UNIVERSIDADE FEDERAL FLUMINENSE

PAULA YAMADA BÜRKLE

Um Método de Pós-processamento de Regras de Associação com Base nas Relações de Dependência entre os Atributos

NITERÓI

2006

UNIVERSIDADE FEDERAL FLUMINENSE

PAULA YAMADA BÜRKLE

Um Método de Pós-processamento de Regras de Associação com Base nas Relações de Dependência entre os Atributos

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre. Área de concentração: Otimização Combinatória e Inteligência Artificial.

Orientadora:
Ana Cristina Bicharra Garcia

NITERÓI

2006

Um Método de Pós-processamento de Regras de Associação com Base nas Relações de Dependência entre os Atributos

Paula Yamada Bürkle

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre.

Aprovada por:

Profa. Ph.D. Ana Cristina Bicharra Garcia / UFF - Universidade Federal Fluminense (Presidenta)

Prof. D.Sc. Flávio Miguel Varejão / UFES - Universidade Federal do Espírito Santo

---

Prof. D.Sc. Luiz Satoru Ochi / UFF - Universidade Federal Fluminense

Niterói, 05 de outubro de 2006.

À minha família.

Agradecimentos

À Profa. Ana Cristina, pela orientação, e por ter contribuído imensamente para o meu crescimento profissional e pessoal.

Ao Prof. Ferraz, pelo exemplo, apoio e incentivo.

À equipe do laboratório ADDLabs, pela torcida, em especial Adriana, Evelyn e Rafael, que me apoiaram na caminhada até aqui.

À todos os amigos e colegas da UFF que compartilharam a construção deste trabalho, em especial, Eduardo Corrêa, Adriana Bechara, Miguel, Zé Luíz e Cristiano.

Aos meus pais Yoshimi e Ricardo, pelo apoio incondicional. Às minhas irmãs Renata e Luisa, pela força e pela torcida, e ao restante da minha família.

Ao meu amado Sávio, pelo amor, carinho e respeito.

Resumo

A Mineração de Dados tem como objetivo a descoberta de conhecimento a partir de bases de dados de grandes dimensões. Apesar desse processo ter despontado por seu desempenho em diversos domínios, em alguns casos o resultado gerado pode ser muito grande ou muito complexo. Este é um problema em específico da técnica de Regras de Associação, na qual o número de regras geradas muitas vezes ultrapassa o limite de regras humanamente manipulável.

Neste trabalho propomos uma nova abordagem de poda semântica de regras de associação, baseada no uso do conhecimento prévio do domínio, representado pelas relações de dependência entre os atributos. O método proposto tem como objetivo facilitar a análise e interpretação das regras de associação, eliminando a redundância das regras mineradas, e selecionando aquelas de maior impacto para o usuário.

Dentre os principais resultados deste trabalho destacam-se a proposta e implementação do método de poda de regras de associação DMcut. Os experimentos realizados sobre quatro bases de dados de domínio público mostram os potenciais benefícios de sua utilização.

Abstract

Data mining is a process that aims at knowledge discovery from within great extension databases. Although it has been increasingly adopted due to its good performance in several domains, in some cases the results that are generated may be too large or too complex. This is a problem that is specific to the Association Rules technique, which often generates an amount of rules that exceed the limit amount of rules that are humanly viable to manipulate.

In this present paper we propose a new semantic pruning approach for association rules based on previous domain knowledge, which is represented by attribute inter-dependency relations. The proposed method is aimed at facilitating analysis and comprehension of the rules, by means of eliminating redundancy within the mined rules, and by selecting those that have greater impact for the user's needs.

Among the main results of the present study is the proposal and implementation of the DMcut association rules pruning method. The experiments that were conducted on four public domain databases reveal the potential benefits of using the method.

Keywords: Data Mining, Association Rules, Post-processing

Palavras-chave

1. Mineração de Dados
2. Regras de Associação
3. Pós-processamento

Glossário

MD	:	Mineração de Dados
KDD	:	Knowledge Discovery in Database
RA	:	Regra de Associação
DW	:	Data Warehousing

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Com o avanço na área da tecnologia da informação, a maioria das transações efetuadas pelas organizações, mesmo as mais simples, como uma ligação telefônica ou o uso de um cartão de crédito, é armazenada em meio digital. Lojas de vendas a varejo (como as lojas de departamento), por exemplo, costumam coletar e armazenar os dados relativos a cada compra efetuada por seus clientes. Não é raro encontrar empresas desse tipo executando milhões de operações diariamente. Como conseqüência, o número e o tamanho dos bancos de dados no mundo cresce a cada dia. Estima-se que a quantidade de informação armazenada no mundo dobre a cada dois anos.

Esta aparente facilidade de se capturar o "know-how" das empresas torna-se um desafio quando se quer tirar proveito dessa grande massa de dados. Métodos tradicionais de análise manual e interpretação, como consultas SQL e planilhas de cálculo tornam-se inviáveis à medida que o volume e a complexidade das bases de dados crescem dramaticamente.

Desta forma, surgiu a necessidade de teorias e ferramentas que auxiliassem os humanos a extrair informação útil (conhecimento) de bases de dados de grandes dimensões, o que deu origem à área de pesquisa da descoberta de conhecimento a partir de bases de dados (KDD - Knowledge Discovery in Databases) ou Mineração de Dados [3]. O processo KDD é o processo global, não trivial, de identificação de padrões válidos, novos, potencialmente úteis, e compreensíveis dos dados [4]. Este processo de natureza interativa e iterativa consiste em transformar uma grande massa de dados, muito volumosa para ser compreendida e digerida facilmente, em outra forma que seja mais compacta, mais abstrata, ou mais útil para o usuário [1].

O processo KDD envolve uma seqüência de passos que tem a aplicação dos algoritmos de Mineração de Dados como uma etapa particular do processo. Os algoritmos de mineração se dividem entre as abordagens preditivas e as descritivas. A primeira abordagem encontra pa-

drões para predizer o comportamento futuro de novas intâncias, enquanto a segunda abordagem procura por padrões que descrevam os dados de uma forma compreensível para o ser humano [1].

Dentre as abordagens descritivas, a técnica de Regras de Associação, introduzida em [5], tem despertado grande interesse por pesquisadores e tem sido bastante utilizada em aplicações comerciais. Uma regra de associação é uma implicação da forma A $\Rightarrow$ B que descreve uma correlação entre os eventos A e B, para A e B conjuntos disjuntos de itens ou atributos de dados. O problema da mineração de regras de associação consiste em gerar todos os padrões de relacionamentos entre itens ou atributos que ocorrem com uma determinada freqüência em uma base de dados. A sua popularidade é devida, em grande parte, à forma bastante simples, fácil de ser compreendida, como as regras de associação são representadas.

1.1 Motivação e Objetivo

Ao mesmo tempo que a técnica de Regras de Associação é interessante porque gera todos os padrões intrísecos ao conjunto de dados, esta característica é também a razão da grande quantidade de dados gerada no formato de regras, dificultando consideravelmente sua análise pelo usuário. Na prática, o número de regras produzidas pelos algoritmos pode ser tão elevado, que se fica perante um novo problema de mineração: extrair conhecimento das regras de associação produzidas [6].

Diante desse contexto, diversas pesquisas têm sido realizadas com o objetivo de facilitar a análise e interpretação das regras mineradas, como o desenvolvimento de métodos de pós-processamento.

Dentre as técnicas de pós-processamento encontram-se as abordagens de poda de regras de associação, que têm como finalidade eliminar do conjunto de regras mineradas as regras não interessantes e/ou redundantes para o usuário. Os maiores esforços concentram-se nas abordagens de poda sintática de regras de associação, baseadas exclusivamente na estrutura das regras e nos dados utilizados no seu processo de extração.

Enquanto as abordagens de poda sintática são necessárias para eliminar boa parte das anomalias inerentes ao conjunto de regras mineradas, devido a sua natureza exclusivamente estatística, essas técnicas não são precisas, ou seja, podem eliminar regras que não deveriam, e deixar regras que não são de interesse do usuário. Isso porque a medida de interesse de uma regra é dependente do usuário e do domínio da aplicação - uma regra considerada interessante para um usuário, pode ser irrelevante para outro.

As abordagens semânticas de poda levam em consideração o conhecimento prévio do domínio na avaliação das regras de associação, o que permite um corte, geralmente em menor quantidade de regras, porém de maior qualidade (mais preciso).

Neste trabalho, investigamos o uso do conhecimento prévio do domínio, representado pelas relações de dependência entre os atributos, como fator de eliminação de regras. As relações de dependência entre os atributos são relações de dependência existentes entre objetos do domínio da aplicação, e introduzidas nos dados pelo engenheiro de conhecimento.

No processo de descoberta de conhecimento, durante a fase do pré-processamento, o engenheiro de conhecimento pode, além de eliminar a inconsistência dos dados, enriquecê-los. O enriquecimento dos dados envolve a criação de novos atributos como uma função dos dados existentes, com o objetivo de incorporar características importantes sobre o domínio do problema. Esta atividade torna-se necessária, uma vez que os dados como estão armazenados podem não apresentar todas as representações desejáveis.

Geralmente, os bancos de dados não apresentam redundância. Porém, quando utilizamos esses dados para análise, pode ser útil ter diferentes visões da mesma informação [7].

Por exemplo, no domínio da medicina, a medida IMC (Índice de Massa Corporal), calculada como peso/altura^2, é bastante utilizada por médicos e pesquisadores, por estar altamente relacionada com o risco de doenças, como a diabetes e doenças cardíacas. Um processo de extração de conhecimento sobre uma base de dados nesse domínio certamente levaria em consideração essa informação, e, na sua ausência, o engenheiro de conhecimento criaria o atributo IMC a partir dos atributos originais Peso e Altura.

Em outro exemplo, considerando uma base de dados que representa objetos físicos do mundo real, atributos derivados como Área, Volume, Superfície, etc, podem ser tão informativos quanto as dimensões primárias dos objetos.

O conhecimento das relações de dependência entre os atributos pode revelar a redundância nas regras de associação mineradas, como será demonstrado no cenário ilustrativo apresentado na próxima seção.

1.1.1 Cenário Ilustrativo

Considere um banco de dados com milhões de registros contendo informações sobre hábitos de compra de clientes, e essa base contém inicialmente os seguintes atributos: **Peso**, **Altura** e **Item comprado**. O analista está tentando obter padrões interessantes a partir do comporta-

mento de compra de seus clientes. Para tanto, ele minera regras de associação que possuem no conseqüente o atributo **Item comprado**.

Na preparação dos dados, foi incluído o atributo **IMC**, calculado a partir dos atributos originais **Peso** e **Altura**. Adicionalmente foi incluído o atributo **Status** que indica a condição de uma pessoa dado seu IMC, tal que "abaixo do peso" ($IMC \leq 19$), "no peso normal" ($20 \leq IMC \leq 24$) e "acima do peso" ($IMC \geq 25$). O analista sabe que esses indicadores estão associados ao consumo de produtos do gênero "diet e light", e cria os novos atributos para obter padrões que incluam esse tipo de informação.

A base de dados descrita é representada pela tabela 1.1.

ID	Peso	Altura	IMC	Status	Item comprado
1	84	1,77	27	acima do peso	lasanha
2	83	1,86	24	no peso normal	coca
⋮	⋮	⋮	⋮	⋮	⋮
15.966	79	1.72	27	acima do peso	legumes
15.967	90	1.82	27	acima do peso	coca
15.968	81	1.72	27	acima do peso	coca
15.969	79	1.69	28	acima do peso	chocolate
15.970	58	1.68	21	acima do peso	fanta
⋮	⋮	⋮	⋮	⋮	⋮

Tabela 1.1: Exemplo de uma base de dados sobre hábitos de compras de clientes.

O analista espera obter como resposta regras de comportamento associado a um IMC específico ou a faixas. Entretanto, há casos onde avaliando os resultados ele nota que regras com **IMC** e com **Status** tornam-se redundantes.

Considere que na população em questão, existam quatro valores de IMC mapeados para o valor "acima do peso" do atributo **Status**: 26, 27, 28 e 29; e quatro valores de IMC mapeados para o valor "no peso normal" do mesmo atributo: 21, 22, 23 e 24.

Considere que as seguintes regras foram mineradas:

R1: (IMC=21) $\wedge$ (Status="no peso normal") $\Rightarrow$ (Item=coca) (16% Sup, 67% Conf)

R2: (IMC=21) $\Rightarrow$ (Item=coca) (16% Sup, 67% Conf)

Onde "Sup" (Suporte) e "Conf" (Confiança) representam a força da regra. Os indicadores Suporte e Confiança serão vistos em detalhes no Capítulo 2.

A regra R1 possui a seguinte interpretação: os indivíduos que estão no peso normal e possuem o IMC igual a 21 compram coca. A informação sobre o status do indivíduo é redundante

com a informação relativa ao IMC - todos os indivíduos de IMC igual a 21 estão no peso normal. Desta forma, o analista opta por eliminar a regra R1 e ficar com a regra R2, que é mais simples, e equivalente à regra R1.

Agora, considere outro subconjunto das regras mineradas:

R3: (Status="no peso normal") $\Rightarrow$ (Item=fanta) (60% Sup, 75% Conf)

R4: (IMC=21) $\Rightarrow$ (Item=fanta) (20% Sup, 63% Conf)

R5: (IMC=22) $\Rightarrow$ (Item=fanta) (20% Sup, 63% Conf)

R6: (IMC=23) $\Rightarrow$ (Item=fanta) (20% Sup, 63% Conf)

R7: (IMC=24) $\Rightarrow$ (Item=fanta) (16% Sup, 67% Conf)

Existem regras associando o item "fanta" com todos os IMCs que mapeiam para o valor "no peso normal" do atributo Status (R4 a R7); e o valor "no peso normal" também está associado ao item "fanta" (R3). À luz da relação de dependência entre os atributos **IMC** e **Status**, fica clara a redundância das regras acima. Esta condição reflete o cenário em que a população está equilibradamente contribuindo para a regra de associação mais geral (R3). Intuitivamente o analista sente que deveria ficar com a regra mais geral e descartar as regras relativas ao IMC, por meio de um processo de generalização. Através desse processo, o analista reduziu a associação descoberta ao seu aspecto mais relevante: os indivíduos no peso normal, de uma maneira geral, bebem fanta.

Um outro cenário levaria a uma decisão oposta. Abaixo nós focamos em outro subconjunto das regras mineradas:

R8: (Status="acima do peso") $\Rightarrow$ (Item=coca diet) (60% Sup, 60% Conf)

R9: (IMC=26) $\Rightarrow$ (Item=coca diet) (60% Sup, 100% Conf)

R10: (IMC=27) $\Rightarrow$ (Item=pepsi) (12% Sup, 100% Conf)

R11: (IMC=28) $\Rightarrow$ (Item=pepsi) (12% Sup, 100% Conf)

R12: (IMC=29) $\Rightarrow$ (Item=pepsi) (16% Sup, 100% Conf)

Neste caso, existe uma associação do item "coca diet" com o IMC 26, e com o status "acima do peso". Todos os demais IMCs que mapeiam para o status "acima do peso" aparecem associados ao item "pepsi". Esta segunda condição reflete o cenário em que há um desbalanceamento da população. Mais uma vez, à luz das relações de dependência entre os atributos **Status** e **IMC**, fica clara a redundância das regras R8 e R9. Certamente é o caso em que a regra mais

geral (R8) incorporou o suporte e a confiança da regra mais específica (R9). Intuitivamente o analista sente que deveria ficar com a regra mais específica e descartar a regra relativa a status, por meio de um processo de especialização. Através desse processo, o analista reduziu a associação descoberta ao seu aspecto mais relevante: dos indivíduos acima do peso, somente aqueles de IMC igual a 26 bebem coca diet.

O processo humano de abstração dos resultados envolve a seleção, guiada pelo conhecimento do domínio, das regras de maior interesse pelo usuário. No exemplo apresentado, seis das doze regras analisadas foram descartadas por serem consideradas redundantes ou de menor impacto pelo analista. Anand et al. [8] discutem o papel do usuário no processo de mineração dos dados. Segundo os autores, um processo ideal seria aquele que retornasse "algo que fosse interessante e útil para o usuário". Contudo, esse processo é inviável a medida que a quantidade de conhecimento que pode ser descoberto muitas vezes excede a quantidade de dados. Desta forma, a participação do usuário torna-se essencial no processo da extração de conhecimento. Os autores argumentam que se o usuário possui um conhecimento prévio do domínio o sistema deveria ser capaz de utilizá-lo.

Este estudo tem como objetivo aplicar o conhecimento prévio das relações de dependência entre os atributos, para eliminar regras que são redundantes ou que possuem significado menos relevante ou de menor impacto quando comparadas às outras.

1.2 Metodologia

Nesta dissertação é proposto um método de poda semântica de regras, denominado DMcut, no qual as relações de dependência entre os atributos são utilizadas como guia na redução do conjunto de regras mineradas. Para a apresentação do método proposto é utilizado um formalismo baseado na lógica de predicados. O processo de poda das regras é descrito em detalhes através de um pseudo-algoritmo.

Com o objetivo de validar o método proposto, são descritos os resultados obtidos através da implementação e teste do mesmo sobre bases de dados reais de diferentes áreas, como Medicina, Ciências Sociais, e Marketing e Venda. As regras de associação utilizadas como entrada para o método de poda, foram geradas considerando-se possíveis domínios de aplicação das bases de dados selecionadas.

Para evidenciar a contribuição específica da abordagem proposta, realizamos uma comparação dos resultados obtidos com abordagens sintáticas de poda de regras, neste caso representadas pelas medidas de interesse objetivas, devido a popularidade dessas medidas junto ao meio

acadêmico e comercial. Neste estudo, consideramos como critério de desempenho dos métodos de poda de regras de associação a quantidade (número de regras eliminadas) e a qualidade (precisão) do corte.

1.3 Mapa da Tese

Esta dissertação é composta pelos seguintes capítulos:

- Capítulo 1: Introdução. Neste capítulo foi apresentado o tema central da dissertação.
- Capítulo 2: Mineração de Dados e Regras de Associação. Este capítulo fornece uma visão geral da Mineração de Dados e da técnica de Regras de Associação.
- Capítulo 3: Trabalhos Correlatos. Este capítulo apresenta os principais trabalhos relacionados existentes na literatura.
- Capítulo 4: O Método Proposto. Neste capítulo apresenta-se o método proposto, e o algoritmo para eliminar as regras de associação redundantes e/ou irrelevantes para o usuário, com base nas relações de dependência entre os atributos.
- Capítulo 5: Experimentos Realizados. Este capítulo apresenta uma série de experimentos realizados em quatro bases de dados de domínio público.
- Capítulo 6: Análise dos Resultados. Este capítulo é dedicado à analise e discussão dos resultados obtidos com a aplicação do método proposto nas bases selecionadas. Além disso, é apresentado um processo de poda de regras de associação, com a utilização do método proposto em conjunto com as abordagens sintáticas.
- Capítulo 7: Conclusões e Trabalhos Futuros. Finalmente, neste capítulo são apresentadas as conclusões e idéias para futuros trabalhos.

Capítulo 2

Mineração de Dados e Regras de Associação

O objetivo deste capítulo é apresentar uma visão geral do processo de Mineração de Dados e da técnica de Regras de Associação. A seção 2.1 apresenta o processo de Mineração de Dados, sendo abordadas as etapas do processo referentes a Seleção, Pré-processamento, Transformação, Mineração de Dados e Avaliação e Interpretação. Em seguida, a técnica de Mineração de Dados denominada Regras de Associação é apresentada na seção 2.2.

2.1 Mineração de Dados

A extração de conhecimento em bases de dados, geralmente referenciada na literatura como Knowledge Discovery in Databases (KDD), ou Mineração de Dados (MD), é uma área de pesquisa que vem crescendo rapidamente. Os principais fatores que têm contribuído para o desenvolvimento nesta área englobam as mudanças no ambiente de negócio e a evolução da computação. Atualmente, para se tornarem competitivas, as empresas necessitam focar em estratégias como conhecer melhor seus clientes, predizer os planos dos competidores líderes no mercado, e descobrir novos nichos de clientes em potencial. A informatização do dia a dia tem resultado na coleta e armazenamento de todo tipo de dados por bancos, empresas de cartão de crédito, sistemas de reservas, pontos de vendas eletrônicos, etc. Uma viagem turística, por exemplo, pode gerar dados sobre os hábitos do turista, preferências de empresas aéreas, aluguel de carros, cartões de crédito, operadoras de telefonia, etc. Essas informações, originalmente coletadas para fins operacionais, são cada vez mais exploradas pelas organizações para o gerenciamento de seus negócios. Dados coletados em transações de pontos de vendas eletrônicos podem ser utilizados para descobrir relações interessantes entre itens que foram comprados ao

mesmo tempo, permitindo um melhor planejamento de campanhas promocionais. Está claro que implícitas nessa grande massa de dados existem informações valiosas que podem ser utilizadas para melhorar as decisões de negócios, além de outras aplicações. Dessa forma, novas ferramentas de análise e extração de conhecimento devem ser utilizadas nos processos decisórios.

Adicionalmente, o crescimento da realização de *Data Warehousing* (DM) [9] pelas empresas vem impulsionando o desenvolvimento na área. A criação de repositórios de dados, conhecidos como *Data Warehousings*, tem como objetivo armazenar dados limpos, agregados e consolidados que possam ser analisados por ferramentas OLAP (*On-Line Analytical Processing* ou Processamento Analítico *On-Line*). As ferramentas OLAP permitem gerar respostas rápidas a consultas complexas de natureza tipicamente multidimensional. Os *Data Warehousings* fornecem o material básico para a Mineração de Dados: bancos de dados limpos e bem-documentados [10], reduzindo consideravelmente o tempo de realização do processo de MD [11].

A realização de *Data Warehousing* é considerado o primeiro passo em direção a extração de conhecimento em bases de dados. Contudo, por serem orientadas a consultas, ou seja, dirigidas pelos usuários, as ferramentas utilizadas para analisar um *Data Warehousing* podem impedir que padrões escondidos nos dados sejam encontrados de forma "inteligente" [11]. Desta forma, surge a demanda por técnicas de análise dirigidas por computador, que possibilitem a extração automática (ou semi-automática) de novos conhecimentos a partir de um grande repositório de dados [12]. Ferramentas OLAP são direcionadas para simplificar e suportar a análise interativa dos dados, enquanto o objetivo do KDD é automatizar este processo o máximo que for possível [1].

Diante desse contexto, diversos estudos têm sido realizados no campo de pesquisa da Mineração de Dados. Fayyad et. al. [4] definiram o processo de Extração de Conhecimento de Bases de Dados como o processo global, não trivial, de identificação de padrões válidos, novos, potencialmente úteis, e compreensíveis dos dados.

Os **padrões** podem significar construir um modelo dos dados, ou, mais comum, produzir qualquer descrição de alto nível dos mesmos. Para serem **válidos**, os padrões descobertos devem possuir algum grau de certeza, que garantam que os exemplos cobertos e os casos relacionados ao padrão encontrado sejam aceitáveis. Como **novo** entende-se que os padrões devem ser previamente desconhecidos pelo usuário. Os padrões **potencialmente úteis** são aqueles que podem trazer algum benefício. Finalmente, o significado dos padrões deve ser **compreensível** para o usuário, então este deve ser capaz de analisar os padrões imediatamente, ou após um

pós-processamento [1].

O processo KDD é de natureza iterativa, pois pode ser repetido várias vezes objetivando melhores resultados numa próxima iteração, e é considerado de natureza interativa, porque envolve a participação de diversas classes de usuários, dentre as quais estão: os especialistas do domínio, responsáveis por fornecer apoio para a execução do processo; o analista, responsável pela execução do processo; e o usuário final, que utiliza o conhecimento extraído no processo.

2.1.1 O Processo de Mineração de Dados

O processo KDD, esquematizado na figura 2.1, envolve uma seqüencia de passos que tem a aplicação dos algoritmos de mineração de dados como uma etapa particular do processo. Dentre esses passos estão: Seleção, Pré-processamento, Transformação, Mineração de Dados, e Interpretação e Avaliação.

Em [11], os autores consideram uma fase anterior ao processo de Seleção dos dados, denominada Identificação do Problema, que se refere ao conhecimento do domínio e identificação do problema. Nesta fase é realizado o estudo do domínio da aplicação e a definição de objetivos e metas a serem alcançadas no processo KDD. O conhecimento sobre o domínio é utilizado nesta fase inicial para definição das principais metas, objetivos e restrições, além de fornecer um subsídio para todas as etapas do processo de Extração de Conhecimento.

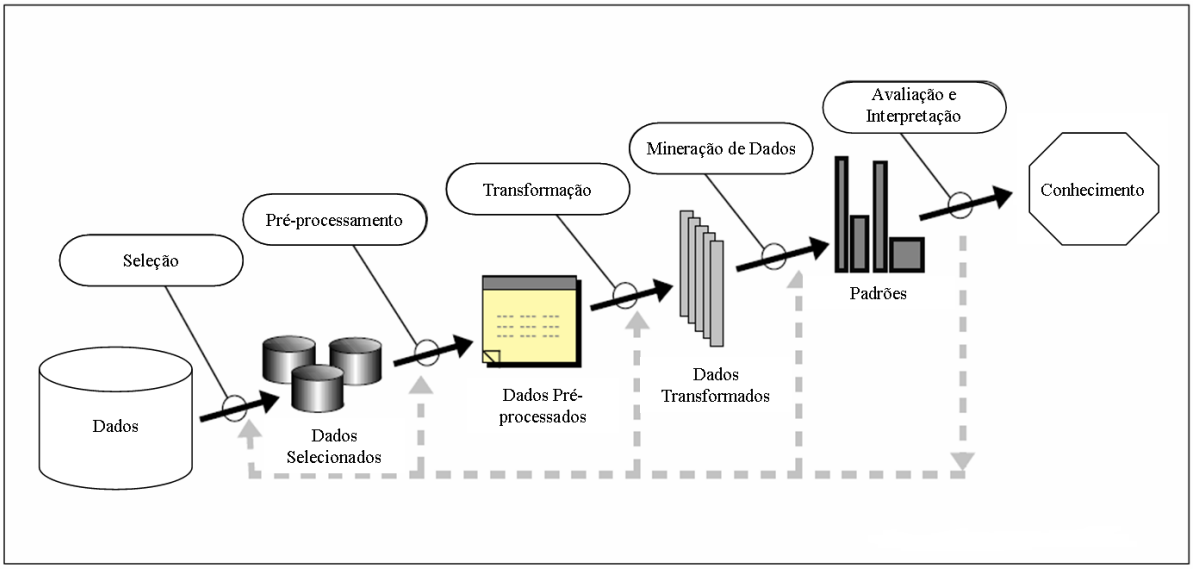


Figura 2.1: Uma visão geral dos passos que compõem o processo KDD, retirado de [1].

