primeiras idades pela substituição do valor 0 (zero) dos anos "0954", "0951" e "0912" pelo valor 1, em seguida calculou-se as idades pela diferença entre o ano de 2000 e o ano corrigido na data de nascimento, sendo os valores do

atributo idade para as três primeiras idades atualizados através de consulta SQL, como por exemplo:

```
update junano
set idade = '3046'
where idade = '3104';
```

Já nas datas em DT_NASC de ano 1680, 1648 e 1399 percebe-se que foram cometidos erros de digitação, então como eram apenas 3 instâncias dentre 574.307 que apresentavam este problema, optou-se por eliminar as 3 (três) instâncias em questão por apresentarem dados errôneos. Assim, a tabela JunAno passa a conter 574.304 instâncias.

O atributo CPF_PAC (CPF do paciente) também apresenta valores não muito confiáveis, já que muitos destes são preenchidos com zero, um mesmo paciente apresenta dois CPFs por algum dígito trocado ou vários pacientes apresentam um mesmo CPF. Assim, diante da dificuldade de validação desses dados optou-se por eliminar este atributo.

Outro atributo que mereceu atenção especial em função dos seus valores distintos foi o CAR_INT (Caráter de internação) que na tabela JunAno apresenta os seguintes valores: 01, 02, 03, 04, 05, 06, 07, 08, 09, 20, 27, 29. Assim verificou-se junto ao especialista do domínio o significado do valor até então desconhecido o 02. Concluindo-se que este deveria ser corrigido para 20 devido a mudança de código confirmada pelo especialista, sendo atualizados então 127 instâncias através da consulta SQL:

```
update junano
set car_int = '20'
where car_int = '02';
```

Decidiu-se então criar e discretizar alguns atributos a partir dos já existentes. Assim tem-se como atributos discretizados os descritos na Tabela 4.14:

TABELA 4.14 – Atributos Discretizados

Atributo	**Descrição**	**Tipo de Dado**	**Tamanho**	**Formato**
CAT_MCOB	Categoria de Motivo de Cobrança	Texto	4	Dois primeiros caracteres são letras e os dois últimos são numéricos
CAT_CID	Categoria da classificação internacional de doenças	Texto	4	Dois primeiros caracteres são letras e os dois últimos são numéricos
CATEGORIA	Faixa etária do paciente	Texto	7	Todos os caracteres são letras
SEMANA_INT	Dia da semana em que o paciente foi internado	Texto	7	Todos os caracteres são letras
SEMANA_SAI	Dia da semana em que o paciente recebeu alta	Texto	7	Todos os caracteres são letras

- O atributo CAT_MCOB (categoria de motivo de cobrança), como exibido na Tabela 4.15, foi discretizado a partir dos valores do atributo original MOT_COB (motivo de cobrança) segundo critérios definidos pelo especialista do domínio.

TABELA 4.15 – Descrição do Atributo Categoria de Motivo de Cobrança

CAT_MCOB	DESCRIÇÃO	MOTIVOS
MC01	Alta	11-19
MC02	Permanência maior que 30 dias	21-25
MC03	Transferência	31-39
MC04	Óbito com autópsia	41-44
MC05	Óbito sem autópsia	51-54
MC06	Alta por reoperação	61-68
MC07	Alta da prematuridade e permanência do recém-nascido	71

- O atributo CAT_CID (categoria da classificação internacional de doenças), conforme o apresentado na Tabela 4.16, foi discretizado a partir dos valores do atributo original DIAG_PRI (diagnóstico principal) que apresenta o código de CID (Classificação Internacional de Doenças). Tendo sido consultada a tabela DAIH150.dbf (tabela CID) assim como também o especialista do domínio para determinar algumas combinações de categorias.

TABELA 4.16 – Descrição do Atributo Categoria de Classificação Internacional de Doenças

CAT_CID	DESCRIÇÃO	CAUSAS
GP01	Algumas doenças infecciosas e parasitárias	A00-B99
GP02	Neoplasias (Tumores)	C00-D489
GP03	Doenças do sangue e alguns transtornos imunitários	D50-D899
GP04	Doenças endócrinas nutricionais e metabólicas	E00-E90
GP05	Transtornos mentais e comportamentais	F00-F99
GP06	Doenças do sistema nervoso	G00-G998
GP07	Doenças do olho e anexos	H00-H599
GP08	Doenças do ouvido e da apófise mastóide	H60-H959
GP09	Doenças do aparelho circulatório	I00-I99
GP10	Doenças do aparelho respiratório	J00-J998
GP11	Doenças do aparelho digestivo	K00-K938
GP12	Doenças da pele e do tecido subcutâneo	L00-L998
GP13	Doenças do sist. osteomuscular e tecido conjuntivo	M00-M999
GP14	Doenças do aparelho geniturinário	N00-N999
GP15	Gravidez, parto e puerpério	O00-O998
GP16	Algumas afecções originadas no periodo pré-natal	P00-P969
GP17	Malformações congênitas, deformações e anomalias cromossômicas	Q00-Q999
GP18	Sintomas e achados anormais de exames clínicos e laboratoriais	R00-R99
GP19	Lesões, envenen. e algumas outras conseq. de causas externas	S00-T983
GP20	Causas externas de morbidade e mortalidade	V01-Y98
GP21	Fatores que influenciam o estado de saúde, contato c/ serv. de saúde	Z00-Z999

- O atributo CATEGORIA (faixa etária), como visualizado na Tabela 4.17 foi discretizado a partir dos valores disponíveis no atributo já existente IDADE, tendo sido os intervalos de valores definidos pelo especialista do domínio.

TABELA 4.17 – Descrição do Atributo Categoria

ATRIBUTO	VALORES	INTERVALOS	Significado
CATEGORIA	CRIANCA	2000 - 3012	Até 12 anos
	JOVEM	3013 - 3021	De 13 a 21 anos
	ADULTO	3022 - 3059	De 22 a 59 anos
	IDOSO	>= 3060	Maior que 59 anos

Ressalta-se que, para a discretização dos atributos CAT_MCOB, CAT_CID e CATEGORIA foi criado um programa em Delphi 5, pois diante da quantidade de instâncias e da variedade dos valores para estes atributos as consultas em SQL consumiam horas para serem realizadas.

- Os atributos SEMANA_INT (dia da semana em que o paciente foi internado) e SEMANA_SAI (dia da semana em que o paciente recebeu alta) foram discretizados a partir das datas contidas nos atributos originais DT_INT (data de internação do paciente) e DT_SAI (data de alta do paciente) respectivamente, tendo sido criado um programa em Delphi 5 para realizar esta discretização de modo a apresentar o dia da semana, de domingo a sábado, em que o paciente foi internado (SEMANA_INT) e que recebeu alta (SEMANA_SAI). Estes atributos são importantes para verificar a freqüência do dia da semana em que determinados hospitais realizam internações ou liberam os pacientes.

Para a composição ou construção de atributos foi preciso a orientação dos especialistas do domínio, sendo então adicionados os atributos apresentados na Tabela 4.18.

TABELA 4.18 – Atributos Compostos

Atributo	Descrição	Tipo de Dado	Tamanho	Formato
VAL_TOTAL	Valor total da AIH	Número	12	Com duas casas decimais
DIAS_INT	Dias de internação	Número	3	Sem casas decimais

- O atributo VAL_TOTAL (valor total da AIH) foi composto a partir da soma dos valores dos atributos originais da tabela de valores de AIH (DSMS160.dbf) que envolvem custo (VALSH, VALSP, VALSADT, VALOPM, VALSANG, VALRN, VALUTI, VALACO, VALUTINN, VALTRANSP e VALNEURO). Esta combinação foi realizada através da seguinte consulta SQL:

```
Update JunAno
Set VAL_TOTAL = VALSH + VALSP + VALSADT + VALOPM +
VALSANG+VALRN+VALUTI+VALACO+VALUTINN+VALTRANSP+VALNEURO;
```

- O atributo DIAS_INT (dias de internação do paciente) foi composto a partir da diferença entre as datas apresentadas nos atributos DT_SAI (data de alta do paciente) e DT_INT (data de internação do paciente). Para tanto foi realizada a seguinte consulta SQL:

```
Update JunAno
Set DIAS_INT= DT_SAI - DT_INT;
```

Tendo-se então as informações referentes às AIHs e aos valores das mesmas tornou-se necessário obter mais informações sobre os hospitais. Assim foram analisadas as tabelas de movimento dos hospitais (DSMS040.dbf), de *bureaux* (Biros.xls) e a de leitos de hospitais (Leitos.xls).

A tabela de *bureaux* continha 38 instâncias cujo valor no atributo CGC apresentava formato diferente, estes valores foram então corrigidos e além disso apresentava um CGC de hospital ("98110000000149") duplicado com nomes de *bureaux* diferentes. Para resolver este problema foi preciso consultar o especialista do domínio que recomendou manter o nome do *bureau* mais antigo. Assim foi eliminada uma instância da tabela de *bureaux*.

Após este tratamento na tabela de *bureaux*, esta foi combinada com a tabela de movimento dos hospitais (DSMS040.dbf) através da seguinte consulta SQL utilizando o recurso criar tabela do Access:

```
Select *
From DSMS040 LEFT JOIN Biros
On DSMS040.CGC_HOSP = Biros.CGC;
```

Tendo-se formado agora a tabela DSMS040Biros com 392 instâncias, as instâncias cujos hospitais não estão relacionados com nenhum *bureau* tiveram o atributo BIRO preenchido com 'NENHUM' para que não apresentasse valores ausentes. Esta atualização foi realizada através da aplicação da seguinte consulta SQL do Access:

```
Update DSMS040Biros
Set BIRO = 'NENHUM'
Where BIRO = NULL;
```

Em seguida, precisava-se das informações da tabela de leitos de hospitais (Leitos) que foi combinada através da seguinte consulta SQL utilizando o recurso criar tabela do Access:

```
Select *
From DSMS040Biros LEFT JOIN Leitos
On DSMS040Biros.CGC_HOSP = Leitos.CGC_HOSP;
```

Através do recurso de consulta do Access de localizar não coincidentes verificou-se que três CGCs não constavam na DSMS040Biros sendo então acrescentadas as instâncias da tabela Leitos que apresentavam estes CGCs. Após este acréscimo de instâncias a tabela DSMS040BirosLeitos foi renomeada para Hospitais contendo 401 instâncias.

Um novo atributo referente ao porte do hospital é acrescentado à tabela Hospitais sendo discretizado a partir dos valores contidos no atributo QTE_LEITOS (número de leitos do hospital), tendo sido os intervalos de valores definidos pelo especialista do domínio, conforme visualizado na Tabela 4.19.

TABELA 4.19 – Descrição do Atributo Porte do Hospital

ATRIBUTO	VALORES	INTERVALOS
PORTE_HOSP	PORTE1	<= 50 leitos
	PORTE2	51 – 100 leitos
	PORTE3	101 – 500 leitos

A partir de então foi feita a combinação de JunAno com Hospitais através da seguinte consulta SQL utilizando o recurso criar tabela do Access:

```
Select *
From JunAno LEFT JOIN Hospitais
On JunAno.CGC = Hospitais.CGC_HOSP;
```

Após o pré-processamento geral realizado na tabela JunAno, decidiu-se de acordo com relatórios fornecidos pela SES e sugestão do especialista do domínio selecionar um subconjunto de AIHs de JunAno referentes aos três procedimentos de maior freqüência no RS no ano de 2000, listados na Tabela 4.20, para investigar características destas AIHs em relação ao seu respectivo custo.

TABELA 4.20 – Descrição dos Três Procedimentos de Maior Freqüência

PROC_REA	Descrição
35001011	Parto Normal
76500225	Doença Pulmonar Obstrutiva Crônica
77500113	Insuficiência Cardíaca

A seleção do subconjunto de AIHs com estes procedimentos foi feita através da consulta SQL utilizando o recurso criar tabela do Access:

```
Select *
From JunAno
Where Proc_Rea = '35001011' OR Proc_Rea = '76500225' OR Proc_Rea = '77500113';
```

A nova tabela chamada JunAnoProcs que comporta agora apenas AIHs que apresentam estes três procedimentos com 107.661 instâncias, precisa ter as AIHs reapresentadas eliminadas, sendo mantidas àquelas da primeira apresentação por recomendação dos especialistas do domínio, pois a presença das reapresentadas alteraria qualquer levantamento estatístico.

Desta forma, a tabela JunAnoProcs é classificada em ordem crescente pelos seguintes atributos: CGC + N_AIH + DCIH + IDENT + DT_SAI + APRES em seguida, é criada uma nova tabela de nome JunAnoSemReap a partir da cópia da estrutura da tabela JunAnoProcs.

Na nova tabela JunAnoSemReap é definida uma chave primária composta pelos atributos: CGC + N_AIH + DCIH + IDENT + DT_SAI e realizada uma consulta acréscimo sobre a tabela JunAnoProcs de modo a acrescentar somente os registros exclusivos à nova tabela JunAnoSemReap. Assim, a tabela JunAnoSemReap comporta 106.679 instâncias. Efetuou-se uma cópia desta tabela pois a estrutura da mesma precisou ter os atributos considerados irrelevantes e redundantes (os que se repetiam em várias tabelas e aqueles que originaram os discretizados e os compostos) excluídos, bem

como teve a inclusão de um novo atributo, denominado CUSTO (com tipo de dado texto e tamanho de 5 caracteres).

A partir da tabela JunAnoSemReap foram geradas três tabelas cada uma contendo as AIHs que apresentavam um dos procedimentos em particular. Esta seleção foi realizada através de uma consulta em SQL manipulando o código do procedimento em questão e utilizando o recurso criar tabela do Access, por exemplo:

```
Select *
From JunAnoSemReap
Where PROC_REA = '35001011;
```

Então foram obtidas as seguintes tabelas: tbParto (41.106 instâncias), tbPulmo (39.520 instâncias) e tbCardio (26.053 instâncias).

O atributo CUSTO foi discretizado nas categorias BAIXO, MEDIO e ALTO a partir dos valores contidos no atributo VAL_TOTAL, tendo sido os intervalos de valores definidos pelo especialista do domínio em função do valor de tabela de cada procedimento realizado.

TABELA 4.21 – Descrição dos Valores dos Três Procedimentos de Maior Freqüência

PROCEDIMENTO REALIZADO	**VALOR DE TABELA (R$)**	**CUSTO**		
		BAIXO	**MEDIO**	**ALTO**
Parto Normal	205,00	<= 307,5	>307,50 e <= 410,00	> 410,00
Doenca Pulmonar Obstrutiva Crônica	310,00	<= 465,00	> 465,00 e <= 620,00	> 620,00
Insuficiencia Cardíaca	429,53	<= 644,29	> 644,29 e <= 859,06	> 859,06

Retomando-se a subfase de entendimento dos dados sobre estes subconjuntos selecionados de instâncias foi realizado um levantamento estatístico sobre os principais atributos das tabelas em questão, conforme apresentado nas Tabelas 4.22 e 4.23, utilizando para isto algumas funções estatísticas disponíveis no Excel 97, tais como:

- MÍNIMO : retorna o valor mínimo de um conjunto de valores.
- MÁXIMO: retorna o valor máximo de um conjunto de valores.
- MÉDIA: retorna a média aritmética de um conjunto de valores.
- DESVPAD: retorna o desvio padrão a partir de um conjunto de valores.

TABELA 4.22 – Levantamento Estatístico sobre o Valor Total da AIH

Tabela	**Atributo: VAL_TOTAL (R$)**			
	MÍNIMO	**MÁXIMO**	**MÉDIA**	**DESVIO PADRÃO**
tbParto	0,00	3.744,80	225,96	105,06
tbPulmo	0,00	152.248,74	440,90	1.148,78
tbCardio	0,00	175.746,70	514,38	1.700,85

Ao serem apresentados estes dados ao especialista do domínio constatou-se que tanto o valor mínimo R$ 0,00 (zero) quanto os valores máximos absurdos encontrados

para o valor total da AIH aparecem apenas nas apresentadas à SES, porém quando estas AIHs são enviadas ao Ministério da Saúde é feito um levantamento sobre as mesmas e corrigido então o valor total. Segundo um dos especialistas estes problemas possivelmente devem ocorrer por erros de digitação, problemas de importação de tabelas ou falhas de processamento cometidas pelos programas dos *bureaux*. Os valores efetivamente pagos pelo SUS como verificado posteriormente são apresentados a seguir:

- Para 3 (três) AIHs da tabela de parto normal com valor total igual a R$ 0,00 (zero) tem-se os valores corrigidos para R$ 205,00 (para duas AIHs) e R$ 225,00 (uma AIH).

- Para 1 (uma) AIH da tabela de doença pulmonar obstrutiva crônica com valor total igual a R$ 0,00 (zero) tem-se o valor total da AIH corrigido para R$ 388,17.

- Para 6 (seis) AIHs da tabela de insuficiência cardíaca com valor total igual a R$ 0,00 (zero) tem-se os valores corrigidos para todas de R$ 422,11.

- Para a AIH de parto normal cujo valor total é de R$ 3.744,80, este é corrigido para R$ 325,02.

- Para a AIH de doença pulmonar obstrutiva crônica cujo valor é de R$ 152.248,74, este é corrigido para R$ 532,60.

- Para a AIH de insuficiência cardíaca cujo valor total é de R$ 175.746,70, este é corrigido para R$ 605,13.

TABELA 4.23 – Levantamento Estatístico sobre o Número de Dias de Internação

Tabela	**Atributo: DIAS_INT**			
	MÍNIMO	**MÁXIMO**	**MÉDIA**	**DESVIO PADRÃO**
tbParto	0	152	1,61	1,56
tbPulmo	0	381	5,78	5,63
tbCardio	0	371	5,96	5,98

Quanto ao apresentado na Tabela 4.23 o valor mínimo 0 (zero) para o número de dias de internação ocorre quando o paciente ficou internado horas do mesmo dia no hospital. E no caso dos valores máximos encontrados, segundo os especialistas são AIHs a serem auditadas através dos prontuários diretamente no hospital para verificar o que ocorreu, pois estas excedem a média de internação encontrada. Estes dados despertaram o interesse para a criação de novos critérios de bloqueio considerando se o número de dias de internação está por exemplo 50% acima ou abaixo da média, bloqueando as AIHs com tal dado.

Juntamente com os especialistas estas AIHs foram investigadas na base de dados de AIHs pagas e verificou-se que elas são todas de identificação 1, ou seja normal, não justificando tantos dias de internação.

Decidiu-se também fazer alguns levantamentos quanto ao dia da semana de internação e de alta do paciente nas três tabelas em questão. Ressaltando-se que isto só foi possível devido a elaboração de um programa para realizar a discretização dos valores encontrados na forma de data nos atributos data de internação (DT_INT) e data

de alta do paciente (DT_SAI) gerando assim os valores dos atributos referentes ao dia da semana de internação (SEMANA_INT) e dia da semana de alta do paciente (SEMANA_SAI).

Assim, considerando-se todas as instâncias da tabela tbParto foram realizadas algumas investigações conforme visualizado nas Tabelas 4.24, 4.25 e 4.26.