Os passos que compõem o processo KDD podem ser descritos da seguinte maneira:

Seleção Nesta etapa, são selecionados o conjunto de dados, e o conjunto de atributos sobre os quais o processo de descoberta vai atuar. Apenas os dados que são relevantes para o foco da aplicação corrente são selecionados. Neste momento, os dados podem estar em diferentes estruturas e formatos, como tabelas relacionais e arquivos, o que dificulta sua obtenção pelo analista.

Pré-processamento Os bancos de dados reais são geralmente incompletos e cheios de ruídos. É nesta etapa que as inconsistências (dados nulos, inválidos, ou repetidos) dos dados são eliminadas.

Transformação Consiste na manipulação dos dados com o objetivo de aumentar sua utilidade para a tarefa de mineração. Algumas transformações comuns que podem ser aplicadas aos dados são: resumo, por exemplo, quando dados sobre vendas são agrupados para formar resumos diários; transformação de tipos; normalização de atributos contínuos, colocando seus valores em intervalos definidos; e a criação de novos atributos a partir dos atributos originais, por exemplo, quando o atributo Volume é criado a partir das dimensões primárias de um objeto.

As etapas de Seleção, Pré-processamento e Transformação são comumente consideradas como uma única fase, referenciada na literatura como Pré-processamento dos dados.

Mineração de Dados Nesta etapa é realizada a escolha, a configuração e execução de um ou mais algoritmos para extração de conhecimento. A escolha da tarefa de Mineração de Dados e do algoritmo a serem empregados é direcionada ao cumprimento dos objetivos definidos na etapa de Identificação do Problema [11].

As tarefas possíveis de um algoritmo de extração de padrões podem ser agrupadas em atividades preditivas e descritivas. A primeira abordagem encontra padrões para predizer o comportamento futuro de novas instâncias, enquanto a segunda abordagem encontra padrões que descrevem os dados de uma forma compreensível para o ser humano [1]. Os dois principais tipos de tarefas para predição são classificação e regressão. A classificação consiste na predição de um valor categórico, por exemplo, predizer se um cliente é bom ou mau pagador. Na regressão, o atributo a ser predito é um valor contínuo, por exemplo, prever a cotação do dólar em determinado mês. Algumas das tarefas de descrição são Regras de Associação, Agrupamentos (*Clustering*) e Sumarização. Essas tarefas são descritas em detalhes na Seção 2.1.2

Uma vez escolhida a tarefa a ser empregada, deve-se definir o algoritmo de extração e a posterior configuração de seus parâmetros. Aspectos que merecem atenção são: a linguagem

de representação dos padrões a serem encontrados, por exemplo, se o objetivo é realizar uma classificação, podem-se utilizar algoritmos indutores de Árvores de Decisão ou regra de produção [11], e a complexidade da solução encontrada pelo algoritmo de extração, que deve ser suficiente para representar o conceito embutido nos dados.

Avaliação e Interpretação Uma vez produzidos os padrões na etapa anterior, os resultados são apresentados para a avaliação e interpretação do usuário. O conhecimento extraído pode ser utilizado na resolução de problemas da vida real, seja por meio de um Sistema Inteligente ou de um ser humano como apoio a algum processo de tomada de decisão [11].

Nesta etapa, possivelmente faz-se necessário retornar aos passos anteriores para a obtenção de resultados melhores numa próxima iteração, caso o conhecimento extraído não seja de interesse do usuário final ou não cumpra com os objetivos propostos.

Freqüentemente os resultados produzidos pelas ferramentas de extração de padrões são muito grandes e/ou muito complexos. Muitos dos padrões gerados podem não ser importantes, relevantes ou interessantes para o usuário. Desta forma, é de vital importância o desenvolvimento de técnicas de apoio no sentido de fornecer aos usuários apenas os padrões interessantes. Essas técnicas de apoio são conhecidas como técnicas de Pós-processamento, e são o objeto de estudo deste trabalho - especificamente, o pós-processamento de Regras de Associação. As técnicas de pós-processamento serão vistas em detalhes no capítulo 3.

2.1.2 Principais Tarefas de Mineração de Dados

Existem diversas tarefas para Mineração de Dados, dentre as quais podemos destacar duas classes de tarefas principais: a predição e a descrição. Como discutido anteriormente, a predição envolve a utilização de algumas variáveis do banco de dados para predizer valores futuros ou desconhecidos de uma variável de interesse. Essas atividades visam principalmente à tomada de decisões. Já as atividades de descrição procuram por padrões que descrevam os dados de uma forma compreensível pelos humanos. Esta tarefa tem como foco o suporte à decisão.

Os objetivos das tarefas de predição e descrição podem ser alcançados utilizando-se uma variedade de métodos de Mineração de Dados. Utilizando-se do mesmo artifício adotado em [1], será apresentado um exemplo simples de forma a tornar os conceitos discutidos nesta seção mais concretos. A figura 2.2 apresenta um conjunto de dados artificial bidimensional contendo 22 casos. Cada ponto do gráfico representa um cliente para o qual foi concedido um empréstimo bancário em determinado tempo passado. O eixo horizontal representa a receita anual do

cliente, enquanto o eixo vertical representa o total de despesas do mesmo. Os dados foram classificados em duas classes: os x's representam os clientes que ficaram em débito com seus empréstimos, e os o's representam os clientes que honraram com seus empréstimos.

A tarefa de *classificação* consiste em aprender uma função que mapeia um conjunto de dados de entrada, em uma classe de saída, dentre um conjunto de classes predefinido. No exemplo da figura 2.2, o gráfico apresenta uma divisão simples dos dados em duas classes de regiões. Futuramente, o banco pode querer utilizar essa classificação para automaticamente decidir se deve conceder empréstimos a clientes.

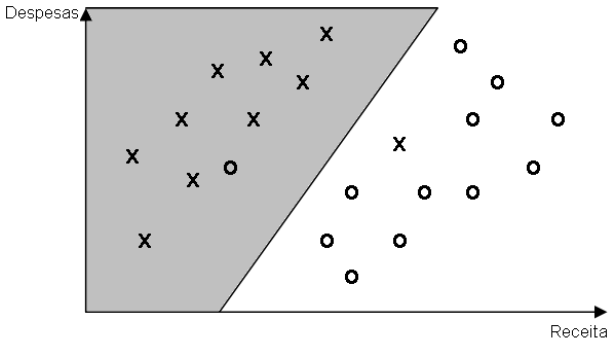


Figura 2.2: Uma classificação linear simples para o conjunto de dados dos empréstimos bancários. A região sombreada representa a classe *não conceder o empréstimo*.

A tarefa de *regressão* é conceitualmente similar à de classificação. A diferença está no atributo a ser predito que neste caso assume valores contínuos. O gráfico da figura da 2.3 apresenta o resultado de uma regressão linear simples, no qual o total de despesas é modelado como uma função linear da receita anual de um cliente.

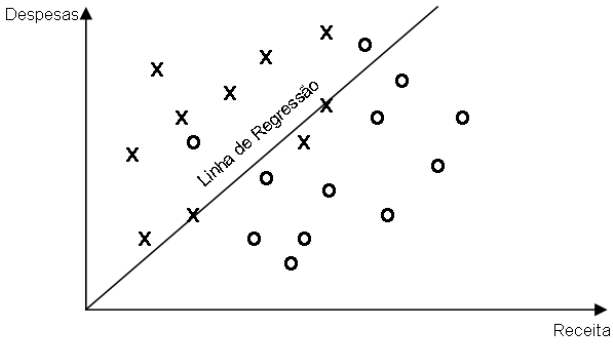


Figura 2.3: Uma regressão linear simples para o conjunto de dados dos empréstimos bancários.

A tarefa de Regras de Associação consiste na identificação de padrões intrínsecos ao con-

junto de dados. De forma geral, uma regra de associação caracteriza o quanto a presença de um conjunto de atributos implica na presença de um outro conjunto disjunto de atributos da mesma base de dados [13]. Uma regra de associação é uma implicação da forma LHS $\Rightarrow$ RHS, onde LHS e RHS são, respectivamente, o lado esquerdo (*Left Hand Side*) e o lado direito (*Right Hand Side*) da regra, definidos por conjuntos disjuntos de atributos de dados.

Nos dados sobre os empréstimos bancários, um exemplo prático de uma regra de associação seria: Clientes que possuem a receita anual superior a 30 salários mínimos e as despesas anuais inferiores a 25 salários mínimos costumam honrar com os empréstimos bancários.

A técnica de Regras de Associação será vista em detalhes na seção 2.2.

Clustering (ou Agrupamento) é uma tarefa descritiva que procura encontrar um conjunto finito de categorias ou agrupamentos que descrevem os dados. Os agrupamentos são conjuntos homogêneos de objetos, cujos elementos são semelhantes entre si, e diferentes em relação aos objetos de outros grupos. Os agrupamentos podem ser mutuamente exclusivos e exaustivos ou podem consistir de uma representação mais rica, como hierarquias e categorias sobrepostas [1]. A figura 2.4 exibe um possível agrupamento dos dados sobre os empréstimos bancários em três categorias. As classes originais (representadas por x's e o's nas figuras anteriores) foram substituídas pelo sinal + para indicar que a pertinência de classes não é considerada neste caso.

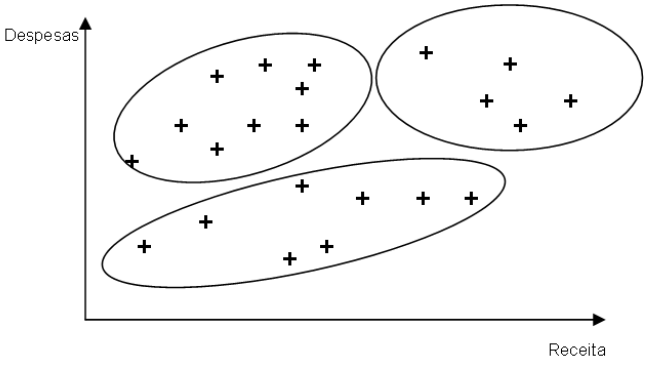


Figura 2.4: Um agrupamento simples do conjunto de dados dos empréstimos bancários.

A *sumarização* envolve métodos para encontrar uma descrição compacta para um subconjunto de dados. Um exemplo simples de sumarização é a definição da média e do desvio padrão de todos os campos. Os métodos mais sofisticados envolvem a derivação de regras resumidas e técnicas de visualização [1].

Dentre as tarefas para Extração de Conhecimento em Bases de Dados, a técnica de Regras de Associação é uma das mais utilizadas, e está presente em um grande número de sistemas de Mineração de Dados. A técnica foi inicialmente proposta por Agrawal, Imielinsky e Swami em

Maio de 1993 [5]. A sua popularidade é devida, em grande parte, à forma bastante simples, fácil de ser compreendida, como as regras de associação são representadas.

2.2 Mineração de Regras de Associação

A técnica de Regras de Associação permite encontrar padrões de relacionamentos entre itens de dados que ocorrem com determinada freqüência em uma base de dados transacional (unidimensional), relacional (multidimensional), ou ainda em outros tipos de repositórios de dados. Uma regra de associação é uma implicação da forma A $\Rightarrow$ B, para A e B conjuntos disjuntos de itens de dados. Denomina-se o conjunto A de **antecedente**, ou de LHS (*Lef Hand Side*), e o conjunto B de **conseqüente** da regra, ou de RHS (*Right Hand Side*).

Tradicionalmente, os indicadores Suporte e Confiança são utilizados para medir a importância de uma regra. O **Suporte** corresponde à freqüência com que A e B ocorrem em toda a base. Já a **Confiança** é dada pela freqüência com que B ocorre, dentre as instâncias que contêm A. O Suporte é interpretado como uma medida da significância estatística de uma regra, enquanto a Confiança pode ser interpretada como uma medida da força da regra.

Uma das aplicações típicas da técnica de Regras de Associação é a análise de transações de compras (*Market Basket Analysis*). As transações consistem de itens comprados juntos por um determinado cliente em determinada compra, ou em determinado período. Regras do tipo: “75% dos clientes que compram sardinha em lata também compram milho em conserva”, poderiam auxiliar o gerente de um supermercado a tomar decisões do tipo: o que colocar à venda, como projetar encartes e layout de prateleiras, etc.

Outro exemplo onde a regra de associação pode ser utilizada, é o caso de uma base de dados escolar, relacionando alunos e disciplinas. Um exemplo prático de uma regra de associação neste domínio seria: “70% dos alunos inscritos em Sistemas Operacionais também estão inscritos em Estruturas de Dados”. Essa afirmação pode ser utilizada pelo diretor do curso para alocar recursos como salas de aula e professores.

Agrawal et al. [5] propuseram um modelo matemático, onde as regras de associação devem atender a valores mínimos de Suporte e Confiança especificados pelo usuário.

2.2.1 Declaração Formal do Problema

As regras de associação podem ser formalmente definidas da seguinte forma (adaptado de [5]):

Seja I = {i_1, i_2,..., i_n} um conjunto de itens que podem assumir valores binários 0 ou 1 (falso ou verdadeiro), conforme representem a presença ou não de um item em particular. Seja D um conjunto de transações, em que cada transação T corresponde a um conjunto de itens tal que T $\in$ I. Considera-se ainda que um conjunto de itens A está contido numa transação T, se todos os itens do conjunto têm valor "verdadeiro" na transação, ou seja, fazem parte dessa mesma transação. Uma regra de associação R pode ser representada por uma expressão da forma: A $\Rightarrow$ B, onde A $\in$ I, B $\in$ I e A $\cap$ B = $\emptyset$. Uma regra de associação A $\Rightarrow$ B possui suporte s se s% das transações em D satisfazem A e B, e confiança c, se c% das transações em D que contêm A também contêm B.

Um exemplo de uma regra de associação é uma expressão do tipo: "Dentre os clientes que possuem casa própria e estão empregados, 90% costumam ter o empréstimo aprovado". Esta regra de associação é representada como:

(Casa-própria=sim) $\wedge$ (Emprego=sim) $\Rightarrow$ (Empréstimo=aprovado)

onde 90% representa a Confiança dessa regra.

2.2.2 Extração de Regras

Existe uma variedade de algoritmos de extração de regras de associação. O primeiro algoritmo proposto, denominado *Apriori*, divide o problema da extração de regras em duas fases [5]. Na primeira fase são gerados todos os conjuntos de itens, chamados conjuntos freqüentes, que possuem o Suporte maior ou igual ao Suporte mínimo especificado. A segunda fase consiste em gerar todas as regras de associação que possuam Confiança maior ou igual a Confiança mínima especificada.

Uma vez gerados os conjuntos freqüentes, a solução para o segundo problema é trivial. A fase de extração dos conjuntos freqüentes exige grande esforço computacional, devido ao número exponencial de possíveis conjuntos freqüentes ($2^{|I|}$). Várias estratégias para extração de conjuntos freqüentes têm sido propostas e aprimoradas na literatura. Exemplos podem ser encontrados em [13, 14, 15].

Quase todos os algoritmos de regras de associação consistem nessa estratégia de duas fases [16]. Contudo, existe um algoritmo bem sucedido, denominado *MagnumOpus*, que utiliza uma estratégia diferente para gerar imediatamente um grande conjunto de todas as regras de associação. Nesta dissertação, além do algoritmo *Apriori*, não serão abordados os demais algoritmos de extração de regras, uma vez que este estudo tem o foco no pós-processamento das mesmas, não importando a forma como as regras foram geradas.

No esquema seguinte (Figura 2.5) está representado o processo de descoberta de regras de associação pelo algoritmo *Apriori*, a partir de uma base de dados de transações de compra que obedecem a um suporte mínimo de 50%.

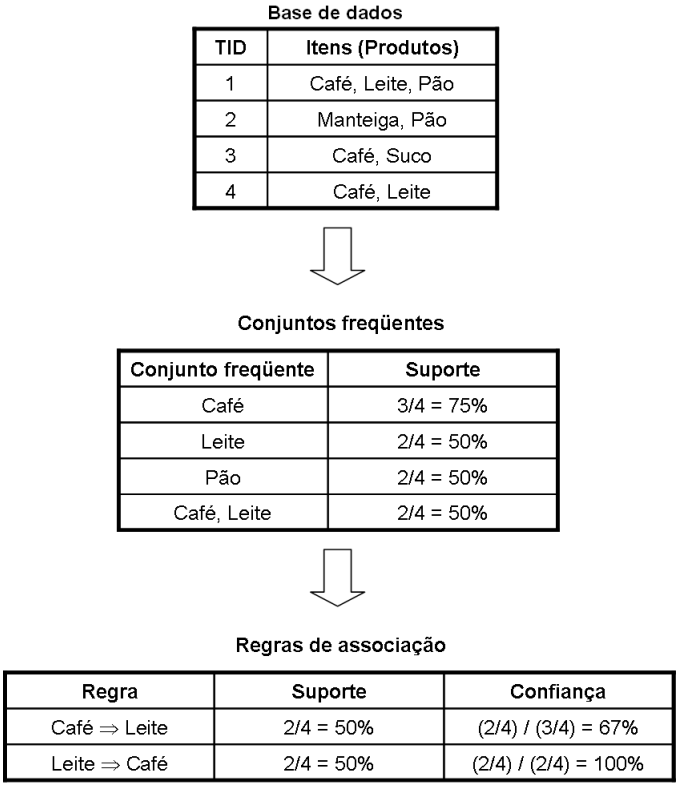


Figura 2.5: Etapas da geração de regras pelo algoritmo *Apriori*.

A primeira tabela da figura 2.5 representa a base de dados de transações de compras. A primeira coluna da tabela apresenta o número de identificação da transação (TID), enquanto a segunda coluna contém os produtos comprados em cada transação. A primeira fase deste algoritmo (transição da tabela "Base de dados" para a tabela "Conjuntos freqüentes") determina e seleciona as associações de itens que ocorrem com maior freqüência nas transações. Observe que neste ponto do algoritmo o produto "Manteiga" não obteve o valor de Suporte mínimo e não aparece nos conjuntos freqüentes da segunda tabela (Suporte("Manteiga") = 1/4 = 25%). Em seguida, com base nos conjuntos freqüentes, geram-se todas as regras de associação que obedeçam aos valores mínimos de Suporte e de Confiança (transição da tabela "Conjuntos freqüentes" para a tabela "Regras de associação").

Normalmente trabalha-se com valores baixos de Suporte e valores elevados de Confiança. O Suporte mínimo é exigido de forma a garantir a significância estatística dos padrões gerados, além de restringir a quantidade de regras na saída do algoritmo de mineração. Valores de Confiança altos são importantes para garantir uma elevada coesão entre os os itens analisados. Uma

Confiança baixa não refletiria qualquer padrão de comportamento.

As figuras 2.6 e 2.7 exibem a relação entre os valores mínimos de Suporte e Confiança, respectivamente, considerados pelo algoritmo *Apriori* na mineração de regras a partir da base *Weather*[1]. Estabeleceu-se o número máximo de 1000 regras para ambas as medidas.

A figura 2.6 mostra como o número de regras produzidas pode variar à medida que se aumenta o valor mínimo do Suporte. O gráfico estabiliza para um valor de Suporte a partir de 0,3. Na figura 2.7 pode-se observar que o aumento do valor mínimo da Confiança também é repercutido no número de regras produzidas, embora esse efeito comece a ser sentido para valores acima de 0,5. Pequenas variações no valor dessas medidas podem resultar em variações expressivas no número de regras produzidas.

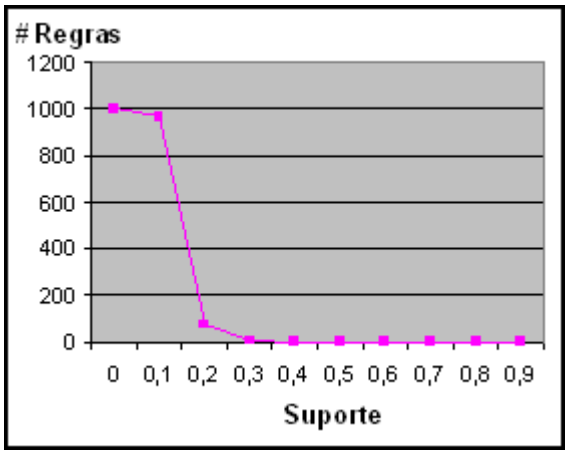


Figura 2.6: Relação entre o número de regras mineradas e diferentes valores de Suporte.

Na mineração de regras de associação baseada no modelo Suporte/Confiança, como é conhecido na literatura, uma regra é considerada forte, ou seja, contém informação interessante, se possui valores elevados de Suporte e Confiança. Assim, o Suporte e a Confiança atuam como medidas de interesse no processo de extração das regras de associação.

Contudo, sabe-se que, na prática, o modelo Suporte/Confiança pode produzir milhares, e até centenas de milhares de regras, dificultando a sua avaliação pelo usuário. Além disso, grande parte dos resultados produzidos contém regras óbvias, redundantes, ou até mesmo contraditórias [17, 18].

Diante desse contexto, diversos autores pesquisaram mudanças no modelo Suporte/Confiança para lidar com esse problema. No próximo capítulo vamos apresentar as principais contribui-

[1] *Weather*: arquivo de dados contendo informações para decidir quando praticar determinado esporte dadas as condições do tempo. Freqüentemente utilizado pela comunidade de aprendizagem de máquina para análise empírica de seus algoritmos.

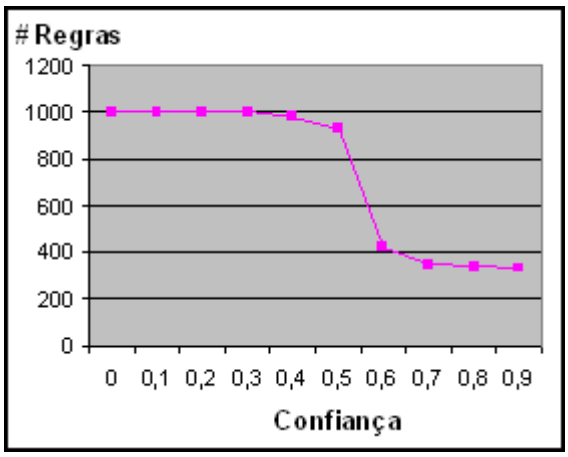


Figura 2.7: Relação entre o número de regras mineradas e diferentes valores de Confiança.

ções nessa área.

Capítulo 3

Trabalhos Correlatos

Como discutido nos capítulos anteriores, a tarefa de Regras de Associação freqüentemente gera uma grande quantidade de regras, muitas das quais podem não ser importantes, relevantes ou interessantes para o usuário. Geralmente o usuário está a procura de um pequeno conjunto de padrões interessantes. Assim, o desenvolvimento de métodos de apoio no sentido de fornecer aos usuários apenas os padrões mais interessantes torna-se imprescindível para a aplicação prática da técnica.

Diversos métodos para melhorar a análise e a compreensão das regras de associação têm sido pesquisados na literatura. Podemos dividir esses métodos em duas linhas de pesquisa principais. A primeira utiliza técnicas de pós-processamento dos resultados produzidos pelo modelo Suporte/Confiança introduzido no capítulo anterior. Já a outra linha procura utilizar um critério mais rico e discriminativo na mineração das regras de associação.

Seja na integração de um critério considerado mais interessante no processo de extração das regras, ou na transformação dos resultados produzidos pelo modelo básico na fase do pós-processamento dos resultados, todas essas tentativas têm em comum um único objetivo: elevar o nível do conhecimento extraído. Neste capítulo são apresentadas as principais contribuições na área.

3.1 Mineração de Regras de Associação não Redundantes

Freqüentemente, o número de regras produzidas pelo modelo tradicional de mineração de regras de associação pode crescer rapidamente, especialmente quando valores baixos de freqüência são exigidos. Segundo Zaki [19], dentre as regras mineradas, o número de regras redundantes é

maior do que se acreditava - mais precisamente, este valor é exponencial ao tamanho do maior conjunto freqüente.

Como veremos no restante deste capítulo, o termo “regra de associação redundante” possui definições variadas na comunidade de Mineração de Dados. Contudo, existe um consenso geral (com o qual concordamos) que diz que uma regra de associação R’ é redundante, se pode ser deduzida a partir de outra regra R presente no conjunto de regras e, por algum motivo, R’ é menos relevante do que R.

Para Zaki [19], as regras mais relevantes são as regras mais gerais. Uma regra R1 é mais geral do que uma regra R2, se R2 pode ser gerada adicionando-se itens ao antecedente ou ao conseqüente de R1. A regra R2 é considerada redundante se Suporte(R1) = Suporte(R2) e Confiança(R1) = Confiança(R2).

O autor apresenta uma nova abordagem para minerar regras de associação não redundantes, baseada no conceito de conjuntos freqüentes fechados (*closed frequent itemsets*). Um conjunto freqüente fechado é um subconjunto do conjunto freqüente correspondente. Este subconjunto é necessário e suficiente para capturar todas as informações sobre o conjunto freqüente. Os conjuntos freqüentes fechados são o menor subconjunto representativo dos conjuntos freqüentes sem que haja perda de informação.

O algoritmo considera apenas regras entre os conjuntos freqüentes fechados adjacentes. A partir do conjunto de regras gerado, pode-se inferir todas as regras de asociação possíveis através de operações como transitividade e incremento. Assim, apenas um pequeno subconjunto das regras, fácil de se compreender, é apresentado para o usuário, que pode seletivamente derivar outras regras de interesse.

Bastide et al. [20] também se basearam no conceito de conjuntos freqüentes fechados para minerar regras de associação não redundantes. Neste trabalho, os autores consideram uma regra R1: X1 $\Rightarrow$ Y1 redundante se existe uma regra R2: X2 $\Rightarrow$ Y2, com Suporte(R1) = Suporte(R2) e Confiança(R1) = Confiança(R2), tal que X2 $\subseteq$ X1 e Y1 $\subseteq$ Y2. Desta forma, os autores propõem um algoritmo para extrair somente as regras que possuem o antecedente mínimo e o conseqüente máximo, chamadas de *minimal non-redundant association rules*, que são consideradas as mais úteis e relevantes.

Um estudo preliminar sobre a utilização de conjuntos freqüentes fechados na mineração de regras pode ser encontrado em [21].

3.2 Métodos de Pós-processamento

O pós-processamento é uma fase importante no processo de descoberta de conhecimento. Nesta estapa, os resultados produzidos pelas ferramentas de mineração são manipulados com o objetivo de melhorar a qualidade do conhecimento descoberto. De acordo com [22], a fase do pós-processamento consiste dos seguintes passos:

- Poda: elimina regras redundantes e/ou não interessantes;
- Resumo: cria um resumo das regras de associação descobertas, destacando as regras mais gerais ou mais abstratas;
- Agrupamento: agrupa as regras de acordo com determinada característica, que pode ser semântica como a relevância econômica, a distribuição no tempo, etc;
- Visualização: esta abordagem consiste no desenvolvimento de técnicas de visualização que auxiliem a análise das regras potencialmente interessantes;

A seguir serão descritos em detalhes esses passos.

3.2.1 Poda das Regras Redundantes e/ou não Interessantes

Esta fase tem como objetivo eliminar as regras que apresentam significado redundante e/ou não interessante para o usuário.