TABELA 4.24 – Levantamento do Número de Instâncias por Dia da Semana para Internações e Altas de Pacientes nas AIHs de Parto Normal

Valores	Atributo: SEMANA_INT	Atributo: SEMANA_SAI
DOMINGO	5703	5416
SEGUNDA	6114	5977
TERCA	5839	5992
QUARTA	5816	5859
QUINTA	5981	5908
SEXTA	5912	5966
SÁBADO	5741	5988

TABELA 4.25 – Levantamento Estatístico do Número de Dias de Internação considerando o Dia da Semana da Internação de Pacientes para as AIHs de Parto Normal

Valores	Atributo:SEMANA_INT		
	Mínimo (DIAS_INT)	Máximo (DIAS_INT)	Média (DIAS_INT)
DOMINGO	0	152	1,60
SEGUNDA	0	33	1,60
TERCA	0	123	1,62
QUARTA	0	31	1,60
QUINTA	0	122	1,60
SEXTA	0	34	1,60
SABADO	0	35	1,63

TABELA 4.26 – Levantamento Estatístico do Número de Dias de Internação considerando o Dia da Semana da Alta de Pacientes para as AIHs de Parto Normal

Valores	Atributo: SEMANA_SAI		
	Mínimo (DIAS_INT)	Máximo (DIAS_INT)	Média (DIAS_INT)
DOMINGO	0	122	1,60
SEGUNDA	0	32	1,65
TERCA	0	31	1,58
QUARTA	0	32	1,61
QUINTA	0	34	1,60
SEXTA	0	152	1,60
SABADO	0	123	1,61

Também foram realizadas algumas verificações nas tabelas tbPulmo e tbCardio considerando todas as instâncias que apresentavam o caráter de internação: Urgência/ Emergência, com AIH emitida após a internação. O levantamento realizado é exibido nas Tabelas 4.27, 4.28 e 4.29.

TABELA 4.27 – Levantamento do Número de Instâncias por Dia da Semana para Internações e Altas nas AIHs de Doença Pulmonar Obstrutiva Crônica e Insuficiência Cardíaca

Valores	**Atributo: SEMANA_INT**		**Atributo: SEMANA_SAI**	
	tbPulmo	**tbCardio**	**tbPulmo**	**tbCardio**
DOMINGO	2748	1711	2568	1619
SEGUNDA	6824	4508	5478	3388
TERCA	5600	3728	4673	3075
QUARTA	5138	3377	4225	2938
QUINTA	4828	3128	4815	3311
SEXTA	4868	3135	6074	3946
SABADO	3046	2022	5219	3332

Como é desproporcional o número de internações feitas no sábado e no domingo em relação aos demais dias da semana para as AIHs de insuficiência cardíaca com caráter de internação de urgência/ emergência, com AIH emitida após a internação, apresentou-se estes dados ao especialista que os considera significativos se foram os mesmos pacientes que se internaram durante uns três a quatro dias receberam alta e depois de três ou quatro dias tornaram a se internar novamente. Muitas vezes fazem isto por não terem dinheiro para comprar remédios, por exemplo. Assim, mereceria uma investigação mais profunda do que aconteceu.

TABELA 4.28 – Levantamento Estatístico do Número de Dias de Internação considerando o Dia da Semana de Internação para as AIHs de Doença Pulmonar Obstrutiva Crônica e Insuficiência Cardíaca

Valores	**Atributo:SEMANA_INT**					
	Mínimo (DIAS_INT)		**Máximo (DIAS_INT)**		**Média (DIAS_INT)**	
	tbPulmo	tbCardio	tbPulmo	tbCardio	tbPulmo	tbCardio
DOMINGO	0	0	79	48	5,93	5,90
SEGUNDA	0	0	72	85	5,51	5,57
TERCA	0	0	52	52	5,50	5,75
QUARTA	0	0	65	52	5,65	5,75
QUINTA	0	0	381	371	6,01	6,33
SEXTA	0	0	76	368	5,93	6,32
SABADO	0	0	377	43	6,19	6,15

TABELA 4.29 – Levantamento Estatístico do Número de Dias de Internação considerando o Dia da Semana de Alta para as AIHs de Doença Pulmonar Obstrutiva Crônica e Insuficiência Cardíaca

Valores	**Atributo:SEMANA_SAI**					
	Mínimo (DIAS_INT)		**Máximo (DIAS_INT)**		**Média (DIAS_INT)**	
	tbPulmo	tbCardio	tbPulmo	tbCardio	tbPulmo	tbCardio
DOMINGO	0	0	381	69	5,45	5,38
SEGUNDA	0	0	60	94	5,95	6,15
TERCA	0	0	79	368	6,29	6,66
QUARTA	0	0	377	65	6,23	6,23
QUINTA	0	0	76	371	5,60	6,14
SEXTA	0	0	377	99	5,55	5,63
SABADO	0	0	59	48	5,26	5,36

A seguir da Tabela 4.30 até a 4.39 tem-se os resultados do levantamento feito sobre o número de instâncias nas tabelas tbParto, tbPulmo e tbCardio em relação aos valores distintos assumidos pelos seus atributos simbólicos.

TABELA 4.30 – Levantamento do Número de Instâncias por Especialidade da AIH

Atributo: ESPEC				
Valores Distintos	**Descrição**	**Número de Instâncias**		
		tbParto	**tbPulmo**	**tbCardio**
01	Cirurgia Geral	2	-	-
02	Obstetrícia	41.101	-	-
03	Clínica Médica	3	39.519	26.048
07	Pediatria	-	1	5

- não apresenta AIH com este valor

TABELA 4.31 – Levantamento do Número de Instâncias por Caráter de Internação

Atributo: CAR_INT				
Valores Distintos	**Descrição**	**Número de Instâncias**		
		tbParto	**tbPulmo**	**tbCardio**
01	Eletiva	1123	432	425
03	Urgência/ Emergência, com AIH emitida antes da internação	6473	6030	4010
04	Internação em AIH de alta complexidade fora do Estado	2	1	1
05	Urgência/ Emergência, com AIH emitida após a internação	33507	33052	21609
06	Acidente no local de trabalho ou a serviço da empresa	-	1	1
08	Outros tipos de acidentes de trânsito	-	-	1
09	Outros tipos de lesões e envenenamento	-	4	2
20	Urgência/ Emergência em unidade de referência	1	-	4

- não apresenta AIH com este valor

TABELA 4.32 – Levantamento do Número de Instâncias por Categoria

Atributo: CATEGORIA			
Valores Distintos	**Número de Instâncias**		
	tbParto	**tbPulmo**	**tbCardio**
CRIANCA	3	1	3
JOVEM	13305	27	83
ADULTO	27795	8907	5614
IDOSO	3	30585	20353

Analisando os dados da Tabela 4.32 juntamente com o especialista, em especial para a tabela de parto normal (tbParto), verificou-se consultando a base de AIHs pagas que foi efetuado o pagamento das mesmas mesmo considerando o critério de rejeição da AIH pelo SUS que não aceita parto em paciente com idade fora da faixa de 12 a 55 anos, o que poderia ser verificado se ocorreu por exemplo, alguma falha no sistema do DATASUS (Departamento de Informática do SUS) quanto à aplicação do bloqueio.

TABELA 4.33 – Levantamento do Número de Instâncias por Categoria de Classificação Internacional de Doenças

Atributo: CAT_CID				
Valores Distintos	**Descrição**	**Número de Instâncias**		
		tbParto	**tbPulmo**	**tbCardio**
GP01	Algumas doenças infecciosas e parasitárias	-	11	7
GP02	Neoplasias (Tumores)	-	7	1
GP03	Doenças do sangue e alguns transtornos imunitários	2	2	2
GP04	Doenças endócrinas nutricionais e metabólicas	-	7	5
GP05	Transtornos mentais e comportamentais	-	1	-
GP06	Doenças do sistema nervoso	1	10	23
GP07	Doenças do olho e anexos	1	1	-
GP09	Doenças do aparelho circulatório	2	30	25901
GP10	Doenças do aparelho respiratório	1	39401	71
GP11	Doenças do aparelho digestivo	-	15	8
GP12	Doenças da pele e do tecido subcutâneo	-	-	4
GP13	Doenças do sist. osteomuscular e tecido conjuntivo	-	18	4
GP14	Doenças do aparelho geniturinário	3	9	15
GP15	Gravidez, parto e puerpério	41.093	1	1
GP16	Algumas afecções originadas no periodo pré-natal	2	-	-
GP18	Sintomas e achados anormais em exames clínicos e laboratoriais	-	-	2
GP19	Lesões, envenen. e algumas outras conseq. de causas externas	-	5	4
GP20	Causas externas de morbidade e mortalidade	-	1	1
GP21	Fatores que influenciam o estado de saúde, contato c/ serv. de saúde	1	1	4

- não apresenta AIH com este valor

Sabendo-se que o atributo categoria de CID foi discretizado a partir do atributo de diagnóstico principal (DIAG_PRI), cabe ressaltar que o procedimento realizado e o

diagnóstico principal devem estar relacionados, sendo que isto não acontece em alguns casos.

Esta relação entre procedimento realizado e diagnóstico principal é uma questão que pode ser transformada em critério de bloqueio de AIH já que é de conhecimento que nem sempre os valores destes atributos são compatíveis.

TABELA 4.34 – Levantamento do Número de Instâncias por Categoria de Motivo de Cobrança

Atributo: CAT_MCOB				
Valores Distintos	**Descrição**	**Número de Instâncias**		
		TbParto	**tbPulmo**	**tbCardio**
MC01	Alta	40689	37571	24.123
MC02	Permanência maior que 30 dias	-	105	95
MC03	Transferência	22	178	251
MC04	Óbito com autópsia	1	141	141
MC05	Óbito sem autópsia	1	1524	1443
MC06	Alta por reoperação	6	1	-
MC07	Alta da prematuridade e permanência do recém-nascido	387	-	-

- não apresenta AIH com este valor

TABELA 4.35 – Levantamento do Número de Instâncias por Natureza do Hospital

Atributo: NAT				
Valores Distintos	**Descrição**	**Número de Instâncias**		
		tbParto	**tbPulmo**	**tbCardio**
10	Hospital próprio	-	1	-
20	Hospital contratado	7114	9538	5404
50	Hospital municipal	3105	2471	1785
60	Hospital filantrópico	25464	25124	16102
61	Hospital filantrópico isento pela IN 01/97 SRF	3208	1425	1187
63	Hospital filantrópico parcialmente isento	42	58	51
70	Hospital universitário de ensino	84	92	99
90	Hospital universitário com FIDEPS	2089	811	1425

- não apresenta AIH com este valor

TABELA 4.36 – Levantamento do Número de Instâncias pelo Porte do Hospital

Atributo:PORTE_HOSP				
Valores Distintos	**Descrição**	**Número de Instâncias**		
		tbParto	**tbPulmo**	**tbCardio**
PORTE1	<= 50 leitos	7972	13012	6872
PORTE2	51 – 100 leitos	12021	13252	7588
PORTE3	101 – 500 leitos	21113	13256	11593

TABELA 4.37 – Levantamento do Número de Instâncias por *Bureau*

Atributo:BIRO			
Valores Distintos	**Número de Instâncias**		
	tbParto	**tbPulmo**	**tbCardio**
Biro1	17318	19922	12961
Biro2	3027	3789	2157
Biro3	4007	2743	1837
Biro4	537	693	469
Biro5	2099	3358	1376
NENHUM	14118	9015	7253

Para a tabela tbParto o atributo SEXO apresenta obviamente o valor 3 (feminino) e o atributo IDENT apresenta o valor 1 (AIH normal). Já para as tabelas tbPulmo e tbCardio tem-se o visualizado na Tabela 4.38.

TABELA 4.38 – Levantamento do Número de Instâncias por Identificação da AIH e pelo Sexo do Paciente

Atributo	**Valores Distintos**	**Descrição**	**Número de Instâncias**	
			tbPulmo	**tbCardio**
IDENT	1	AIH normal	39519	26045
	5	AIH de longa permanência e FTP	1	8
SEXO	1	Masculino	23228	10744
	3	Feminino	16292	15309

Quanto ao número de hospitais distintos apresentados nas três tabelas tem-se o descrito na Tabela 4.39.

TABELA 4.39 – Levantamento do Número de Hospitais

Tabela	**Número de CGCs Distintos de Hospitais**
tbParto	309
tbPulmo	325
tbCardio	326

Diante dos valores máximos encontrados nas três tabelas para o atributo valor total da AIH (VAL_TOTAL), conforme o apresentado na Tabela 4.22, decidiu-se fazer um levantamento estatístico para alguns hospitais que apresentavam os valores mais altos em cada uma das tabelas, obtendo-se o seguinte:

a)Para a tabela tbParto

Na tabela de AIHs de parto normal foram selecionados dois hospitais com o valor total de AIH alto, listados na Tabela 4.40.

TABELA 4.40 – Dados de AIHs com Maior Valor Total para Parto Normal

Características	CGC do Hospital	
	CGCHosp1	CGCHosp2
Município do hospital	Rio Pardo	Caçapava do Sul
Porte do hospital	PORTE3	PORTE2
Bureau	Biro1	Biro1
Categoria de motivo de cobrança	Alta	Alta
Especialidade da AIH	Obstetrícia	Obstetrícia
Identificação da AIH	AIH normal	AIH normal
Caráter de internação	Urgência/ Emergência com AIH emitida após a internação	Urgência/ Emergência com AIH emitida após a internação
Número de dias de internação	3	3
Categoria de Classificação Internacional de Doenças	Gravidez, parto e puerpério	Gravidez, parto e puerpério
Valor total da AIH (R$)	3.744,80	3.292,97

- Considerando-se o CGC = "CGCHosp1" tem-se que a média do valor total para as AIHs de parto normal é de R$ 483,10.

- Já para o CGC = "CGCHosp2" tem-se que a média do valor total para as AIHs de parto normal é de R$ 574,37.

- Ao serem apresentados estes dados ao especialista verificou-se que devido a média do valor total das AIHs de parto normal ser de R$ 225,96, faz-se necessária uma melhor investigação sobre as AIHs destes dois hospitais, já que o de CGC = "CGCHosp1" apresenta o primeiro valor total mais alto de AIH, além de outras 97 instâncias com custo alto. Já o de CGC = "CGCHosp2" apresenta o segundo valor total mais alto de AIH, além de outras 19 instâncias com custo alto.

- Na base de dados das AIHs pagas observou-se que a AIH cujo valor total é de R$ 3.744,80 teve o mesmo corrigido para R$ 325,02 assim como também a AIH de valor total R$ 3.292,97 sofreu uma correção para R$ 268,26.

Considerando-se que na Tabela 4.40 é o mesmo *bureau* que está relacionado aos hospitais daquelas AIHs, resolveu-se realizar outras investigações relacionando *bureau* com estatísticas do valor total e do custo da AIH, considerando-se todas as AIHs de parto normal, conforme o apresentado nas Tabelas 4.41 e 4.42.

TABELA 4.41 – Levantamento Estatístico do Valor Total de AIHs de Parto Normal considerando o *Bureau*

BIRO	Atributo: VAL_TOTAL (R$)		
	MÍNIMO	MÁXIMO	MÉDIA
Biro1	0,00	3744,80	232,94
Biro2	194,78	615,33	216,69
Biro3	194,78	597,05	208,92
Biro4	194,78	295,02	203,84
Biro5	194,78	381,32	207,83
NENHUM	194,78	1585,32	227,77

TABELA 4.42 – Levantamento do Número de Instâncias de cada *Bureau* em Relação ao Custo para AIHs de Parto Normal

BIRO	Atributo: CUSTO					
	BAIXO		MÉDIO		ALTO	
Biro1	16.959	97,93%	92	0,53%	267	1,54%
Biro2	3.017	99,67%	9	0,30%	1	0,03%
Biro3	3.990	99,58%	15	0,37%	2	0,05%
Biro4	537	100,00%	0	0,00%	0	0,00%
Biro5	2.096	99,86%	3	0,14%	0	0,00%
NENHUM	14.019	99,30%	67	0,47%	32	0,23%

Observa-se que o *bureau* Biro1 é o que apresenta maior número de AIHs com custo alto e custo médio.

b)Para a tabela tbPulmo

Na tabela de AIHs de doença pulmonar obstrutiva crônica foram selecionados dois hospitais com o valor total de AIH alto, listados na Tabela 4.43.

TABELA 4.43 – Dados de AIHs com Maior Valor Total para Doença Pulmonar Obstrutiva Crônica

	CGC do Hospital	
Características	**CGCHosp3**	**CGCHosp4**
Município do Hospital	Rio Grande	Camaquã
Porte do Hospital	PORTE3	PORTE3
Bureau	NENHUM	Biro1
Categoria de motivo de cobrança	Alta	Alta
Especialidade da AIH	Clínica médica	Clínica médica
Identificação da AIH	AIH normal	AIH normal
Caráter da internação	Urgência/ Emergência com AIH emitida após a internação	Urgência/ Emergência com AIH emitida após a internação
Número de dias de internação	13	13
Categoria de Classificação Internacional de Doenças	Doenças do aparelho respiratório	Doenças do aparelho respiratório
Valor total da AIH (R$)	152.248,74	51.600,47

- Para o CGC = "CGCHosp3" a média do valor total das AIHs de Doença Pulmonar Obstrutiva Crônica é de R$ 4.368,28.

- Já para o CGC = "CGCHosp4" a média do valor total das AIHs de Doença Pulmonar Obstrutiva Crônica é de R$ 1.011,47.

- Analisando estes dados juntamente com o especialista verificou-se que em função da média do valor total de AIHs de doença pulmonar obstrutiva crônica ser de R$ 440,90, também existe a necessidade de uma melhor investigação das AIHs destes dois hospitais, pois o de CGC = "CGCHosp3" apresenta os quatro primeiros valor total mais altos além de outras 17 instâncias com custo alto. E o de CGC = "CGCHosp4" apresenta o quinto valor total mais alto, além de outras 38 instâncias de custo alto.

- Na base de dados de AIHs pagas a AIH de valor total R$ 152.248,74 teve o mesmo corrigido para R$ 532,60. E a AIH de valor total R$ 51.600,47 teve este corrigido para R$ 628,25.

Para a tabela de doença pulmonar obstrutiva crônica também resolveu-se realizar investigações relacionando o bureau com estatísticas do valor total e custo da AIH, considerando todas as AIHs deste procedimento, conforme visualizado nas Tabelas 4.44 e 4.45.

TABELA 4.44 – Levantamento Estatístico do Valor Total de AIHs de Doença Pulmonar Obstrutiva Crônica considerando o *Bureau*

BIRO	**Atributo: VAL_TOTAL (R$)**		
	MÍNIMO	**MÁXIMO**	**MÉDIA**
Biro1	0,00	51.600,47	427,36
Biro2	388,17	4.498,17	432,96
Biro3	388,17	1.724,75	420,17
Biro4	388,17	627,36	395,23
Biro5	388,17	1.246,26	402,36
NENHUM	122,15	152.248,74	498,32

TABELA 4.45 – Levantamento do Número de Instâncias de cada *Bureau* em Relação ao Custo para AIHs de Doença Pulmonar Obstrutiva Crônica

BIRO	**Atributo: CUSTO**					
	BAIXO		**MÉDIO**		**ALTO**	
Biro1	18.586	93,29%	896	4,50%	440	2,21%
Biro2	3.445	90,92%	203	5,36%	141	3,72%
Biro3	2.481	90,45%	169	6,16%	93	3,39%
Biro4	688	99,28%	4	0,58%	1	0,14%
Biro5	3.256	96,96%	80	2,38%	22	0,66%
NENHUM	7.674	85,12%	715	7,93%	626	6,94%

Observa-se que o *bureau* Biro1 é o que apresenta mais AIHs com custo alto e custo médio.

c) Para a tabela tbCardio

Na tabela de AIHs de insuficiência cardíaca foram selecionados três hospitais com o valor total de AIH alto, listados na Tabela 4.46.

TABELA 4.46 – Dados de AIHs com Maior Valor Total para Insuficiência Cardíaca

	CGC do Hospital		
Características	**CGCHosp3**	**CGCHosp4**	**CGCHosp5**
Município do Hospital	Rio Grande	Camaquã	Passo Fundo
Porte do Hospital	PORTE3	PORTE3	PORTE3
Bureau	NENHUM	Biro1	NENHUM
Categoria de motivo de cobrança	Alta	Alta	Permanência maior que 30 dias
Especialidade da AIH	Clínica médica	Clínica médica	Clínica médica
Identificação da AIH	AIH normal	AIH normal	AIH normal
Caráter da internação	Urgência/Emergência com AIH emitida após a internação	Urgência/Emergência com AIH emitida após a internação	Urgência/ Emergência com AIH emitida após a internação
Número de dias de internação	15	10	27
Categoria de Classificação Internacional de Doenças	Doenças do aparelho circulatório	Doenças do aparelho circulatório	Doenças do aparelho circulatório
Valor total da AIH (R$)	175.746,70	39.055,73	4.533,07

- Para o CGC = “CGCHosp3” a média do valor total das AIHs de Insuficiência Cardíaca é de R$ 2.479,43.

- Já para o CGC = “CGCHosp4” a média do valor total das AIHs de Insuficiência Cardíaca é de R$ 964,96.

- No caso do CGC = “CGCHosp5” a média do valor total das AIHs de Insuficiência Cardíaca é de R$ 743,60.

- Através da análise realizada sobre estes dados com o auxílio do especialista verificou-se que também é preciso realizar investigações mais profundas sobre as AIHs destes três hospitais, já que a média do valor total das AIHs de insuficiência cardíaca é de R$ 514,38 e o hospital de CGC = “CGCHosp3” apresenta os sete primeiros valores mais altos para o valor total da AIH, além de outras 51 instâncias de custo alto. O hospital de CGC = “CGCHosp4” apresenta o oitavo valor total mais alto, além de outras 16 instâncias com custo alto. Enquanto o CGC = “CGCHosp5” apresenta o vigésimo primeiro valor total mais alto sendo considerado como categoria de motivo de cobrança a permanência maior que 30 dias quando o paciente ficou internado por 27 dias, e este hospital é responsável ainda por outras 152 instâncias com custo alto.

- Na tabela de AIHs pagas verificou-se que a AIH de valor total R$ 175.746,70 teve o mesmo corrigido para R$ 605,13. Já a AIH de valor total R$ 39.055,73 sofreu correção para R$ 508,70. E a AIH de valor total R$ 4.533,07 teve este corrigido para R$ 3.053,93.

Para as AIHs de insuficiência cardíaca também foram realizadas investigações quanto a relação do *bureau* com estatísticas do valor total e custo da AIH, para todas as AIHs, sendo apresentadas nas Tabelas 4.47 e 4.48.

TABELA 4.47 – Levantamento Estatístico do Valor Total de AIHs de Insuficiência Cardíaca considerando o *Bureau*

BIRO	**Atributo: VAL_TOTAL (R$)**		
	MÍNIMO	**MÁXIMO**	**MÉDIA**
Biro1	0,00	39055,73	471,21
Biro2	422,11	4532,11	513,89
Biro3	422,11	1929,67	456,61
Biro4	422,11	713,56	432,36
Biro5	422,11	759,43	439,46
NENHUM	193,55	175746,7	625,80

TABELA 4.48 – Levantamento do Número de Instâncias de cada *Bureau* em Relação ao Custo para AIHs de Insuficiência Cardíaca

BIRO	**Atributo: CUSTO**					
	BAIXO		**MÉDIO**		**ALTO**	
Biro1	12.406	95,72%	272	2,10%	283	2,18%
Biro2	1.953	90,54%	74	3,43%	130	6,03%
Biro3	1.764	96,03%	57	3,10%	16	0,87%
Biro4	468	99,79%	1	0,21%	0	0,00%
Biro5	1.368	99,42%	8	0,58%	0	0,00%
NENHUM	6.176	85,15%	489	6,74%	588	8,11%

Verifica-se que o *bureau* Biro1 é o que apresenta maior número de AIHs com custo alto e custo médio.

Após estas investigações volta-se a dar continuidade à subfase de seleção de dados no que se refere a atributos. Para as três tabelas foram selecionados os atributos considerados mais relevantes de acordo com os especialistas do domínio a fim de se tentar descobrir características comuns das AIHs em relação ao CUSTO.

Ressalta-se que, todas as vezes em que se precisou definir os atributos relevantes, isto constituiu-se em um processo demorado em função das diferentes opiniões e interesses dos especialistas, tornando-se necessárias várias visitas à SES para participar de reuniões com os mesmos tentando-se definir os atributos. E, finalmente, foram mantidos apenas os atributos listados na Tabela 4.49.

TABELA 4.49 – Atributos Relevantes

Atributo	Tipo de Dado	Tamanho
ESPEC	Texto	2
CGC	Texto	14
IDENT	Texto	1
SEXO	Texto	1
CAR_INT	Texto	2
NAT	Texto	2
CATEGORIA	Texto	7
PORTE_HOSP	Texto	6
BIRO	Texto	50
CAT_CID	Texto	4
CAT_MCOB	Texto	4
CUSTO	Texto	5

Para realizar um teste de mineração de dados, os dados precisaram ser preparados também de acordo com os requisitos da ferramenta See5, deste modo foi preciso criar um arquivo .names e outro .data para cada tabela.

A criação do arquivo .names é bastante simples pois utilizando um editor de texto como o Bloco de Notas deve-se listar os atributos com seus respectivos tipo de dados para o See5 e valores assumidos, além de ter que ser definido o atributo de classe ou classificador, que neste experimento é o CUSTO. Sendo o arquivo salvo com a extensão .names.

Já para a criação do arquivo .data, é preciso se certificar de que o atributo de classe seja o último listado na estrutura da tabela, então é feita a exportação da tabela original que no caso está no Access para o Excel (com extensão .xls). Abrindo esta tabela no Excel pode-se salvá-la com extensão .csv para que os atributos sejam separados por vírgula, em seguida este arquivo deve ser aberto em um editor de texto como o WordPad , pois no bloco de notas não é possível abrí-lo devido ao tamanho do arquivo, e eliminada a primeira linha do arquivo que contém os nomes dos atributos. Em seguida, salva-se o arquivo ainda com extensão .csv sendo depois renomeado para .data.

a) Para a tabela tbCardio

Tem-se o arquivo tbCardio.names exibido na Figura 4.1 cujo atributo de classe é o CUSTO definido no início do arquivo e com os valores assumidos listados no final do mesmo.

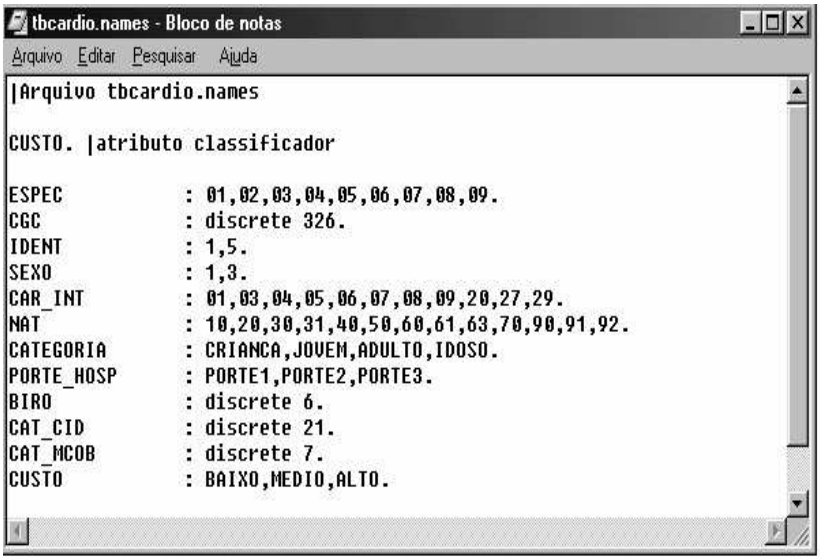

FIGURA 4.1 – Arquivo tbCardio.names

4.1.3 Mineração de Dados

Quanto às regras geradas a partir do See5, é preciso entender a apresentação de alguns elementos na listagem das mesmas, descritos a seguir, conforme o visualizado na Figura 4.2:

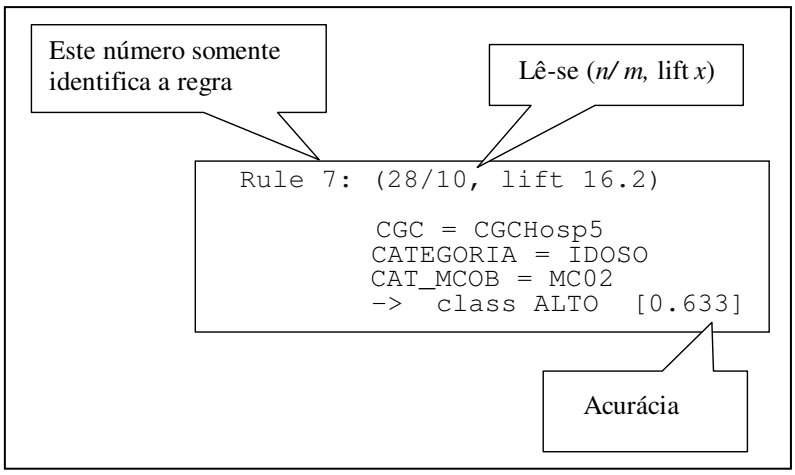

FIGURA 4.2 – Exemplo de Regra em See5

Em *n/ m* tem-se que *n* é o número de casos de treinamento cobertos pela regra e *m*, quando aparecer, mostra quantos destes não pertencem a classe predita pela regra.

A acurácia é estimada pela razão de Laplace:

$$Acurácia = \frac{(n - m + 1)}{n + 2}$$

O *lift* baseia-se na acurácia e na freqüência relativa da classe predita, dado pela seguinte fórmula:

$$lift = \frac{acurácia}{freqüência\ relativa\ da\ classe}$$

onde a freqüência relativa da classe corresponde a razão entre o número de instâncias da classe e o número total de instâncias de treinamento.

Assim para a mineração de dados a partir do arquivo tbCardio.data destacam-se as seguintes regras apresentadas na Figura 4.3:

```
Rule 4: (8/2, lift 20.2)
        CGC = CGCHosp6
        CAT_MCOB = MC02
        ->  class MEDIO  [0.700]


Rule 6: (23/5, lift 19.5)
        CGC = CGCHosp5
        SEXO = 1
        CAT_MCOB = MC02
        ->  class ALTO  [0.760]

Rule 7: (28/10, lift 16.2)
        CGC = CGCHosp5
        CATEGORIA = IDOSO
        CAT_MCOB = MC02
        ->  class ALTO  [0.633]

Rule 8: (49/21, lift 14.6)
        CGC = CGCHosp5
        CAT_MCOB = MC02
        ->  class ALTO  [0.569]
```

FIGURA 4.3 – Regras Geradas para tbCardio.data

Tentou-se aplicar o atributo de classe CUSTO para a mineração de dados das demais tabelas de parto normal e doença pulmonar obstrutiva crônica, porém o See5 não gerou nenhuma regra.

Foram também realizadas outras minerações considerando outras combinações de atributos para as três tabelas, porém não foi gerada nenhuma regra aparecendo apenas a classe *default* BAIXO, referente ao custo baixo das AIHs que é o predominante nestas tabelas.

4.1.4 Pós-Processamento

Para a tabela de insuficiência cardíaca, das 8 regras geradas quatro se destacaram, sejam elas:

Regra 4 – Se o CGC do hospital é "CGCHosp6" e a categoria de motivo de cobrança é permanência maior que 30 dias, então o custo é médio. Esta regra apresenta 70% de acurácia.

Regra 6 – Se o CGC do hospital é "CGCHosp5" e o sexo do paciente é masculino e a categoria de motivo de cobrança é permanência maior que 30 dias, então o custo é alto. Esta regra apresenta 76% de acurácia.

Regra 7 – Se o CGC do hospital é "CGCHosp5" e a categoria do paciente é idoso e a categoria de motivo de cobrança é permanência maior que 30 dias, então o custo é alto. Esta regra apresenta 63,3% de acurácia.

Regra 8 – Se o CGC do hospital é "CGCHosp5" e a categoria de motivo de cobrança é permanência maior que 30 dias, então o custo é alto. Esta regra apresenta 56,9 % de acurácia.

Nota-se que nas quatro regras a categoria de motivo de cobrança é a mesma, permanência maior que 30 dias, sendo que para o hospital apresentado na regra 4 o custo da AIH é médio e para as demais regras que consideram outro hospital o custo da AIH é alto. Para um dos especialistas isto mereceria uma investigação junto aos hospitais para averigüar o ocorrido.

4.2 Experimento 2

Neste experimento deseja-se identificar as características comuns das AIHs de parto normal e de doença pulmonar obstrutiva crônica utilizando o atributo *bureau* (BIRO) para classificar as instâncias.

4.2.1 Entendimento do Domínio do Problema

Trata-se do mesmo contexto do experimento 1, porém o objetivo é encontrar as características comuns das AIHs da tabela de parto normal e de doença pulmonar obstrutiva crônica quanto ao *bureau* associado ao hospital que apresentou as AIHs. Sendo assim, o atributo de classe a ser considerado é o BIRO.

O problema de mineração de dados (MD) continua sendo o de classificação e da mesma forma a técnica bem como a ferramenta de MD continuam sendo regras de classificação e See5 respectivamente.

4.2.2 Pré-Processamento

São utilizadas as tabelas de parto normal e de doença pulmonar obstrutiva crônica já pré-processadas conforme o descrito na seção 4.1.2 porém as alterações realizadas foram em relação ao posicionamento do atributo de classe na estrutura destas tabelas, que no caso é o atributo BIRO sendo posicionado como último atributo na lista de atributos da estrutura das tabelas, e o fato de serem ignorados alguns atributos como o SEXO (para tbParto), por ser de conhecimento que todas as AIHs eram de pacientes do sexo feminino e o CGC (para tbParto e tbPulmo) por não contribuir para a geração de regras significativas, tendo isto sido observado após alguns testes de mineração em que o CGC foi considerado.

Também foram gerados os arquivos .names e .data para cada uma das tabelas em questão com o mesmo procedimento descrito na seção 4.1.2.

a) Para a tabela tbParto

Tem-se o arquivo tbParto.names exibido na Figura 4.4, cujo atributo de classe é o BIRO declarado no início deste arquivo. E tendo seus valores assumidos listados ao final do mesmo.

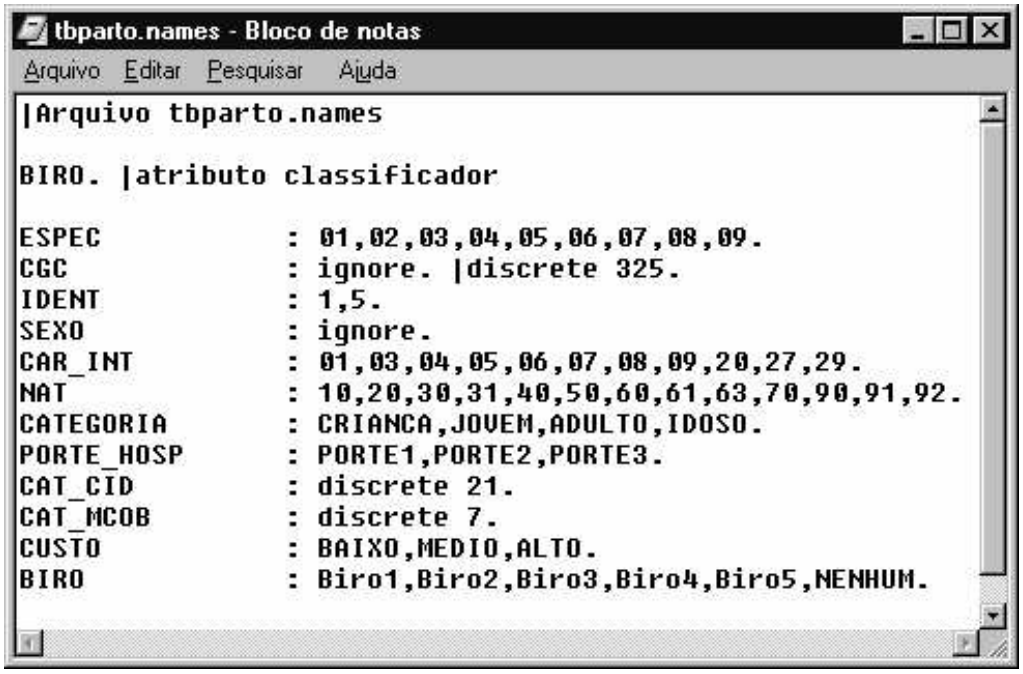

FIGURA 4.4 – Arquivo tbParto.names

b) Para a tabela tbPulmo

Tem-se o arquivo tbPulmo.names exibido na Figura 4.5, cujo atributo de classe também é o BIRO declarado no início deste arquivo com os seus valores assumidos listados na última linha do mesmo.

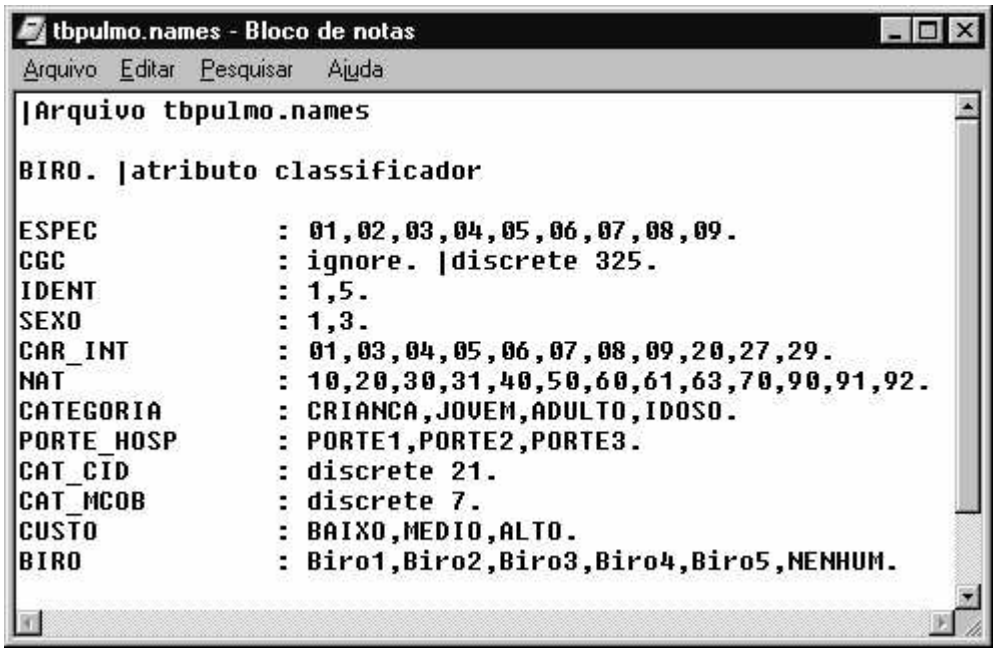

FIGURA 4.5 – Arquivo tbPulmo.names

4.2.3 Mineração de Dados

a) Para o tbParto.data

Submetendo-se então o arquivo tbParto.data ao processo de mineração de dados pelo See5, tem-se a geração de regras sendo que destas destacam-se as seguintes apresentadas na Figura 4.6:

```
Rule 9: (180/82, lift 7.4)
        CAR_INT = 05
        NAT = 50
        PORTE_HOSP = PORTE2
        CAT_MCOB = MC01
        ->  class Biro2  [0.544]

Rule 24: (253/6, lift 10.0)
        CAR_INT = 05
        NAT = 61
        PORTE_HOSP = PORTE2
        ->  class Biro3  [0.973]
```

FIGURA 4.6 – Regras Geradas para tbParto.data

b) Para o tbPulmo.data

Realizando-se a mineração de dados sobre o arquivo tbPulmo.data obteve-se a geração de regras pelo See5 destacando-se as seguintes visualizadas na Figura 4.7:

```
Rule 14: (5, lift 8.9)
        NAT = 60
        CAT_CID = GP02
        ->  class Biro2  [0.857]

Rule 15: (25/3, lift 8.9)
        CAR_INT = 03
        NAT = 60
        PORTE_HOSP = PORTE2
        CAT_MCOB = MC05
        ->  class Biro2  [0.852]

Rule 41: (75/22, lift 10.1)
        NAT = 20
        CATEGORIA = IDOSO
        PORTE_HOSP = PORTE1
        CUSTO = ALTO
        ->  class Biro3  [0.701]
```

FIGURA 4.7 – Regras Geradas para tbPulmo.data

4.2.4 Pós-Processamento

Estas minerações de dados considerando o BIRO (*bureau*) como atributo de classe são importantes, pois ainda não foi considerada nas regras de bloqueio de AIHs a associação do *bureau* com as características das AIHs.