3.2.1.1 Medidas de Interesse

As medidas de interesse desempenham um papel fundamental no processo de avaliação das regras de associação, indicando quais regras devem ser descartadas, ou quais devem ser mais bem exploradas. Estas medidas podem ser divididas em duas categorias: subjetivas e objetivas.

As medidas de interesse subjetivas são aquelas que levam em consideração a opinião do usuário para avaliar a força de uma regra. Em [23], Silberschatz e Tuzhilin apresentam os dois principais fatores a serem considerados para que uma regra de associação seja subjetivamente interessante: “inesperabilidade” e utilidade. Segundo a medida de interesse inesperabilidade, uma regra é considerada interessante se contém uma informação que contradiz a expectativa do usuário, o que depende de suas convicções. Já a medida utilidade considera uma regra interessante se o usuário pode fazer algo útil a partir dela, ou seja, se é possível tirar proveito da regra.

As medidas de interesse objetivas medem a força de uma regra utilizando-se exclusivamente da estrutura das regras e dos dados utilizados no seu processo de extração. Geralmente, o emprego das medidas de interesse objetivas consiste na definição de um valor de corte no início do processo de mineração das regras. As regras que estiverem abaixo desse limiar não são consideradas interessantes para o usuário.

Estas medidas foram desenvolvidas com o objetivo de medir a dependência entre itens de dados, e podem ser utilizadas como forma de filtrar as regras de associação que obedeçam a determinados valores de independência.

Lift ou Interest

Dada uma regra de associação A $\Rightarrow$ B, a medida de interesse Lift [17], também conhecida como Interest, indica o quanto mais freqüente torna-se B quando A ocorre. Esta medida é calculada como o quociente entre a probabilidade conjunta (de A e B) observada, e a probabilidade conjunta sob independência. O valor do Lift para A $\Rightarrow$ B é computado por:

$$\begin{aligned} Lift(A \Rightarrow B) &= Prob(A \cup B)/(Prob(A) \times Prob(B)) \\ &= Suporte(A \cup B)/(Suporte(A) \times Suporte(B)) \end{aligned} \tag{3.1}$$

Observe que a probabilidade conjunta (de A e B) representa o **Suporte real** do conjunto de itens (A $\cup$ B), enquanto a probabilidade conjunta sob independência representa o **Suporte esperado** do mesmo conjunto.

Esta medida varia entre 0 e ∞. Se Lift(A $\Rightarrow$ B) = 1, então A e B são independentes. Para Lift(A $\Rightarrow$ B) > 1, A e B são positivamente dependentes, ou seja, a presença de A aumenta as chances da ocorrência de B. Quando Lift(A $\Rightarrow$ B) < 1, A e B são negativamente dependentes, o que significa que a presença de A diminui as chances da ocorrência de B. A medida tem a seguinte interpretação: quanto maior for o valor do Lift, mais interessante a regra, pois A aumentou (*"lifted"*) B numa maior taxa.

A figura 2.7 apresenta o gráfico da relação entre o número de regras geradas e diferentes valores de Lift, a partir da base *Weather*. Como podemos observar nesse gráfico, o número de regras produzidas varia consideravelmente se houver um acréscimo no valor do Lift considerado. Para esta base, variando-se o valor do Lift de 2,1 para 2,4, o número de regras produzidas é reduzido pela metade.

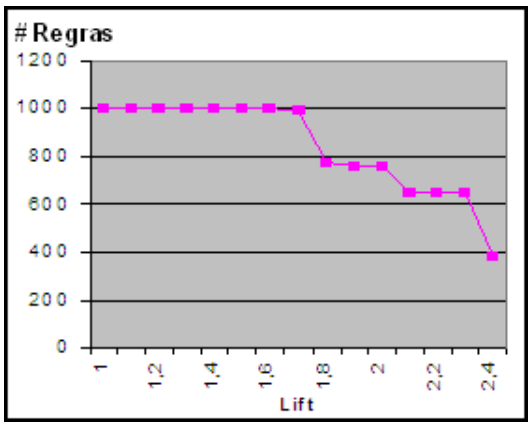


Figura 3.1: Relação entre o número de regras mineradas e diferentes valores de Lift.

Cobertura

Dada uma regra de associação A $\Rightarrow$ B, a medida de interesse Cobertura [24], representa a fração das instâncias cobertas pelo conjunto A. Sua fórmula é dada pelo Suporte do antecedente da regra:

$$Cobertura(A \Rightarrow B) = Suporte(A) \tag{3.2}$$

Esta medida varia entre 0 e 1. A Cobertura pode ser interpretada como medida de generalidade da regra.

Especificidade

Dada uma regra de associação A $\Rightarrow$ B, a medida de interesse Especificidade [24] é dada pela freqüência condicional de que A seja falso dado que B é falso, de acordo com a fórmula:

$$\begin{aligned} Especificidade(A \Rightarrow B) &= f(\overline{A}|\overline{B}) \\ &= \frac{f(\overline{A} \cup \overline{B})}{f(\overline{B})} \end{aligned} \tag{3.3}$$

Onde $f(\overline{A} \cup \overline{B})$ e $f(\overline{B})$ denotam a freqüência relativa associada ao conjunto ($\overline{A} \cup \overline{B}$) e $\overline{B}$, respectivamente. A freqüência de um conjunto $\overline{X}$ é calculada como n($\overline{X}$)/N, onde n($\overline{X}$) denota o número de instâncias nas quais X é falso, e N o número total de instâncias. Esta medida varia entre 0 e 1.

Novidade

A medida Novidade, introduzida em [25], e também conhecida na literatura como *Rule Interest* (RI), PS (letras iniciais do nome do autor), e *leverage*, objetiva identificar o quão inovadora, interessante ou não usual é uma dada regra. Esta medida é calculada como a diferença entre o **Suporte real** e o **Suporte esperado** da regra de associação, de acordo com a fórmula:

$$\begin{aligned} Novidade(A \Rightarrow B) &= Prob(A \cup B) - (Prob(A) \times Prob(B)) \\ &= Suporte(A \cup B) - (Suporte(A) \times Suporte(B)) \end{aligned} \quad (3.4)$$

Observe que o primeiro termo da fórmula é o Suporte da regra, indicando que valores elevados de Suporte apenas são interessantes quando valores de Suporte(A) e/ou Suporte(B) são relativamente pequenos.

Esta medida varia entre -0.25 e 0.25. Se Novidade(A $\Rightarrow$ B) = 0, então A e B são independentes. Para Novidade(A $\Rightarrow$ B) > 0, A e B são positivamente dependentes. Quando Novidade(A $\Rightarrow$ B) < 0, A e B são negativamente dependentes. Quanto maior o valor da medida, mais interessante é a regra.

Pode-se observar, mais uma vez (Figura 3.2), que a definição de um valor mínimo da Novidade tem implicações no número de regras de associação geradas pelos algoritmos.

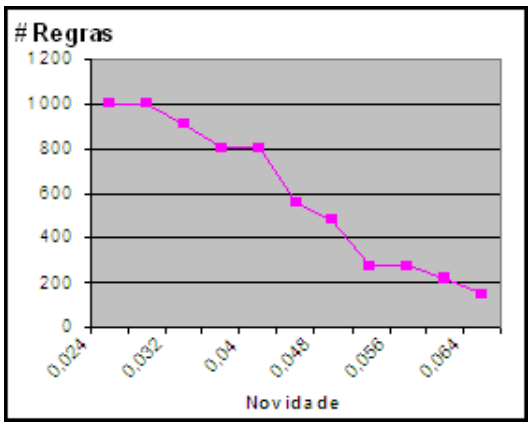


Figura 3.2: Relação entre o número de regras mineradas e diferentes valores de Novidade.

Convicção

Tanto o Lift quanto a Novidade, tal como são definidas, possuem como característica o fato de serem medidas simétricas em relação aos lados da regra. A medida Convicção, definida em

[17], permite avaliar uma regra de associação considerando-se o sentido da implicação, isto é, Convicção(A $\Rightarrow$ B) $\neq$ Convicção(B $\Rightarrow$ A). O valor da Convicção para A $\Rightarrow$ B é calculado por:

$$\text{Convicção}(A \Rightarrow B) = \frac{Suporte(A) \times Suporte(\overline{B})}{Suporte(A \cup \overline{B})} \tag{3.5}$$

O valor da Convicção pode variar entre 0 e ∞, apresentando o valor 1 quando os conjuntos A e B são independentes, e ∞ quando o valor da Confiança for igual a 100%.

Teste do χ^2 (chi-quadrado)

O teste do χ^2 é outra medida estatística utilizada para medir a independência e/ou correlação entre itens. O teste é baseado na comparação entre as freqüências observadas com a freqüência esperada correspondente. Quanto mais próximos forem os valores dessas duas freqüências, maior é a evidência de que os itens analisados são independentes. Desta forma, o teste do χ^2 é utilizado para testar a significância do desvio com relação aos valores esperados.

Seja f_o a freqüência observada, e f a freqüência esperada. O valor do χ^2 é definido como:

$$\chi^2 = \sum \frac{(f_o - f)^2}{f} \tag{3.6}$$

Para uma regra de associação (A $\Rightarrow$ B), a freqüência observada é dada por Suporte(A $\cup$ B), enquanto a freqüência esperada é dada por Suporte(A) $\times$ Suporte(B). Desta forma, o teste do χ^2 para uma regra de associação (A $\Rightarrow$ B) é calculado como:

$$\chi^2(A \Rightarrow B) = \frac{(Suporte(A \cup B) - Suporte(A) \times Suporte(B))^2}{Suporte(A) \times Suporte(B)} \tag{3.7}$$

Se χ^2 = 0, então A e B são independentes. Para χ^2 > 0, A e B são positivamente dependentes. Quando χ^2 < 0, A e B são negativamente dependentes. Quanto maior o valor da medida, mais interessante é a regra.

3.2.1.2 Rule Covers

Toivonen et al. [6] apresentam um método de redução do número de regras através da formação de um subconjunto das regras mineradas, denominado de *rule covers*. Este subconjunto das regras é formado de maneira que, para cada instância no banco de dados, existe uma regra que cobre esta instância no subconjunto somente se existe uma regra que cobre esta mesma instância no conjunto original das regras.

Neste sentido, é proposto um algoritmo para podar o conjunto de regras mineradas, baseado na estruturas das regras (*structural rule covers*). Um conjunto de regras $\Delta \subseteq \Gamma$ é um *structural rule cover* de Γ, se para todas as regras do tipo (X $\Rightarrow$ Y) $\in \Delta$ não existe uma regra do tipo (X' $\Rightarrow$ Y) $\in \Gamma$, tal que X' $\subseteq$ X. Os autores apresentam um exemplo que pode ajudar a compreender essa idéia. Considerando que, a partir de uma base de dados contendo informações sobre os cursos de informática para os quais os alunos se inscreveram em determinado período, e as seguintes regras mineradas:

R1: (Programação em C) $\wedge$ (Base de Dados) $\Rightarrow$ (Comunicação de Dados) (2% Sup, 90% Conf)

R2: (Programação em C) $\wedge$ (Base de Dados) $\wedge$ (Curso de Utilização do Computador) $\Rightarrow$ (Comunicação de Dados) (1% Sup, 90% Conf)

A regra R2 é redundante porque é mais especializada do que a regra R1, e não contém informação adicional com relação à primeira. Considera-se que a primeira regra "cobre" a segunda regra.

3.2.1.3 Métodos que Utilizam Informação de Taxonomia

As taxonomias refletem uma caracterização coletiva ou individual de como os itens podem ser hierarquicamente classificados [26]. Na figura 3.3 é apresentado um pequeno exemplo de uma taxonomia. Nesse exemplo, pode-se verificar que: jaqueta é uma roupa de inverno, calça de esqui é uma roupa de inverno, roupa de inverno é um tipo de roupa, camisa é um tipo de roupa, sapato é um tipo de calçado e bota é um tipo de calçado.

A informação da taxonomia pode ser utilizada para encontrar associações entre itens presentes em diferentes níveis da hierarquia, e não apenas entre os itens contidos no nível inferior da taxonomia, como ocorre em regras de associação que não se utilizam de taxonomias [2]. Por exemplo, a partir da taxonomia da figura 3.3 podemos inferir uma regra do tipo: "pessoas que compram calçados tendem a comprar jaquetas", sem necessidade de utilizar as regras: "pessoas que compram sapatos também compram jaquetas" e "pessoas que compram botas também compram jaquetas".

Srikant e Agrawal [2] propõem dois algoritmos para encontrar associações entre itens presentes em diferentes níveis da hierarquia, e argumentam que o uso da informação da taxonomia é de grande valia, uma vez que: primeiro, é possível descobrir regras interessantes com classes de itens que individualmente não possuem Suporte mínimo e, segundo, a informação da taxonomia pode ser utilizada na poda de regras redundantes.

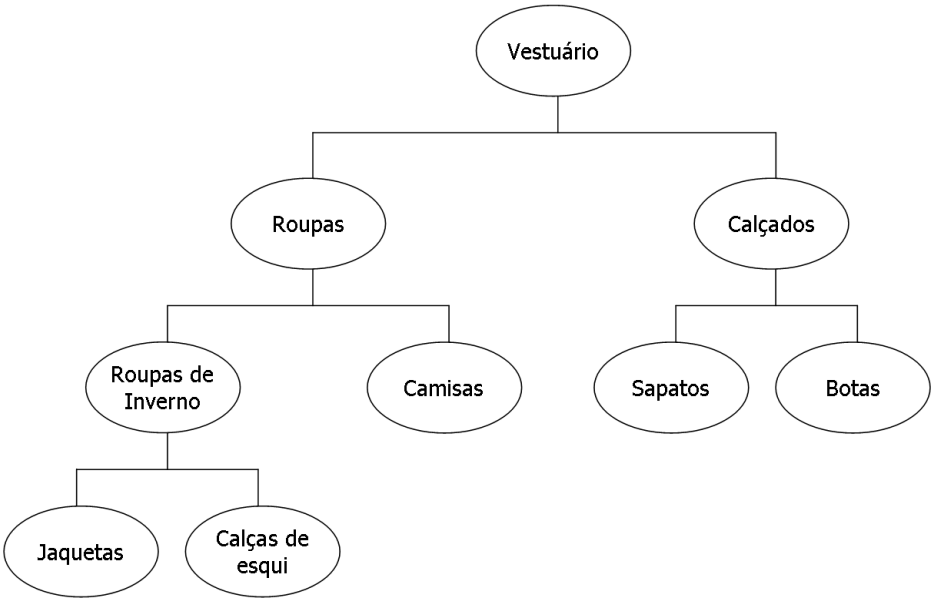


Figura 3.3: Exemplo de uma taxonomia para vestuário, retirado de [2].

Inicialmente os algoritmos geram todas as regras de associação gerais que possuem Suporte e Confiança maiores do que os valores mínimos especificados. A definição para uma regra de associação geral é uma implicação da forma X $\Rightarrow$ Y, para X e Y conjuntos disjuntos de itens, tal que X e Y contêm itens em qualquer nível da hierarquia, e nenhum item de Y é ancestral de qualquer item de X.

Em seguida as regras redundantes são eliminadas de acordo com uma medida de interesse baseada na informação da taxonomia. Basicamente, uma regra é considerada redundante quando não apresenta informação adicional em relação a sua forma mais geral (regra ancestral). Por exemplo, considere a taxonomia apresentada na figura 3.3 na qual "Calçados" é pai de "Sapatos", e a seguinte regra minerada:

Calçados $\Rightarrow$ Jaquetas (8% Sup, 70% Conf)

Se em torno de um quarto das vendas de calçados é de sapatos, nós esperamos que a regra (Sapatos $\Rightarrow$ Jaquetas) tenha 2% de Suporte e 70% de Confiança. Se o Suporte e a Confiança são próximos dos valores esperados, a regra é considerada redundante, já que não possui um comportamento significativamente diferente de sua generalização.

A partir dessa noção pode-se dizer que uma regra é interessante se a mesma possui valores de Suporte ou Confiança maior ou igual δ vezes o valor de um Suporte ou de uma Confiança esperada, para algum valor δ especificado pelo usuário.

Domingues [27] investiga o uso de taxonomias para reduzir o volume de regras extraídas.

Considere a taxonomia apresentada na figura 3.3, na qual “Calçados” é pai de “Sapatos” e “Botas”, e as seguintes regras mineradas:

Sapatos $\Rightarrow$ Jaqueta

Botas $\Rightarrow$ Jaqueta

Uma vez que a associação com o item “Jaqueta” está presente para todos os filhos do item “Calçados”, o método proposto por Domingues substitui as regras referentes a “Sapatos” e “Botas” pela regra mais geral:

Calçados $\Rightarrow$ Jaqueta

Dentre as abordagens que utilizam o conhecimento prévio do domínio como guia no processo de poda, os métodos apresentados por Srikant e Agrawal [2] e Domingues [27], com a utilização de taxonomias, são os que mais se aproximam do presente estudo, uma vez que, a taxonomia é um tipo de relação de dependência entre os dados. Contudo, neste trabalho, a avaliação da relevância das regras, ou melhor, o critério de escolha entre as regras mais especializadas e suas generalizações, é realizada de maneira diferente.

No método proposto por Srikant e Agrawal [2], a escolha do valor do interesse mínimo δ é realizada de maneira subjetiva (de acordo com a experiência do usuário), o que não garante que haja uma grande redução do volume de regras. Além do mais, o método de Srikant e Agrawal não resolve o problema da redundância semântica das regras, uma vez que nem sempre elimina todas as regras mais específicas e nunca considera o processo de especialização, no qual a regra mais geral é eliminada em prol das regras mais específicas.

Já o método proposto por Domingues [27] considera apenas o processo de generalização para os casos extremos, ou seja, quando todas as especializações estão presentes no conjunto de regras mineradas. A abordagem proposta também tem ação sobre os casos intermediários, mais comuns em aplicações reais.

3.2.2 Resumo e Agrupamento

Ainda que a fase de poda possa diminuir consideravelmente o número de regras de associação, este número pode continuar muito grande para ser manipulado pelo usuário.

A fase de resumo cria um subconjunto das regras representativo do conhecimento extraído. Geralmente, este processo tem como objetivo obter os conceitos mais gerais das regras de associação produzidas. Como argumentado em [6, 2], as regras mais gerais ou abstratas são preferíveis.

Em [28], Liu et al. falam sobre a existência de um pequeno subconjunto dos padrões que refletem todos os padrões existentes nos dados. Este subconjunto especial das regras, denominado de *regras DS* (*Direction Setting (DS) rules*), fornece um resumo das relações de comportamento descobertas. Utilizando este resumo, o usuário pode focar em aspectos essenciais do domínio, e seletivamente visualizar os detalhes relevantes (*regras não DS*).

As *regras DS* são encontradas com a utilização do teste estatístico χ^2, da seguinte forma: considere as regras de associação R1: (A $\Rightarrow$ C) e R2: (B $\Rightarrow$ C). Se o teste χ^2 indica que A e B estão positivamente correlacionados com C, então a regra R3: (A $\wedge$ B $\Rightarrow$ C) não é supreendente, uma vez que R3 intuitivamente segue R1 e R2. Desta forma, R1 e R2 são consideradas *regras DS* porque elas indicam a direção de R3, que, por sua vez, é considerada uma *regra não DS* porque segue a direção de R1 e R2.

Agrupar as regras de associação descobertas de acordo com certa característica pode ser útil na análise dos resultados. Esta técnica permite que o usuário analise grupos de interesse baseado em determinado critério. O critério de agrupamento pode ser semântico, como sugerido em [17], ou baseado em alguma medida estatística, como proposto em [6].

Constatando que mesmo o número de regras no subconjunto *rule covers* continuavam numerosas, Toivonen et al. [6] complementaram o estudo propondo o agrupamento das regras. Neste estudo a medida de distância entre duas regras de asssociação é baseada no número de instâncias cobertas pelos itens que compõem as regras. Esta medida é calculada da seguinte forma:

$$\begin{aligned} d(A \Rightarrow B, C \Rightarrow D) &= |(m(AB) \cup m(CD))/m(ACD)| \\ &= |(m(AB)| + |m(CD)| - 2 \times |m(ACD)| \end{aligned}$$

Onde |m(X)| é o número de instâncias da base de dados que possuem os itens presentes no conjunto X. No experimento realizado com a base de dados sobre os cursos de informática, o método encontrou agrupamentos bastante intuitivos, por exemplo, o agrupamento especializado na área de Sistemas Operacionais composto por regras envolvendo os cursos VAX/VMS e Sistemas Operacionais Distribuídos.

Desta forma, Toivonen et al. facilitam a análise e compreensão das regras, primeiro, criando um subconjunto menor das regras (*rule covers*) e, posteriormente, criando agrupamentos das mesmas.

3.2.3 Técnicas de Visualização

Uma vez obtidas as regras de asssociação potencialmente interessantes, as técnicas de visualização permitem a exploração visual dessas regras. O objetivo é facilitar a análise dos resultados pelo usuário, aumentando sua percepção da relevância das regras. Geralmente essas ferramentas fornecem uma representação gráfica das regras, com a utilização de gráficos e tabelas. Além disso, as ferramentas permitem que o usuário navegue pelas regras, direcionado a análise para determinada região do espaço das mesmas.

Trabalhos neste sentido podem ser encontrados em [29, 30, 31].

Capítulo 4

O Método Proposto

Neste capítulo, apresenta-se o método DMcut de pós-processamento de regras de associação. O método tem como objetivo eliminar as regras de associação redundantes ou de menor impacto, com base nas relações de dependência entre os atributos. A seção 4.1 é utilizada para definir as relações de dependência entre os atributos. A seção 4.2 apresenta um formalismo baseado na lógica de predicados, utilizado para motivar e apresentar o método proposto. A mesma seção aborda a questão da generalização e especialização das regras de associação, além de apresentar uma declaração formal do problema. A seção 4.3 é dedicada à apresentação do algoritmo DMcut. Por fim, na seção 4.4 são apresentadas as considerações finais a respeito deste capítulo.

4.1 Relações de Dependência entre Atributos

Uma relação de dependência é definida como o relacionamento entre dois conjuntos de atributos, com respeito a determinada base de dados[1]. Considere D uma base de dados, e X e Y subconjuntos dos atributos de D, sendo Y um conjunto unitário (de um único elemento). Dizemos que Y é dependente de X (ou que X determina Y), se e somente se, existe uma função f : Domínio(X) $\rightarrow$ Domínio(Y) válida para todo valor possível de D. A relação de dependência de Y por X será representada em símbolos por:

$$X \rightarrow Y$$

Chamaremos o atributo do subconjunto Y de atributo dependente, e os atributos do subconjunto X de atributos determinantes.

[1]Este tipo de dependência entre os atributos é conhecido por "dependência funcional" na literatura de Bancos de Dados.

Como exemplo, considere o atributo **IMC** (Índice de Massa Corporal), calculado a partir dos atributos **Peso** e **Altura**. A relação de dependência entre o atributo **IMC** e os atributos **Peso** e **Altura** é definida por f : Domínio({Peso, Altura}) $\rightarrow$ Domínio({IMC}), tal que $f\,(IMC) = Peso/Altura^2$. A dependência de **IMC** por **Peso** e **Altura** é representada em símbolos por:

$\{Peso,\ Altura\} \rightarrow \{IMC\}$,

Onde **IMC** é o atributo dependente, e **Peso** e **Altura** são os atributos determinantes.

4.1.1 Grau de Dependência

Dizemos que a relação de dependência $X \rightarrow Y$ é de grau *n*, quando existirem *n* funções que mapeiam os elementos do domínio de X para os elementos do domínio de Y, passando por *n*-1 domínios diferentes. A relação de dependência de grau *n* de Y por X será representada em símbolos por:

$X \overset{n}{\rightarrow} Y$

Como exemplo, considere o atributo **Status** calculado a partir do atributo **IMC**, tal que **Status** é uma discretização de **IMC**. A relação de dependência entre **Status** e **IMC** é de grau 1 (n=1), porque existe uma função g : Domínio({IMC}) $\rightarrow$ Domínio({Status}), tal que $g\,(Status)$ = {“abaixo do peso”, Se $IMC \leq 19$; “no peso normal”, Se $20 \leq IMC \leq 24$; “acima do peso”, Se $IMC \geq 25$}, que mapeia os elementos de {IMC} para os elementos de {Status}, passando por 0 (*n*-1) domínios diferentes.

Já a relação de dependência entre **Status** e os atributos **Peso** e **Altura** é de grau 2 (n=2), porque existem duas funções, g e f (f : Domínio({Peso, Altura}) $\rightarrow$ Domínio({IMC})), que mapeiam os elementos do domínio de {Peso, Altura} para os elementos do domínio de {Status}, passando por 1 (*n*-1) domínio diferente (Domínio(IMC)). A relação de dependência entre o atributo Status e os atributos Peso e Altura é representada em símbolos por:

$\{Peso,\ Altura\} \overset{2}{\rightarrow} \{Status\}$

Neste trabalho, estudamos o uso do conhecimento prévio do domínio, representado pelas relações de dependência entre os atributos, para eliminar regras redundantes ou de menor impacto para o usuário.

4.2 O Modelo

Seja D uma relação pertencente a uma base de dados multidimensional. Considere x e y dois atributos de D. Seja a dependência de x e y dada por f: Domínio(x) $\rightarrow$ Domínio(y). Considere y_i um elemento do domínio de y, e x_{ij} um elemento do domínio de x, tal que f(x_{ij}) = y_i. Considere X_{ij} = {x = x_{ij}} e Y_i = {y = y_i} dois conjuntos de condições definidas sobre os atributos x e y de D, respectivamente.

Considere P(h, A) o predicado: a instância h satisfaz as condições do conjunto A. Para A um conjunto de condições definidas sobre atributos de D.

Intuitivamente, podemos representar logicamente uma regra de associação A $\Rightarrow$ B como uma implicação de probabilidade p: P(h, A) $\xrightarrow{p}$ P(h, B), para A e B conjuntos disjuntos de condições definidas sobre atributos de D. A implicação significa que, se uma instância satisfaz as condições de A, então a instância satisfaz as condições de B com uma probabilidade p.

Por motivos de simplicidade, neste estudo, vamos omitir a probabilidade p, e representar logicamente as regras de associação como uma implicação comum.

A partir da relação de dependência entre os atributos x e y, mais especificamente, entre os conjuntos X_{ij} e Y_i, temos que, se uma instância satisfaz a condição do conjunto X_{ij}, então esta instância também satisfaz a condição do conjunto Y_i, o que pode ser traduzido logicamente pela expressão: P(h, X_{ij}) $\rightarrow$ P(h, Y_j).

A seguir são apresentados os tipos de redundância de regras de associação que são o objeto de estudo deste trabalho.