Para a tabela de parto normal dentre 28 regras geradas destacam-se duas, sejam elas:

Regra 9 – Se caráter de internação é urgência/ emergência, com AIH emitida após a internação e natureza do hospital é hospital municipal e o porte do hospital é porte2 (de 51 a 100 leitos) e categoria de motivo de cobrança é a alta do paciente, então o *bureau* é Biro2. Esta regra apresenta 52,9 % de acurácia.

Regra 24 – Se o caráter de internação é urgência/ emergência, com AIH emitida após a internação e natureza do hospital é hospital filantrópico isento e porte do hospital é porte2 (de 51 a 100 leitos), então o *bureau* é Biro3. Esta regra apresenta 97,3 % de acurácia.

Já para a tabela de doença pulmonar obstrutiva crônica dentre 43 regras geradas três mereceram destaque, sejam elas:

Regra 14 – Se natureza do hospital é hospital filantrópico e categoria de classificação internacional de doenças é neoplasias (tumores), então *bureau* é Biro2. Esta regra apresenta 85,7% de acurácia.

Regra 15 – Se caráter de internação é urgência/ emergência, com AIH emitida antes da internação e natureza do hospital é hospital filantrópico e porte do hospital é porte2 (de 51 a 100 leitos) e categoria de motivo de cobrança é óbito sem autópsia, então o *bureau* é Biro2. Esta regra apresenta 85,2% de acurácia.

Regra 41 – Se natureza do hospital é hospital contratado e categoria é idoso e porte do hospital é porte1 (até 50 leitos) e custo é alto, então o *bureau* é Biro3. Apresentando 70.1% de acurácia.

Estas regras mostram as principais características das AIHs de alguns *bureaux*, considerando-se as da tabela de parto normal a as de doença pulmonar obstrutiva crônica, sendo para esta última percebido a presença de tumores e óbitos sem autópsia para AIHs cujo *bureau* é o Biro2, e a combinação porte1 (até 50 leitos) e custo alto para algumas AIHs cujo *bureau* é o Biro3. Após a análise por parte dos especialistas sobre as regras em questão, os mesmos consideraram a necessidade de serem feitas investigações sobre as AIHs correspondentes.

4.3 Experimento 3

Considerando-se o problema de descoberta de padrões temporais sobre a mesma base de dados de AIHs, relata-se então o pré-processamento realizado por Lucas (2002). Este relato é bastante interessante no sentido de mostrar a diferença no tratamento dos dados aplicando-se uma mineração de dados temporais já que esta depende das relações entre as seqüências e sub-seqüências dos eventos, enquanto que a mineração de dados até então discutida considera os dados como coleções desordenadas de eventos ignorando o seu aspecto temporal.

De acordo com Lucas (2002), entende-se como fases do pré-processamento para a mineração de dados temporais as seguintes: seleção, limpeza, ordenação e transformação de dados.

Em termos de seleção de dados para escolher o conjunto de dados temporais mais significativo é preciso determinar três tipos de atributos: o principal (aquele que se apresenta em várias ocorrências ao longo do tempo), o temporal (refere-se às diversas ocorrências temporais do atributo principal) e o elementar (consiste nos elementos contidos em cada ocorrência temporal do atributo principal).

Assim na base de dados de AIHs foram selecionadas as tabelas de movimentos das AIHs mensais sendo estas combinadas formando a tabela correspondente ao movimento de AIHs do ano de 2000, com 574.352 instâncias. Em seguida, foram selecionados inicialmente os seguintes atributos: CPF do paciente (principal), a data de internação do paciente (temporal), o procedimento realizado e o diagnóstico principal (elementares).

Como foi verificado que CPF do paciente apresentava valores ausentes em várias instâncias ou mais de um paciente apresentava o mesmo CPF a solução seria eliminar as instâncias com estes problemas, o que representaria uma grande perda de informação. Deste modo, verificou-se que o atributo principal mais apropriado seria o nome do paciente concatenado com a data de nascimento do mesmo para evitar problemas de homonímia.

Em termos de limpeza de dados resolveu-se eliminar da tabela de movimento de AIHs as instâncias correspondentes às AIHs bloqueadas na tabela de controle de

AIHs através da comparação dos valores de atributos comuns as duas tabelas que são os relacionados ao número de AIH, CGC do hospital e procedimento realizado.

Após a equivalência foram eliminadas 5.519 instâncias da tabela de movimento de AIHs consideradas bloqueadas totalizando a atual tabela de movimento de AIHs com 568.833 instâncias.

Quanto à fase de ordenação, na metodologia de mineração de dados temporais (MDT) apresentada em Lucas (2002), é necessário agrupar as instâncias na tabela através de uma ordenação da mesma pelo atributo principal seguido do atributo temporal. Assim, a tabela de movimento de AIHs foi ordenada pelos nome do paciente concatenado com a data de nascimento seguido da data de internação.

Finalmente, na fase de transformação os dados ordenados foram transformados em seqüências temporais. Tendo sido utilizado para isto a ferramenta *Intelligent Miner* da IBM por disponibilizar de forma totalmente automatizada a realização desta fase.

4.4 Considerações finais sobre os experimentos

A partir dos experimentos realizados constatou-se que em termos de pré-processamento para qualquer aplicação a fase de entendimento dos dados é o primeiro passo a ser feito. Quanto à fase de seleção de dados observou-se que inicialmente é feita a seleção das tabelas relevantes ao problema.

Posteriormente, devem ser efetuadas as operações de limpeza necessárias, considerando-se que estas precisam ocorrer antes de qualquer integração de tabelas e da realização de operações de transformação de dados.

Caso seja necessário integrar tabelas, considera-se a possibilidade de serem realizadas novamente operações de limpeza sobre o conjunto de dados já integrado, para eliminar, por exemplo, duplicatas que só seriam visualizadas após a integração.

As operações de transformação de dados podem ser realizadas no conjunto de dados integrado como sendo parte do tratamento geral sobre este conjunto de dados que possa ser aproveitado para qualquer objetivo. Após o tratamento geral realizado sobre o conjunto de dados integrado podem ser realizadas as seleções de instâncias e de atributos conforme o objetivo a ser alcançado.

Destaca-se que novas operações de transformação de dados como discretizações ou composições de atributos podem ser necessárias para objetivos específicos de mineração, eventualmente seguida de nova seleção de atributos e mesmo da re-execução de tarefas da fase de entendimento dos dados, como por exemplo um levantamento estatístico sobre atributos discretizados ou sobre novos atributos criados.

Considera-se então que o tratamento geral realizado sobre o conjunto integrado é importante na medida em que disponibiliza um conjunto de dados preparado que posteriormente poderá ser submetido a seleção de instâncias e de atributos para qualquer objetivo, reduzindo assim o tempo gasto em operações de limpeza e transformação na experimentação de novas minerações sobre o mesmo conjunto de dados.

Além disso, conforme o que é comentado nas referências bibliográficas e de fato comprovado nos experimentos deste trabalho, ressalta-se que a interatividade com o especialista do domínio é fundamental na realização do processo de DCBD, principalmente para auxiliar no entendimento dos dados e na tomada de decisões sobre o quê e de que forma pré-processar. O caráter iterativo do processo de DCBD também é naturalmente vivenciado e deve ser explorado no sentido de se buscar algo novo e interessante, tendo-se que muitas vezes retornar a fase de pré-processamento, repetindo ou efetuando de uma maneira diferente algumas operações da mesma, mesmo depois de já terem sido realizadas algumas minerações sobre o conjunto de dados manipulado.

5 Conclusões

As metodologias para o processo de descoberta de conhecimento em banco de dados (DCBD) envolvem basicamente as fases de entendimento do domínio do problema, pré-processamento, mineração de dados e pós-processamento. Sendo que a maior parte do esforço e tempo consumido durante o processo concentra-se na fase de pré-processamento, o que foi claramente vivenciado durante os experimentos realizados sobre a base de dados real das AIHs da SES do RS.

Entretanto, pouco detalhamento sobre o processo de pré-processamento é descrito nas metodologias para DCBD, assim como também trabalhos específicos sobre o assunto são menos encontrados que sobre técnicas de mineração de dados, por exemplo.

Assim, a partir das pesquisas realizadas sobre pré-processamento foram encontrados alguns métodos e técnicas para realizar as operações de entendimento, seleção, limpeza e transformação de dados. Porém, existem também outras formas de se realizar estas mesmas operações que são utilizar o conhecimento do especialista do domínio e juntamente com alguns recursos de determinadas ferramentas tais como Excel, Access e Appenda, além de se poder implementar algum programa que personalize as necessidades do especialista.

Devido a acessibilidade aos especialistas envolvidos no problema e ao fato de ter sido utilizada para os experimentos uma base de dados que apresentava muitas inconsistências tornou-se necessário explorar o conhecimento destes especialistas durante todo o processo de descoberta, mas principalmente na fase de entendimento do domínio do problema e na de pré-processamento.

Enfatizando-se a fase de pré-processamento, comprovou-se que esta depende profundamente do objetivo a ser alcançado e que não existe uma ordem rígida quanto à execução de suas operações de entendimento, seleção, limpeza e transformação de dados. O pré-processamento realizado nos experimentos 1 e 2 constituiu-se no seguinte:

a) Entendimento dos Dados

Foram utilizados alguns recursos dos *sofwares* Excel e Access para realizar o levantamento sobre o número de atributos e instâncias das tabelas envolvidas, além de serem verificados os tipos de dados dos atributos com seus respectivos formatos e tamanho. Além disso, algumas estatísticas básicas foram efetuadas sobre os dados em questão encontrando-se algumas distorções.

b) Seleção de Tabelas

Trabalhando-se com a base de dados de AIHs de 2000, foram selecionadas as tabelas de movimento de AIHs, de valores de AIHs, de movimento dos hospitais, de *bureaux,* de leitos de hospitais e a de classificação internacional de doenças. Houve a necessidade de integrar as tabelas primeiramente de estruturas diferentes (por exemplo, a de movimento de AIHs e a de valores de AIHs, mês-a-mês) através de consultas SQL do Access e depois as de mesma estrutura através do programa Appenda, formando a base de dados do ano todo.

c) Seleção de Atributos

Após várias entrevistas realizadas com os especialistas, estes auxiliaram na seleção de atributos relevantes das tabelas selecionadas sendo 18 atributos da tabela de movimento de AIHs, 11 atributos da tabela de valores de AIHs, 4 da tabela de movimento dos hospitais, 2 atributos da tabela de *bureaux*, 5 atributos da tabela do número de leitos do hospital e 3 atributos da tabela de classificação internacional de doenças.

Após a realização de outras operações de pré-processamento foi definido para a tabela a ser submetida à mineração de dados, o subconjunto de 12 atributos relevantes sendo todos do tipo simbólico ou discreto.

Para a seleção de atributos seguiu-se a orientação dos especialistas, excluindo diretamente os atributos irrelevantes na estrutura da tabela pelo Access, sempre tendo o cuidado de manter uma cópia da tabela original caso fossem necessários novos pré-processamentos.

d) Seleção de Instâncias

A partir das recomendações dos especialistas foram selecionadas somente as instâncias de AIHs que apresentavam como procedimento realizado um dos seguintes: parto normal, doença pulmonar obstrutiva crônica e insuficiência cardíaca. Considerados os três primeiros procedimentos de maior freqüência dentre as AIHs de 2000.

Depois foram selecionadas as AIHs de cada um destes três procedimentos realizados em separado formando-se assim três novas tabelas sendo uma para cada procedimento.

Destaca-se que todas estas seleções foram realizadas através de consultas SQL do Access.

e) Eliminação de Dados Errôneos

As instâncias que apresentavam valores errôneos no ano da data de nascimento de um paciente foram excluídas considerando-se que eram apenas três instâncias dentro de 574.307.

Alguns atributos tais como o CPF do paciente foi eliminado pois para algumas instâncias este atributo apresentava valores zerados, em outras se tratava do mesmo paciente, porém com CPF diferente, e em outras tratava-se de pacientes diferentes apresentando o mesmo CPF.

f) Padronização de Dados

Alguns problemas foram encontrados na idade dos pacientes, devido a erros na digitação do ano na data de nascimento, que precisaram ser corrigidos para depois serem recalculadas manualmente as idades dos pacientes considerando o ano de 2000, tendo sido feitas as atualizações na tabela de AIHs através de consulta SQL no Access.

Na tabela de *bureaux* foram encontrados alguns CGCs com formato diferente também sendo corrigidos.

g) Eliminação de Duplicatas

Como se dispunha de recursos para detecção e eliminação de duplicatas no Access então estes foram utilizados, não sendo necessário aplicar nenhum dos algoritmos apresentados sobre isto.

Para a detecção de duplicatas foi utilizado o recurso do assistente de consulta localizar duplicatas disponível no Access e para a eliminação das mesmas primeiramente as instâncias foram classificadas por uma chave primária composta pelos atributos que poderiam conter valores duplicados, em seguida foi criada uma nova tabela a partir da cópia da estrutura daquela que apresentava duplicatas, definindo-se uma chave primária. Finalmente, através de uma consulta acréscimo do Access foram incluídas apenas instâncias exclusivas com todos os atributos da tabela original.

Assim puderam ser eliminadas as duplicatas do mês de janeiro e as AIHs reapresentadas encontradas na tabela que continha apenas os três procedimentos de maior freqüência.

h) Tratamento de Valores Ausentes

Vários atributos das tabelas selecionadas apresentavam todos ou a grande maioria de seus valores ausentes, em alguns casos por não existir informação e em outros porque o atributo não tinha mais utilidade, porém ainda fazia parte da estrutura da tabela. Então, estes atributos com valores ausentes foram excluídos por terem sido considerados irrelevantes pelos especialistas do domínio.

Após a integração das tabelas de *bureaux* com a de movimento de hospitais apareceram para algumas instâncias valor ausente para o atributo biro pelo fato de que nem todos os hospitais estão associados a um *bureau*, sendo assim para estas instâncias o atributo teve o valor preenchido com “NENHUM” para que não continuasse com valores ausentes. Esta atualização foi feita através de uma consulta SQL.

h) Discretização de Atributos

Para a discretização foi de total necessidade a definição de valores e de intervalos feita pelos especialistas do domínio, não sendo preciso utilizar os algoritmos listados para isto.

Assim os atributos de categoria do motivo de cobrança, categoria (faixa etária), porte do hospital e custo foram discretizados a partir dos critérios considerados pelos especialistas do domínio. Já o atributo categoria da classificação internacional de doenças teve seus intervalos de valores definidos a partir de consultas feitas a tabela de classificação internacional de doenças, a qual apresentava as categorias e seus respectivos códigos de doenças, mesmo assim foi preciso ainda o auxílio do especialista do domínio para combinar determinadas categorias.

Para serem efetuadas as discretizações utilizou-se consultas de atualização SQL para os atributos porte do hospital e custo. Quanto à discretização dos demais atributos foi preciso criar um programa para realizar a mesma, pois a quantidade de instâncias era de 574.304 e a variedade de valores daqueles atributos era muito grande tornando as consultas SQL muito lentas.

i) Composição de Atributos

Tornou-se necessário construir novos atributos referentes ao valor total da AIH e ao número de dias de internação do paciente. Após ter sido confirmado pelos especialistas os atributos originais que seriam utilizados para a composição destes novos atributos, foram realizadas consultas de atualização SQL do Access considerando que para o valor total da AIH foi preciso somar os atributos referentes a valores e para o número de dias de internação foi preciso calcular a diferença entre a data de alta do paciente e a data de internação do mesmo.

Ressalta-se que as operações de normalização de dados e de conversões de valores simbólicos para numéricos não foram necessárias para os experimentos.

Um pré-processamento especial foi realizado considerando as exigências da ferramenta de mineração utilizada, o See5, tendo sido considerado como atributo de classe no experimento 1 o atributo custo e no experimento 2 o biro.

Já no experimento 3 as fases de pré-processamento consideradas são seleção, limpeza, ordenação e transformação de dados. Na seleção foram escolhidos os atributos principal, o temporal e o elementar na mesma base de dados de AIHs. A limpeza caracterizou-se pela eliminação de instâncias que correspondiam às AIHs bloqueadas, a ordenação consiste no agrupamento das instâncias considerando o atributo principal seguido do atributo temporal. E a transformação de dados envolve a conversão dos dados ordenados em seqüências temporais.

5.1 Dificuldades Encontradas

A principal dificuldade para o entendimento do domínio do problema e posterior pré-processamento foi a inconsistência dos dados.

A definição dos objetivos para a mineração de dados assim como também a determinação dos atributos relevantes para a mesma também foi uma tarefa muito difícil e demorada para ser realizada devido às diferentes opiniões dos especialistas envolvidos para definir as necessidades deles mesmos. E sem uma definição clara do objetivo não se tem idéia do pré-processamento que deverá ser realizado, o que conseqüentemente leva a uma considerável perda de tempo.

Algumas dúvidas referentes a significados dos atributos das tabelas utilizadas não foram esclarecidas pelos especialistas.

Alguns resultados encontrados considerados fora da normalidade não puderam ter um retorno imediato, visto que era preciso auditar diretamente nos hospitais através de levantamento dos prontuários para conferência dos dados.

O tempo consumido na realização dos experimentos foi muito grande por envolver várias visitas aos especialistas para esclarecimento de dúvidas, além do fato que a manipulação de uma base de dados real muito grande também tornou lenta a efetivação de parte do pré-processamento.

5.2 Contribuições

A principal contribuição deste trabalho é apresentar o que existe em termos de métodos e técnicas de pré-processamento considerando-se todas as fases do mesmo e mostrar através dos experimentos realizados que a interatividade com o especialista do domínio é de fundamental importância para qualquer tomada de decisão no que deve ser feito como pré-processamento, já que o mesmo varia conforme o objetivo que se quer alcançar.

Além disso, comprovou-se que de fato o pré-processamento é a fase que requer a maior parte do tempo de todo o processo de descoberta de conhecimento em banco de dados.

Entretanto, observou-se que uma continuação do trabalho de mineração com o mesmo conjunto de dados tenderia a uma diminuição percentual do esforço dedicado ao pré-processamento, uma vez que parte do mesmo, como o entendimento dos dados, por exemplo, não precisaria ser refeito, bem como algumas das discretizações, mesmo com a utilização de outras técnicas de modelagem.

E para o projeto desenvolvido com a SES do RS foi possível mostrar alguns resultados merecedores de investigações mais profundas sobre o que realmente ocorreu em determinadas AIHs, assim como também foi despertado o interesse por análises de outros elementos relacionados as AIHs tais como os *bureaux*, o porte do hospital que as emitiu, a freqüência de internações e alta de pacientes por dia da semana e o custo da AIH.

5.3 Trabalhos Futuros

Os dois experimentos realizados utilizaram a técnica de geração de regras através do *software* See5. O pré-processamento necessário para a aplicação de outras técnicas, como por exemplo, as de *clustering,* seria uma primeira extensão ao trabalho realizado.

Outro trabalho futuro seria a aplicação de uma ou várias combinações de técnicas e métodos automáticos de pré-processamento, apresentados no capítulo 3, para verificar os resultados obtidos para os mesmos objetivos descritos nos experimentos.

E, no contexto do projeto com a SES do RS e a partir do estudo realizado sobre pré-processamento apresentado neste trabalho pretende-se implementar uma ferramenta de pré-processamento que reúna todas as operações necessárias para o mesmo no que diz respeito ao entendimento, à seleção, à limpeza e à transformação de dados. Sendo que, na operação de discretização a ferramenta possibilite ao usuário definir os valores e os intervalos conforme as requisições do seu problema. Esta ferramenta também deverá conter um módulo para a detecção de homonímias, considerando a fonética da língua portuguesa, de modo que, nomes como Luiz e Luis, por exemplo, sejam tratados como homônimos.

Anexo 1

Métodos de Seleção de Atributos

Neste anexo são apresentados os métodos de seleção de atributos do tipo filtro e *wrapper*, assim como também os de uma seleção híbrida, considerando as abordagens supervisionada e não-supervisionada dos mesmos. A seguir são descritos os métodos mais conhecidos, conforme Baranauskas (2001), Dash; Liu (1997), Riaño (1997) e Liu; Yu (2002).

1) Métodos de seleção de atributos do tipo filtro com abordagem supervisionada

FOCUS

Este algoritmo utiliza a estratégia de busca completa e como critério de avaliação medidas de consistência.

O algoritmo FOCUS de Almuallim; Dietterich (1991) foi originalmente desenvolvido para domínios booleanos sem a presença de dados errôneos.

Segundo Dash; Liu (1997), o método FOCUS pode manipular dados simbólicos, além de múltiplas classes e de gerar um subconjunto ótimo de atributos. Porém, não é apto para trabalhar com um conjunto de dados muito grande, já que este algoritmo examina cada atributo individualmente, depois pares de atributos, triplas e assim por diante, ou seja, ele examina exaustivamente todos os subconjuntos de atributos, selecionando o subconjunto mínimo de atributos que seja suficiente para determinar a classe das instâncias.

Após encontrar um subconjunto mínimo de atributos relevantes, as instâncias de treinamento são filtradas para remover todos os atributos irrelevantes. Em seguida, as instâncias filtradas são então fornecidas a um algoritmo de aprendizagem, como o ID3 para construir uma árvore de decisão (ALMUALLIM; DIETTERICH, 1991).

A complexidade de tempo no pior caso de FOCUS é de $O\ (npm^{p})$, onde n é o número de instâncias, p é o número de atributos relevantes ($p \leq m$), e m é o número de atributos. Assim, este método não é apropriado quando o número de atributos m aumenta visto que a quantidade de memória necessária é exponencial ao número de atributos (RIAÑO, 1997).

Método de Schlimmer

Este algoritmo é considerado uma variação de FOCUS e também adota a estratégia de busca completa e como critério de avaliação medidas de consistência.

O método de Schlimmer utiliza um esquema de busca sistemática como procedimento de geração e o critério de inconsistência como a função de avaliação. A busca pelo subconjunto ótimo é do tipo em largura ou amplitude (*breadth-first*) sendo bastante rápida por ser utilizada uma função heurística. Esta função heurística é uma medida de confiança baseada na intuição de que a probabilidade que uma inconsistência será observada é proporcional a percentagem de valores que tenham sido menos

observados considerando o subconjunto de atributos. Considera-se ainda que todos os superconjuntos de um subconjunto não confiável são também não confiáveis (DASH;LIU, 1997).

Conforme Dash; Liu (1997), o algoritmo de Schlimmer pode manipular dados simbólicos, além de múltiplas classes e gerar um subconjunto ótimo se certas considerações são válidas, mas não é apropriado para trabalhar com um conjunto de dados muito grande nem com dados errôneos.

MIFES-1

Do mesmo modo que o método de Schlimmer, o algoritmo MIFES-1 também é considerado uma variação de FOCUS, utilizando a estratégia de busca completa e como critério de avaliação medidas de consistência.

Este método é semelhante ao FOCUS no seu processo de seleção de atributos. Ele representa o conjunto de instâncias na forma de uma matriz com classes binárias e atributos simbólicos com valores booleanos, cada elemento desta corresponde a uma combinação única de uma instância positiva (classe = 1) e uma instância negativa (classe = 0). Sendo que, um atributo "cobre" um elemento da matriz se ele assume valores opostos para uma instância positiva e uma instância negativa associada ao elemento. Este algoritmo busca por um conjunto de atributo de cobertura com $N - 1$ atributos iniciando a partir de um conjunto com todos os N atributos, e realiza iterações até nenhuma redução no tamanho do conjunto de atributos de cobertura ser atingido (DASH;LIU, 1997). Para tanto, segundo Oliveira; Vincentelli (1992) é utilizada uma árvore de busca.

De acordo com Dash; Liu (1997), o algoritmo MIFES-1 pode manipular múltiplas classes e gerar um subconjunto ótimo de atributos, porém não é apropriado para conjuntos de dados grandes nem com a presença de valores errôneos.

RELIEF

Este método utiliza a estratégia de busca heurística e como critério de avaliação as medidas de distância.

O algoritmo RELIEF de Kira; Rendell (apud KONONENKO, 1994, p. 02) foi originalmente criado para tratar com atributos simbólicos e contínuos, domínios booleanos e problemas de classificação binária, além de poder manipular dados errôneos e atributos correlacionados. Complementarmente, considera-se que este método é apropriado para manipular um conjunto de dados muito grande mas não gera um subconjunto ótimo de atributos (DASH; LIU, 1997). Tendo sido realizado o experimento do RELIEF aplicando em seguida um classificador *naive-bayes*[9] (KONONENKO, 1994).

O método RELIEF é baseado no peso de relevância de atributos inspirado por algoritmos de aprendizagem baseada em instância. Assim, a partir de um conjunto de instâncias de treinamento é escolhida uma amostra de instâncias sendo o número de instâncias desta definido pelo usuário. O RELIEF obtém aleatoriamente esta amostra de

[9] Segundo Han (2001), o classificador *naive-bayes* assume que o efeito do valor de um atributo em uma determinada classe é independente dos valores de outros atributos. Este classificador pode ser comparado em desempenho com classificadores de árvore de decisão e de redes neurais, além de assumir alta acurácia e velocidade quando aplicado a grandes conjuntos de dados.

instâncias, e para cada uma ele encontra instâncias mais próximas da instância em questão, sendo uma da mesma classe e outra da classe oposta, o que é conseguido utilizando uma medida de distância Euclidiana.

A partir disto, o algoritmo RELIEF atualiza os pesos de relevância dos atributos que são inicializados com zero considerando que um atributo é mais relevante se o mesmo distingue entre uma determinada instância e a sua mais próxima pertencente à classe oposta, e menos relevante se este distingue entre uma instância particular e a sua mais próxima da mesma classe. Após realizar este processo para todas as instâncias da amostra, o algoritmo escolhe todos os atributos que tem um peso de relevância maior ou igual ao de um determinado limiar (*threshold*). Este limiar pode ser automaticamente avaliado através de uma função que utiliza o número de instâncias da amostra, assim como também pode ser determinado por inspeção (todos os atributos com pesos de relevância positivo são selecionados).

Segundo Riaño (1997), a complexidade de tempo do algoritmo RELIEF é de *O (kmn)*, onde *k* é o número de iterações, *m* é o número de atributos e *n* é o número de instâncias, tendo-se assim um tempo linear considerando o número de atributos e o número de instâncias da amostra.

A maior limitação de RELIEF está na manipulação de atributos redundantes, pois com a presença destes o algoritmo gera um conjunto de atributos cujo tamanho não é considerado ótimo. Porém, isto pode ser resolvido por uma subseqüente busca exaustiva sobre os subconjuntos de todos os atributos selecionados por RELIEF (DASH; LIU, 1997).

Quanto a limitação do RELIEF em trabalhar apenas com duas classes, esta é resolvida a partir de uma extensão do algoritmo chamada RELIEF-F que além disso também manipula valores ausentes de atributos.

Método *B & B (Branch and Bound)*

Este método utiliza a estratégia de busca completa e como critério de avaliação as medidas de distância.

Complementarmente, Dash; Liu (1997) mostram que o método *B & B* pode manipular dados contínuos e simbólicos, múltiplas classes e gera um subconjunto ótimo de atributos se certas considerações são válidas.

Segundo Narendra; Fukunaga (apud DASH; LIU, 1997, p. 10), o algoritmo *B & B*, tenta satisfazer dois critérios: primeiro, que o subconjunto de atributos selecionado seja tão pequeno quanto possível; segundo, que um limite seja colocado no valor do critério de avaliação. Assim, *B & B* inicia a busca a partir do conjunto original de atributos e atua removendo atributos do mesmo. E, devido a utilização de um valor limite no critério de avaliação esta busca torna-se bastante rápida.

Ressalta-se que o critério de avaliação neste método exige o princípio de monotonicidade, desta forma qualquer subconjunto de atributos para o qual o valor é menor que o limite deve ser removido a partir do espaço de busca. Geralmente, os critérios de avaliação mais utilizados são: distância de *Mahalanobis,* a função discriminante, o critério *Fisher*, a distância *Bhattacharya*, e a divergência.

Método BFF

Da mesma forma que o método *B & B*, este método também utiliza a estratégia de busca completa e como critério de avaliação as medidas de distância.

Segundo Dash; Liu (1997), o método BFF manipula dados contínuos e simbólicos, problemas com múltiplas classes e gera um subconjunto ótimo de atributos.

O método BFF é proposto por Xu; Yan; Chang (apud DASH; LIU, 1997, p. 11), e apresenta uma estratégia de busca modificada para resolver o problema da busca por um caminho ótimo em uma árvore *weighted* através da estratégia de busca *best-first* conhecida na área de inteligência artificial.

O algoritmo BFF garante o melhor subconjunto globalmente sem busca exaustiva, para qualquer critério que satisfaça o princípio de monotonicidade.

Método de Bobrowski

Este método também utiliza a estratégia de busca completa e como critério de avaliação as medidas de distância.

Conforme Dash; Liu (1997), o método de Bobrowski pode manipular dados contínuos e simbólicos, problemas com múltiplas classes e gera um subconjunto ótimo de atributos se certas considerações forem válidas.

O método de Bobrowski (apud DASH; LIU, 1997, p. 11) permite que o coeficiente de homogeneidade possa ser utilizado na medida do grau de dependência linear entre algumas medidas, e também adota o princípio de monotonicidade. Entretanto, este método também pode ser apropriadamente convertido a um algoritmo de seleção de atributos por implementá-lo como um critério de avaliação para *B & B* com regressão (*backtracking*) ou como um procedimento de geração *best-first*.

SFG (*Sequential Forward Generation*)

Este método utiliza a estratégia de busca heurística e como critério de avaliação as medidas de informação.

O algoritmo SFG inicia com um conjunto vazio *S*, e adiciona atributos seqüencialmente a partir do conjunto original *F*. Sendo que, em cada etapa de seleção, o melhor atributo é escolhido de acordo com algum critério de informação. O atributo escolhido é então adicionado dentro de *S* e removido de *F*. Assim, *S* aumenta e *F* é reduzido. O SFG pode gerar uma lista ordenada de atributos de modo que o primeiro atributo escolhido seja o mais relevante e o último atributo escolhido seja o menos relevante.

DTM (*Decision Tree Method*)

Semelhante ao SFG, o DTM também utiliza a estratégia de busca heurística e como critério de avaliação as medidas de informação.

De acordo com Dash; Liu (1997), o método DTM pode manipular dados contínuos e simbólicos, problemas com múltiplas classes e conjuntos de dados grandes, porém não gera um subconjunto ótimo de atributos. E, pode melhorar bastante a aprendizagem baseada em instâncias.

Igualmente ao SFG, o método DTM gera um subconjunto de atributos, porém utiliza o critério de entropia para avaliar cada subconjunto selecionado. Neste algoritmo, o C4.5 é executado sobre o conjunto de treinamento, e os atributos que finalmente aparecem na árvore de decisão podada são selecionados como o melhor subconjunto, ou seja, a união dos subconjuntos de atributos que aparecem no caminho para qualquer nó folha na árvore podada consiste no melhor subconjunto selecionado.

Método de Koller; Sahami

Assim como o SFG e o DTM, o método de Koller; Sahami (apud DASH; LIU, 1997, p. 12) também utiliza a estratégia de busca heurística e como critério de avaliação as medidas de informação.

Segundo Dash; Liu (1997), o método de Koller; Sahami pode manipular dados simbólicos, problemas com múltiplas classes e um conjunto de dados grande, porém não gera um subconjunto ótimo de atributos. E, o efeito da seleção de atributos por este método pode ser verificado através da realização de uma classificação utilizando um classificador *naive-bayes* e o C4.5 como algoritmos de indução (KOLLER; SAHAMI, 1996).

Este método considera que um atributo seja irrelevante ou redundante se apresentar pouca ou nenhuma informação adicional além do que é conhecido pelos demais atributos, devendo assim ser eliminado. Isto é conseguido através de uma aproximação do *blanket* de *Markov* , isto é, um subconjunto de atributos *T* é um *blanket* de *Markov* para um atributo f_i se, dado *T*, f_i é condicionalmente independente do rótulo de classe e de todos os atributos que não estão em *T* (incluindo o próprio f_i).

MDLM (*Minimum Description Length Method*)

O algoritmo MDLM utiliza a estratégia de busca completa e como critério de avaliação as medidas de informação.

De acordo com Dash; Liu (1997), o algoritmo MDLM pode manipular dados contínuos e simbólicos e problemas com múltiplas classes, mas não gera um subconjunto ótimo de atributos.

Este método tenta eliminar os atributos irrelevantes e/ ou redundantes considerando o seguinte: caso os atributos em um subconjunto *V* possam ser expressos como uma função *F* dos atributos independentes de classe em outro subconjunto *U*, então uma vez que os valores dos atributos no subconjunto *U* sejam conhecidos o subconjunto de atributos *V* torna-se desnecessário.

O algoritmo em questão realiza uma busca exaustiva sobre todos os possíveis subconjuntos (2^N) e gera o subconjunto satisfazendo o MDLC (*Minimum Description Length Criterion*).

Método POE + ACC (*Probability of Error & Average Correlation Coefficient*)

Este método utiliza a estratégia de busca heurística e como critério de avaliação as medidas de dependência.

Segundo Dash; Liu (1997), o método POE + ACC pode manipular dados contínuos e simbólicos e problemas com múltiplas classes, mas não gera um subconjunto ótimo de atributos.

No POE + ACC, o primeiro atributo selecionado é aquele com menor probabilidade de erro (P_e). O próximo atributo selecionado é o atributo que produz a soma mínima de pesos de P_e e coeficiente médio de correlação (ACC), e assim por diante.

O ACC corresponde a média dos coeficientes de correlação do atributo candidato com os atributos previamente selecionados neste ponto.

Este método pode ordenar todos os atributos a partir da soma de peso de importância, e um número requerido de atributos (M) é utilizado como o critério de parada.

PRESET

Assim como o POE + ACC, o método PRESET utiliza a estratégia de busca heurística e como critério de avaliação as medidas de dependência.

Conforme Dash; Liu (1997), o método PRESET pode manipular dados contínuos e simbólicos, problemas com múltiplas classes e um conjunto de dados grande, porém não gera um subconjunto ótimo de atributos.

Este método utiliza o conceito de *rough set* , ou seja, ele primeiro encontra um subconjunto de um conjunto *S* e remove todos os atributos que não aparecem no subconjunto. Então, os atributos são ordenados baseados em sua significância. Considerando-se que, a significância de um atributo é uma medida que expressa quão importante um atributo é em relação a classificação. Sendo esta medida baseada na dependência de atributos.

LVF (*Las Vegas Filter*)

O método LVF utiliza a estratégia de busca não-determinística ou aleatória e como critério de avaliação medidas de consistência.

De acordo com Dash; Liu (1997), o algoritmo LVF pode manipular dados simbólicos, problemas com múltiplas classes, dados errôneos, um conjunto de dados grande e gera um subconjunto ótimo de atributos. E, para verificar a eficiência deste método na seleção de atributos, os dados com o subconjunto de atributos selecionado pode ser submetido em seguida a algoritmos de indução de árvores de decisão como C4.5 e ID3 (LIU; SETIONO, 1996).

Este método inicia com um subconjunto selecionado aleatoriamente e realiza uma busca aleatória sobre o espaço de subconjuntos utilizando um algoritmo *Las Vega*s que faz escolhas probabilísticas para ajudar a obter mais rapidamente uma solução ótima e utiliza uma medida de consistência que tenta encontrar um número mínimo de

atributos capaz de diferenciar as classes tão consistentemente quanto o conjunto completo de atributos o faz.

Conforme Liu; Yu (2002), uma medida de inconsistência é definida como duas instâncias tendo os mesmos valores de atributos, mas pertencendo a classes diferentes.

Para cada subconjunto candidato é calculado um valor de inconsistência baseado na consideração que o rótulo de classe mais frequente entre estas instâncias tomando este subconjunto de atributos consiste no rótulo de classe mais provável. Assim um determinado limiar (*threshold*) é fixado no início (geralmente 0), e qualquer subconjunto tendo uma taxa de inconsistência maior que esta, é rejeitado. Além desta medida de consistência, o algoritmo LVF utiliza outro parâmetro predefinido como um critério de parada, que é o número máximo de subconjuntos aleatoriamente gerados.

Diante da manipulação de um conjunto de dados muito grande o LVF pode consumir muito tempo para a verificação da consistência do mesmo, o que pode ser resolvido por uma versão incremental deste algoritmo chamada LVI (*Las Vegas Incremental*), que reduz bastante o número de inconsistência conseguido pela redução do dado utilizado para verificação de inconsistência, ao mesmo tempo em que mantém a qualidade dos subconjuntos de atributos gerados.

Já uma outra variação de LVF, chamada QBB, resolve o problema que LVF tem de ser rápido somente no início do processo de seleção. O algoritmo QBB combina LVF e ABB, de modo que no início da busca o LVF pode remover eficientemente alguns atributos irrelevantes e quando o ABB atua sobre o novo conjunto de dados, o número de atributos é menor que o do começo da busca. Entretanto, QBB não pode garantir que o subconjunto final selecionado seja ótimo devido ao fato que a saída de LVF não necessariamente é o superconjunto do subconjunto ótimo.

2) Métodos de seleção de atributos do tipo filtro com abordagem não-supervisionada

SBUD (*Sequential Backward for Unsupervised Data*)

Este método utiliza a estratégia de busca heurística e as medidas de informação como critério de avaliação.

O algoritmo SBUD adota a medida de entropia como critério de informação para ordenar os atributos e a estratégia de busca seqüencial de retorno (*backward*) é usada para gerar diferentes subconjuntos de atributos a partir do conjunto completo. A medida de entropia ordena os atributos de acordo com a importância destes em relação ao conceito que representam ou a tarefa de *clustering*. O valor de entropia de todas as instâncias do conjunto de dados varia dentro de uma faixa de zero a um. Como esta medida está relacionada com a homogeneidade da distribuição das instâncias, tem-se que, se as instâncias são mais uniformemente distribuídas, o valor de entropia é mais próximo de 1 (um). Senão, o valor é mais próximo de 0 (zero).

A remoção de atributos do conjunto de dados torna a distribuição mais uniforme, assim o valor de entropia aumenta, sendo que o aumento é menor se for removido um atributo menos importante que um mais importante.

O método SBUD é considerado apropriado para encontrar atributos importantes sem a informação de classe. E pára quando o número pré-definido de atributos relevantes é alcançado.

Método de Mitra

Este método utiliza a estratégia de busca heurística e as medidas de distância como critério de avaliação.