Definição 4.2.1 *(Redundância das regras na forma 1). Se existe uma regra da forma $X_{ij} \wedge C \Rightarrow Y_i$, esta regra é redundante. Para C um conjunto de condições definidas sobre atributos de D.*

A justificativa vem da lógica de primeira ordem, onde:

Uma regra de associação que apresenta a forma da expressão:

$$X_{ij} \wedge C \Rightarrow Y_i, \tag{4.1}$$

expressa logicamente por P(h, X_{ij}) $\wedge$ P(h, C) $\rightarrow$ P(h, Y_i), pode ser deduzida a partir da relação de dependência existente entre os conjuntos X_{ij} e Y_i, de acordo com o seguinte argumento válido:

Premissa 1 : $P(h,\ X_{ij}) \rightarrow P(h,\ Y_i)$
Conclusão : $P(h,\ X_{ij}) \wedge P(h,\ C) \rightarrow P(h,\ Y_i)$

De acordo com este argumento, a partir da premissa 1 pode ser concluído logicamente que $P(h,\ X_{ij}) \wedge P(h,\ C) \rightarrow P(h,\ Y_i)$ tem que ser também verdadeiro.

Observe que se $P(h,\ X_{ij})$ for verdadeiro, então $P(h,\ Y_i)$ seria verdadeiro pela premissa 1, e teríamos $P(h,\ X_{ij}) \wedge P(h,\ C) \rightarrow$ V (verdadeiro), que é verdadeiro. Se $P(h,\ X_{ij})$ for falso, aplicando-se a conjunção F (falso) $\wedge\ P(h,\ C) \equiv$ F, teríamos F $\rightarrow\ P(h,\ Y_i)$, que também é verdadeiro.

Exemplo 4.2.1 *Considere a relação de dependência entre os atributos **IMC** e **Status**, {IMC} $\xrightarrow{1}$ {Status}, definida pela função f: Domínio(IMC) → Domínio(Status), tal que f(24) = "normal". A relação de dependência entre os conjuntos (IMC=24) e (Status="normal") pode ser expressa logicamente pela implicação: P(h, {IMC=24}) → P(h, {Status="normal"}). Uma regra de associação do tipo:*

(IMC=24) ∧ (Idade=50) ∧ (Sexo="F") ⇒ (Status="normal"),

que apresenta a forma da expressão 4.1, e que pode ser expressa logicamente pela implicação: P(h, {IMC=24}) ∧ P(h, {Idade=50, Sexo="F"}) → P(h, {Status=" normal"}), é redundante porque pode ser deduzida da relação de dependência {IMC} $\xrightarrow{1}$ {Status}, de acordo com o seguinte argumento válido:

Premissa 1 : P(h, {IMC=24}) → P(h, {Status="normal"})
Conclusão : P(h, {IMC=24}) ∧ P(h, {Idade=50, Sexo="F"}) →
P(h, {Status="normal"})

Desta forma, podemos concluir que as regras que apresentam a forma da expressão 4.1 podem ser eliminadas, uma vez que representam uma relação previamente conhecida dos dados - as relações de dependência entre os atributos.

Definição 4.2.2 *(Redundância das regras na forma 2). Se existem duas regras das formas R_1: $X_{ij} \wedge A \Rightarrow B$ e R_2: $X_{ij} \wedge Y_i \wedge A \Rightarrow B$, então a regra R_2 é redundante. Para A e B conjuntos de condições definidas sobre atributos de D.*

A justificativa vem da lógica de primeira ordem, onde:

Uma regra de associação que apresenta a forma da expressão:

$$X_{ij} \wedge Y_i \wedge A \Rightarrow B, \tag{4.2}$$

expressa logicamente por P(h, X_{ij}) $\wedge$ P(h, Y_i) $\wedge$ P(h, A) $\rightarrow$ P(h, B), pode ser deduzida a partir da relação de dependência existente entre os conjuntos X_{ij} e Y_i, e de uma regra da forma da regra R_1, expressa logicamente por P(h, X_{ij}) $\wedge$ P(h, A) $\rightarrow$ P(h, B), de acordo com o seguinte argumento válido:

Premissa 1 : $P(h,\ X_{ij}) \rightarrow P(h,\ Y_i)$
Premissa 2 : $P(h,\ X_{ij}) \wedge P(h,\ A) \rightarrow P(h,\ B)$
Conclusão : $P(h,\ X_{ij}) \wedge P(h,\ A) \wedge P(h,\ Y_i) \rightarrow P(h,\ B)$

De acordo com este argumento, a partir das premissas 1 e 2 pode ser concluído logicamente que P(h, X_{ij}) $\wedge$ P(h, A) $\wedge$ P(h, Y_i) $\rightarrow$ P(h, B) tem que ser também verdadeiro.

Observe que, se P(h, X_{ij}) ou P(h, A) for falso, a conclusão será sempre verdadeira, uma vez que $P(h,\ X_{ij}) \wedge P(h,\ A) \wedge P(h,\ Y_i) \equiv F$ e F $\rightarrow$ P(h, B) é sempre verdadeiro. Se P(h, X_{ij}) e P(h, A) forem verdadeiros, pela premissa 1, P(h, Y_i) é verdadeiro, e, pela premissa 2, P(h, B) também é verdadeiro, e assim teríamos $P(h,\ X_{ij}) \wedge P(h,\ A) \wedge P(h,\ Y_i) \equiv V \rightarrow P(h,\ B) \equiv V$, que é verdadeiro.

Exemplo 4.2.2 *Considere a relação de dependência entre os atributos **IMC** e **Status** definida no exemplo 4.2.1. Considere as seguintes regras mineradas:*

R_1 : *(IMC=24) $\wedge$ (Idade=50) $\Rightarrow$ (Seguro-saúde="sim") e*
R_2 : *(IMC=24) $\wedge$ (Status="normal") $\wedge$ (Idade=50) $\Rightarrow$ (Seguro-saúde="sim")*

A regra R_2 que apresenta a forma da expressão 4.2, e que pode ser expressa logicamente pela implicação P(h, {IMC=24}) $\wedge$ P(h, {Status="normal"}) $\wedge$ P(h, {Idade=50}) $\rightarrow$ P(h, {Seguro-saúde="sim"}), é redundante porque pode ser deduzida a partir da regra R_1 e da relação de dependência {IMC} $\xrightarrow{1}$ {Status}, de acordo com o seguinte argumento válido:

Premissa 1 : P(h, {IMC=24}) $\rightarrow$ P(h, {Status="normal"})
Premissa 2 : P(h, {IMC=24}) $\wedge$ P(h, {Idade=50}) $\rightarrow$ P(h, {Seguro-saúde="sim"})
Conclusão : P(h, {IMC=24}) $\wedge$ P(h, {Idade=50}) $\wedge$ P(h, {Status="normal"}) $\rightarrow$ P(h, {Seguro-saúde="sim"})

O mesmo raciocínio pode ser utilizado para justificar o mesmo tipo de redundância no conseqüente da regra, ou seja, dadas duas regras das formas: R_1: A $\Rightarrow$ X_{ij} $\wedge$ B e R_2: A $\Rightarrow$ X_{ij} $\wedge$ Y_i $\wedge$ B, podemos concluir que a regra R_2 é redundante.

Observe que de acordo com a técnica de Regras de Associação, e com a relação de dependência existente entre os atributos x e y, podemos afirmar que sempre que uma regra da forma da expressão 4.2 for minerada, sua forma mais simples, ou seja, sem o elemento referente ao atributo dependente, também estará presente no conjunto de regras. Desta forma podemos seguramente eliminar as regras da forma da expressão 4.2.

Definição 4.2.3 *(Redundância das regras na forma 3). Se existem duas regras das formas R_1: $X_{ij} \wedge A \Rightarrow B$ e R_2: $X_{ij} \wedge A \Rightarrow Y_i \wedge B$, então a regra R_2 é redundante.*

A justificativa vem da lógica de primeira ordem, onde:

Uma regra de associação que apresenta a forma da expressão:

$$X_{ij} \ \wedge \ A \ \Rightarrow \ Y_i \ \wedge \ B, \tag{4.3}$$

expressa logicamente por P(h, X_{ij}) $\wedge$ P(h, A) $\rightarrow$ P(h, Y_i) $\wedge$ P(h, B), pode ser deduzida a partir da relação de dependência existente entre os conjuntos X_{ij} e Y_i, e de uma regra da forma da regra R_1, expressa logicamente por P(h, X_{ij}) $\wedge$ P(h, A) $\rightarrow$ P(h, B), de acordo com o seguinte argumento válido:

Premissa 1 : $P(h,\ X_{ij}) \ \rightarrow \ P(h,\ Y_i)$
Premissa 2 : $P(h,\ X_{ij}) \ \wedge \ P(h,\ A) \ \rightarrow \ P(h,\ B)$
Conclusão : $P(h,\ X_{ij}) \ \wedge \ P(h,\ A) \ \rightarrow \ P(h,\ Y_i) \ \wedge \ P(h,\ B)$

De acordo com este argumento, a partir das premissas 1 e 2 pode ser concluído logicamente que P(h, X_{ij}) $\wedge$ P(h, A) $\rightarrow$ P(h, Y_i) $\wedge$ P(h, B) tem que ser também verdadeiro.

Observe que, se P(h, X_{ij}) ou P(h, A) for falso, a conclusão será sempre verdadeira, uma vez que $P(h,\ X_{ij}) \ \wedge \ P(h,\ A) \ \equiv \ F$ e F $\rightarrow$ P(h, Y_i) $\wedge$ P(h, B) é sempre verdadeiro. Se P(h, X_{ij}) e P(h, A) forem verdadeiros, pela premissa 1, P(h, Y_i) é verdadeiro, e, pela premissa 2, P(h, B) também é verdadeiro, e assim teríamos $P(h,\ X_{ij}) \ \wedge \ P(h,\ A) \ \equiv \ V \ \rightarrow \ P(h,\ Y_i) \ \wedge \ P(h,\ B) \ \equiv \ V$, que é verdadeiro.

Exemplo 4.2.3 *Considere a relação de dependência entre os atributos **IMC** e **Status** definida no exemplo 4.2.1. Considere as seguintes regras mineradas:*

R_1 : (IMC=24) $\wedge$ (Idade=50) $\Rightarrow$ (Seguro-saúde="sim") e
R_2 : (IMC=24) $\wedge$ (Idade=50) $\Rightarrow$ (Status="normal") $\wedge$ (Seguro-saúde="sim")

A regra R_2 que apresenta a forma da expressão 4.3, e que pode ser expressa logicamente pela implicação P(h, {IMC=24}) $\wedge$ P(h, {Idade=50}) $\rightarrow$ P(h, {(Status="normal")}) $\wedge$ P(h,

{Seguro-saúde="sim"}), é redundante porque pode ser deduzida a partir da regra R_1 e da relação de dependência {IMC} $\xrightarrow{1}$ {Status}, de acordo com o seguinte argumento válido:

Premissa 1 : P(h, {IMC=24}) → P(h, {Status="normal"})
Premissa 2 : P(h, {IMC=24}) ∧ P(h, {Idade=50}) →
P(h, {Seguro-saúde="sim"})
Conclusão : P(h, {IMC=24}) ∧ P(h, {Idade=50}) →
P(h, {Status="normal"}) ∧ P(h, {Seguro-saúde="sim"})

Observe mais uma vez que, de acordo com a técnica de Regras de Associação, e com a relação de dependência existente entre os atributos x e y, podemos afirmar que sempre que uma regra da forma da expressão 4.3 for minerada, sua forma mais simples, ou seja, sem o elemento referente ao atributo dependente, também estará presente no conjunto de regras. Desta forma podemos seguramente eliminar as regras da forma da expressão 4.3.

Definição 4.2.4 *(Redundância de regras das formas 4 e 5). Se existem duas regras das formas R_1: $Y_i \wedge A \Rightarrow B$ e R_2: $X_{ij} \wedge A \Rightarrow B$, a regra R_2 é redundante.*

A justificativa vem da lógica de primeira ordem, onde:

De maneira intuitiva, considerando a relação de dependência existente entre os atributos x e y, uma regra de associação da forma da expressão:

$$Y_i \ \wedge \ A \ \Rightarrow \ B, \tag{4.4}$$

pode ser expressa logicamente por ∀ X ((P(h, X) → P(h, Y_i)) → (P(h, X) ∧ P(h, A) → P(h, B))). Ou seja, uma regra de associação da forma da expressão 4.4 é equivalente a uma regra da forma X ∧ A ⇒ B, para todo conjunto X = $\{x = x' \mid f(x') = y_i\}$.

Assim, uma regra de associação da forma da expressão:

$$X_{ij} \ \wedge \ A \ \Rightarrow \ B, \tag{4.5}$$

expressa logicamente por P(h, X_{ij}) ∧ P(h, A) → P(h, B), pode ser deduzida a partir da relação de dependência existente entre os conjuntos X_{ij} e Y_i, e de uma regra da forma da regra expressão 4.4, de acordo com o seguinte argumento válido:

Premissa 1 : $P(h,\ X_{ij}) \ \rightarrow \ P(h,\ Y_i)$
Premissa 2 : $\forall X \ (\ (P(h,\ X) \rightarrow \ P(h,\ Y_i)) \ \rightarrow \ (P(h,\ X) \ \wedge \ P(h,\ A) \ \rightarrow P(h, B)))$
Conclusão : $P(h,\ X_{ij}) \ \wedge \ P(h,\ A) \ \rightarrow \ P(h,\ B)$

De acordo com este argumento, a partir das premissas 1 e 2 pode ser concluído logicamente que $P(h,\ X_{ij}) \wedge P(h,\ A) \rightarrow P(h,\ B)$ tem que ser também verdadeiro. A prova da validade deste argumento é apresentada a seguir.

1.	$P(h,\ X_{ij}) \rightarrow P(h,\ Y_i)$	Premissa
2.	$\forall X\ (P(h,\ X) \rightarrow P(h,\ Y_i)) \rightarrow$ $(P(h,\ X) \wedge P(h,\ A) \rightarrow P(h, B))$	Premissa
3.	$(P(h,\ X_{ij}) \rightarrow P(h,\ Y_i)) \rightarrow$ $(P(h,\ X_{ij}) \wedge P(h,\ A) \rightarrow P(h, B))$	2. Instanciação universal : X_{ij} é a nova const.
4.	$P(h,\ X_{ij}) \wedge P(h,\ A) \rightarrow P(h, B)$	1,3. ($\rightarrow$)

Exemplo 4.2.4 *Considere a relação de dependência entre os atributos **IMC** e **Status** definida no exemplo 4.2.1. Considere as seguintes regras mineradas:*

R_1 : (Status="normal") $\wedge$ (Idade=50) $\Rightarrow$ (Seguro-saúde="sim") e

R_2 : (IMC=24) $\wedge$ (Idade=50) $\Rightarrow$ (Seguro-saúde="sim")

A regra R_2 que apresenta a forma da expressão 4.5, e que pode ser expressa logicamente pela implicação P(h, {IMC=24}) $\wedge$ P(h, {Idade=50}) $\rightarrow$ P(h, {Seguro-saúde="sim"}), é redundante porque pode ser deduzida a partir da regra R_1 e da relação de dependência {IMC} $\xrightarrow{1}$ {Status}, de acordo com o seguinte argumento válido:

Premissa 1	:	*P(h, {IMC=24}) $\rightarrow$ P(h, {Status="normal"})*
Premissa 2	:	*$\forall$X (P(h, X) $\rightarrow$ P(h, {Status="normal"})) $\rightarrow$ (P(h, X) $\wedge$ P(h, {Idade=50}) $\rightarrow$ P(h, {Seguro-saúde="sim"}))*
Conclusão	:	*P(h, {IMC=24}) $\wedge$ P(h, {Idade=50}) $\rightarrow$ P(h, {Seguro-saúde="sim"})*

Para resolver esse tipo de redundância, avaliamos a relevância das regras, e descartamos aquelas menos relevantes ou de menor impacto para o usuário. Na próxima seção apresentamos como é realizado o processo de avaliação das regras nas formas das expressões 4.4 e 4.5, e como é feita a escolha entre os processos de generalização e especialização das mesmas.

4.2.1 Generalização e Especialização das Regras

Vimos na seção anterior que existe uma redundância semântica entre uma regra na forma da expressão 4.4, a qual chamaremos de regra mais geral, e um conjunto de regras que apresentam a forma da expressão 4.5, as quais faremos referência como regras mais específicas. Para resolver esse tipo de redundância, propomos os processos de generalização e especialização das regras. O processo de generalização se dá quando a regra mais geral é considerada mais relevante, ou de maior impacto, do que as regras mais específicas. Neste caso, eliminamos

as regras mais específicas em prol da regra mais geral. No caso contrário, quando as regras mais específicas apresentam maior relevância do que a regra mais geral, aplicamos o processo de especialização, no qual ficamos com as regras mais específicas e descartamos a regra mais geral.

De maneira intuitiva, no processo de generalização, valorizamos a generalidade das relações descobertas, enquanto que, no processo oposto, valorizamos a capacidade discriminatória das mesmas.

A avaliação das regras é feita da seguinte forma. Quando o comportamento descrito pela regra mais geral independe dos atributos determinantes, ou melhor, quando a regra mais geral descreve um comportamento uniforme da população, consideramos a regra mais geral de maior relevância. No caso de uma população desbalanceada, onde o comportamento descrito pela regra mais geral não é uniforme (válido) para todos os valores possíveis dos atributos determinantes, as regras mais específicas são consideradas as mais relevantes.

O exemplo a seguir ilustra um cenário de escolha entre os processos de generalização e especialização das regras.

Exemplo 4.2.5 *Dado um banco de dados contendo informações sobre pacientes de uma clínica, considere a relação de dependência entre os atributos* ***IMC*** *(Índice de Massa Corporal) e* ***Status****: {IMC} → {Status}, definida por* f *: Domínio({IMC}) → Domínio({Status}), tal que:* $f(38)$*=“obeso”.*

Considere as seguintes regras mineradas:

R1: (Status=“obeso”) ⇒ (Pressão-sanguínea=“alta”)

R2: (IMC=37) ⇒ (Pressão-sanguínea=“alta”)

R3: (IMC=38) ⇒ (Pressão-sanguínea=“alta‘”)

Neste exemplo a regra mais geral, representada pela regra R1, descreve uma associação entre os indivíduos de status obeso e a pressão sanguínea alta. As regras R2 e R3 descrevem a mesma associação da regra R1 para os indivíduos de IMC igual a 37 e 38, respectivamente.

Se o comportamento descrito pela regra R1 for uniforme em relação ao IMC, ou seja, os indivíduos obesos têm a pressão alta não importando o valor do IMC, então aplicamos o processo de generalização. No caso de uma população desbalanceada onde, dos indivíduos obesos, apenas aqueles de IMC 37 e 38 têm a pressão alta, então aplicamos o processo de especialização.

A noção do quanto um comportamento é considerado uniforme será dada pelas medidas CRg e CD+ descritas a seguir.

4.2.1.1 CRg

O indicador CRg mede a generalidade da regra mais geral com relação aos atributos que determinam o atributo dependente. Esta medida é baseada na medida de interesse Cobertura [24], que representa a fração das instâncias cobertas pelo antecedente da regra, podendo ser considerada como medida de generalidade da regra. O valor da Cobertura de uma regra é dado pelo suporte do antecedente dessa regra.

Definição 4.2.5 *(CRg). Seja D uma base de dados multidimensional. Seja R_i uma regra de associação da forma da expressão 4.4, e S_i = {r_{ij}|j=1..n} o conjunto de regras correspondentes na forma da expressão 4.5, obtidas a partir de D. O valor da medida CRg para R_i e S_i é dado por:*

$$CRg(R_i, S_i) = \frac{\sum_{j=1..n} Suporte(Antecedente(r_{ij}))}{Suporte(Antecedente(R_i))} \tag{4.6}$$

Intuitivamente, quanto maior for a medida CRg, maior a representatividade das instâncias cobertas pelas regras mais específicas com relação às instâncias cobertas pela regra mais geral, o que significa um comportamento uniforme da população. Esta medida varia entre 0 e 1.

A medida CRg pode ser interpretada como a probabilidade condicional de uma instância satisfazer o antecedente de uma das regras mais específicas, dado que a instância satisfaz o antecedente da regra mais geral.

Utilizando o exemplo 4.2.5 apresentado no início desta seção, se o valor calculado para a medida CRg for alto (maior do que um valor mínimo especificado pelo usuário), significa que a maior parte dos indivíduos obesos tem o IMC igual a 37 ou a 38. Desta forma, o comportamento descrito pela regra mais geral é considerado uniforme em relação ao IMC, uma vez que, os valores de IMC 37 e 38 também estão associados à pressão sanguínea alta.

O valor da medida CRg pode ser calculado a partir dos valores de Suporte e Confiança da regra mais geral e das regras mais específicas. Para tanto, utilizamos a fórmula da medida Confiança para calcular o suporte do antecedente de uma regra R como:

$$Suporte(Antecedente(R)) = \frac{Suporte(R)}{Confianca(R)} \tag{4.7}$$

Substituindo o suporte do antecedente das regras na fórmula 4.6, de acordo com a expressão 4.7, a medida CRg é dada por:

$$CRg(R_i, S) = \left(\sum_{j=1..n} \frac{Suporte(r_{ij})}{Confianca(r_{ij})} \right) / \left(\frac{Suporte(R_i)}{Confianca(R_i)} \right) \tag{4.8}$$

4.2.1.2 $\mathbf{CD^+}$

A medida CD^+ indica se a regra mais geral continua válida se considerarmos apenas as instâncias não cobertas pelas regras mais específicas. Esta medida é utilizada quando o valor da medida CRg for abaixo do valor mínimo especificado pelo usuário.

Esta medida é baseada na medida Confiança, e a mesma fórmula é utilizada, com a inclusão de uma restrição δ. A restrição δ restringe as instâncias consideradas no cálculo da Confiança da regra mais geral, para aquelas não cobertas pelas regras mais específicas.

Definição 4.2.6 *(CD^+). Seja D uma base de dados multidimensional. Seja R_i uma regra de associação da forma da expressão 4.4, e S_i = {r_{ij}|j=1..n} o conjunto de regras correspondentes na forma da expressão 4.5, obtidas a partir de D. O valor da Confiança para R_i e S_i é dado por:*

$$CD^+(R_i, S_i) = \frac{Suporte(A \cap Y_i \cap B \cap \overline{(\bigcup X_{ij} \cap A)})}{Suporte(A \cap Y_i \cap \overline{(\bigcup X_{ij} \cap A)})} \tag{4.9}$$

Onde a condição $\overline{(\bigcup X_{ij} \cap A)}$ representa a restrição δ, que significa não considerar as instâncias cobertas pelas regras mais específicas.

Se a regra mais geral continuar válida com a inclusão da restrição δ, ou seja, se o valor da medida CD^+ for acima do valor mínimo especificado pelo usuário, significa que o comportamento da população é uniforme, uma vez que o comportamento descrito pela regra mais geral é válido para quaisquer valores dos atributos que determinam o atributo dependente. Esta medida varia entre 0 e 1.

Podemos substituir a condição $\overline{(\bigcup X_{ij} \cap A)}$ pela condição equivalente: $(\overline{\bigcup X_{ij}} \cup \overline{A})$. A conjunção dessa condição com o conjunto A pode ser calculada como:

$$A \cap (\overline{\bigcup X_{ij}} \cup \overline{A}) = (A \cap \overline{\bigcup X_{ij}}) \cup (A \cap \overline{A}) = A \cap \overline{\bigcup X_{ij}}$$

Substituindo a fórmula acima na expressão 4.9, temos que:

$$CD^{+}(R_i, S_i) = \frac{Suporte(A \cap Y_i \cap B \cap \overline{\bigcup X_{ij}})}{Suporte(A \cap Y_i \cap \overline{\bigcup X_{ij}})}$$

Os suportes da expressão acima podem ser calculados da seguinte maneira:

$$Suporte(A \cap Y_i \cap B \cap \overline{\bigcup X_{ij}}) = Suporte(A \cap Y_i \cap B) - Suporte(A \cap Y_i \cap B \cap \bigcup X_{ij}),\ e$$

$$Suporte(A \cap Y_i \cap \overline{\bigcup X_{ij}}) = Suporte(A \cap Y_i) - Suporte(A \cap Y_i \cap \bigcup X_{ij})$$

Nas expressões acima, podemos substituir o termo $Y_i \cap \bigcup X_{ij}$ pela união dos conjuntos X_{ij} ($\bigcup X_{ij}$), uma vez que: $\bigcup X_{ij} \subseteq Y_i$, e logo, $Y_i \cap \bigcup X_{ij} = \bigcup X_{ij}$. Desta forma, podemos calcular os suportes por:

$$Suporte(A \cap Y_i \cap B \cap \bigcup X_{ij}) = Suporte(A \cap B \cap \bigcup X_{ij}),\ e$$
$$Suporte(A \cap Y_i \cap \bigcup X_{ij}) = Suporte(A \cap \bigcup X_{ij})$$

Substituindo o cálculo dos suportes acima na fórmula da CD^{+}, temos que:

$$CD^{+}(R_i, S_i) = \frac{Suporte(A \cap Y_i \cap B) - Suporte(A \cap B \cap \bigcup X_{ij})}{Suporte(A \cap Y_i) - Suporte(A \cap \bigcup X_{ij})}$$

Das definições das medidas Suporte e Confiança, podemos deduzir que:

$$Suporte(A \cap Y_i \cap B) = Suporte(R_i),$$
$$Suporte(A \cap B \cap \bigcup X_{ij}) = \sum_{j=1..n} Suporte(r_{ij}),$$
$$Suporte(A \cap Y_i) = Suporte(Antecedente(R_i)),\ e$$
$$Suporte(A \cap \bigcup X_{ij}) = \sum_{j=1..n} Suporte(Antecedente(r_{ij}))$$

Substituindo os termos acima na expressão da CD^{+}, temos que:

$$CD^{+}(R_i, S_i) = \frac{Suporte(R_i) - \sum_{j=1..n} Suporte(r_{ij})}{Suporte(Antecedente(R_i)) - \sum_{j=1..n} Suporte(Antecedente(r_{ij}))}$$

Assim como a medida CRg, o valor da medida CD^+ também pode ser calculado a partir dos valores de Suporte e Confiança da regra mais geral e das regras mais específicas. Para tanto, substituimos o suporte dos antecedentes na expressão da medida CD^+, de acordo com a fórmula 4.7. Desta forma, a medida CD^+ é calculada como:

$$CD^+(R_i, S_i) = \frac{Suporte(R_i) - \sum_{j=1..n} Suporte(r_{ij})}{\frac{Suporte(R_i)}{Confianca(R_i)} - \sum_{j=1..n} \frac{Suporte(r_{ij})}{Confianca(r_{ij})}} \tag{4.10}$$

4.2.2 Declaração Formal do Problema

A seguir é apresentada uma declaração formal do problema do pós-processamento de regras de associação, com base nas relações de dependência entre os atributos.

Definição 4.2.7 *(Problema do Pós-processamento de Regras de Associação) Seja R um conjunto de regras de associação obtidas a partir de uma base de dados multidimensional D. Seja RD o conjunto das relações de dependência entre os atributos da forma* $X \xrightarrow{n} Y$*, para X e Y conjuntos de atributos de D (sendo Y um conjunto unitário). O problema do pós-processamento de regras de associação, com base nas relações de dependência entre os atributos consiste em:*

1. *Eliminar do conjunto R todas as regras de associação redundantes que apresentam as formas das expressões 4.1, 4.2 e 4.3.*
2. *Para cada regra* R_i *da forma da expressão 4.4, e o conjunto* S_i *correspondente de regras da forma da expressão 4.5, redundantes com* R_i*, avaliar a relevância das regras com a utilização das medidas CRg e* CD^+*, e eliminar do conjunto R aquelas (ou aquela) de menor relevância, ou de menor impacto para o usuário.*

4.3 O Algoritmo DMcut

Nesta seção descrevemos o algoritmo DMcut, responsável por resolver o problema do pós-processamento de regras de associação descrito na seção anterior. O algoritmo DMcut, ilustrado no Algoritmo 1, recebe como entrada os seguintes parâmetros: um conjunto *R* de regras de associação, um conjunto *RD* das relações de dependência entre os atributos, o valor *GrauMax* que indica o grau máximo de dependência entre os atributos a ser considerado, e os valores mínimos para as medidas CRg e CD^+, *CRgMin* e CD^+*Min*, respectivamente. O algoritmo retorna um conjunto filtrado das regras de associação.

Algoritmo 1 DMcut

Require: <R, RD, n, CRgMin, CD$^+$Min>
1: R' := poda-regras-1(R, RD)
2: R' := poda-regras-2(R', RD, *esquerdo*)
3: R' := poda-regras-2(R', RD, *direito*)
4: R' := poda-regras-3(R', RD)
5: $\widehat{S}$:= gera-subconjuntos(R', *direito*)
6: **for all** subconjunto S tal que $S \in \widehat{S}$ **do**
7: G := gera-regras-gerais(S, RD)
8: **for all** r tal que $r \in G$ **do**
9: **for** n $= 1$ até *GrauMax* **do**
10: E := gera-regras-específicas(S, RD, r, n)
11: crg := calcula-CRg(r, E)
12: **if** crg $\geq$ CRgMin **then**
13: R' := elimina-regras(E, R') {Processo de generalização}
14: **else**
15: cd$^+$:= calcula-CD$^+$(r, E)
16: **if** cd$^+$ $\geq$ CD$^+$Min **then**
17: R' := elimina-regras(E, R') {Processo de generalização}
18: **else**
19: R' := elimina-regras(r, R') {Processo de especialização}
20: n := GrauMax {"Pula" para a próxima regra mais geral presente em G}
21: **end if**
22: **end if**
23: **end for**
24: **end for**
25: **end for**
26: **return** R'

Inicialmente o conjunto R' que irá armazenar as regras de associação filtradas recebe o resultado da função poda-regras-1. Esta função elimina as regras que apresentam a forma da expressão 4.1. A função poda-regras-1 é descrita pelo Algoritmo 2. Em seguida, as regras que apresentam a forma da expressão 4.2, para o antecedente (lado esquerdo) e o conseqüente (lado direito) da regra, são eliminadas do conjunto R' pela função poda-regras-2, descrita pelo Algoritmo 3. No próximo passo do algoritmo, as regras que apresentam a forma da expressão 4.3 são eliminadas do conjunto R' pela função poda-regras-3, descrita pelo Algoritmo 4.

Após eliminar do conjunto R' as regras redundantes que apresentam as formas das expressões 4.1, 4.2 e 4.3, o algoritmo segue para resolver a redundância das regras nas formas das expressões 4.4 e 4.5.

O próximo passo do algoritmo consiste em gerar os subconjuntos presentes em R' de regras que apresentam o mesmo conseqüente. A geração dos subconjuntos é realizada pela função gera-subconjuntos, descrita detalhadamente em [32]. Para cada subconjunto S gerado, a fun-

ção gera-regras-gerais gera o conjunto G das regras mais gerais (da forma da expressão 4.4) presentes em S. Uma descrição da função gera-regras-gerais é apresentada no Algoritmo 5.

Em seguida, para cada regra mais geral r $\in$ G, e para o grau de dependência variando de 1 até *GrauMax*, é gerado o conjunto E das regras mais específicas (da forma da expressão 4.3), presentes no conjunto S, que são redundantes com r. A geração do conjunto das regras mais específicas é realizada pela função gera-regras-específicas, descrita pelo Algoritmo 6.

A partir da regra mais geral, e do conjunto correspondente das regras mais específicas, a função calcula-CRg calcula o valor da medida CRg de acordo com a fórmula 4.8. Se o valor da medida CRg for maior ou igual ao valor mínimo especificado pelo usuário (*CRgMin*), as regras mais específicas são eliminadas do conjunto R'(processo de generalização).

O processo de generalização é realizado através da função elimina-regras, que elimina do conjunto R' as regras presentes em E e retorna o resultado.

Caso contrário, o valor para a medida CD^+ é calculado pela função calcula-CD^+. A função calcula-CD^+ utiliza a fórmula 4.10 para calcular a medida CD^+ a partir da regra mais geral e do conjunto correspondente das regras mais específicas. Se este valor for maior ou igual ao valor mínimo especificado pelo usuário, mais uma vez aplicamos o processo de generalização das regras.

Caso o valor da medida CD^+ seja menor do que o valor mínimo especificado pelo usuário, o processo executado é o de especialização das regras. Desta vez, o conjunto R' recebe o resultado da função elimina-regras que elimina do conjunto R' a regra r. Por fim, na linha 26, o algoritmo retorna o conjunto R'.

4.3.1 Função poda-regras-1

A função poda-regras-1 é apresentada no Algoritmo 2. Esta função recebe como parâmetro um conjunto de regras de associação *R*, e um conjunto de relações de dependência entre os atributos *RD*.

Inicialmente, o conjunto R' que irá armazenar o conjunto filtrado de regras de associação recebe o conjunto R. Em seguida, para cada regra r presente em R, o algoritmo verifica se a regra possui no conseqüente apenas um item. Caso verdadeiro, se o atributo deste item i for um atributo dependente, o algoritmo verifica se os itens que o determinam estão presentes no antecedente da regra. Caso verdadeiro, a regra r é removida do conjunto R', e o algoritmo "pula" para a próxima regra presente em R'. Por fim, o algoritmo retorna o conjunto R'.

Algoritmo 2 Função poda-regras-1

```
Require: <R,RD>
 1: R' := R
 2: for all r tal que r ∈ R' do
 3:    if RHS(s) contém apenas um item i then
 4:       if atributo de i é um atributo dependente then
 5:          if itens que determinam i estão presentes em LHS(r) then
 6:             R' := R' - r
 7:             "pula" para a próxima regra r
 8:          end if
 9:       end if
10:    end if
11: end for
12: return R'
```

4.3.2 Função poda-regras-2

A função poda-regras-2 é apresentada no Algoritmo 3. Esta função recebe como parâmetro um conjunto de regras de associação *R*, um conjunto de relações de dependência entre os atributos *RD*, e o lado *l* da regra a ser analisado.

Algoritmo 3 Função poda-regras-2

```
Require: <R,RD,l>
 1: R' := R
 2: for all r tal que r ∈ R' do
 3:    for all i tal que i ∈ lado(l,r) do
 4:       if atributo de i é um atributo dependente then
 5:          if itens que determinam i estão presentes em lado(l,r) then
 6:             R' := R' - r
 7:             "pula" para a próxima regra r
 8:          end if
 9:       end if
10:    end for
11: end for
12: return R'
```

Inicialmente, o conjunto R' que irá armazenar o conjunto filtrado de regras de associação recebe o conjunto R. Em seguida, para cada regra r presente em R, o algoritmo percorre os itens do lado *l* da regra. Se o atributo do item corrente i for um atributo dependente, o algoritmo verifica se os itens que determinam i estão presentes no lado *l* da regra. Caso verdadeiro, a regra r é removida do conjunto R', e o algoritmo "pula" para a próxima regra presente em R'. Por fim, o algoritmo retorna o conjunto R'.

4.3.3 Função poda-regras-3

A função poda-regras é apresentada no Algoritmo 4. Esta função recebe como parâmetro um conjunto de regras de associação *R*, e um conjunto de relações de dependência entre os atributos *RD*.

Algoritmo 4 Função poda-regras-3

```
Require: <R,RD>
 1: R' := R
 2: for all r tal que r ∈ R' do
 3:    if RHS(r) tem mais de um item then
 4:       for all i tal que i ∈ RHS(r) do
 5:          if atributo de i é um atributo dependente then
 6:             if itens que determinam i estão presentes em LHS(r) then
 7:                R' := R' - r
 8:                "pula" para a próxima regra r
 9:             end if
10:          end if
11:       end for
12:    end if
13: end for
14: return R'
```

Inicialmente, o conjunto R' que irá armazenar o conjunto filtrado de regras de associação recebe o conjunto R. Em seguida, para cada regra r presente em R, que possui mais de um item no conseqüente, o algoritmo percorre os itens do conseqüente da regra. Se o atributo do item corrente i for um atributo dependente, o algoritmo verifica se os itens que especificam este item estão presentes no antecedente da regra. Caso verdadeiro, a regra r é removida do conjunto R', e o algoritmo "pula" para a próxima regra presente em R'. Por fim, o algoritmo retorna o conjunto R'.

4.3.4 Função gera-regras-gerais

A função gera-regras-gerais é apresentada no Algoritmo 5. Esta função recebe como parâmetro um conjunto de regras de associação *R*, e um conjunto das relações de dependência entre os atributos *RD*. A função retorna o conjunto das regras mais gerais (da forma da expressão 4.4) presentes em R.

Inicialmente, o conjunto R' que irá armazenar o conjunto de regras de associação gerais recebe o conjunto vazio ($\emptyset$). Em seguida, para cada regra r presente em R, o algoritmo percorre os itens do antecedente da regra. Se o atributo do item corrente for um atributo dependente, a

Algoritmo 5 Função gera-regras-gerais

```
Require: <R, RD>
  R' := ∅
  for all r tal que r ∈ R do
    for all item tal que item ∈ LHS(r) do
      if atributo de item é um atributo dependente then
        R' := R' ∪ r
        "pula" para a próxima regra
      end if
    end for
  end for
  return R'
```

regra é adicionada ao conjunto R', e o algoritmo "pula" para a próxima regra presente em R. Por fim, o algoritmo retorna o conjunto R'.

4.3.5 Função gera-regras-específicas

A função gera-regras-específicas é apresentada no Algoritmo 6. Esta função recebe como parâmetro um conjunto de regras de associação *R*, um conjunto de relações de dependência entre os atributos *RD*, uma regra mais geral *rg*, e o grau de dependência entre os atributos a ser considerado *n*. A função retorna o conjunto das regras mais específicas (da forma da expressão 4.5) presentes em R, que são redundantes com rg.

Algoritmo 6 Função gera-regras-específicas

```
Require: <R,RD,rg,n>
  R' := ∅
  y := atributo dependente de rg
  X := atributos que determinam y considerando o grau de dependência n
  for all r tal que r ∈ R do
    if LHS(r) possui os atributos de X then
      v := calcula-valor(y, r);
      if v ≠ valor do atributo y em rg then
        "pula" para a próxima regra
      end if
      for i = 0 to número-itens(LHS(r)) do
        if (atributo de r.itens[i] ∉ X and r.itens[i] ∉ rg) or (atributo de r.itens[i] = y) then
          "pula" para a próxima regra
        end if
      end for
      R' := R' ∪ r
    end if
  end for
  return R'
```

Inicialmente, o conjunto R' que irá armazenar o conjunto de regras mais específicas recebe o conjunto vazio (∅). Em seguida, a variável y recebe o atributo dependente da regra geral e a variável X recebe o conjunto dos atributos que especificam y. Neste passo, apenas as relações de dependência de grau n são consideradas.

O próximo passo do algoritmo é percorrer todas as regras presentes em R. Para cada regra r em R, o algoritmo verifica se a regra possui os atributos que especificam y no antecedente. Caso verdadeiro, a variável v recebe o resultado da função calcula-valor. Esta função calcula o valor do atributo y a partir dos atributos de X presentes em r. Se o valor calculado for diferente do valor do atributo y de rg, então a regra r não é redundante com rg, e o algoritmo "pula" para a próxima regra em R.

No caso contrário, o algoritmo continua a verificação da regra r. O próximo passo consiste em verificar se os demais elementos da regra r são os mesmos da regra rg. Para tanto, o algoritmo percorre os itens do antecedente da regra r e faz a seguinte verificação: se o atributo do item não é um dos que especificam y, e este item não existe na regra rg, ou se o atributo do item é o atributo y. Caso verdadeiro a regra r não é redundante com rg, e o algorimto "pula" para a próxima regra r. Se algoritmo chegar à linha 15, a regra r é redundante com rg e é adicionada ao conjunto R'. Por fim, o algoritmo retorna o conjunto R'.

4.4 Considerações Finais

Neste capítulo mostramos como automatizar o processo de abstração das regras de associação, considerando o conhecimento de mundo representado pelas relações de dependência entre os atributos.

Inicialmente, definimos os tipos de redundância que podemos identificar e eliminar com a utilização das relações de dependência.

Em seguida, apresentamos os processos de generalização e especialização das regras como forma de eliminar as redundâncias das regras nas formas das expressões 4.4 e 4.5. Vimos que a escolha entre os dois processos é baseada nas medidas CRg e CD^{+}, que são bastante intuitivas.

Vale dizer que, ao utilizar essas medidas estatísticas no processo de escolha, assumimos que os dados utilizados refletem com fidelidade o contexto da aplicação. Ou seja, assumimos que todos os elementos existentes no domínio estão representados, e com uma distribuição real dos dados. Uma abordagem mais subjetiva poderia levar em consideração outras informações do domínio da aplicação, como a cardinalidade e/ou a distribuição dos elementos envolvidos.

Por exemplo, se existem cinco tipos diferentes de refrigerante à venda, e apenas “coca” e “fanta” aparecem nas regras mineradas, então talvez o processo de especialização seria o mais indicado. Ou ainda, utilizando a distribuição dos elementos, se “coca” e “fanta” representam 90% das vendas de refrigerante, talvez seria melhor ficar com a regra mais geral relativa à refrigerante.

Quanto mais completo for o conhecimento do domínio incorporado na análise, mais próximo chegamos ao processo humano de abstração das regras, e maior a segurança na escolha. Entretanto, achamos que a aplicabilidade do método proposto seria maior se o conhecimento de mundo utilizado fosse mais simples. Observe que a cardinalidade e a distribuição dos elementos, quando não são de conhecimento explícito do domínio, podem ser difíceis de se determinar.

Por fim, neste capítulo foi apresentado o algoritmo DMcut, proposto para filtrar as regras de associação na fase do pós-processamento do conhecimento extraído. O algoritmo proposto tem como objetivo eliminar a redundância das regras mineradas, reduzindo a quantidade, e, portanto, a complexidade, do conjunto de regras de associação.

Capítulo 5

Experimentos Realizados

5.1 Implementação

O método DMcut foi implementado na linguagem C++, com a utilização do compilador Microsoft Visual C++ 6.0.

5.1.1 Formato das Regras de Associação

A implementação do método DMcut lê as regras de associação e suas medidas de interesse a partir de arquivos no formato de saída dos aplicativos Weka[1] [33] e ADDMiner [2].

5.1.2 Formato das Relações de Dependência entre os Atributos

As relações de dependência entre os atributos são lidas a partir de um arquivo ASCII cuja extensão é *.dep. O arquivo é formado por uma série de comandos "@atributo" que especificam o nome e o tipo do atributo, seguido de um comando "@dependencia" que especifica os atributos que determinam este atributo, e a regra de derivação. Os atributos que determinam o atributo dependente são escritos entre parênteses e separados por vírgulas, e a regra de derivação é delimitada por chaves.

[1] O ambiente Weka é uma coleção de algoritmos de aprendizagem de máquina para aplicações de Mineração de Dados. O Weka foi desenvolvido na Universidade de Waikato (www.waikato.ac.nz) na Nova Zelândia dentro da filosofia GNU (General Public License) de código fonte aberto, e possui ferramentas para pré-processamento de dados, classificação, regressão, agrupamento, Regras de Associação e visualização.

[2] O ADDMiner, desenvolvido pelo laboratório ADDLabs (www.addlabs.uff.br), é um aplicativo que disponibiliza ferramentas para a Mineração de Dados com a utilização da técnica de Regras de Associação. Dentre as principais características deste aplicativo estão: além das medidas Suporte e Confiança, gera também as principais medidas de interesse objetivas encontradas na literatura, e permite controlar os atributos que podem aparecer no antecedente e no conseqüente das regras mineradas.

A regra de derivação especifica como o valor do atributo dependente é calculado e contém a seguinte forma:

[<condições:v:{s|n}>]; v:{s|n}

Ou seja, a regra de derivação contém zero ou mais declarações do tipo: *condições:v:{s|n}*, onde: *condições* especifica as condições para que o atributo assuma o valor *v*, e *{s|n}* indica se *v* deve ser avaliado como uma expressão, tal que *s* = sim, *n* = não. O último termo, separado por ponto e vírgula (*v:{s|n}*) indica o valor do atributo dependente caso nenhuma das condições seja satisfeita. Mais uma vez, o parâmetro *{s|n}* indica se *v* deve ser avaliado como uma expressão.

Na Figura 5.1 é apresentado um exemplo de um arquivo de relações de dependência entre atributos, contendo a relação de dependência {Peso, Altura} $\stackrel{1}{\rightarrow}$ {IMC}. A primeira linha do arquivo indica que o nome do atributo dependente é **IMC** e que se trata de um atributo do tipo "real". Na segunda linha, **Peso** e **Altura** são indicados como os nomes dos atributos determinantes do atributo **IMC**. Ainda na segunda linha, a regra de derivação especifica que o atributo **IMC** assume o valor "vazio" caso um dos atributos **Peso** e **Altura** assuma o mesmo valor. Caso contrário, o último termo da regra de derivação indica que o atributo **IMC** recebe o resultado da expressão $Peso/Altura^2$.

```
@atributo IMC real
@dependencia (Peso, Altura) {Peso=vazio|Altura=vazio :vazio:n;
Peso/(Altura*Altura):s}
```

Figura 5.1: Exemplo de um arquivo de relações de dependência entre os atributos.

5.1.3 Interface Gráfica - Ambiente de Pós-processamento DMcut

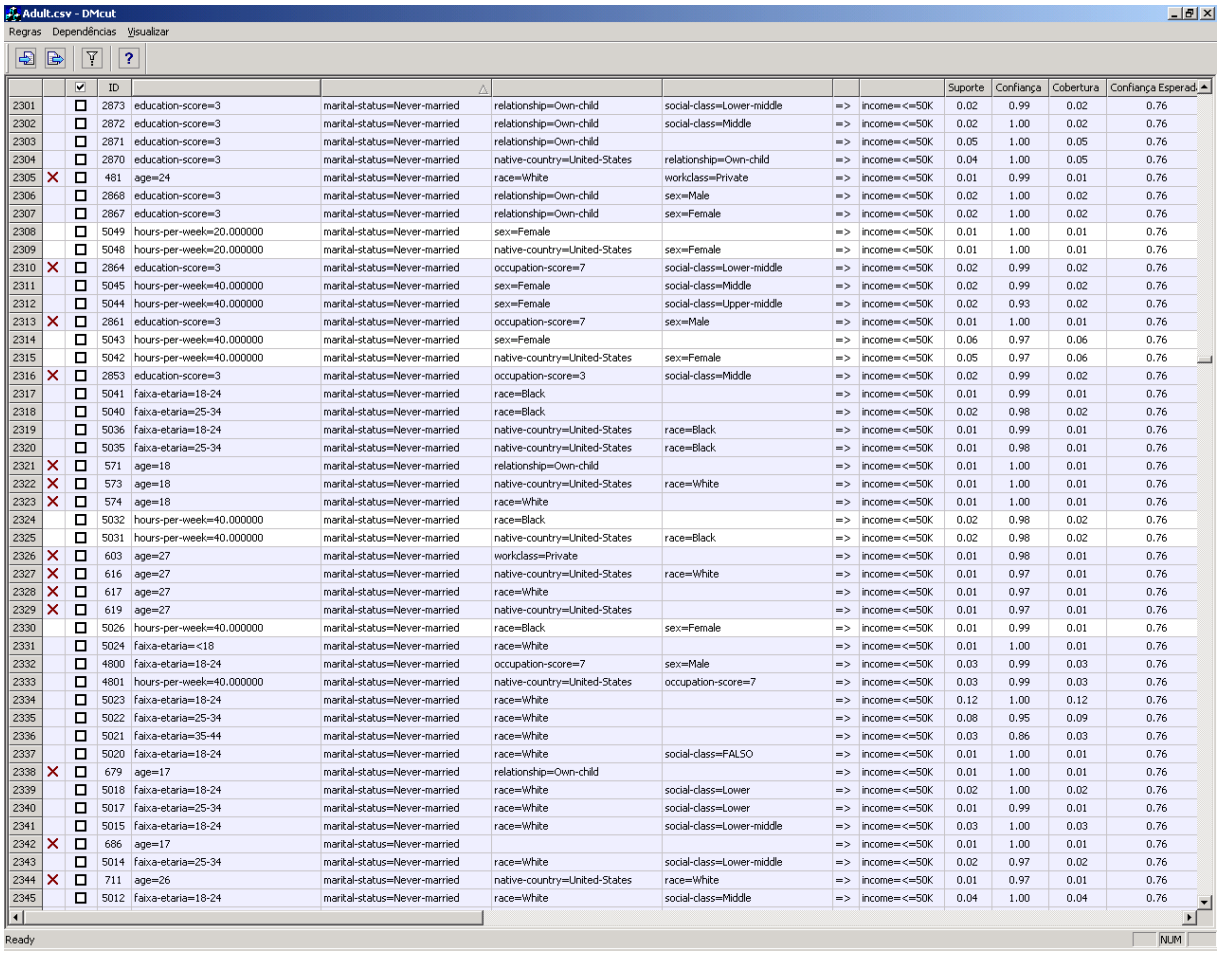
Uma interface gráfica foi desenvolvida para auxiliar os usuários finais na avaliação e identificação das regras de associação interessantes, com a utilização do método proposto em conjunto com as medidas de avaliação das regras. A interface, denominada de Ambiente de Pós-processamento DMcut, foi implementada na linguagem C++, utilizando a biblioteca de classes MFC (Microsoft Foundation Classes), e o compilador Microsoft Visual C++ 6.0.

A interface gráfica consiste num ambiente para exploração de regras de associação. As regras e as medidas de interesse objetivas importadas são visualizadas na forma de uma tabela. O usuário tem a opção de filtrar o que está sendo visualizado, selecionando os itens desejados no antecedente e no conseqüente das regras. A figura 5.2 exibe a tela principal do sistema.

Para aplicar o filtro DMcut, o programa necessita, além de um conjunto de regras de associação no formato de saída do ADDMiner ou Weka, de um arquivo de relações de dependência entre os atributos, como apresentado na seção 5.1.2, e dos valores mínimos para as medidas CRg e CD^{+}.

Ao importar o arquivo de dependências, as regras sobre as quais o filtro DMcut tem ação, ou seja, aquelas que possuem atributos com dependência ou atributos que determinam outros atributos, aparecem em destaque (cor de fundo mais mais escura), como mostra a figura 5.2.

Após a execução do filtro, as regras eliminadas pelo filtro DMcut são indicadas por uma marcação em “x” na segunda coluna da tabela. Os detalhes de um processo de poda ou simplificação, ou ainda, de um processo de generalização/especialização podem ser visualizados com a ajuda de uma janela “popup”, como mostra a janela apresentada na figura 5.3. Esta janela exibe os detalhes de um processo de generalização para determinada regra.