Considerando-se que as instâncias dentro de um *cluster* são similares umas às outras e não-similares às instâncias de outros *clusters,* tem-se que a similaridade refere-se a quão próximas as instâncias estão no espaço, baseado em uma função de distância (HAN; KAMBER, 2001).

Através da busca heurística este algoritmo pode gerar subconjuntos de atributos, e o critério de similaridade de atributos utilizado é o índice de máxima compressão de informação. Esta nova medida permite que o algoritmo seja rápido com complexidade de tempo de $O(M^2N)$, onde M é o número de instâncias e N é o número de atributos no conjunto de dados. Isto torna o algoritmo de Mitra apropriado para manipular conjuntos de dados com dimensionalidade muito alta, além de trabalhar bem na remoção de atributos redundantes.

3) Métodos de seleção de atributos do tipo *wrapper* com abordagem supervisionada

Métodos de Ichino e Sklansky

Os métodos de Ichino; Sklansky (apud DASH; LIU, 1997, p. 17) utilizam a estratégia de busca completa e como critério de avaliação uma medida para classificação (a taxa de erro do classificador). Sendo a seleção de atributos resolvida através de programação de inteiros zero-um.

AMB & B (*Approximate Monotonic Branch and Bound*)

Assim como os métodos de Ichino e Sklansky, o algoritmo AMB & B utiliza a estratégia de busca completa e como critério de avaliação uma medida para classificação (a taxa de erro do classificador).

Este método foi criado para resolver a desvantagem de B & B por permitir critérios de avaliação que não são monotônicos. Neste método o limite de B & B não é seguido rigorosamente para gerar subconjuntos que apareçam sob alguns subconjuntos violando o limite; mas o subconjunto selecionado não deve violar o limite.

BS (*Beam Search*)

O algoritmo BS também utiliza a estratégia de busca completa e como critério de avaliação uma medida para classificação (a taxa de erro do classificador).

Este método é um tipo de busca *best-first* que utiliza uma fila limitada para restringir o escopo da busca. A fila é ordenada do melhor para o pior com os melhores subconjuntos no início da fila. O procedimento de geração toma o subconjunto do começo da fila, e produz todos os subconjuntos possíveis através da adição de um atributo a cada um. Cada subconjunto é colocado em sua posição apropriada de

classificação na fila. Caso não exista limite no tamanho da fila, o BS é uma busca exaustiva, por outro lado, se o limite do tamanho da fila é um, o BS equivale a WSFS.

WSFG (*Wrapper Sequential Forward Generation*)

O algoritmo WSFG utiliza a estratégia de busca heurística e como critério de avaliação uma medida para classificação (a taxa de erro do classificador).

Este método inicia a partir de um conjunto vazio de atributos, e em cada iteração gera novos subconjuntos através da inclusão de um atributo selecionado por algum critério de avaliação.

WSBG (*Wrapper Sequential Backward Generation*)

Assim como o algoritmo WSFG, o método WSBG também utiliza a estratégia de busca heurística e como critério de avaliação uma medida para classificação (a taxa de erro do classificador).

Este método inicia a partir de um conjunto de atributos completo, e em cada iteração gera novos subconjuntos por remover um atributo selecionado por algum critério de avaliação.

SBS-SLASH (*Sequential Backward Selection-SLASH*)

O algoritmo SBS-SLASH utiliza a estratégia de busca heurística e como critério de avaliação uma medida para classificação (a taxa de erro do classificador).

Este método é baseado na observação que quando existe um grande número de atributos, alguns classificadores tais como ID3 e C4.5, freqüentemente não utilizam muitos destes atributos. Assim, o algoritmo SBS-SLASH inicia com o conjunto de atributos completo, mas após cada passo, elimina qualquer atributo não utilizado no que foi aprendido em cada passo.

BDS (*Bi-Directional Search*)

Assim como o algoritmo WSFS, o WSBG e o SBS-SLASH, o método BDS também utiliza a estratégia de busca heurística e como critério de avaliação uma medida para classificação (a taxa de erro do classificador). Sendo que a busca é realizada nas duas direções, ou seja, para frente (*foward*) e para trás (*backward*).

PQSS (*(p,q)Sequential Search*)

O algoritmo PQSS utiliza a estratégia de busca heurística e como critério de avaliação uma medida para classificação (a taxa de erro do classificador).

Este método oferece um certo grau de regressão por permitir adição e deleção para cada subconjunto. Caso PQSS inicie a partir de um conjunto vazio, então ele adiciona mais atributos que ele remove em cada iteração. Caso ele inicie a partir de um conjunto de atributos completo, então remove mais atributos e adiciona menos atributos em cada iteração.

Método Schemata

Assim como o algoritmo PQSS, o método Schemata utiliza a estratégia de busca heurística e como critério de avaliação uma medida para classificação (a taxa de erro do classificador).

Este método inicia a partir de um conjunto vazio ou de um conjunto completo, e em cada iteração, encontra o melhor subconjunto por remoção ou adição de somente um atributo do/ ao subconjunto.

A avaliação de cada subconjunto é realizada utilizando LOOCV (*leave-one-out cross validation*) que consiste em uma versão especial de validação cruzada, considerada computacionalmente dispendiosa e freqüentemente utilizada em amostras pequenas. Assim, para uma amostra de tamanho *n* uma hipótese é induzida utilizando (*n - 1*) instâncias; a hipótese é então testada na única instância remanescente. Este processo é repetido *n* vezes, sendo que em cada vez é induzida uma hipótese ignorando-se uma única instância (BARANAUSKAS, 2001).

Portanto, no método Schemata ocorre que em cada iteração é selecionado o subconjunto tendo menor erro LOOCV. Este processo continua até que nenhuma mudança de um simples atributo o melhore.

RC (*Relevance in Context*)

A estratégia de busca heurística e uma medida para classificação (a taxa de erro do classificador) como critério de avalição também são utilizados por RC.

Segundo Domingos (1997), o algoritmo RC pode manipular dados contínuos e simbólicos, valores errôneos e múltiplas classes.

Este método considera o fato que alguns atributos serão relevantes somente em algumas partes do espaço. Assim, decide quanto a relevância de atributos em instâncias específicas.

Método de Queiros e Gelsema

Semelhante ao RC, o método de Queiros; Gelsema (apud DASH; LIU, 1997, p.17) também utiliza a estratégia de busca heurística e uma medida para classificação (a taxa de erro do classificador) como critério de avaliação.

Este método é semelhante ao WSFG, porém sugere que em cada iteração, cada atributo seja avaliado sob várias definições por considerar as diferentes iterações com o conjunto de atributos previamente selecionado. As duas definições consideradas são: sempre assumir independência de atributos (não considerar os atributos previamente selecionados) e nunca assumir a independência (considerar os atributos previamente selecionados). Como critério de avaliação é utilizada a taxa de erro do classificador bayesiano.

Existem ainda outros métodos *wrapper* que utilizam a estratégia de busca não-determinística ou aleatória e como critério de avaliação a taxa de erro do classificador, sejam eles conforme Dash; Liu (1997) e Liu; Yu (2002): o RGSS (*Random Generation Plus Sequential* Algorithm), que inicia gerando um subconjunto aleatório e executando o WSFG e o WSBG a partir de um subconjunto aleatório; o RMHC-PF1(*Random Mutation Hill Climbing-Prototype and Feature Selection*) que seleciona tanto instâncias

quanto atributos simultâneamente para o problema de classificação *nearest-neighbor*[10], utilizando um vetor que armazena as instâncias e os atributos, sendo em cada iteração aleatoriamente alterado um *bit* do vetor para produzir o próximo subconjunto; além dos algoritmos GA (*Genetic Algorithm*), SA (*Simulated Annealing*) e LVW (*Las Vegas Wrapper*).

4) Métodos de seleção de atributos do tipo *wrapper* com abordagem não-supervisionada

Método de Devaney; Ram

Este método utiliza a estratégia de busca heurística e medidas para *clustering* como critério de avaliação. Segundo Han; Kamber (2001), a qualidade de um *cluster* pode ser representada pelo diâmetro do mesmo, que corresponde a distância máxima entre duas instâncias quaisquer no *cluster*.

Segundo Devaney; Ram (1997), o método por eles proposto pode manipular dados contínuos e simbólicos, além de múltiplas classes.

O algoritmo de Devaney; Ram (apud LIU; YU, 2002, p. 15) é apropriado para *clustering* hierárquico (conceitual), aplicando a busca *forward* e *backward* seqüencial para gerar subconjuntos candidatos de atributos, e medindo a boa qualidade de cada subconjunto através da medida da boa qualidade de *clusters* obtidos pela aplicação de COBWEB (um algoritmo de *clustering* hierárquico) no conjunto de treinamento. Sendo a boa qualidade de *clusters* medida através da utilização de um critério de avaliação chamado utilidade de categoria.

O critério de utilidade de categoria consiste em combinar características de modelos *wrapper* e filtro, na medida em que funciona como um modelo *wrapper* quando utiliza o algoritmo de aprendizagem para guiar a busca do descritor mas também assemelha-se com um filtro quando a função de avaliação mede uma propriedade própria do dado ao invés de algum tipo de acurácia preditiva (DEVANEY; RAM, 1997).

FSSEM (*Feature Subset Selection and EM clustering*)

Este algoritmo também utiliza a estratégia de busca heurística e medidas para *clustering* como critério de avaliação.

O método FSSEM aplica a busca forward seqüencial e realiza (*wraps*) à seleção do subconjunto de atributos em torno do algoritmo de *clustering* EM (*Expectation-Maximization*). Quando o número k de *clustering* não é conhecido, ele envolve EM-k (clusterização EM com ordem de identificação), que busca por k e pelos *clusters*.

[10] Classificação *nearest-neighbor* é baseada em aprendizagem por analogia. Assim, as amostras de treinamento são descritas por atributos numéricos n-dimensionais. Cada amostra representa um ponto no espaço n-dimensional. Desta forma, todas as amostras de treinamento são armazenadas em um espaço n-dimensional padrão. Quando é fornecida uma amostra desconhecida, o classificador k-*nearest neighbor* busca o espaço padrão pelas k amostras de treinamento que estão mais próximas à amostra desconhecida. Estas k amostras de treinamento são os k vizinhos mais próximos (*nearest neighbors*) da amostra desconhecida. Sendo a proximidade definida por uma distância euclidiana (HAN; KAMBER, 2001).

Para avaliar a boa qualidade de *clusters* o algoritmo FSSEM utiliza o critério de separabilidade *scatter* e o critério de probabilidade máxima, além de aplicar uma função de normalização, para de algum modo, remover o *bias* sobre a dimensão do espaço de atributos.

ELSA (*Evolutionary Local Search Algorithm*)

Assim como o algoritmo de Devaney; Ram e o FSSEM, este algoritmo também utiliza a estratégia de busca heurística e as medidas para *clustering* como critério de avaliação.

Segundo Kim; Street; Menczer (2000), o algoritmo ELSA manipula dados contínuos, valores errôneos e gera um determinado número de *clusters* aplicando o algoritmo K-médias que também é utilizado para avaliar a qualidade do subconjunto de atributos selecionado.

O método ELSA utiliza um algoritmo de seleção local evolucionária que mantém uma população diversa de soluções que aproximam a frente Pareto no espaço objetivo multi-dimensional. Sendo os *clusters* formados a partir da aplicação dos algoritmos K-médias e EM sobre o subconjunto de atributos selecionado.

5) Métodos de seleção de atributos do tipo híbrida com abordagem supervisionada

BBHFS (*Boosting-Based Hybrid Feature Selection*)

Este método utiliza a estratégia de busca heurística e a combinação de medidas de informação e medidas para classificação (acurácia) como critério de avaliação.

O algoritmo BBHFS baseia-se no conceito de *boosting* da teoria de aprendizagem computacional que é considerado como um método para melhorar o desempenho de um sistema de aprendizado. No algoritmo de *boosting*, cada instância de treinamento possui um peso associado e a cada ciclo de iteração, uma hipótese é induzida a partir das instâncias ponderadas. Então, cada instância de treinamento tem seu respectivo peso alterado, dependendo se ela foi ou não classificada corretamente pela hipótese induzida, ressaltando-se que as hipóteses são induzidas seqüencialmente (BARANAUSKAS, 2001).

Além do embasamento em *boosting*, o método BBHFS avalia a boa qualidade de um subconjunto de atributos de cardinalidade *K* através do critério de ganho de informação. O método em questão não utiliza o número de atributos a ser selecionado como critério de parada, mas sim a acurácia de treinamento de um algoritmo de aprendizagem (por exemplo, *naive-bayes*, ID3, k- *nearest-neighbor*) sendo considerada um critério de parada natural.

O algoritmo de aprendizagem é aplicado a cada subconjunto novamente selecionado com cardinalidade *K*, e se a acurácia de treinamento não aumenta quando comparada a do subconjunto anteriormente selecionado com cardinalidade *K - 1*, o algoritmo pára sem incluir o último atributo ao subconjunto de atributos.

Método de Xing et al.

Semelhante ao BBHFS, este método também utiliza a estratégia de busca heurística e a combinação de medidas de informação e medidas para classificação (acurácia) como critério de avaliação.

O método de Xing et al. (apud LIU; YU, 2002, p. 14) adota o ganho de informação para avaliar o subconjunto de atributos, porém os atributos que passam pelo filtro de ganho de informação são entradas para um procedimento de seleção de subconjunto mais intensivo computacionalmente chamado Filtro de *Market Blanket,* que apresenta-se sob a seguinte definição: considerando-se *M* como um conjunto de atributos que não contém um atributo F_i, tem-se que *M* é um *Markov blanket* para F_i se F_i é condicionalmente independente de $(F\ Y\ C) - M - \{F_i\}$ dado *M*, onde *F* é o conjunto de atributos e *C* é o conjunto de classes (PEARL, 1988, p. 97 apud KOLLER; SAHAMI, 1996, p. 03). Assim, o critério de *Markov blanket* remove somente atributos realmente desnecessários como aqueles que são irrelevantes a classe e os que são redundantes.

Além disso, no método de Xing et al., os subconjuntos que apresentam diferentes cardinalidades são comparados utilizando a validação cruzada que é considerada como um método de estimativa da acurácia de um modelo de classificação ou de regressão.

A validação cruzada consiste em dividir o conjunto de treinamento dentro de *k* subconjuntos. Um classificador é então treinado utilizando os $k - 1$ dos subconjuntos, e é avaliado no k^{th} subconjunto. Este processo é repetido *k* vezes, utilizando cada um dos subconjuntos como o subconjunto de validação. Os resultados de validação são então combinados para obter uma estimativa geral da efetividade do procedimento de treinamento (LEWIS, 2001).

6) Método de seleção de atributos do tipo híbrida com abordagem não-supervisionada

Método de Dash; Liu

Este método utiliza a estratégia de busca heurística e a combinação das medidas de informação e medidas para *clustering* como critério de avaliação.

O algoritmo de Dash; Liu (apud LIU; YU, 2002, p. 16) é uma extensão do algoritmo SBUD, na medida em que introduz amostragem aleatória para resolver o problema da escalabilidade do algoritmo SBUD em grandes conjuntos de dados, e que adota o modelo híbrido utilizando o algoritmo de *clustering* K-médias para avaliar a ordenação dos atributos e também para escolher o subconjunto que maximiza a qualidade de *clustering*, o que resolve o problema apresentado no algoritmo SBUD de seleção arbitrária do subconjunto ótimo de atributos.

O método proposto por Dash; Liu (apud DASH; LIU, 2000, p. 02 - 06) consiste em dois passos: Primeiro, os atributos são ordenados de acordo com a relevância dos mesmos para *clustering*, para tanto é utilizada uma medida baseada em entropia que permite a atribuição do grau de importância para o atributo. Segundo, um subconjunto de atributos relevantes é selecionado utilizando-se uma função de critério para *clustering* que é a mesma para qualquer número de atributos. Além disso, um novo

método escalável baseado em amostragem aleatória é introduzido para grandes conjuntos de dados.

Dependendo do número de instâncias manipulado tem-se que a entropia (*E*) é calculada pelas seguintes fórmulas:

a) Para duas instâncias X_{i_1} e X_{i_2}

$$E = -S_{i_1,i_2} \log S_{i_1,i_2,} - (1 - S_{i_1,i_2}) \log(1 - S_{i_1,i_2})$$

onde *E* assume o valor máximo de 1.0 para $S_{i_1}, S_{i_2} = 0.5$

e o valor mínimo de 0.0 para $S_{i_1}, S_{i_2} = 0.0$ e $S_{i_1}, S_{i_2} = 1.0$

b) Para N instâncias

$$E = -\sum_{i_1=1}^{N} \sum_{i_2=1}^{N} (S_{i_1,i_2} \times \log S_{i_1,i_2} + (1 - S_{i_1,i_2}) \times \log(1 - S_{i_1,i_2}))$$

onde S_{i_1}, S_{i_2} assume valores dentro do intervalo de 0.0 a 1.0.

Para dados numéricos é utilizada a distância Euclidiana para calcular a medida de similaridade (S_{i_1}, S_{i_2}) que é dada pela fórmula:

$$S_{i_1,i_2} = e^{-\alpha \times D_{i1,i2}}$$

onde α é um parâmetro. Em um espaço multidimensional, a distância D_{i_1,i_2} para dados numéricos é dada pela fórmula:

$$D_{i_1,i_2} = \left[\sum_{k=1}^{M} \left(\frac{x_{i_1}k - x_{i_2}k}{\max_k - \min_k} \right)^2 \right]^{1/2}$$

O intervalo na k^{th} dimensão é normalizado por dividí-lo pelo intervalo máximo ($max_k - min_k$) antes de calcular a distância.

Tratando-se de atributos nominais é utilizada a distância de Hamming para calcular a similaridade que é dada pela fórmula:

$$S_{i_1,i_2} = \frac{\sum_{k=1}^{M} | x_{i_1}k = x_{i_2}k |}{M}$$

onde $| x_{i_1}k = x_{i_2}k |$

é 1 se $x_{i_1}k = x_{i_2}k$ e 0 caso contrário.

Para a ordenação de atributos a entropia (*E*) é utilizada da seguinte forma: cada atributo é removido e *E* é calculada. Caso a remoção de um atributo resulte em um *E* mínimo, isto significa que o atributo em questão é menos importante; e vice-versa.

Diante de um conjunto de dados muito grande o método descrito acima passa a ser referenciado dentro de outro método escalável baseado em amostragem aleatória, que consiste em inicializar todos os atributos com ordem 0, gerar amostras aleatórias e executar o método descrito no parágrafo anterior sobre cada amostra para produzir as ordenações de atributos, adicionar as ordenações de atributos correspondentemente e após o processo ser realizado em todas as *p* amostras aleatórias é obtida a ordenação final dos atributos.

Tratando-se da seleção do subconjunto de atributos relevantes são consideradas duas alternativas por Dash; Liu (2000): primeiro, caso seja conhecido o número necessário de atributos relevantes, utiliza-se somente este, começando com o mais importante. Segundo, pode-se utilizar um algoritmo de *clustering* para escolher o subconjunto que maximiza a qualidade de *clustering*.

Na primeira opção é preciso ter um conhecimento prévio, caso contrário torna-se impraticável. Já a segunda, corresponde a utilização de um método *wrapper*, sendo que os atributos a serem manipulados já estão ordenados de acordo com o seu grau de importância e assim a tarefa de buscar através de um espaço de subconjunto de atributos de 2^M , onde M é o número total de atributos, é evitada. O objetivo então é executar um algoritmo de *clustering* nos atributos selecionados e escolher o subconjunto que produz melhor qualidade de *cluster*.