			ID							Suporte	Confiança	Cobertura	Confiança Esperad.
2301		☐	2873	education-score=3	marital-status=Never-married	relationship=Own-child	social-class=Lower-middle	=>	income=<=50K	0.02	0.99	0.02	0.76
2302		☐	2872	education-score=3	marital-status=Never-married	relationship=Own-child	social-class=Middle	=>	income=<=50K	0.02	1.00	0.02	0.76
2303		☐	2871	education-score=3	marital-status=Never-married	relationship=Own-child		=>	income=<=50K	0.05	1.00	0.05	0.76
2304		☐	2870	education-score=3	marital-status=Never-married	native-country=United-States	relationship=Own-child	=>	income=<=50K	0.04	1.00	0.05	0.76
2305	✕	☐	481	age=24	marital-status=Never-married	race=White	workclass=Private	=>	income=<=50K	0.01	0.99	0.01	0.76
2306		☐	2868	education-score=3	marital-status=Never-married	relationship=Own-child	sex=Male	=>	income=<=50K	0.02	1.00	0.02	0.76
2307		☐	2867	education-score=3	marital-status=Never-married	relationship=Own-child	sex=Female	=>	income=<=50K	0.02	1.00	0.02	0.76
2308		☐	5049	hours-per-week=20.000000	marital-status=Never-married	sex=Female		=>	income=<=50K	0.01	1.00	0.01	0.76
2309		☐	5048	hours-per-week=20.000000	marital-status=Never-married	native-country=United-States	sex=Female	=>	income=<=50K	0.01	1.00	0.01	0.76
2310	✕	☐	2864	education-score=3	marital-status=Never-married	occupation-score=7	social-class=Lower-middle	=>	income=<=50K	0.02	0.99	0.02	0.76
2311		☐	5045	hours-per-week=40.000000	marital-status=Never-married	sex=Female	social-class=Middle	=>	income=<=50K	0.02	0.99	0.02	0.76
2312		☐	5044	hours-per-week=40.000000	marital-status=Never-married	sex=Female	social-class=Upper-middle	=>	income=<=50K	0.02	0.93	0.02	0.76
2313	✕	☐	2861	education-score=3	marital-status=Never-married	occupation-score=7	sex=Male	=>	income=<=50K	0.01	1.00	0.01	0.76
2314		☐	5043	hours-per-week=40.000000	marital-status=Never-married	sex=Female		=>	income=<=50K	0.06	0.97	0.06	0.76
2315		☐	5042	hours-per-week=40.000000	marital-status=Never-married	native-country=United-States	sex=Female	=>	income=<=50K	0.05	0.97	0.06	0.76
2316	✕	☐	2853	education-score=3	marital-status=Never-married	occupation-score=3	social-class=Middle	=>	income=<=50K	0.02	0.99	0.02	0.76
2317		☐	5041	faixa-etaria=18-24	marital-status=Never-married	race=Black		=>	income=<=50K	0.01	0.99	0.01	0.76
2318		☐	5040	faixa-etaria=25-34	marital-status=Never-married	race=Black		=>	income=<=50K	0.02	0.98	0.02	0.76
2319		☐	5036	faixa-etaria=18-24	marital-status=Never-married	native-country=United-States	race=Black	=>	income=<=50K	0.01	0.99	0.01	0.76
2320		☐	5035	faixa-etaria=25-34	marital-status=Never-married	native-country=United-States	race=Black	=>	income=<=50K	0.01	0.98	0.01	0.76
2321	✕	☐	571	age=18	marital-status=Never-married	relationship=Own-child		=>	income=<=50K	0.01	1.00	0.01	0.76
2322	✕	☐	573	age=18	marital-status=Never-married	native-country=United-States	race=White	=>	income=<=50K	0.01	1.00	0.01	0.76
2323	✕	☐	574	age=18	marital-status=Never-married	race=White		=>	income=<=50K	0.01	1.00	0.01	0.76
2324		☐	5032	hours-per-week=40.000000	marital-status=Never-married	race=Black		=>	income=<=50K	0.02	0.98	0.02	0.76
2325		☐	5031	hours-per-week=40.000000	marital-status=Never-married	native-country=United-States	race=Black	=>	income=<=50K	0.02	0.98	0.02	0.76
2326	✕	☐	603	age=27	marital-status=Never-married	workclass=Private		=>	income=<=50K	0.01	0.98	0.01	0.76
2327	✕	☐	616	age=27	marital-status=Never-married	native-country=United-States	race=White	=>	income=<=50K	0.01	0.97	0.01	0.76
2328	✕	☐	617	age=27	marital-status=Never-married	race=White		=>	income=<=50K	0.01	0.97	0.01	0.76
2329	✕	☐	619	age=27	marital-status=Never-married	native-country=United-States		=>	income=<=50K	0.01	0.97	0.01	0.76
2330		☐	5026	hours-per-week=40.000000	marital-status=Never-married	race=Black	sex=Female	=>	income=<=50K	0.01	0.99	0.01	0.76
2331		☐	5024	faixa-etaria=<18	marital-status=Never-married	race=White		=>	income=<=50K	0.01	1.00	0.01	0.76
2332		☐	4800	faixa-etaria=18-24	marital-status=Never-married	occupation-score=7	sex=Male	=>	income=<=50K	0.03	0.99	0.03	0.76
2333		☐	4801	hours-per-week=40.000000	marital-status=Never-married	native-country=United-States	occupation-score=7	=>	income=<=50K	0.03	0.99	0.03	0.76
2334		☐	5023	faixa-etaria=18-24	marital-status=Never-married	race=White		=>	income=<=50K	0.12	1.00	0.12	0.76
2335		☐	5022	faixa-etaria=25-34	marital-status=Never-married	race=White		=>	income=<=50K	0.08	0.95	0.09	0.76
2336		☐	5021	faixa-etaria=35-44	marital-status=Never-married	race=White		=>	income=<=50K	0.03	0.86	0.03	0.76
2337		☐	5020	faixa-etaria=18-24	marital-status=Never-married	race=White	social-class=FALSO	=>	income=<=50K	0.01	1.00	0.01	0.76
2338	✕	☐	679	age=17	marital-status=Never-married	relationship=Own-child		=>	income=<=50K	0.01	1.00	0.01	0.76
2339		☐	5018	faixa-etaria=18-24	marital-status=Never-married	race=White	social-class=Lower	=>	income=<=50K	0.02	1.00	0.02	0.76
2340		☐	5017	faixa-etaria=25-34	marital-status=Never-married	race=White	social-class=Lower	=>	income=<=50K	0.01	0.99	0.01	0.76
2341		☐	5015	faixa-etaria=18-24	marital-status=Never-married	race=White	social-class=Lower-middle	=>	income=<=50K	0.03	1.00	0.03	0.76
2342	✕	☐	686	age=17	marital-status=Never-married			=>	income=<=50K	0.01	1.00	0.01	0.76
2343		☐	5014	faixa-etaria=25-34	marital-status=Never-married	race=White	social-class=Lower-middle	=>	income=<=50K	0.02	0.97	0.02	0.76
2344	✕	☐	711	age=26	marital-status=Never-married	native-country=United-States	race=White	=>	income=<=50K	0.01	0.97	0.01	0.76
2345		☐	5012	faixa-etaria=18-24	marital-status=Never-married	race=White	social-class=Middle	=>	income=<=50K	0.04	1.00	0.04	0.76

Figura 5.2: Tela principal da interface gráfica.

O programa permite a ordenação e a eliminação das regras considerando uma das medidas

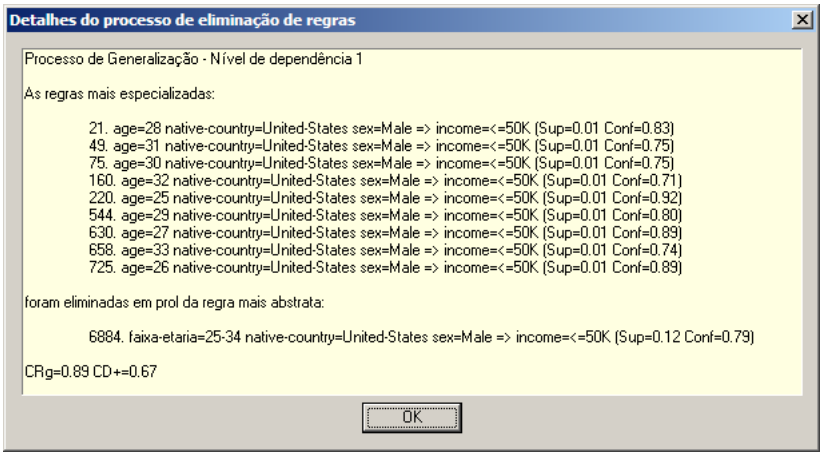


Figura 5.3: Janela da interface com os detalhes de um processo de generalização.

de interesse objetivas disponíveis. Desta forma, o usuário pode combinar o uso do filtro DMcut com as medidas de interesse objetivas das regras.

As regras de associação selecionadas podem ser exportadas para arquivo ASCII.

5.2 Bases de Dados

Esta seção descreve as bases de dados de domínio público utilizadas nos experimentos realizados.

5.2.1 Base de Dados do Censo Americano (Adult)

A base de dados Adult, originalmente conhecida como Census Income Database, encontra-se disponível em [34]. Esta base contém dados extraídos do censo americano pelo U.S. Census Bureau, nos anos de 1994 e 1995. Cada registro apresenta informações relativas a um indivíduo entrevistado pelo censo.

Os dados consistem de 48.842 registros descritos por 15 atributos, que incluem, entre outros, a idade, educação, estado civil e ocupação de cada pessoa. Os dados fornecem um atributo alvo (**income**) que indica se o entrevistado possui renda anual superior a US$ 50.000,00.

5.2.2 Base de Dados da Aterosclerose (STULONG)

A base de dados STULONG [35] contém informações sobre um estudo da doença aterosclerose.

A aterosclerose é uma doença marcada pela formação de placas de gordura que impedem

a passagem do sangue nas artérias. Este problema pode ocorrer em qualquer artéria, sendo as mais freqüentes as coronarianas e cerebrais [36].

Esta doença vascular é uma das mais estudadas em todos os países do mundo, pois é a principal causa de morte no mundo ocidental [36].

Em meados da década de 70, um projeto visando a prevenção da aterosclerose foi iniciado na antiga Tchecoslováquia. Este projeto foi batizado de STULONG - Longitudinal Study of Atherosclerosis Risk Factors [35], e durou ao todo mais de vinte anos.

Um dos principais objetivos da pesquisa era identificar os fatores de risco prevalentes na população sobre a qual a doença incide com maior freqüência - homens de meia idade. 2.370 indivíduos que atendiam esse perfil foram selecionados para os exames iniciais, dos quais 1.417 compareceram. Os resultados desses exames foram armazenados em meio digital e disponibilizados em [35].

Para aplicar o método proposto, selecionamos dados contendo 22 atributos dos 244 atributos que foram de fato coletados nos exames inicias. As informações incluem: dados pessoais (idade, sexo, estado civil, etc), dados sobre a função e a responsabilidade no local de trabalho, hábitos gerais (movimentação, consumo de cigarros e álcool), e resultados de exames físicos e bioquímicos (peso, altura, pressão sanguínea, colesterol, etc).

5.2.3 Base de Dados sobre Contratos de Trabalho (Labor)

A base de dados Labor, disponível em [34], contém informações sobre contratos de trabalho na indústria Canadense. É uma pequena base de dados de 57 registros e 18 atributos. Cada registro apresenta informações relativas a determinado contrato. As informações contidas nesta base incluem: número de horas trabalhadas por semana, principais benefícios oferecidos, como participação no custeio de plano de saúde e de serviços de odontologia, número de dias pagos de férias, etc, além de um atributo alvo (**class**) que indica se o contrato em questão é ou não razoável.

5.2.4 Base de Dados dos Automóveis (Car)

A base de dados Car, disponível em [34], contém informações sobre uma avaliação da aceitabilidade de automóveis baseada no preço e nas características técnicas do veículo. A base consiste de 1.728 registros, cada registro representando as características e a avaliação de um carro, com 7 atributos no total. As informações contidas nesta base incluem: preço, custo da manuntenção,

número de portas, capacidade, tamanho do bagageiro, a segurança (baixa, média ou alta), além de um atributo alvo (**class**) que indica a aceitabilidade do carro.

5.3 Pré-processamento dos Dados

Neste estudo, simulamos a análise dos dados considerando possíveis domínios de aplicação das bases de dados selecionadas. Escolhemos o contexto de uma empresa de seguros de saúde para minerar regras sobre a base STULONG; o uso da base Adult por usuários de uma empresa de marketing; a base Labor sendo utilizada por uma pessoa buscando informações sobre empregabilidade e o mercado de trabalho; e, por último, o uso da base Car com o objetivo de melhorar as vendas de uma concessionária de automóveis.

Desta forma, na fase de pré-processamento dos dados, realizamos a criação de novos atributos a partir dos atributos originais, tal como achamos que seria necessário em uma análise real sobre os dados inseridos nesses contextos.

A seguir serão apresentados os atributos criados para cada base de dados, e as relações de dependência entre os atributos resultantes desse processo.

Base de dados Adult

Para a base de dados Adult criamos o atributo **Classe-social** a partir dos atributos originais **education** e **ocupation**, referentes ao nível de escolaridade e a ocupação principal do entrevistado. O nível de escolaridade e a ocupação são considerados fatores socio-econômicos e são utilizados pelo índice ISP (Index of Social Position) [37] para determinar a classe social de uma pessoa.

Com a utilização do índice ISP [37], a classe social é definida da seguinte maneira: a escolaridade e a ocupação são pontuadas de acordo com as tabelas 5.2 e 5.3, respectivamente, e aplicadas na fórmula:

$$\text{Pontuação ISP } = (\text{Pontuação ocupação } * \ 7) + (\text{Pontuação escolaridade } * \ 4) \qquad (5.1)$$

A partir da pontuação ISP, a classe social pode ser definida de acordo com a tabela 5.1.

A classe social tem um papel importante no comportamento do consumidor, e é amplamente utilizada para o desenvolvimento de estratégias de marketing [38].

Faixa da pontuação ISP	Classe social
11-17	Alta
18-31	Média-alta
32-47	Média
48-63	Média-baixa
64-77	Baixa

Tabela 5.1: Sistema de classificação da classe-social

Escolaridade	Pontuação
Masters degree(MA-MS-MEng-MEd-MSW-MBA)	1
Doctorate degree(PhD-EdD)	1
Prof school degree(MD-DDS-DVM-LLB-JD)	1
Bachelors degree(BA-AB-BS)	2
Associates degree-occup/vocational	2
Associates degree-academic program	2
Some college but no degree	3
High school graduate	4
10th grade	5
11th grade	5
12th grade no diploma	5
7th and 8th grade	6
9th grade	6
Children	7
5th or 6th grade	7
Less than 1st grade	7
1st 2nd 3rd or 4th grade	7

Tabela 5.2: Pontuação da escolaridade

A relação de dependência entre os atributos **ISP**, **education** e **ocupation** é de grau um, e pode ser representada em símbolos por: {education, ocupation} $\xrightarrow{1}$ {ISP}. A relação de dependência entre os atributos **ISP** e **Classe-social** também é de grau um, e pode ser representada em símbolos por: {ISP} $\xrightarrow{1}$ {Classe-social}.

Já a relação de dependência dos atributos **Classe-social**, **education** e **ocupation** é de grau dois, uma vez que existe um domínio intermediário referente à pontuação ISP. Esta relação é representada em símbolos por: {education, ocupation} $\xrightarrow{2}$ {Classe-social}.

Adicionalmente, criamos o atributo discreto **Faixa-etária** a partir do atributo contínuo **age**. As faixas foram divididas considerando grupos de pessoas que vivenciaram em comum um ambiente social, político, histórico, e econômico, também considerado como uma estratégia de marketing em [38]. A relação de dependência dos atributos **Faixa-etária** e **age** é representada em símbolos por: {age} $\xrightarrow{1}$ {Faixa-etária}.

Ocupação	**Pontuação**
Executive admin and managerial	1
Professional specialty	2
Adm support including clerical	3
Armed Forces	3
Sales	4
Precision production craft & repair	5
Technicians and related support	5
Farming forestry and fishing	6
Protective services	6
Machine operators assmblrs & inspctrs	6
Other service	7
Handlers equip cleaners etc	7
Transportation and material moving	7
Private household services	7

Tabela 5.3: Pontuação da ocupação

A figura 5.4 apresenta o arquivo das relações de dependências para esta base de dados.

```
@atributo ISP int
@dependencia (education,ocupation) {;(ocupation ∗ 7) + (education ∗
4):s}

@atributo Classe-social string
@dependencia (ISP) {ISP ≥ 11 and ISP ≤ 17:Alta:n,ISP ≥ 18 and ISP ≤
31 :Média-alta:n,ISP ≥ 32 and ISP ≤ 47:Média:n,ISP ≥ 48 and ISP ≤
63:Média-baixa:n,ISP ≥ 64 and ISP ≤ 77:Baixa:n;<vazio>:s}

@atributo Faixa-etária string
@dependencia (age) {age < 18:<18:n,age ≥ 18 and age ≤ 24 :18-24:n,age
≥ 25 and age ≤ 34:25-34:n,age ≥ 35 and age ≤ 44:35-44:n,age ≥
45 and age ≤ 54:45-54:n,age ≥ 55 and age ≤ 64:55-64:n,age ≥
65:≥65:n;<vazio>:s}
```

Figura 5.4: Arquivo das relações de dependência para a base de dados Adult

Base de dados STULONG

Para a base de dados STULONG criamos o atributo **IMC** (Índice de Massa Corporal), calculado a partir dos atributos originais **weight** e **heigth**, que indicam o peso e a altura do paciente, respectivamente. O IMC é calculado através da fórmula $IMC = Peso/Altura^2$. A discretização do atributo **IMC** deu origem ao atributo **Status**.

Em seguida, o atributo **Faixa-etária** foi criado como uma discretização do atributo **age**, que fornece a idade do paciente.

O IMC é utilizado por médicos e pesquisadores que estudam a obesidade, e está altamente correlacionado com o risco de doenças como a diabetes e doenças cardíacas [39]. A informação da faixa-etária é utilizada por empresas de seguro de saúde para encaixar os clientes em seus planos. Nós consideramos um caso real de faixas-etárias utilizadas por uma empresa de plano de saúde.

Na figura 5.5 é apresentado o arquivo das relações de dependência para esta base de dados.

```
@atributo IMC real
@dependencia (weight, heigth) {weight=<vazio>|heigth=<vazio>:<vazio>:
n; weight/(heigth*heigth):s}

@atributo Status string
@dependencia (IMC) {IMC ≤ 18.5 :abaixo do peso:n, IMC > 18.5 and
IMC ≤ 24.9 :no peso normal:n, BMI > 24.9 and IMC ≤ 29.9 :acima
do peso:n, IMC > 29.9 and IMC ≤ 34.9 :obeso:n, IMC > 39.9 :obeso
mórbido:n; <vazio>:n}

@atributo Faixa-etária string
@dependencia (age) {age ≤ 18 :0-18:n, age > 18 and age ≤ 23
:19-23:n, age > 23 and age ≤ 28 :24-28:n, age > 28 and age ≤ 33
:29-33:n, age > 33 and age ≤ 38 :34-38:n, age > 38 and age ≤ 43
:39-43:n, age > 43 and age ≤ 48 :44-48:n, age > 48 and age ≤
53 :49-53:n, age > 53 and age ≤ 58 :54-58:n, age > 58 :≥59:n;
<vazio>:n}
```

Figura 5.5: Arquivo das relações de dependência para a base de dados STULONG.

As relações de dependência desta base de dados são representadas em símbolos por: {weight, heigth} $\stackrel{1}{\rightarrow}$ {IMC}, {weight, heigth} $\stackrel{2}{\rightarrow}$ {Status}, {IMC} $\stackrel{1}{\rightarrow}$ {Status}, e {age} $\stackrel{1}{\rightarrow}$ {Faixa-etária}.

Base de dados Labor

Para a base de dados Labor, nós criamos três atributos novos: **Benefício-saúde**, **Ausência-remunerada** e **Aumento-salário2**.

Os dois primeiros atributos são referentes aos benefícios oferecidos pela empresa. O atributo **Benefício-saúde** foi criado a partir dos atributos originais **contribution-to-dental-plan** e **contribution-to-health-plan**, que indicam a participação da empresa no custeio dos serviços de odontologia e no custeio do plano de saúde com cobertura médica, respectivamente. Este atributo assume os valores *sim*, se a empresa tiver alguma participação (50% ou total) no custeio dos serviços de odontologia, e custear 100% do plano de saúde com cobertura médica, no caso contrário, o atributo assume o valor *não*.

O atributo **Ausência-remunerada** indica se a empresa remunera o funcionário no caso de ausência. A ausência pode ser por motivos de férias, ou inaptidão para o trabalho. O atributo assume o valor *sim* no caso da empresa conceder férias remuneradas correspondente ao número de dias médio ou acima da média concedida por outras empresas, indicado pelo atributo **vacation**, e remunera o funcionário no caso de inaptidão para o trabalho, este último indicado pelo atributo **longterm-disability-assistance**.

Adicionalmente, criamos o atributo **Aumento-salário2** obtido através dos atributos originais **wage-increase-first-year** e **wage-increase-second-year**, que indicam, respectivamente, o aumento de salário no primeiro e no segundo ano de serviço. O atributo **Aumento-salário2** indica o aumento de salário total nesses dois primeiros anos.

A figura 5.7 apresenta o arquivo das relações de dependências para esta base de dados.

```
@atributo Benefício-saúde string
(contribution-to-dental-plan, contribution-to-health-plan)
contribution-to-dental- plan == ? or contribution-to-health-plan
== ? :?:n,contribution-to-dental-plan != none and
contribution-to-health-plan == full :sim:n;não:n

@atributo Ausência-remunerada string
(vacation,longterm-disability-assistance) vacation == ? or
longterm-disability -assistance == ? :?:n,vacation != below_average
and longterm-disability-assistance == yes :sim:n;não:n

@atributo Aumento-salário2 real
@dependencia (wage-increase-first-year,wage-increase-second-year) ;
wage-increase-first-year + wage-increase-second-year :s
```

Figura 5.6: Arquivo das relações de dependência para a base de dados Labor

As relações de dependência desta base de dados são representadas em símbolos por: {contribution-to-dental-plan,contribution-to-health-plan} $\xrightarrow{1}$ {Benefício-saúde}, {vacation, longterm-disability-assistance} $\xrightarrow{1}$ {Ausência-remunerada}, e {wage-increase-first-year, wage-increase-second-year} $\xrightarrow{1}$ {Aumento-salário2}.

Base de dados Car

Para a base de dados Car, nós criamos o atributo **Conforto** (baixo, médio, alto) a partir dos atributos originais **doors**, que indica o número de portas do carro, e **lug_boot**, que indica o tamanho do bagageiro. Este atributo representa uma avaliação do conforto que o carro proporciona, uma característica importante para a aceitabilidade de um carro. O conforto é considerado *baixo* se o carro for de 2 ou 3 portas, ou possui o bagageiro pequeno. O conforto é *alto* se o

carro possuir o bagageiro espaçoso e tiver mais de 4 portas. Nos demais casos o conforto é considerado *médio*.

A figura 5.6 apresenta o arquivo das relações de dependências para esta base de dados.

```
@atributo Conforto string
@dependencia (doors, lug_boot) {doors == 3 :baixo:n,doors == 4 and
lug_boot == med :médio:n,doors == 4 and lug_boot == big :alto:n,doors
== 5-more and lug_boot == med :médio:n,doors == 5-more and lug_boot ==
big :alto:n,doors == 2:baixo:n,lug_boot == small:baixo:n;<vazio>:n}
```

Figura 5.7: Arquivo das relações de dependência para a base de dados Car

A relação de dependência desta base de dados é representada em símbolos por: {doors, lug_boot} $\xrightarrow{1}$ {Conforto}.

A tabela 5.4 exibe as características gerais das bases de dados selecionadas. A primeira coluna desta tabela apresenta os nomes das bases de dados. Enquanto a segunda coluna fornece o número de atributos originais, o número de registros de cada base de dados é apresentado pela terceira coluna. A quarta coluna apresenta os nomes dos novos atributos criados. A última coluna indica o nome do atributo alvo.

Base de dados	**# Atributos**	**# Registros**	**Atributos criados**	**Atributo alvo**
Adult	15	32.561	ISP Classe-social Faixa-etária	income
STULONG	22	1.417	IMC Status Faixa-etária	-
Labor	18	57	Aumento-salário2 Ausência-remunerada Benefício-saúde	class
Car	7	1.728	Conforto	class

Tabela 5.4: Características das bases de dados selecionadas.

5.4 Mineração das Regras de Associação

Com o auxílio do aplicativo ADDMiner foram geradas as regras de associação utilizadas nos experimentos. Os valores mínimos de Suporte e Confiança utilizados são apresentados na tabela 5.5.

A primeira coluna desta tabela apresenta os nomes das bases de dados. A segunda coluna indica o número máximo de elementos que uma regra pode ter. A última coluna da tabela é

dedicada ao número total de regras mineradas para cada uma das configurações.

Ao especificarmos os valores de Suporte, consideramos o Suporte mínimo tal que fossem mineradas regras com os atributos criados e com os atributos originais utilizados para a criação dos novos atributos. Observe que esta medida foi necessária, uma vez que o método proposto tem ação somente sobre as regras que apresentam atributos que possuem relação de dependência.

Para todos os experimentos utilizamos o valor *default* da Confiança no aplicativo ADDMiner.

Base de dados	**# Elementos**	**Suporte**	**Confiança**	**# Regras mineradas**
Adult	5	0,01	0,70	7.015
STULONG	3	0,05	0,70	2.798
Labor	5	0,05	0,70	5.652
Car	3	0,05	0,70	140

Tabela 5.5: Mineração das regras de associação para as bases de dados selecionadas.

Capítulo 6

Análise dos Resultados

Neste capítulo apresentamos e discutimos os resultados obtidos com o filtro de regras de associação DMcut sobre as bases de dados selecionadas.

Para cada base de dados utilizada, selecionamos alguns exemplos de regras que foram eliminadas por apresentarem as formas das expressões 4.1, 4.2 e 4.3. Adicionalmente, apresentamos e discutimos alguns processos de generalização e especialização das regras.

Os resultados são apresentados no formato apresentado pela tabela 6.1.

A primeira linha é utilizada para identificar o tipo do processo (Poda de regras da forma das expressões 4.1, 4.2 ou 4.3, Processo de generalização ou Processo de especialização). A primeira coluna (**Id**) apresenta o identificador da regra. Os itens que formam o antecedente da regra aparecem na segunda coluna (**Antecedente**) separados por vírgulas, enquanto o conseqüente da regra é apresentado na terceira coluna (**Conseqüente**). As últimas duas colunas (**Sup**) e (**Conf**) fornecem os valores para as medidas Suporte e Confiança.

Tipo do Processo				
Id	**Antecedente**	**Conseqüente**	**Sup**	**Conf**
xxx	(xxx=xxx), . . . (xxx=xxx)	(xxx=xxx)	x,xx	x,xx
xxx	(xxx=xxx), . . . (xxx=xxx)	(xxx=xxx)	x,xx	x,xx
xxx	(xxx=xxx), . . . (xxx=xx)	(xxx=xxx)	x,xx	x,xx
⋮	⋮	⋮	⋮	

Tabela 6.1: Formato de apresentação dos resultados

O filtro DMcut foi aplicado considerando-se os valores de CRg e CD^+ iguais ao valor mínimo da Confiança especificada na fase de mineração, ou seja, 70%. No restante deste capítulo, faremos referência a este valor como uma medida da confiança mínima do usuário. Consideramos o grau máximo de dependência igual a 2.

6.1 Base de dados do Censo Americano

Com a base de dados do censo americano foram eliminadas no total 1.212 regras. Esse valor representa 17,3% do total de regras mineradas e 19,0% do total das regras que apresentavam dependência (uma regra apresenta dependência quando possui um atributo dependente ou um atributo que determina outro atributo da base).

Do conjunto de regras mineradas, 321 regras que apresentavam a forma da expressão 4.2 foram eliminadas. As redundâncias das regras nas formas 1 e 3 não se aplicam para esta base de dados, uma vez que estas formas incluem um atributo dependente no conseqüente das regras, e o atributo alvo (**income**) desta base não é um atributo dependente.

A tabela 6.2 fornece dois exemplos de regras que apresentavam a forma da expressão 4.2.

Poda de regras da forma da expressão 4.2				
Id	**Antecedente**	**Conseqüente**	**Sup**	**Conf**
67	(age=30), (Faixa-etária=25-34), (workclass=Private)	(income=<=50K)	0,02	0,81
2518	(education-score=1), (Faixa-etária=45-54), (occupation-score=2), (Classe-social=Média-alta)	(income=>50K)	0,01	0,73

Tabela 6.2: Regras da base Adult na forma da expressão 4.2

No primeiro exemplo da tabela 6.2, a regra de **Id** 67, que possui a seguinte interpretação: os entrevistados de 30 anos de idade, na faixa dos 25 a 34 anos, e pertencem à classe de trabalho privada, possuem uma receita anual inferior a U$ 50.000,00. A informação sobre a faixa-etária é redundante com o dado sobre a idade do entrevistado. A regra pôde ser eliminada porque sua forma mais simplificada, sem o item (Faixa-etária=25-34), estava presente no conjunto das regras mineradas.

O segundo exemplo da mesma tabela apresenta a regra de **Id** 2518. Esta regra possui a seguinte interpretação: os entrevistados na faixa dos 45 a 54 anos, que possuem os níveis de escolaridade e ocupação referentes às pontuações 1 e 2, respectivamente, possuem uma receita

anual superior a U$ 50.000,00. A informação sobre a classe social é redundante com os dados sobre a escolaridade e ocupação do entrevistado. A regra pôde ser eliminada porque sua forma mais simplificada, sem o item (Classe-social=Média-alta), estava presente no conjunto das regras mineradas.

Os 388 processos de generalização eliminaram ao todo 877 regras. Um exemplo de generalização baseada na relação de dependência {age} $\rightarrow$ {Faixa-etária} é apresentado na tabela 6.3. A tabela 6.4 apresenta um exemplo de generalização baseada na relação de dependência {education-score, occupation-score} $\stackrel{2}{\rightarrow}$ {Classe-social}.

De acordo com a tabela 6.3, existe uma regra mais específica para cada uma das idades que mapeiam para o valor "25-34" do atributo Faixa-etária, exceto para a idade 34. Neste caso, a medida CRg indica a probabilidade do entrevistado ter entre 25 e 33 anos, dado que ele está na faixa dos 25 a 34 anos, é do sexo masculino e nativo dos EUA. O valor calculado para a medida CRg foi 89%, um valor acima da confiança mínima do usuário. Desta forma, o processo de generalização foi aplicado, e as regras relativas ao atributo **age** foram eliminadas em prol da regra mais geral com **Faixa-etária**.

No exemplo da tabela 6.4, de acordo com a medida CRg, a probabilidade do entrevistado possuir a escolaridade e ocupação que aparecem nas regras mais específicas, dado que ele nunca foi casado, pertence à classe social média, e à classe de trabalho privada, é de 52%, um valor abaixo da confiança mínima do usuário. Entretanto, a medida CD^+ igual a 94% indica que a regra mais geral continua válida para os demais casos de escolaridade e ocupação dos mesmos entrevistados. Assim, no processo de generalização, as regras relativas aos atributos **education-score** e **occupation-score** foram eliminadas em prol da regra mais geral com **Classe-social**.

Dentre as regras mais gerais, 14 foram eliminadas através dos processos de especialização. A tabela 6.5 apresenta um exemplo de um processo especialização com base na relação de dependência {education-score, occupation-score} $\stackrel{2}{\rightarrow}$ {Classe-social}.

No exemplo da tabela 6.5, a medida CRg possui um valor abaixo do valor de confiança mínima do usuário. Ou seja, a probabilidade de um entrevistado possuir a escolaridade e a ocupação que aparecem nas regras mais específicas, dado que ele pertecence à classe social média, é baixa. O valor da medida CD^+ igual 63% indica que a regra mais geral não é válida se considerarmos apenas os entrevistados que pertencem à classe social média, e que não possuem a escolaridade e ocupação presentes nas regras mais específicas. Desta forma, foi aplicado o processo de especialização, onde a regra referente à classe social foi eliminada em prol das regras mais específicas com escolaridade e ocupação.

Processo de generalização CRg=0,89				
Id	**Antecedente**	**Conseqüente**	**Sup**	**Conf**
As regras mais especializadas:				
21	(age=28), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,83
49	(age=31), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,75
75	(age=30), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,75
160	(age=32), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,71
220	(age=25), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,92
544	(age=29), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,80
620	(age=27), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,89
658	(age=33), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,74
725	(age=26), (sex=male), (native-country=United-States)	(income=<=50K)	0,01	0,89
Foram eliminadas em prol da regra mais geral:				
6884	(Faixa-etária=25-34), (sex=male), (native-country=United-States)	(income=<=50K)	0,12	0,79

Tabela 6.3: Processo de generalização baseado na relação de dependência {age} → {Faixa-etária} da base Adult.

6.2 Base de dados da Aterosclerose

Com a base de dados da aterosclerose foram eliminadas no total 371 regras. Esse valor representa 13,26% do total de regras mineradas e 39,81% do total das regras que apresentavam dependência.

Do conjunto de regras mineradas, 237 regras que apresentavam a forma das expressões 4.1, 4.2 e 4.3 foram eliminadas. A tabela 6.6 fornece um exemplo para cada forma de redundância

Processo de generalização CRg=0,52 CD$^+$=0,94				
Id	**Antecedente**	**Conseqüente**	**Sup**	**Conf**
As regras mais especializadas:				
829	(education-score=3), (Estado-civil=nunca-foi-casado), (occupation-score=3), (Classe-trabalho=privada)	(income=<=50K)	0,01	0,99
830	(education-score=3), (marital-status=Never-married), (occupation-score=4), (workclass=Private)	(income=<=50K)	0,01	0,99
964	(education-score=4), (marital-status=Never-married), (occupation-score=3), (workclass=Private)	(income=<=50K)	0,01	0,99
Foram eliminadas em prol da regra mais geral:				
1518	(marital-status=Never-married), (Classe-social=média), (workclass=Private)	(income=<=50K)	0,07	0,97

Tabela 6.4: Processo de generalização baseado na relação de dependência {education-score, occupation-score} $\xrightarrow{2}$ {Classe-social} da base Adult.

Processo de especialização CRg=0,69 CD$^+$=0,63				
Id	**Antecedente**	**Conseqüente**	**Sup**	**Conf**
A regra mais geral:				
6995	(Classe-social=média)	(income=<=50K)	0,20	0,77
Foi eliminada em prol das regras mais especializadas:				
2946	(education-score=3), (occupation-score=3)	(income=<=50K)	0,04	0,89
2958	(education-score=3), (occupation-score=4)	(income=<=50K)	0,02	0,79
2972	(education-score=3), (occupation-score=5)	(income=<=50K)	0,03	0,72
3422	(education-score=4), (occupation-score=3)	(income=<=50K)	0,04	0,88
3439	(education-score=4), (occupation-score=4)	(income=<=50K)	0,03	0,81

Tabela 6.5: Processo de especialização baseado na relação de dependência {education-score, occupation-score} $\xrightarrow{2}$ {Classe-social} da base Adult.

eliminada nessa fase.

O primeiro exemplo da tabela 6.6 apresenta a regra de **Id** 1134, da forma da expressão 4.1, com a seguinte interpretação: os indivíduos casados, de IMC igual a 28 estão acima do peso. Esta regra é redundante porque pode ser deduzida a partir da relação de dependência entre os

Poda de regras da forma das expressões 4.1, 4.2 e 4.3				
Id	**Antecedente**	**Conseqüente**	**Sup**	**Conf**
1134	(IMC=28), (marital_status=married)	(Status=acima do peso)	0,06	1,00
57	(age=41), (Faixa-etária=39-43)	(marital_status=married)	0,05	0,87
2142	(IMC=24)	(phys_act_after_job=moderate activity), (Status=normal)	0,09	0,71

Tabela 6.6: Regras da base STULONG nas formas das expressões 4.1, 4.2 e 4.3

atributos **IMC** e **Status**, da qual já se sabe que os indivíduos de IMC igual a 28 estão acima do peso.

A segunda regra da tabela 6.6, a regra de **Id** 57, da forma da expressão 4.2, possui a seguinte interpretação: os indivíduos na faixa dos 39 a 43 anos de idade, que têm 41 anos, são casados. A informação sobre a faixa-etária do indivíduo é redundante com o dado sobre a idade. A regra pôde ser eliminada porque sua forma mais simplificada, sem o item (Faixa-etária=39-43), estava presente no conjunto das regras mineradas.

No terceiro exemplo da mesma tabela, de **Id** 2142, apresenta a forma da expressão 4.3, e pode ser interpretada da seguinte maneira: os indivíduos que possuem o IMC igual a 24, praticam atividade física moderada após o trabalho e estão no peso normal. A informação sobre o status dos indivíduos é redundante com a informação sobre o IMC. A regra pôde ser eliminada porque sua forma mais simplificada, sem o item (Status=normal), estava presente no conjunto das regras mineradas.

Os 68 processos de generalização eliminaram ao todo 99 regras. Um exemplo de generalização baseada na relação de dependência {age} $\stackrel{1}{\rightarrow}$ {Faixa-etária} é apresentado na tabela 6.7. Em seguida, a tabela 6.8 apresenta um exemplo de generalização baseada na relação de dependência {IMC} $\stackrel{2}{\rightarrow}$ {Status}.

Podemos observar na tabela 6.7 que a regra mais geral, referente ao atributo **Faixa-etária**, possui três regras mais específicas com o atributo **age**, para os valores: 46, 47 e 48. A medida CRg indica a probabilidade do paciente ter entre 46 e 48 anos, dado que ele está na faixa dos 44 aos 48 anos e é casado. O valor calculado para a medida CRg foi 68%, um valor abaixo da confiança mínima do usuário. Contudo, a medida CD^{+} igual a 77% apresenta um valor acima da confiança mínima do usuário, o que significa que a regra continua válida para os demais indivíduos casados com idade entre 44 e 48 anos. Desta forma, o processo de generalização foi aplicado, e as regras relativas ao atributo **age** foram eliminadas em prol da regra mais geral com o atributo **Faixa-etária**.

Processo de generalização CRg=0,68 CD$^+$=0,77				
Id	**Antecedente**	**Conseqüente**	**Sup**	**Conf**
As regras mais especializadas:				
80	(age=46), (marital_status=married)	(phys_act_after_job=moderate activity)	0,05	0,70
106	(age=47), (marital_status=married)	(phys_act_after_job=moderate activity)	0,07	0,74
65	(age=48), (marital_status=married)	(phys_act_after_job=moderate activity)	0,06	0,75
Foram eliminadas em prol da regra mais geral:				
540	(Faixa-etária=44-48), (marital_status=married)	(phys_act_after_job=moderate activity)	0,26	0,74

Tabela 6.7: Processo de generalização baseado na relação de dependência {age} $\xrightarrow{1}$ {Faixa-etária} da base STULONG.

Processo de generalização CRg=0,84				
Id	**Antecedente**	**Conseqüente**	**Sup**	**Conf**
As regras mais especializadas:				
2135	(IMC=22)	(phys_act_after_job=moderate activity)	0,05	0,71
2131	(IMC=23)	(phys_act_after_job=moderate activity)	0,08	0,72
2139	(IMC=24)	(phys_act_after_job=moderate activity)	0,09	0,71
Foram eliminadas em prol da regra mais geral:				
2147	(Status=no peso normal)	(phys_act_after_job=moderate activity)	0,27	0,71

Tabela 6.8: Processo de generalização baseado na relação de dependência {IMC} $\xrightarrow{2}$ {Status} da base STULONG.

No exemplo da tabela 6.8, existem três regras mais específicas que apresentam os valores 22, 23 e 24 do atributo **IMC**, que mapeiam para o valor “no peso normal” do atributo **Status**. De acordo com a medida CRg, a probabilidade do paciente ter o IMC igual a 22, 23 ou 24, dado que ele está no peso normal, é de 84%. Esse valor é acima do valor de confiança mínima do usuário, o que resultou no processo de generalização das regras.

Dentre as regras mais gerais, 12 foram eliminadas através dos processos de especialização. A tabela 6.9 apresenta um exemplo de um processo especialização com base na relação de dependência {IMC} $\xrightarrow{2}$ {Status}.

No exemplo da tabela 6.9, a medida CRg possui um valor abaixo do valor de confiança mínima do usuário. A probabilidade de um paciente possuir o IMC igual a 25 ou a 27, dado que ele bebe cerveja, e está no peso normal, é de 50%. O valor 68% da medida CD$^+$, abaixo da confiança mínima do usuário, indica que a regra mais geral não é válida se considerarmos os indivíduos que apresentam essas características, e o IMC diferente de 25 e 27. Desta forma, foi aplicado o processo de especialização, onde a regra referente ao atributo **Status** foi eliminada

Processo de especialização CRg=0,50 CD$^+$=0,68				
Id	**Antecedente**	**Conseqüente**	**Sup**	**Conf**
A regra mais geral:				
2428	(beer=yes), (Status=acima do peso)	(job_transp_duration=around 1/2 hour)	0,25	0,70
Foi eliminada em prol das regras mais especializadas:				
2426	(IMC=25) (beer=yes)	(job_transp_duration=around 1/2 hour)	0,07	0,70
2424	(IMC=27) (beer=yes)	(job_transp_duration=around 1/2 hour)	0,05	0,75

Tabela 6.9: Processo de especialização baseado na relação de dependência {IMC} $\stackrel{2}{\rightarrow}$ {Status} da base STULONG.

em prol das regras mais específicas com o atributo **IMC**.

6.3 Base de dados sobre Contratos de Trabalho

Com a base de dados Labor foram eliminadas no total 380 regras. Esse valor representa 6,72% do total de regras mineradas e 7,34% do total das regras que apresentavam dependência.

Do conjunto de regras mineradas, 99 regras que apresentavam a forma da expressão 4.2 foram eliminadas. As redundâncias das regras nas formas 1 e 3 não se aplicam para esta base de dados, uma vez que estas formas incluem um atributo dependente no conseqüente das regras, e o atributo alvo (**class**) desta base não é um atributo dependente.

A tabela 6.10 fornece dois exemplos de regras que apresentavam a forma da expressão 4.2.

Poda de regras da forma da expressão 4.2				
Id	**Antecedente**	**Conseqüente**	**Sup**	**Conf**
1904	(Aumento-salário2=7.50), (wage-increase-first-year=3.50), (wage-increase-second-year=4.00)	(class=good)	0,07	0,80
2163	(Benefício-saúde=não), (contribution-to-dental-plan=none), (contribution-to-health-plan=none), (wage-increase-second-year=0.00)	(class=bad)	0,05	1,00

Tabela 6.10: Regras da base Labor na forma da expressão 4.2

No primeiro exemplo da tabela 6.10, a regra de **Id** 1904 possui a seguinte interpretação: contratos que prevêem aumento total de 7,50% nos dois primeiros anos, sendo 3,50% no primeiro ano, e 4,00% no segundo, possuem uma boa aceitabilidade. A informação sobre o aumento total nos dois primeiros anos é redundante com os dados sobre os aumentos no primeiro

e no segundo ano. A regra pôde ser eliminada porque sua forma mais simplificada, sem o item (Aumento-salário2=7.50), estava presente no conjunto das regras mineradas.

O segundo exemplo da tabela 6.10 apresenta a regra de **Id** 2163. Esta regra pode ser interpretada da seguinte forma: contratos nos quais a empresa não fornece benefício saúde, que não têm participação da empresa nos custeios dos serviços de odontologia e do plano de saúde, e que não prevê aumento de salário no segundo ano, possuem aceitabilidade baixa. A informação de que a empresa não fornece seguro de saúde é redundante com a informação de não haver serviços de odontologia e plano de saúde com cobertura médica. A regra pôde ser eliminada porque sua forma mais simplificada, sem o item (Benefício-saúde=não), estava presente no conjunto das regras mineradas.

Os 250 processos de generalização eliminaram ao todo 256 regras. A seguir é apresentado um exemplo para cada relação de dependência existente entre os dados. A tabela 6.11 apresenta um exemplo de generalização baseada na relação de dependência {contribution-to-health-plan, contribution-to-dental-plan} $\stackrel{1}{\rightarrow}$ {Benefício-saúde}. A tabela 6.12 apresenta um exemplo de generalização baseada na relação de dependência {vacation,longterm-disability-assistance} $\stackrel{1}{\rightarrow}$ {Ausência-remunerada}. Por último, um exemplo de generalização baseada na relação de dependência {wage-increase-first-year,wage-increase-second-year} $\stackrel{1}{\rightarrow}$ {Aumento-salário2} é apresentado na tabela 6.12.

Processo de generalização CRg=0,80				
Id	**Antecedente**	**Conseqüente**	**Sup**	**Conf**
A regra mais especializada:				
5478	(contribution-to-dental-plan=full), (contribution-to-health-plan=full), (longterm-disability-assistance=yes)	(class=good)	0,07	1,00
Foi eliminada em prol da regra mais geral:				
5504	(Benefício-saúde=sim), (longterm-disability-assistance=yes)	(class=good)	0,07	0,80

Tabela 6.11: Processo de generalização baseado na relação de dependência {contribution-to-health-plan,contribution-to-dental-plan} $\stackrel{1}{\rightarrow}$ {Benefício-saúde} da base Labor.

No processo de generalização da tabela 6.11, a medida CRg indica uma probabilidade de 80% da empresa oferecer total participação no custeio dos serviços de odontologia e no custeio do plano de saúde, dado que a empresa fornece benefício saúde, e remunera o funcionário no caso de inaptidão para o trabalho. Este valor é acima da confiança mínima do usuário. Desta forma, o processo de generalização foi aplicado, e a regra mais específica foi eliminada em prol da regra mais geral com o atributo **Benefício-saúde**.

Processo de generalização CRg=1,00				
Id	**Antecedente**	**Conseqüente**	**Sup**	**Conf**
As regras mais especializadas:				
5411	(longterm-disability-assistance=yes), (vacation=average)	(class=good)	0,05	0,75
5438	(longterm-disability-assistance=yes), (vacation=generous)	(class=good)	0,09	0,83
Foram eliminadas em prol da regra mais geral:				
5649	(Ausência-remunerada=sim),	(class=good)	0,14	0,80

Tabela 6.12: Processo de generalização baseado na relação de dependência {vacation,longterm-disability-assistance} $\stackrel{1}{\rightarrow}$ {Ausência-remunerada} da base Labor.

No exemplo da tabela 6.12, de acordo com a medida CRg, a probabilidade da empresa conceder férias remuneradas correspondente ao número de dias médio ou acima da média concedida por outras empresas, dado que a empresa remunera o funcionário no caso de inaptidão para o trabalho, e emprega a política de ausência remunerada, é de 100%. Assim, no processo de generalização, as regras relativas aos atributos **longterm-disability-assistance** e **vacation** foram eliminadas em prol da regra mais geral com **Ausência-remunerada**.

Processo de generalização CRg=0,43 CD$^+$=0,75				
Id	**Antecedente**	**Conseqüente**	**Sup**	**Conf**
A regra mais especializada:				
1420	(wage-increase-first-year=4.50), (wage-increase-second-year=4.50)	(class=good)	0,05	1,00
Foi eliminada em prol da regra mais geral:				
5636	(Aumento-salário2=9.00)	(class=good)	0,11	0,86

Tabela 6.13: Processo de generalização baseado na relação de dependência {wage-increase-first-year,wage-increase-second-year} $\stackrel{1}{\rightarrow}$ {Aumento-salário2} da base Labor.

No exemplo da tabela 6.13, de acordo com a medida CRg, a probabilidade do contrato prever aumentos de salário de 4,50%, a cada ano, nos dois primeiros anos de serviço, dado que o aumento de salário total nos dois primeiros anos é de 9,00%, é de 43%. Este valor é abaixo da confiança mínima do usuário. Entretanto, a medida CD$^+$, igual a 75%, indica que a regra mais geral continua válida para os demais casos de aumento de salário no primeiro e no segundo ano de serviço dos mesmos contratos. Assim, no processo de generalização, as regras mais específicas foram eliminadas em prol da regra mais geral com **Aumento-salário2**.

Dentre as regras mais gerais, 29 foram eliminadas através dos processos de especialização. A tabela 6.14 apresenta um exemplo de um processo especialização com base na relação de dependência {vacation,longterm-disability-assistance} $\stackrel{1}{\rightarrow}$ {Ausência-remunerada}.

Processo de especialização CRg=0,50 CD$^+$=0,67				
Id	**Antecedente**	**Conseqüente**	**Sup**	**Conf**
A regra mais geral:				
3661	(Ausência-remunerada=não), (cost-of-living-adjustment=none)	(class=bad)	0,09	0,83
Foi eliminada em prol das regras mais especializadas:				
3615	(cost-of-living-adjustment=none), (longterm-disability-assistance=no) (vacation=below_average)	(class=bad)	0,05	1,00

Tabela 6.14: Processo de especialização baseado na relação de dependência {vacation,longterm-disability-assistance} $\xrightarrow{1}$ {Ausência-remunerada} da base Labor.

No exemplo da tabela 6.14, a medida CRg possui um valor abaixo do valor de confiança mínima do usuário. Ou seja, é baixa a probabilidade da empresa em questão não remunerar o funcionário no caso de inaptidão para o trabalho, e conceder férias remuneradas correspondente ao número de dias abaixo da média concedida por outras empresas, dado que o contrato não prevê a remuneração no caso de ausência e ajuda de custo de vida. O valor da medida CD$^+$ igual 67%, também abaixo da confiança mínima do usuário, indica que a regra mais geral não é válida se considerarmos apenas os contratos que não prevêem a remuneração no caso de ausência e ajuda de custo de vida, e que não satisfazem às regras mais específicas (remunera o funcionário no caso de inaptidão para o trabalho, e concede férias remuneradas correspondente ao número de dias na média ou acima da média concedida por outras empresas). Desta forma, foi aplicado o processo de especialização, onde a regra referente ao atributo **Ausência-remunerada** foi eliminada em prol da regra mais específica.

6.4 Base de dados dos Automóveis

Com a base de dados Car foram eliminadas no total 18 regras. Esse valor representa 12,86% do total de regras mineradas e 14,40% do total das regras que apresentavam dependência.

Do conjunto de regras mineradas, 9 regras que apresentavam a forma da expressão 4.2 foram eliminadas. As redundâncias das regras nas formas 1 e 3 não se aplicam para esta base de dados, uma vez que estas formas incluem um atributo dependente no conseqüente das regras, e o atributo alvo (**class**) desta base não é um atributo dependente.

A tabela 6.15 fornece dois exemplos de regras que apresentavam a forma da expressão 4.2.

As duas regras da tabela 6.15 possuem a interpretação parecida: carros com o conforto baixo, de 2 portas, que possuem o bagageiro pequeno, para a regra de **Id** 8, ou médio, para

Poda de regras da forma da expressão 4.2				
Id	**Antecedente**	**Conseqüente**	**Sup**	**Conf**
8	(Conforto=baixo), (doors=2), (lug_boot=small)	(class=unacc)	0,11	1,00
10	(Conforto=baixo), (doors=2), (lug_boot=med)	(class=unacc)	0,11	1,00

Tabela 6.15: Regras da base Car na forma da expressão 4.2

a regra de **Id** 10, têm a aceitabilidade ruim. Nas duas regras a informação sobre o conforto baixo é redundante com as informações sobre o número de portas e o tamanho do bagageiro do veículo. Neste caso, as regras foram eliminadas porque suas formas mais simplificadas, sem o item (Conforto=baixo), estavam presentes no conjunto das regras mineradas.

Os 3 processos de generalização eliminaram ao todo 9 regras. A tabela 6.16 apresenta um exemplo de generalização baseada na relação de dependência {doors,lug_boot} $\stackrel{1}{\rightarrow}$ {Conforto}.

Processo de generalização CRg=0,95				
Id	**Antecedente**	**Conseqüente**	**Sup**	**Conf**
As regras mais especializadas:				
7	(doors=2), (lug_boot=small)	(class=unacc)	0,11	1,00
9	(doors=2), (lug_boot=med)	(class=unacc)	0,11	1,00
11	(doors=2), (lug_boot=big)	(class=unacc)	0,11	1,00
27	(doors=3), (lug_boot=small)	(class=unacc)	0,11	1,00
29	(doors=3), (lug_boot=med)	(class=unacc)	0,11	1,00
31	(doors=3), (lug_boot=big)	(class=unacc)	0,11	1,00
44	(doors=4), (lug_boot=small)	(class=unacc)	0,11	1,00
Foram eliminadas em prol da regra mais geral:				
138	(Conforto=baixo),	(class=unacc)	0,78	1,00

Tabela 6.16: Processo de generalização baseado na relação de dependência {doors,lug_boot} $\stackrel{1}{\rightarrow}$ {Conforto} da base Car.

No processo de generalização da tabela 6.16, a medida CRg indica uma probabilidade de 95% do carro apresentar uma das combinações de número de portas e tamanho do bagageiro encontradas nas regras mais específicas, dado que o carro possui o conforto baixo. Este valor

é acima da confiança mínima do usuário. Desta forma, o processo de generalização foi aplicado, e as regras mais específicas foram eliminadas em prol da regra mais geral com o atributo **Conforto**.

6.5 Resumo dos Resultados

A tabela 6.17 apresenta um resumo dos resultados obtidos pelo método proposto sobre as quatro bases de dados selecionadas. As bases de dados são identificadas pelo nome na primeira coluna da tabela. A segunda coluna apresenta o número de regras de associação mineradas. A terceira coluna fornece o número de regras consideradas redundantes e, portanto, eliminadas pelo método proposto. Os percentuais de regras eliminadas sobre o total de regras mineradas e sobre o conjunto de regras que apresentam atributos com relações de dependência encontram-se, respectivamente, na quarta e na quinta colunas.

Base de dados	**# Regras mineradas**	**# Regras eliminadas**	**% Total**	**% Com dep.**
Adult	7.015	1.212	17,30	19,00
STULONG	2.798	371	13,26	39,81
Labor	4.497	380	6,72	7,34
Car	140	18	12,86	14,40

Tabela 6.17: Resumo dos resultados obtidos pelo método proposto sobre as quatro bases de dados selecionadas.

Os números da tabela 6.17 mostram que o filtro DMcut teve ação, eliminando regras redundantes, sobre as quatro bases selecionadas. O filtro DMcut obteve bons percentuais de redução, chegando a quase 40% para a base STULONG. Desta forma, mostramos que o conhecimento prévio das relações de dependência entre os atributos pode ser utilizado para reduzir o tamanho do conjunto de regras a ser apresentado para o usuário.

Contudo, o número de regras eliminadas pode não ser suficiente para que o usuário possa manipular os resultados. No caso da base Adult, por exemplo, um percentual de 19% parece não ser significativo enquanto ainda restam 5.803 regras para serem avaliadas. De fato, o método proposto não é por si só suficiente, e sim um complemento às técnicas existentes. Quando o usuário tem o conhecimento prévio das relações de dependência entre os atributos, o filtro DMcut deve ser utilizado para melhorar a qualidade dos resultados gerados.

6.6 Análise Comparativa entre as Medidas de Interesse Objetivas

Ao generalizar/especializar, o método proposto seleciona as regras mais relevantes para o usuário, eliminando a redundância nas regras mineradas. Este processo reflete o processo humano de abstração dos resultados, no qual somente as regras não interessantes são descartadas.

Nesta seção realizamos uma análise comparativa entre o filtro DMcut e as medidas de interesse objetivas de regras de associação. O objetivo é evidenciar a contribuição específica da abordagem proposta. Nossa hipótese é que o filtro DMcut realiza um corte mais preciso das regras quando comparado às medidas de interesse objetivas, uma vez que o processo de poda é guiado pelo conhecimento prévio do domínio.

Nos experimentos realizados selecionamos quatro das principais medidas de interesse objetivas existentes na literatura: Convicção [17], Especificidade [24], Lift [17] e Novidade [25].

No primeiro experimento, veremos que o uso dessas medidas não exclui a necessidade da abordagem proposta. A tabela 6.18 apresenta os resultados obtidos pelas quatro medidas de interesse sobre as regras mineradas a partir da base de dados Adult, com a utilização dos valores de Suporte e Confiança apresentados na tabela 5.5. Para cada medida de interesse, foram considerados três valores de corte distintos. O resultado é apresentado como se segue. A primeira coluna fornece o nome da medida de interesse utilizada. Na segunda coluna está o valor de corte considerado. A terceira coluna é dedicada ao percentual de regras eliminadas pela medida, enquanto o número de regras deixadas (consideradas interessantes segundo a medida) é apresentado na quarta coluna. A última coluna apresenta o percentual de regras eliminadas pelo filtro DMcut, dentre as regras consideradas interessantes segundo a medida de interesse.