Assim para o propósito de selecionar o subconjunto de atributos relevantes, Dash; Liu (2000) utiliza o algoritmo de *clustering* k-médias para encontrar os *clusters* utilizando o subconjunto de atributos ordenados e em seguida utiliza um critério que não varia diante de transformações lineares não-singulares nos dados. Este critério mede a razão entre a dispersão inter-*cluster* e intra-*cluster*, de modo que quanto maior a razão, maior é a qualidade do *cluster*.

Anexo 2

Métodos de Seleção de Instâncias

Neste anexo são apresentados alguns métodos de seleção de instâncias do tipo filtro e *wrapper*.

Quanto aos métodos de seleção de instâncias do tipo filtro, tem-se alguns citados por Riaño (1997) que são úteis para reduzir instâncias quando o conjunto de dados de entrada está sendo classificado com um método que é baseado em uma função de distância tal como um classificador *nearest-neighbor*. E como método de seleção de instâncias do tipo *wrapper* é citado por Blum; Langley (1997) o *windowing*. A seguir são descritos tais métodos.

1) Métodos de seleção de instâncias do tipo filtro

IFOCUS

O algoritmo IFOCUS realiza uma busca exaustiva por um subconjunto mínimo de instâncias tal que este seja capaz de representar as instâncias que não estão no subconjunto. Este método inicia com um subconjunto vazio e trabalha adicionando instâncias uma por uma. Uma estratégia de busca em largura ou amplitude (*breadth-first*) é seguida com o auxílio de uma fila onde pares de conjuntos são colocados. O primeiro conjunto representa o subconjunto de instâncias que estão sendo consideradas em cada momento, e o segundo conjunto é utilizado para conter as instâncias que não estão no primeiro, e portanto, que tendem a ser incorporadas ao primeiro conjunto.

O custo de tempo deste algoritmo é $O\ (mpn^p)$ onde m o número de atributos, p é o menor número de instâncias relevantes e n o número de instâncias no conjunto de treinamento.

Existe ainda uma variação de IFOCUS chamada IFOCUS2 que inclui uma instância x ao conjunto mínimo de instâncias relevantes A, permitindo não só a eliminação de x de futuras considerações, mas também a eliminação de outras instâncias próximas a x.

Forward

O algoritmo *forward* é um método que utiliza a busca heurística, e baseia-se no algoritmo de aprendizagem baseada em instâncias IB2, onde as instâncias do conjunto de treinamento são incluídas no conjunto de descrição de conceito de instâncias se a instância mais próxima do conjunto de descrição de conceito pertence a mesma classe ou não.

Este método inicia com um conjunto vazio de instâncias relevantes, A. Então as instâncias do conjunto de treinamento são requeridas a serem relevantes, de acordo com a seguinte definição: “Uma instância é irrelevante a uma certa classe se a instância relevante mais próxima desta pertence a mesma classe” (RIAÑO, 1997, p.63)

Assim, satisfazendo esta definição as instâncias em questão podem ser incluídas em A. As instâncias relevantes são incorporadas em A, e o processo é repetido até todas as instâncias do conjunto de treinamento serem irrelevantes em relação a outras em A.

O custo deste algoritmo no pior caso – quando todas as instâncias são relevantes e em cada iteração somente uma instância é incorporada a A – é de

$$O(\frac{n(n+1)^2}{6}m)$$

onde n é o número de instâncias na amostra, e m é o número de atributos.

ISET

Este algoritmo também realiza uma busca heurística para seleção de instâncias. Inicialmente, o ISET computa os conflitos entre instâncias de classes diferentes através da equação :

$$c_{ij} = \begin{cases} 1 \text{ se } d(x_i,x_j) < d(x_i, n_i) \text{ e } classe\ (x_i) = classe\ (x_j) \\ 0 \text{ caso contrário} \end{cases}$$

Onde d é uma função de distância, e n_i representa a instância mais próxima a x_i dentro de uma classe diferente. Esta equação traduz a idéia de relevância da definição acima descrita ao conceito de conflito. Assim, uma instância x_i está em conflito com uma segunda instância x_j se elas pertencem a mesma classe e não existe outra instância de uma classe diferente entre elas. Isto é verdade se uma delas é irrelevante.

Na medida em que os conflitos são obtidos e eles são representados na forma de uma matriz, a i^{th} linha que tem mais conflitos com instâncias que não estão em B identifica a instância que é incorporada no conjunto de instâncias relevantes, A.

IRET

O algoritmo IRET implementa um método de convergência para seleção de instâncias.

Este algoritmo realiza uma aproximação *rough* para cada relevância de instância através da consideração das instâncias próximas. Para delimitar quais são as instâncias em torno de uma determinada instância, o IRET calcula um radiano que é uma percentagem π da diferença entre a maior e a menor distância entre duas instâncias na amostra. As instâncias da mesma classe que x diminuem a relevância de x, e as instâncias em uma classe diferente a aumentam.

IPRO1

Este algoritmo é um método probabilístico para seleção de instâncias que toma um subconjunto aleatório de instâncias onde cada instância no conjunto de treinamento tem a mesma probabilidade de ser parte ou não deste subconjunto.

O novo subconjunto é comparado com a melhor alternativa atual A, e se o formador melhora o último, o novo subconjunto é tomado como a melhor alternativa, e o anterior que tinha sido considerado melhor é ignorado. O processo é repetido um número pré-definido de *k* vezes, e ao final o melhor subconjunto A é retornado.

IPRO2

O algoritmo IPRO2 também é um método probabilístico para seleção de instâncias que realiza uma busca binária do número de instâncias relevantes.

Este método considera os casos extremos: quando nenhuma instância é relevante e o caso em que todas as instâncias são relevantes. Sendo este último mais promissor para seleção de instâncias A. Então, o algoritmo IPRO2 estuda o caso n_3 onde somente metade das instâncias são relevantes. Assim, se o estudo tem êxito e um conjunto descritivo de instâncias relevantes é encontrado com somente a metade das instâncias, o conjunto mais promissor é substituído com um B encontrado, e o número superior de n_2 atributos relevantes é definido para a cardinalidade dele. Caso o estudo não tenha êxito, o conjunto mais promissor é alterado, e o número mais baixo de n_1 instâncias relevantes é definido para a metade. O processo é repetido até os limites superior e inferior serem os mesmos e então o conjunto mais promissor conter as instâncias selecionadas.

Em geral, o algoritmo IPRO2 gera aleatoriamente um número pré-definido de conjuntos *K*, todos com uma cardinalidade n_3, e testa se todos eles descrevem uma percentagem do conjunto de treinamento que esteja acima de um determinado limiar γ pré-definido.

2) Método de seleção de instâncias do tipo *wrapper*

Windowing

Consiste em uma técnica de caráter *wrapper* que identifica uma subamostra aleatória de instâncias (*window*) do conjunto de dados original e a utiliza como o conjunto de treinamento para realizar o processo de aprendizagem sobre a mesma, para induzir uma árvore de decisão inicial, então utiliza esta árvore para classificar todos as demais instâncias. A partir das instâncias classificadas incorretamente (*misclassified*), o método seleciona outro conjunto aleatório para aumentar a amostra original, constrói uma nova árvore de decisão, e assim por diante, repetindo o processo até obter uma árvore que classifique corretamente todos os dados de treinamento.

O método *windowing* consiste então em reduzir o tempo de aprendizagem através da construção de uma árvore inicial a partir de uma pequena amostra (10 a 20%) das instâncias de treinamento. Contudo este tempo de aprendizagem pode ser maior se estiverem sendo manipulados dados com valores errôneos.

Anexo 3

Métodos de Discretização de Atributos

Neste anexo são apresentados alguns métodos de discretização de atributos contínuos. Em geral, os métodos de discretização são aplicados sobre um determinado atributo dividindo o intervalo de valores contínuos ou combinando os intervalos adjacentes. E estas duas formas de discretização, por divisão e por combinação, podem ser agrupadas como supervisionadas ou não-supervisionadas dependendo do uso da informação de classe.

Dentre os métodos de discretização considerados como de divisão (*splitting*), alguns como ID3, D2 e MDLP utilizam uma medida de discretização bastante conhecida que é a entropia. Segundo Shannon; Weaver (apud HUSSAIN et al., 1999, p. 10), a entropia de um atributo X é calculada por:

$$\mathrm{H(X)} = -\sum_x p_x \log p_x$$

onde x representa um valor de X e p_x sua probabilidade estimada de ocorrência. A entropia consiste assim na quantidade média de informação por evento, sendo a informação de um evento é definida como:

$$I(x) = -\log p_x$$

A informação é alta para eventos pouco prováveis e baixa caso contrário. Contudo, a entropia H é maior quando cada evento é equi-provável, isto é,

$$p_{x_i} = p_{x_j}$$

para todo *i*, *j*; e é menor quando $p_x = 1$ para um evento e 0 para outros eventos.

A seguir é apresentada uma breve descrição dos métodos de discretização por divisão e por combinação, conforme Hussain et al. (1999) e Kohavi; Sahami (1996):

1) Métodos de discretização por divisão

Equal-width

Este é um método não-supervisionado em que a faixa de valores contínuos de um atributo é igualmente dividida dentro de intervalos de mesma largura e cada intervalo é associado a um valor discreto distinto.

Equal-frequency

Também é um método não-supervisionado em que cada intervalo contém aproximadamente o mesmo número de instâncias de treinamento.

1R

Trata-se de um método supervisionado que consiste em após a classificação dos valores contínuos, dividí-los dentro de um número de intervalos independentes e ajustar os limites baseado no conhecimento do valor do atributo de classe associado aos valores contínuos. Cada intervalo deve conter uma quantidade mínima de instâncias, com exceção do intervalo final que deverá conter as instâncias restantes que não foram agrupadas em nenhum intervalo.

D2

Utiliza a medida de entropia para encontrar um *cut-point* potencial para dividir uma faixa de valores contínuos dentro de 2 intervalos. Ao contrário de encontrar somente 1 *cut-point,* o método em questão binariza recursivamente faixas e subfaixas até um critério de parada ser alcançado, pois sem isto poderia ser concluído que cada valor distinto é um *cut-point.*

MDLP (*Minimum Description Length Principle*)

O método MDLP aplica recursivamente um particionamento binário até o custo de representar uma nova partição tornar-se maior que o custo de não representá-la.

Formalmente, a informação introduzida por um *cut-point* T para um conjunto *S* de *n* instâncias é aceito se

$$n \cdot Gain(p,T;S) > \log_2(n-1) + \Delta(p,T;S)$$

onde $Gain(p,T;S)$ é calculado através da fórmula:

$$Gain(p,T;S) = Ent(S) - E(p,T;S)$$

e $\Delta(p,T;S)$ é obtido através da fórmula:

$$\Delta(p,T;S) = \log_2(3^k - 2) - (kEnt(S) - k_1 Ent(S_1) - k_2 Ent(S_2))$$

Mantaras

Utiliza uma medida de distância para avaliar os *cut-points*. Considera-se duas partições, P_a e P_b em uma faixa de valores contínuos, cada uma contendo *n* e *m* número de valores do atributo de classe. A distância *Mantaras* entre duas partições deve-se a um simples *cut-point* obtido através de:

$$Dist(P_a, P_b) = \frac{I(P_a \mid P_b) + I(P_b \mid P_a)}{I(P_a \cap P_b)}$$

Desde que,

$$I(P_b \mid P_a) = I(P_b \cap P_a) - I(P_a)$$

$$Dist(P_a, P_b) = 2 - \frac{I(P_a) + I(P_b)}{I(P_a \cap P_b)}$$

Onde,

$$I(P_a) = -\sum_{i=1}^{n} P_i \log_2 P_i$$

$$I(P_b) = -\sum_{j=1}^{m} P_j \log_2 P_j$$

$$I(P_a \cap P_b) = -\sum_{i=1}^{n} \sum_{j=1}^{m} P_{ij} \log_2 P_{ij}$$

$|C_i|$ = quantidade total da classe i

$$P_i = \frac{|C_i|}{|N|}$$

|N| = número total de instâncias

$P_{ij} = P_i \times P_j$

O método escolhe o *cut-point* que minimiza a distância.

ID3

Para encontrar *cut-points* potenciais dentro de uma faixa existente de valores contínuos, este método utiliza a seguinte fórmula:

$$H = -p_{left} \sum_{j=1}^{m} p_{j,left} \log p_{j,left} - p_{right} \sum_{j=1}^{m} p_{j,right} \log p_{j,right}$$

onde m é o número de classes, p_{left} e p_{right} são probabilidades que uma instância está no lado esquerdo ou no lado direito de um *cut-point* respectivamente. Já $p_{j,side}$ representa a probabilidade que uma instância no lado (*left* ou *right*) pertença a classe *j*. Assim o *cut-point* com a entropia mais baixa é escolhido para dividir a faixa dentro de duas partes. A divisão continua em cada parte até um critério de parada ser satisfeito, na verdade ocorre a binarização de uma faixa em qualquer divisão.

C4.5

Conforme o descrito em Kohavi; Sahami (1996), o algoritmo de indução de árvore de decisão C4.5 pode também ser aplicado como método de discretização na fase de pré-processamento. Neste caso, o C4.5 é aplicado a cada atributo contínuo separadamente construindo uma árvore completa para este atributo e então aplicando uma poda para encontrar um número apropriado de nós na árvore que corresponde ao número de intervalos de discretização, caracterizando-se assim como uma abordagem *bottom-up*.

Após a construção da árvore para um determinado atributo os valores de entrada em cada nó da árvore induzida são usados como valores de entrada para uma discretização deste atributo contínuo.

Zeta

É uma medida de intensidade de associação entre o atributo de classe e o atributo a ser discretizado. Pode ser definido como a acurácia máxima alcançada quando cada valor de um atributo prediz um valor de classe diferente. Para um atributo com *k* valores, tem-se *k-1 cut-points* potenciais. Um *cut-point* com o valor mais alto de Z , calculado pela fórmula:

$$Z = \frac{\sum_{i=1}^{k} n_{f(i),i}}{N} \times 100\%$$

é selecionado se nenhum par vizinho de partições prediz a mesma classe. Este método continua binarizando sub-faixas da mesma forma até o critério de parada ser alcançado.

Adaptive Quantizer

Neste método a faixa de valores contínuos de cada atributo é dividida em duas partições por *equal-frequency* ou por *equal-width*. A divisão é testada pela execução de um classificador para verificar se com a mesma houve melhora na acurácia. A binarização continua sobre as subfaixas e o *cut-point* que apresenta a taxa mínima é selecionado. Devido a necessidade do treinamento de um classificador, o tempo consumido é maior que sem o uso do mesmo.

Quanto aos métodos de combinação (*merging*), estes consistem em quatro etapas:

a) Classificar os valores;

b) Encontrar o melhor par de intervalos adjacentes;

c) Combinar o par dentro de 1 (um) intervalo e

d) Parar quando o critério de parada escolhido é satisfeito.

Os métodos que realizam este tipo de discretização utilizam a estatística X^2 como uma medida de avaliação. Esta medida X^2 conduz um teste significativo do relacionamento entre os valores de um atributo e do atributo de classe. Assim a estatística X^2 testa a hipótese que dois intervalos adjacentes de um atributo são independentes da classe. Caso sejam independentes, estes devem ser combinados, senão devem permanecer separados. A fórmula para obter X^2 é:

$$X^2 = \sum_{i=1}^{2} \sum_{j=1}^{p} \frac{(A_{ij} - E_{ij})^2}{E_{ij}}$$

onde:

p = número de classes;
A_{ij} = número de valores distintos no *ith* intervalo, *jth* classe;
E_{ij} = freqüência esperada de $Aij = (Ri \times Cj) / N$
Ri = número de instâncias no *ith* intervalo, calculado através de

$$R_i = \sum_{j=1}^{p} A_{ij}$$

C_j = número de exemplos na *jth* classe, calculado através de

$$C_j = \sum_{i=1}^{m} A_{ij}$$

N = número total de exemplos, calculado através de

$$N = \sum_{j=1}^{p} C_j$$

2) Métodos de discretização por combinação

ChiMerge

Trata-se de um método supervisionado que considera inicialmente cada valor distinto do atributo como um intervalo. O teste X^2 é realizado para cada par de intervalos adjacentes. Os intervalos adjacentes com o valor X^2 mínimo são combinados até o critério de parada ser satisfeito. Um valor maior do nível de significância para o teste X^2 causa superdiscretização, enquanto um valor mais baixo causa subdiscretização. Algumas recomendações são feitas em relação ao nível de significância, que pode ser definido entre 0.90 e 0.99, e ao parâmetro *max-interval,* que pode ser definido como 10 ou 15.

Chi2

Segundo Liu; Setiono (1995), esse algoritmo discretiza atributos numéricos assim como também seleciona os atributos relevantes. Consiste em duas fases:

a) Na primeira fase, inicializa um alto nível de significância, por exemplo 0.5, para todos os atributos numéricos sujeitos à discretização. Cada atributo é classificado de acordo com seus valores, então é calculado o valor X^2 para cada par de intervalos adjacentes, em seguida o par de intervalos adjacentes com o valor X^2 mais baixo é combinado. A combinação continua até todos os pares de intervalos terem valores X^2 excedendo o parâmetro determinado pelo nível de significância. Esse processo da primeira fase é repetido com um decremento do nível de significância até uma taxa de inconsistência, δ ser excedida no dado discretizado.

b) Na segunda fase, é inicializado o nível de significância com o valor imediatamente superior ao último valor assumido do nível de significância no final da primeira fase. Cada atributo *i* é associado a um nível de significância [*i*], e tomado para combinação. A verificação da consistência é conduzida após cada combinação de atributo. Caso a taxa de inconsistência não seja excedida, o nível de significância [*i*] é decrementado para o próximo atributo *i* envolvido na combinação, senão o atributo *i* não será envolvido na próxima combinação. Este processo continua até nenhum valor

de atributo poder ser combinado. Ao final da segunda fase, se um atributo é combinado para somente um valor, isto significa que este atributo não é relevante na representação do conjunto de dados original.

Este algoritmo pode ser aplicado a dados com atributos de diferentes tipos (discretos e numéricos)

ConMerge

É um método similar ao *Chi2* já que também usa a medida estatística X^2 e a inconsistência. Sendo que, em vez de considerar um atributo por vez, o *ConMerge* escolhe o menor valor X^2 entre os intervalos de todos os atributos contínuos.

Glossário

Acurácia: constitui um fator importante na avaliação do sucesso de mineração de dados. Quando aplicada a dados, a acurácia refere-se a taxa de valores corretos no dado. Quando aplicada a modelos, a acurácia consiste no grau de ajuste entre o modelo e o dado, ela verifica então quão as predições do modelo estão livres de erros.

Atributo: também chamado de variável, característica, propriedade ou campo possui valores pertencentes a um conjunto pré-definido de valores que são dependentes do problema, e que descrevem algum aspecto da instância.

Bias: de um modo geral significa tendenciosidade (CAMARÃO, 1994). Corresponde a qualquer preferência de uma hipótese sobre outra, além da simples consistência com as instâncias (RUSSEL; NORVIG, 1995 apud BARANAUSKAS, 2001, p. 23)

Cobertura: a cobertura de um algoritmo de aprendizagem, *L*, é a medida do número de conceitos distintos que podem ser aprendidos a partir de uma amostra de treinamento de tamanho *m* (ALMUALLIM; DIETTERICH, 1991).

Complexidade da amostra: a complexidade da amostra de um algoritmo *L* para um espaço de conceitos *C* é estimada como o menor tamanho de amostra suficiente para habilitar *L* a aprender freqüente e aproximadamente de maneira correta qualquer conceito em *C* (ALMUALLIM; DIETTERICH, 1991).

Critério de Máxima Semelhança (*Maximum Likelihood*): esta medida verifica o quão apropriado o dado está aos parâmetros e ao modelo dado (DY; BRODLEY, 2000).

Instância: também denominada exemplo, fato ou registro, é um conjunto ordenado de atributos que descreve um objeto de interesse.