Medida	Corte	% Regras eliminadas	# Regras deixadas	% DMcut
Especificidade	0,95	5,86%	6603	17,91%
	0,97	11,19%	6230	18,49%
	0,99	38,16%	4338	17,99%
Lift	1,20	41,71%	4089	16,70%
	1,30	72,83%	1905	17,75
	1,40	93,92%	426	2,86%
Novidade	0,00	9,15%	6373	17,00%
	0,01	91,81%	574	1,86%
	0,02	98,21%	125	0%
Convicção	1,10	17,57%	5782	14,25%
	1,20	21,17%	5529	16,56%
	1,30	24,54%	5293	20,58%

Tabela 6.18: Resultados obtidos pelas medidas de interesse com a base Adult.

Nota-se, na última coluna da tabela 6.18, que o emprego das medidas de interesse não elimina por completo as regras consideradas redundantes pela abordagem proposta, o que justifica que o filtro DMcut é necessário como um complemento ao uso dessas medidas.

Podemos observar na mesma tabela, que uma pequena variação no valor de corte das medidas resulta numa variação expressiva no número de regras eliminadas. Por exemplo, uma variação de apenas 0,1 no valor de corte do Lift fez com que o número de regras eliminadas praticamente duplicasse. Este é um inconveniente no uso das medidas de interesse objetivas - a escolha do valor de corte é feita de acordo com a experiência do usuário (de maneira subjetiva), o que não garante a qualidade na eliminação das regras.

O próximo experimento objetiva demonstrar que as medidas de interesse objetivas eliminam regras que são de interesse do usuário, e não eliminam regras que deveriam eliminar. Para tanto, assumimos as seguintes premissas: i) as regras eliminadas pelo filtro DMcut são redundantes; ii) as regras consideradas mais relevantes pelo filtro DMcut nos processos de generalização e especialização das regras são de interesse do usuário.

Para tanto, utilizamos o formato de uma tabela, semelhante a estrutura de uma matriz de confusão. As tabelas, chamadas de tabelas de desempenho, contêm o cruzamento das classes **D**, das regras que deveriam ser eliminadas, e **ND**, das regras que não deveriam ser eliminadas, com as classes **F**, das regras que foram eliminadas pela medida de interesse objetiva, e **NF**, das regras que não foram eliminadas.

A diagonal secundária das tabelas exibe o número de casos em que a medida de interesse objetiva não atuou corretamente, eliminando regras que não deveria (**ND** x **F**), ou mantendo regras que deveriam ser eliminadas (**D** x **NF**).

As tabelas são referentes ao uso das quatro medidas de interesse selecionadas sobre a base de dados Adult. Para cada medida de interesse objetiva foram considerados quatro valores distintos de corte. As tabelas de desempenho referentes às bases de dados STULONG, Labor e Car encontram-se no **Apêndice A** desta dissertação.

Com a ajuda das tabelas de desempenho, podemos constatar que as medidas de interesse objetivas não atuaram como deveriam em todos os casos considerados. Apesar de mais eficientes em relação à quantidade de regras eliminadas, essas medidas eliminaram regras que não deveriam e deixaram regras que poderiam ser eliminadas. Este experimento justifica que a nossa abordagem semântica é mais precisa do que as medidas de interesse objetivas.

Em [11] os autores propõem a utilização dos aspectos objetivos de interessabilidade como um primeiro filtro para selecionar regras potencialmente interessantes ao usuário, e posterior-

Convicção (1,10)			**Convicção (1,20)**		
	F	**NF**		**F**	**NF**
D	365	841	**D**	406	800
ND	52	261	**ND**	70	243
Convicção (1,30)			**Convicção (1,40)**		
	F	**NF**		**F**	**NF**
D	436	770	**D**	451	755
ND	83	230	**ND**	97	216

Tabela 6.19: Tabela de desempenho da medida de interesse Convicção sobre a base de dados Adult.

Especificidade (0,95)			**Especificidade (0,97)**		
	F	**NF**		**F**	**NF**
D	16	1190	**D**	40	1166
ND	64	249	**ND**	102	211
Especificidade (0,98)			**Especificidade (0,99)**		
	F	**NF**		**F**	**NF**
D	144	1062	**D**	380	826
ND	125	188	**ND**	171	142

Tabela 6.20: Tabela de desempenho da medida de interesse Especificidade sobre a base de dados Adult.

mente a utilização dos aspectos subjetivos como um filtro final para selecionar regras realmente interessantes. O emprego das medidas de interesse objetivas se faz necessário para eliminar grande parte das anomalias inerentes ao conjunto de regras mineradas. Neste trabalho, propomos a utilização do filtro DMcut antes do emprego das medidas de interesse das regras. O processo de utilização do filtro DMcut em conjunto com as medidas de interesse é apresentado na figura 6.1.

No processo da figura 6.1, inicialmente as relações de dependência entre os atributos são utilizadas pelo filtro DMcut para eliminar a redundância semântica das regras de associação. O resultado é armazenado em um subconjunto das regras de associação (RA potencialmente interessantes), com exceção das regras escolhidas pelo filtro DMcut como representantes de outras regras menos relevantes ou de menor impacto para o usuário. Estas últimas são armazenadas em um subconjunto separado de regras (Regras de associação mais relevantes). Este processo é necessário para garantir que o uso posterior das medidas de interesse não elimine essas regras, garantindo assim a integridade do método proposto. Em seguida o conjunto das regras de associação potencialmente interessantes passa pelo filtro sintático das medidas de interesse objetivas, gerando o subconjunto das regras realmente interessantes. Por fim, este conjunto e o conjunto das regras deixadas pelo DMcut em prol de outras regras, formam o conjunto das

Lift (1,00)			**Lift (1,10)**		
	F	**NF**		**F**	**NF**
D	112	1094	**D**	273	933
ND	22	291	**ND**	81	232
Lift (1,20)			**Lift (1,30)**		
	F	**NF**		**F**	**NF**
D	477	729	**D**	782	424
ND	128	185	**ND**	205	108

Tabela 6.21: Tabela de desempenho da medida de interesse Lift sobre a base de dados Adult.

Novidade (0,00)			**Novidade (0,10)**		
	F	**NF**		**F**	**NF**
D	112	1094	**D**	1206	0
ND	22	291	**ND**	313	0
Novidade (0,20)			**Novidade (0,30)**		
	F	**NF**		**F**	**NF**
D	1206	0	**D**	1206	0
ND	313	0	**ND**	313	0

Tabela 6.22: Tabela de desempenho da medida de interesse Novidade sobre a base de dados Adult.

regras de associação a ser apresentado para o usuário final.

Desse modo, é possível reduzir significativamente o volume das regras geradas, com maior qualidade no corte, ou seja, eliminando a redundância semântica das regras, e preservando as regras de maior impacto para o usuário.

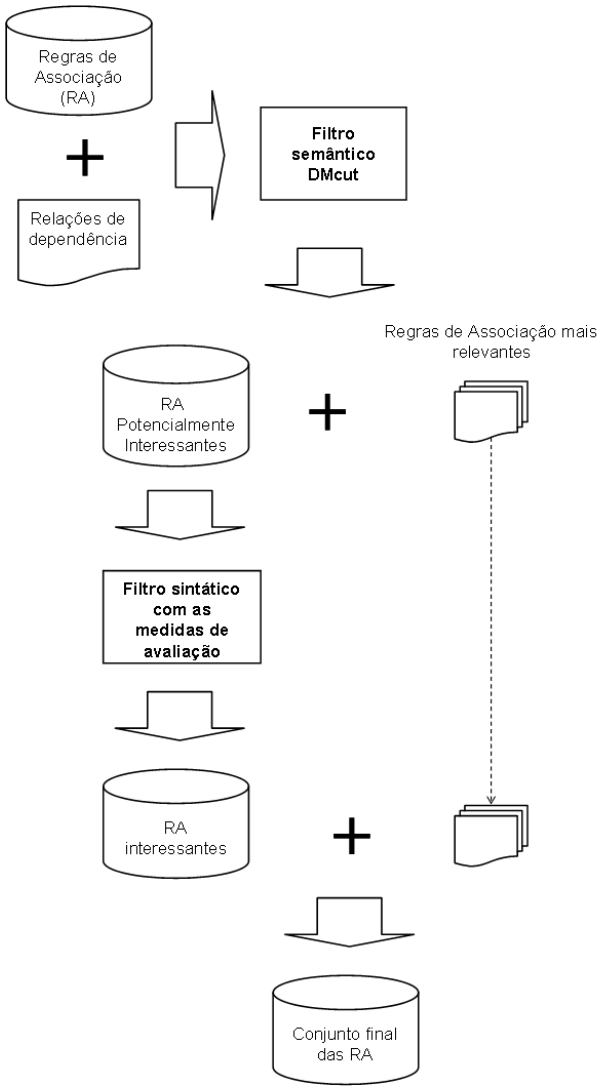


Figura 6.1: Processo para a utilização do filtro DMcut em conjunto com as medidas de interesse objetivas.

Capítulo 7

Conclusões e Trabalhos Futuros

A área da Mineração de Dados se dedica a encontrar algoritmos eficientes para transformar dados, muito volumosos para serem compreendidos facilmente, em informação organizada no formato de regras de associação, árvores de decisão, Agrupamentos (*Clustering*), etc. Contudo, não é raro os processos de mineração gerarem como resultado uma grande quantidade de dados, que continuam sendo um desafio para os humanos. Este é um problema em especial da técnica de Regras de Associação que, freqüentemente, produz uma grande quantidade de regras, muitas das quais não são de interesse do usuário.

Para ajudar a reverter esse quadro, diversos estudos têm sido direcionados ao desenvolvimento de métodos de pós-processamento dos resultados. Estes métodos consistem basicamente em técnicas de poda para eliminar regras não interessantes, técnicas de resumo e agrupamento, e ferramentas de visualização que permitem uma melhor percepção dos resultados pelo usuário.

As técnicas de poda são de vital importância, uma vez que reduzem consideravelmente o número de regras apresentadas para o usuário, facilitando a busca pelos padrões realmente interessantes. Essas técnicas podem ser divididas nas abordagens sintática e semântica. A primeira utiliza exclusivamente da estrutura das regras e dos dados utilizados no seu processo de extração. Já as abordagens semânticas levam em consideração o conhecimento prévio do domínio para avaliar a força de uma regra.

As abordagens sintáticas são mais populares, concentram os maiores esforços na literatura, e estão presentes na maioria das ferramentas comerciais. Contudo, as abordagens sintáticas muitas vezes eliminam regras que não deveriam e deixam regras que não são de interesse do usuário, o que pode parecer arbitrário para um especialista do domínio. Assim, o desenvolvimento de técnicas baseadas no conhecimento do domínio é necessário no sentido de permitir um corte mais preciso das regras.

Neste trabalho foi apresentada uma nova abordagem de pós-processamento das regras, com a utilização das relações de dependência entre os atributos como guia no processo de poda. A contribuição do presente trabalho foi a proposta e implementação do algoritmo DMcut e do ambiente de pós-processamento de regras de associação DMcut. O algoritmo DMcut tem como objetivo eliminar as regras redundantes ou de menor impacto para o usuário. O Ambiente de Pós-processamento DMcut auxilia o uso do filtro DMcut, e provê funcionalidades para a exploração das regras, além de permitir o emprego das medidas de interesse em conjunto com o DMcut.

Para mostrar que o uso das relações de dependência entre os atributos pode reduzir o volume do conjunto de regras mineradas, foram realizados alguns experimentos utilizando o algoritmo DMcut. Os experimentos apresentaram taxas de redução, do volume de um conjunto de regras de associação, na faixa de 6,72% a 17,30%, e na faixa de 7,34% a 39,81% considerando-se apenas as regras sobre as quais o método proposto tem ação. Entretanto, como as relações de dependência entre os atributos são dependentes do domínio, outras aplicações podem gerar taxas de redução maiores ou menores do que as apresentadas nos experimentos.

Os experimentos demonstraram que o algoritmo DMcut corta apenas as regras redundantes ou de menor impacto, uma vez que o processo não é cego. Ou seja, por ser baseada no conhecimento prévio do domínio, a abordagem proposta realiza um corte mais preciso, proporcionando uma maior qualidade no pós-processamento das regras.

A existência de tantos métodos voltados para o problema da grande quantidade de regras de associação produzidas leva a crer que nenhum deles é, por si só, suficiente. Estes trabalhos são complementares a nossa abordagem. Neste estudo, propomos uma metodologia para o uso do filtro DMcut em conjunto com o emprego das medidas de interesse objetivas.

Como trabalhos futuros, primeiro, pretende-se investigar novas abordagens de escolha entre os processos de generalização e especialização das regras, por exemplo, trocando a ordem de aplicação dos indicadores CRg e CD^{+}, ou ainda, incorporando novas informações sobre as relações de dependência entre os atributos.

Pretende-se ainda estender o algoritmo DMcut para ler regras de associação representadas no formato universal PMML (Predictive Model Markup Language), um modelo de regras baseado na linguagem XML [40]. Prevê-se que, no futuro, as maiores aplicações na área de Mineração de Dados exportem e importem os seus modelos de regras (como de outros métodos) para o formato PMML [30].

Intenciona-se ainda utilizar os conceitos aqui apresentados na etapa da extração das regras,

implementando-se um algoritmo que elimine a possibilidade da extração de regras de associação semanticamente redundantes para o usuário.

Por fim, pretende-se investigar novas formas de redundância semântica que possam ser identificadas a partir das relações de dependência entre os atributos.

APÊNDICE A – Tabelas de desempenho das medidas de interesse objetivas

A seguir são apresentadas as tabelas de desempenho referentes ao emprego das medidas de interesse objetivas Convicção, Especificidade, Lift e Novidade em cada uma das combinações de bases de dados e valores de corte utilizados nos experimentos deste trabalho. Os dados apresentados nessas tabelas têm como objetivo dar subsídios às avaliações de desempenho das medidas de interesse objetivas realizadas no capítulo 6.

Convicção (1,10)			**Convicção (1,20)**		
	F	**NF**		**F**	**NF**
D	243	128	**D**	280	91
ND	40	42	**ND**	64	18
Convicção (1,30)			**Convicção (1,40)**		
	F	**NF**		**F**	**NF**
D	298	73	**D**	304	67
ND	73	9	**ND**	77	5

Tabela A.1: Tabela de desempenho da medida de interesse Convicção sobre a base de dados STULONG.

Especificidade (0,95)			**Especificidade (0,97)**		
	F	**NF**		**F**	**NF**
D	151	220	**D**	189	182
ND	80	2	**ND**	82	0
Especificidade (0,99)			**Especificidade (1,00)**		
	F	**NF**		**F**	**NF**
D	214	157	**D**	216	155
ND	82	0	**ND**	82	0

Tabela A.2: Tabela de desempenho da medida de interesse Especificidade sobre a base de dados STULONG.

Lift (1,00)			**Lift (1,10)**		
	F	**NF**		**F**	**NF**
D	54	317	**D**	168	203
ND	17	65	**ND**	73	9
Lift (1,20)			**Lift (1,30)**		
	F	**NF**		**F**	**NF**
D	176	195	**D**	176	195
ND	78	4	**ND**	78	4

Tabela A.3: Tabela de desempenho da medida de interesse Lift sobre a base de dados STULONG.

Novidade (0,00)			**Novidade (0,10)**		
	F	**NF**		**F**	**NF**
D	54	317	**D**	371	0
ND	17	65	**ND**	82	0
Novidade (0,20)			**Novidade (0,30)**		
	F	**NF**		**F**	**NF**
D	371	0	**D**	371	0
ND	82	0	**ND**	82	0

Tabela A.4: Tabela de desempenho da medida de interesse Novidade sobre a base de dados STULONG.

Convicção (1,10)			**Convicção (1,20)**		
	F	**NF**		**F**	**NF**
D	297	83	**D**	297	83
ND	209	66	**ND**	209	66
Convicção (1,30)			**Convicção (1,40)**		
	F	**NF**		**F**	**NF**
D	304	76	**D**	304	76
ND	210	65	**ND**	210	65

Tabela A.5: Tabela de desempenho da medida de interesse Convicção sobre a base de dados Labor.

Especificidade (0,95)			**Especificidade (0,97)**		
	F	**NF**		**F**	**NF**
D	16	364	**D**	64	316
ND	5	270	**ND**	38	237
Especificidade (0,99)			**Especificidade (1,00)**		
	F	**NF**		**F**	**NF**
D	83	297	**D**	83	297
ND	66	209	**ND**	66	209

Tabela A.6: Tabela de desempenho da medida de interesse Especificidade sobre a base de dados Labor.

Lift (1,00)			**Lift (1,10)**		
	F	**NF**		**F**	**NF**
D	0	380	**D**	0	380
ND	0	275	**ND**	0	275
Lift (1,20)			**Lift (1,30)**		
	F	**NF**		**F**	**NF**
D	30	350	**D**	56	324
ND	15	260	**ND**	37	238

Tabela A.7: Tabela de desempenho da medida de interesse Lift sobre a base de dados Labor.

Novidade (0,00)			**Novidade (0,10)**		
	F	**NF**		**F**	**NF**
D	0	380	**D**	378	2
ND	0	275	**ND**	275	0
Novidade (0,20)			**Novidade (0,30)**		
	F	**NF**		**F**	**NF**
D	380	0	**D**	380	0
ND	275	0	**ND**	275	0

Tabela A.8: Tabela de desempenho da medida de interesse Novidade sobre a base de dados Labor.

Convicção (1,10)			**Convicção (1,20)**		
	F	**NF**		**F**	**NF**
D	18	0	**D**	18	0
ND	3	0	**ND**	3	0
Convicção (1,30)			**Convicção (1,40)**		
	F	**NF**		**F**	**NF**
D	18	0	**D**	18	0
ND	3	0	**ND**	3	0

Tabela A.9: Tabela de desempenho da medida de interesse Convicção sobre a base de dados Car.

Especificidade (0,95)			**Especificidade (0,97)**		
	F	**NF**		**F**	**NF**
D	18	0	**D**	18	0
ND	3	0	**ND**	3	0
Especificidade (0,99)			**Especificidade (1,00)**		
	F	**NF**		**F**	**NF**
D	18	0	**D**	18	0
ND	3	0	**ND**	3	0

Tabela A.10: Tabela de desempenho da medida de interesse Especificidade sobre a base de dados Car.

Lift (1,00)			**Lift (1,10)**		
	F	**NF**		**F**	**NF**
D	0	18	**D**	18	0
ND	0	3	**ND**	3	0
Lift (1,20)			**Lift (1,30)**		
	F	**NF**		**F**	**NF**
D	18	0	**D**	18	0
ND	3	0	**ND**	3	0

Tabela A.11: Tabela de desempenho da medida de interesse Lift sobre a base de dados Car.

Novidade (0,00)			**Novidade (0,10)**		
	F	**NF**		**F**	**NF**
D	0	18	**D**	18	0
ND	0	3	**ND**	3	0
Novidade (0,20)			**Novidade (0,30)**		
	F	**NF**		**F**	**NF**
D	18	0	**D**	18	0
ND	3	0	**ND**	3	0

Tabela A.12: Tabela de desempenho da medida de interesse Novidade sobre a base de dados Car.

Referências

[1] FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. *Ai Magazine*, Univ Calif Irvine, Dept Comp & Informat Sci, Irvine, Ca, 92717 Gte Labs Inc, Knowledge Discovery Databases Kdd Project, Tech Staff, Waltham, Ma, 02254, v. 17, p. 37–54, 1996. Disponível em: <citeseer.ist.psu.edu/fayyad96from.html>.

[2] SRIKANT, R.; AGRAWAL, R. Mining generalized association rules. *Proc. of the 21st Int'l Conference on Very Large Databases*, p. 407–419, 1995.

[3] FRAWLEY, W. J.; PIATETSKY-SHAPIRO, G.; MATHEUS, C. J. Knowledge discovery in databases - an overview. *Ai Magazine*, Gte Labs Inc, Distributed Cooperating Learning Syst Project, Waltham, Ma, 02254 Gte Labs Inc, Knowledge Discovery Databases Project, Waltham, Ma, 02254, v. 13, p. 57–70, 1992. Disponível em: <citeseer.ist.psu.edu/frawley92knowledge.html>.

[4] FAYYAD; PIATETSKY-SHAPIRO; SMYTH. From data mining to knowledge discovery: An overview. 1996.

[5] AGRAWAL, R.; IMIELINSKI, T.; SWAMI, A. Mining association rules between sets of items in large databases. In: BUNEMAN, P.; JAJODIA, S. (Ed.). *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*. Washington, D.C.: [s.n.], 1993. p. 207–216.

[6] TOIVONEN, H. et al. *Pruning and grouping of discovered association rules*. 1995. Disponível em: <citeseer.ist.psu.edu/toivonen95pruning.html>.

[7] KNOBBE, A. J. *Multi-Relational Data Mining*. Tese (Doutorado) — Faculteit Wiskunde en Informatica, Universiteit Utrecht, 2004.

[8] ANAND, S. S.; BELL, D. A.; HUGHES, J. G. The role of domain knowledge in data mining. In: *CIKM*. [s.n.], 1995. p. 37–43. Disponível em: <citeseer.ist.psu.edu/anand95role.html>.

[9] GARDNER, S. R. Building the data warehouse. *Commun. ACM*, ACM Press, New York, NY, USA, v. 41, n. 9, p. 52–60, 1998. ISSN 0001-0782.

[10] CABENA PABLO HADJINIAN, R. S. J. V. A. Z. P. *Discovering Data Mining: From Concept to Implementation*. [S.l.]: Prentice Hall, 1997.

[11] RESENDE, S. O. *Sistemas Inteligentes: fundamentos e aplicações*. Editora manole ltda. [S.l.: s.n.], 2003.

[12] BRADLEY, U. M. F. P. S.; MANGASARIAN, O. L. *Data Mining: Overview and Optimization Opportunities*. [S.l.], 1998.

[13] AGRAWAL, R.; SRIKANT, R. Fast algorithms for mining association rules. In: BOCCA, J. B.; JARKE, M.; ZANIOLO, C. (Ed.). *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*. Morgan Kaufmann, 1994. p. 487–499. ISBN 1-55860-153-8. Disponível em: <citeseer.ist.psu.edu/article/agrawal94fast.html>.

[14] HAN, E.; PEI, J.; YIN, Y. Mining frequent patterns without candidate generation. In: *ACM SIGMOD International Conference on Management of Data*. [S.l.: s.n.], 2000.

[15] ORLANDO, S. Adaptative and resource-aware mining of frequent sets. In: *IEEE ICDM International Conference on Data Mining*. [S.l.: s.n.], 2002.

[16] GOETHALS, B. *Survey on Frequent Pattern Mining*. 2003. Disponível em: <http://www.adrem.ua.ac.be/ goethals/software/survey.pdf>.

[17] BRIN, S. et al. Dynamic itemset counting and implication rules for market basket data. In: PECKHAM, J. (Ed.). *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA*. ACM Press, 1997. p. 255–264. Disponível em: <citeseer.ist.psu.edu/brin97dynamic.html>.

[18] PADMANABHAN, B.; TUZHILIN, A. Unexpectedness as a measure of interestingness in knwoledge discovery. *Decision Support Systems*, v. 70, 1997.

[19] ZAKI, M. J. Generating non-redundant association rules. In: *Conference on Knowledge Discovery in Data, Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2000.

[20] BASTIDE, Y. et al. *Mining minimal non-redundant association rules using frequent closed itemsets*. 2000. Disponível em: <citeseer.ifi.unizh.ch/article/bastide00mining.html>.

[21] ZAKI, M. J.; OGIHARA, M. Theoretical foundations of association rules. In: *In Proceedings of 3 rd SIGMOD'98 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'98)*. Seattle, Washington: [s.n.], 1998. Disponível em: <citeseer.ist.psu.edu/zaki98theoretical.html>.

[22] BAESENS S. VIAENE, J. V. B. Post-processing of association rules. *KDD-2000: The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, USA*, p. 20–23, August 2000.

[23] SILBERSCHATZ, A.; TUZHILIN, A. On subjective measures of interestingness in knowledge discovery. In: *1st ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*. [S.l.: s.n.], 1995. p. 275281.

[24] LAVRAC, N.; FLACH, P.; ZUPAN, B. Rule evaluation meaures: A unifying view. 1999.

[25] PIATETSKY-SHAPIRO, G. Analysis and presentation of strong rules. *Knowledge Discovery in Databases, AAAI/MIT Press*, 1991.

[26] ADAMO, J.-M. *Data Mining for Association Rules and Sequential Patterns*. [S.l.]: NY: Springer-Verlag, 2001.

[27] DOMINGUES, M. A. *Generalização de regras de associação*. Dissertação (Mestrado) — USP - São Carlos, 2004. Instituto de Ciências Matemáticas e de Computação.

[28] LIU, B.; HSU, W.; MA, Y. Pruning and summarizing the discovered associations. *Conference on Knowledge Discovery in Data, Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999.

[29] LIU, B. et al. *Visually aided exploration of interesting association rules*. 1999. Disponível em: <citeseer.ifi.unizh.ch/liu99visually.html>.

[30] NEVES, J. M. P. M. D. *AMBIENTE DE PÓS-PROCESSAMENTO PARA REGRAS DE ASSOCIAÇÃO*. Dissertação (Mestrado) — UNIVERSIDADE DO PORTO - FACULDADE DE ECONOMIA, 2002.

[31] WONG, P. C.; WHITNEY, P.; THOMAS, J. Visualizing association rules for text mining. In: *INFOVIS*. [s.n.], 1999. p. 120–123. Disponível em: <citeseer.ist.psu.edu/wong99visualizing.html>.

[32] DOMINGUES, M. A.; REZENDE, S. O. *Descrição de um algoritmo para generalização de regras de associação*. [S.l.], 2004. Relatório Técnico do ICMC/USP - Número 228.

[33] WITTEN, I. H.; FRANK, E. *Data Mining: Practical machine learning tools and techniques*. [S.l.]: Morgan Kaufmann, 2005.

[34] NEWMAN S. HETTICH, C. B. D.; MERZ, C. *UCI Repository of machine learning databases*. 1998. Disponível em: <http://www.ics.uci.edu/∼mlearn/MLRepository.html>.

[35] STULONG Project. 2006. Disponível em: <http://euromise.vse.cz/STULONG>.

[36] SBACV-RJ. 2004. `http://www.sbacvrj.com.br/index.htm`. Acesso em 28/03/2006.

[37] HOLLINGSHEAD, A. B.; REDLICH, F. C. *Social class and mental illness: a community study*. [S.l.]: New York, John Wiley & Sons, Inc., 1958.

[38] HAWKINS, D. I. et al. *Consumer Behavior: Building Marketing Strategy*. 9. ed. [S.l.]: McGraw-Hill/Irwin, 2004.

[39] VOLUNTARY Guidelines for Providers of Weight Loss Products or Services Homepage. 1999. `http://www.consumer.gov/weightloss/index.htm`. Acesso em 05/01/2006.

[40] DATA Mining Group (PMML development). Disponível em: <http://www.dmg.org/>.