Modelo: consiste na descrição dos dados originais da base de dados, e é construído para ser aplicado com sucesso em novos dados para fazer predições sobre valores que faltam ou para descobrir valores esperados. O desenvolvimento de um modelo envolve duas fases: a de treinamento – que é a construção de um novo modelo utilizando geralmente grandes proporções dos dados que serão avaliados – e a de teste – que envolve o experimento do modelo em novos dados previamente não vistos para determinar sua acurácia e características do desempenho físico, sendo geralmente realizada em uma pequena porcentagem dos dados, destinados exclusivamente a este objetivo (BERSON; STEPHEN, 1997 apud NOGUEZ, 2000, p. 27).

Padrão: refere-se a um evento ou combinação de eventos que ocorrem com mais freqüência do que o esperado numa base de dados (BERSON; STEPHEN, 1997 apud NOGUEZ, 2000, p. 27).

Tabela: também conhecida por arquivo ou conjunto de dados, é composta por um conjunto de instâncias.

Taxa de erro de um classificador: corresponde ao quociente entre o número de instâncias incorretamente classificadas sobre o número total de instâncias classificadas.

Validação Cruzada (*Cross Validation*): consiste em dividir o conjunto de treinamento dentro de *k* subconjuntos. Um classificador é então treinado utilizando *k – 1* dos subconjuntos, e é avaliado com o subconjunto não utilizado no treinamento. Este processo é repetido *k* vezes, utilizando cada um dos subconjuntos como o subconjunto de validação. Os resultados de validação são então combinados para obter uma estimativa geral da efetividade do procedimento de treinamento (LEWIS, 2001).

Bibliografia

ALMUALLIM, H.; DIETTERICH, T. G. Learning With Many Irrelevant Features**.** In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, AAAI, 9., 1991. **Proceedings** ... [S.l.: s.n.], 1991. Disponível em: **<**http://citeseer.nj.nec.com/almuallim91learning.html> Acesso em: 20 dez. 2001.

BARANAUSKAS, J. A. **Extração Automática de Conhecimento por Múltiplos Indutores**. 2001. 194 p. Tese (Doutorado em Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, ICMC-USP, São Carlos.

BLUM, A. L.; LANGLEY, P. Selection of relevant features and examples in machine learning. In: ARTIFICIAL INTELLIGENCE, 1997. **Proceedings** ... [S.l.: s.n.], 1997. p. 245-271. Disponível em: <http://citeseer.nj.nec.com/blum97selection.html> Acesso em: 10 set. 2001.

BRACHMAN, R. J.; ANAND, T. The Process of Knowledge Discovery in Databases: a human-centered approach. In: FAYYAD, U. M. (Ed.). **Advances in Knowledge Discovery and Data Mining**. Menlo Park, Califórnia: AAAI Press: The MIT Press, 1996. p. 37-57.

BRUHA, I. Data Mining, KDD, and Knowledge Integration: methodology and a case study. In: INTERNATIONAL CONFERENCE ADVANCES IN INFRASTRUCTURE FOR ELETRONIC BUSINESS, SCIENCE, AND EDUCATION ON THE INTERNET, SSGRR, 2000. **Proceedings** ... [S.l.: s.n.], 2000. Disponível em: <http://www.ssgrr.it/en/ssgrr2000/papers/288.pdf>. Acesso em: 13 jun. 2001.

CABENA, P. et al. **Discovering Data Mining :** from concept to implementation. Upper Saddle River: Prentice Hall PTR, c 1998.

CAMARÃO, P. C. B. **Glossário de Informática**. 2. ed. Rio de Janeiro: LTC, 1994.

CHAPMAN, P. et al. **The CRISP-DM Process Model**. 1999. Disponível em: <http://www.crisp-dm.org>. Acesso em: 25 jun. 2001.

CUROTTO, C. L. **A Strategy of Data Mining using Incremental Induction of Decision Trees**. 2000. Proposta de Tese (Doutorado) – UFRJ, Rio de Janeiro. Disponível em: <http://www.curotto.com/doc/english/thesis/proposal.pdf>. Acesso em: 27 jun. 2001.

DASH, M.; LIU, H. Feature Selection for Classification. **Intelligent Data Analysis - An International Journal**, [S. l.], v. 1, n. 3, 1997. Disponível em: <http://www.public.asu.edu/~huanliu/publications.html> Acesso em: 04 ago. 2002.

DASH, M.; LIU, H. Feature Selection for Clustering. In: PAKDD, 2000. **Proceedings ...** Kyoto: Springer, 2000. p. 110-121. Disponível em: <http://www.public.asu.edu/~huanliu/publications.html> Acesso em: 04 ago. 2002.

DEVANEY, M.; RAM, A. Efficient Feature Selection in Conceptual Clustering. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, ICML, 14., 1997, San Francisco. **Proceedings** ... San Francisco: Morgan Kaufmann, 1997. p. 92-97. Disponível em: <http://citeseer.nj.nec.com/devaney97efficient.html>. Acesso em: 10 out. 2002.

DOMINGOS, P. **Context-Sensitive Feature Selection for Lazy learners**. 1997. Disponível em: <http://citeseer.nj.nec.com/103012.html> Acesso em: 15 ago. 2002.

DY, J. G.; BRODLEY, C. E. Feature Subset Selection and Order Identification for Unsupervised Learning. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 17., 2000. **Proceedings** ... [S. l. : s. n.], 2000. p. 247 –254.

FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From Data Mining to Knowledge Discovery: an overview. In: FAYYAD, U. M. (Ed.). **Advances in Knowledge Discovery and Data Mining**. Menlo Park, California: AAAI Press: The MIT Press, 1996. p. 01-34.

FELDENS, M. A. **Engenharia da Descoberta de Conhecimento em Bases de Dados:** estudo e aplicação na área de saúde. 1997. 90 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre.

FÉLIX, L. C. M. **Data Mining no Processo de Extração de Conhecimento de Bases de Dados**. 1998. 116 f. Dissertação (Mestrado em Ciências) – USP, São Carlos. Disponível em: <http://www.lsi.upc.es/~lcmolina/sc/html/paper/thesismsd.pdf>. Acesso em: 13 jul. 2001.

FURASTÉ, P. A. **Normas Técnicas para o Trabalho Científico:** explicitação das normas da ABNT. 11.ed. Porto Alegre: [s.n.], 2002.

GALHARDAS, H. et al. AJAX: an extensible data cleaning tool. In: CONFERENCE ON MANAGEMENT OF DATA, ACM SIGMOD, 2000. **Proceedings** ... [S.l.: s.n.], 2000. Disponível em: <http://www-caravel.inria.fr/Fmbrepubs_galharda.html> Acesso em: 07 jan. 2002.

GALHARDAS, H. et al. Declaratively Cleaning your Data using AJAX . **Journées Bases de Données Avancées - BDA,** 2000. Disponível em: <http://www-rodin.inria.fr/Fpubsbyyear.html> Acesso em: 07 jan. 2002.

GUPTA, S. K. et al. **Intension Mining:** a new paradigm in knowledge discovery. New Delhi: Department of Computer Science and Engineering, Indian Institute of Technology, 2000. 67 f. (Relatório Técnico IITD/ CSE/ TR2000/ 001). Disponível em: <http://citeseer.nj.nec.com/434044.html>. Acesso em: 18 jun. 2001.

HALMENSCHLAGER, C. **Um algoritmo para indução de árvores e regras de decisão**. 2002. 112 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre.

HAN, J.; KAMBER, M. **Data Mining:** concepts and techniques. San Francisco: Morgan Kaufmann, c 2001.

HERNANDEZ, M.A.; STOLFO, S. J. Real-World Data Is Dirty: data cleansing and the merge/ purge problem. **Data Mining and Knowledge Discovery**, [S. l.], v. 2, n. 1, p. 9-37, 1998. Disponível em: <http://citeseer.nj.nec.com/hernandez98realworld.html> Acesso em: 07 jan. 2002.

HERNANDEZ, M.; STOLFO S. The Merge/Purge Problem for Large Databases. In: SIGMOD, 1995. Disponível em: < http://citeseer.nj.nec.com/stolfo95mergepurge.html> Acesso em: 07 jan. 2002.

HEUSER, C. A. **Projeto de Banco de Dados**. 3. ed. Porto Alegre: Sagra Luzzatto, 2000.

HO, K. M.; SCOTT, P. D. Zeta: a global method for discretization of continuous variables. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, KDD, 3., 1997. **Proceedings ...** [S.l.: s.n.], 1997. p. 191-194. Disponível em: <http://citeseer.nj.nec.com/ho97zeta.html> Acesso em: 13 abr. 2002.

HSU, W. et al. **Exploration Mining in Diabetic Patients Databases:** findings and conclusions. 2000. Disponível em: <http://citeseer.nj.nec.com/327409.html> Acesso em: 10 jun. 2001.

HUSSAIN, F. et al. **Discretization:** an enabling technique. Singapore: School of Computing, National University of Singapore, 1999. 29 p. (Technical Report TRC6/99). Disponível em: <http://www.public.asu.edu/~huanliu/publications.html> Acesso em: 13 abr. 2002.

JHA, G.; HUI, S. C. Data Mining for Risk Analysis and Targeted Marketing. In: PACIFIC RIM INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE ON PRICAI, PRICAI, 5., 1998, Singapura. **Topics in Artificial Intelligence:** proceedings. Berlin: Springer – Verlag, 1998. p. 158-169. (Lecture Notes in Artificial Intelligence, 1531).

JOHN, G. H. **Enhancements to the Data Mining Process**. 1997. 194 f. Tese (Doctor of Philosopfy) – Department of Computer Science, Stanford University, Stanford.

JOHN, G.; KOHAVI, R.; PFIEGER, K. Irrelevant features and the subset selection problem. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 11., 1994. **Proceedings ...** [S.l.: s.n.], 1994. p. 121-129. Disponível em: < http://citeseer.nj.nec.com/john94irrelevant.html> Acesso em: 25 set. 2002.

KIM, Y.; STREET, W.; MENCZER, F. Feature Selection for Unsupervised Learning Via Evolutionary Search. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, ACM SIGKDD, 6., 2000. **Proceedings** ... [S.l.: s.n.], 2000. p. 365 - 369. Disponível em: <http://citeseer.nj.nec.com/kim00feature.html> Acesso em: 11 out. 2002.

KLEMETTINEN, M. **A Knowledge Discovery Methodology for Telecommunication Network Alarm Databases**. 1999. 138 f. Tese (Doutorado em Ciência da Computação) – Universidade de Helsinki, Finland. Disponível em: <http://www.cs.helsinki.fi/TR/A-1999/1>. Acesso em: 13 jun. 2001.

KOHAVI, R.; JOHN, G. Wrappers for Feature Subset Selection. **Artificial Intelligence**, Amsterdam, v. 97, n. 1-2, p. 273-324, 1997. Disponível em: < http://citeseer.nj.nec.com/13663.html> Acesso em: 17 set. 2001.

KOHAVI, R; SAHAMI, M. Error-based and entropy-based discretization of continuous features. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY IN DATABASES, 2., 1996. **Proceedings ...** [S. l.]: AAAI Press, 1996. p. 114-119. Disponível em: <http://citeseer.nj.nec.com/4084.html> Acesso em: 09 ago. 2002.

KOLLER, D.; SAHAMI, M. Toward optimal feature selection. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, ICML, 13., 1996, Bari. **Proceedings ...** Bari: [s. n.], 1996. p. 284-292. Disponível em: < http://citeseer.nj.nec.com/koller96toward.html> Acesso em: 15 set. 2001.

KONONENKO, I. Estimating attributes: analysis and extension of RELIEF. In: EUROPEAN CONFERENCE ON MACHINE LEARNING, 1994. **Proceedings ...** [S. l.]: Morgan Kaufmann, 1994. p. 171-182. Disponível em: < http://citeseer.nj.nec.com/kononenko94estimating.html> Acesso em: 11 out. 2001.

LEE, M. et al. Cleansing data for mining and warehousing. In: DEXA, 1999. Disponível em: <http://citeseer.nj.nec.com/lee99cleansing.html> Acesso em: 07 jan. 2002.

LEE, S. W.; KERSCHBERG, L. A Methodology and Life Cycle Model for Data Mining and Knowledge Discovery in Precision Agriculture. In: INTERNATIONAL CONFERENCE ON SYSTEMS, MAN, AND CYBERNETICS, IEEE SMC, 1998. **Proceedings** ... Disponível em: <http://www.cs.gmu.edu/~swlee/publication.html>. Acesso em: 20 jun. 2001.

LEWIS, D. D. Applying Support Vector Machines to the TREC-2001 Batch Filtering and Routing Tasks. In: TEXT RETRIEVAL CONFERENCE, TREC, 10., 2001, Gaithersburg. **Proceedings** … Gaithersburg: Department of Commerce, National Institute of Standards and Technology, 2001. Disponível em: <http://trec.nist.gov/pubs/trec10/t10_proceedings.html> Acesso em: 11 set. 2002.

LIU, H.; SETIONO, R. A probabilistic approach to feature selection: a filter solution. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, ICML, 13., 1996, Bari. **Proceedings ...** Bari: [s. n.], 1996. p. 319-327. Disponível em: <http://citeseer.nj.nec.com/321378.html> Acesso em: 13 set. 2001.

LIU, H.; SETIONO, R. **Chi2:** feature selection and discretization of numeric attributes. In: INTERNATIONAL CONFERENCE ON TOOLS WITH ARTIFICIAL INTELLIGENCE, IEEE TAI, 7., 1995, Washington D.C., USA. Disponível em: <http://www.public.asu.edu/~huanliu/publications.html> Acesso em: 13 set. 2001.

LIU, H.; YU, L. **Feature Selection for data mining**. [S.l.: s.n.], 2002. Disponível em: <http://bim.im.fju.edu.tw/evergreen/Document/Research/Method/910510Survey-feature-selection.pdf > Acesso em: 15 set. 2002.

LOBO, O. O.; NUMAO, M. Ordered Estimation of Missing Values. In: PACIFIC-ASIA CONFERENCE, 3., PAKDD, 1999, Beijing, China. **Methodologies for Knowledge Discovery and Data Mining**: proceedings. Berlin: Springer-Verlag, 1999. p. 499-503. (Lecture Notes in Artificial Inteligence, 1574).

LUCAS, A. M. **Utilização de Técnicas de Mineração de Dados considerando os Aspectos Temporais**. 2002. 132 f. Dissertação (Mestrado em Informática) – Universidade Federal do Rio Grande do Sul, Porto Alegre.

MORIK, K. et al. **Knowledge Acquisition and Machine Learning:** theory, methods and applications. London: Academic Press, 1993. 305 p.

NEVES, R. C. D. **Estudo de Metodologias de Descoberta de Conhecimento em Banco de Dados**. 2001. 94 f. Trabalho Individual I (PPGC) – Universidade Federal do Rio Grande do Sul, Porto Alegre.

NILSSON, N. J. **Artificial Intelligence:** a new synthesis. [S. l.]: Morgan Kaufmann, 1998.

NOGUEZ, J. H. S. **Técnicas de Mineração de Dados no Processo de Descoberta de Conhecimento em Banco de Dados**. 2000. 47 f. Trabalho Individual I (PPGC) – Universidade Federal do Rio Grande do Sul, Porto Alegre.

OLIVEIRA, A. L. Constructive Induction Using a Non-Greedy Strategy for Feature Selection. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 9., 1992, Aberdeen, Scotland. **Proceedings** ... Aberdeen: Morgan Kaufmann, 1992. p. 355 - 360. Disponível em: <http://citeseer.nj.nec.com/512171.html> Acesso em: 12 set. 2002.

POE, V.; KLAUER, P.; BROBST, S. **Building a Data Warehouse for Decision Support**. 2. ed. Upper Saddle River: Prentice - Hall, 1998.

PYLE, D. **Data Preparation for Data Mining**. San Francisco: Morgan Kaufmann, c 1999.

RAGEL, Arnaud. Preprocessing of Missing Values Using Robust Association Rules. In: EUROPEAN SYMPOSIUM ON PRINCIPLES OF DATA MINING AND KNOWLEDGE DISCOVERY, 2., PKDD, 1998, Nantes, France. **Principles of Data Mining and Knowledge Discovery:** proceedings. Berlin: Springer – Verlag, 1998. p. 414 - 422. (Lecture Notes in Artificial Inteligence, 1510).

RAHM, E.; DO, H**. Data Cleaning:** problems and current approaches. **Bulletin of the IEEE Computer Society Technical Committee on Data Engineering**, [S. l.], v. 23, n. 4, 2000. Disponível em: < http://dol.uni-leipzig.de/pub/2000-45 > Acesso em: 07 jan. 2002.

RIAÑO, D. **Automatic Construction of Descriptive Rules**. 1997. 184 p. PhD thesis (Doctor en Informàtica) - Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Barcelona. Disponível em: <http://www.etse.urv.es/~drianyo/publications/PhD.ps.gz> Acesso em: 10 ago. 2002.

RIO GRANDE DO SUL. Secretaria da Saúde. Coordenação de Regulação das Ações e Serviços de Saúde. **Relatório de Gestão da Assistência no SUS:** Rio Grande do Sul 2000. Porto Alegre, 2001.

SCHLIMMER, J. C. Efficiently Inducing Determinations: a complete and systematic search algorithm that uses optimal pruning. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 1993, San Mateo, CA. **Proceedings ...** San Mateo: Morgan Kaufmann, 1993. p. 284-290. Disponível em: <http://citeseer.nj.nec.com/schlimmer93efficiently.html> Acesso em: 20 set. 2002.

SCHREIBER, A. T. et al. **Knowledge Engineering and Management:** the CommonKADS methodology. Cambridge: MIT, c 2000.

SCOTT, A. C.; CLAYTON, J. E.; GIBSON, E. L. **A Practical Guide to Knowledge Acquisition**. Reading: Addison-Wesley, c 1991.

SHINGHAL, R. **Formal Concepts in Artificial Intelligence:** fundamentals. London: Chapman & Hall, c 1992.

SYED, N. A.; LIU, H.; SUNG, K. K. **From Incremental Learning to Model Independent Instance Selection:** a support vector machine approach. Lower Kent Ridge Road, Singapore: School of Computing, The National University of Singapore, 1999. 31 p. (Technical Report TRA9/ 99). Disponível em: <http:// techrep.comp.nus.edu.sg/techreports/1999/TRA9-99.pdf> Acesso em: 16 abr. 2002.

TRIOLA, M. F. **Introdução à Estatística**. 7. ed. Rio de Janeiro: LTC, c 1999.

TWO CROWS: data mining glossary. 2002. Disponível em: <http://www.twocrows.com/glossary.htm> Acesso em: 27 jul. 2001.

VENTURA, D. **On Discretization as a Preprocessing Step for Supervised Learning Models**. 1995. 67 p. Thesis (Master of Science) – Department of Computer Science of Brigham Young University, Cidade. Disponível em: <http://citeseer.nj.nec.com/ventura95discretization.html> Acesso em: 04 maio 2002.

WILLIAMS, G. J.; HUANG, Z. **Modelling the KDD Process:** a four stage process and four element model. [S. l.]: CSIRO DIT Data Mining, 1996. (Technical Report TR-DM-96013). Disponível em: <http://citeseer.nj.nec.com/292502.html>. Acesso em: 18 jun. 2001.

WITTEN, I. H.; FRANK, E. **Data Mining:** practical machine learning tools and techniques with java implementations. San Francisco: Morgan Kaufmann, c 2000.

ZHENG, Z. **Constructing New Attributes for Decision Tree Learning**. 1996. 253 p. Thesis (Doctor of Philosophy) – Basser Department of Computer Science, The University of Sydney, Sydney. Disponível em: <http: //citeseer.nj.nec.com/zheng96constructing.html> Acesso em: 13 set. 2001.