SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP
Data de Depósito: 20.02.2003
Assinatura: Ana Paula Campos Julio

Ambiente para exploração de regras

Marcos Ferreira de Paula

Orientadora: Profª Drª Solange Oliveira Rezende

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC/USP, como parte dos requisitos para obtenção do título de Mestre em Ciências de Computação e Matemática Computacional.

USP - São Carlos
Fevereiro de 2003

Trabalho realizado com apoio financeiro da CAPES

A Comissão Julgadora:

Profa. Dra. Solange Oliveira Rezende _____

Profa. Dra. Maria Carolina Monard _____

Prof. Dr. Geber Lisboa Ramalho _____

Este documento foi preparado com o formatador de textos LaTeX. O sistema de citações de referências bibliográficas utilizado foi o BibTeX padrão *Chicago*.

Dedicatória

À minha família Geraldo, Ana Rosa e Ligia

Agradecimentos

A meus pais, Geraldo e Ana Rosa, e minha irmã Ligia pelo amor, carinho e respeito tão importantes para que pudesse realizar tudo que sempre desejei.

À professora Solange que foi mais que uma orientadora desde os trabalhos de Iniciação Científica, mas uma grande amiga que ofereceu incentivo, apoio, dedicação e compreensão, tendo contribuído não somente nesse trabalho de mestrado mas também para meu crescimento pessoal.

À Talita por ser tão especial e por todo carinho, comprometimento, dedicação e ajuda em todos os momentos desde a época da graduação.

A todos os meus "irmãos", Luis Carlos, Alex, Tereza, Luiz Fernando, Léa, Walter, Robson, Cristiane, Jaqueline, Edson, Valmir, Daniel, Marcos Aurélio, Roberta, Camila, entre outros alunos e ex-alunos da "mamãe" Solange, que contribuíram na minha formação desde a Iniciação Científica, na elaboração deste trabalho, na redação final desta dissertação e por todos os bons momentos que passamos juntos.

À professora Maria Carolina Monard e aos seus alunos Ronaldo e Gustavo por todas as discussões, dicas e contribuições para a realização deste trabalho.

A todos do LABIC pelos bons momentos e por fazer do laboratório um ambiente agradável, propício para a realização de qualquer trabalho.

A todos os funcionários do ICMC pelo respeito, dedicação e educação.

À CAPES pelo apoio financeiro.

Por fim, a Deus por ter me dado condições para cumprir mais uma etapa da minha vida.

Resumo

Recentemente, o desenvolvimento das tecnologias de coleta e armazenamento de dados, a popularização da Internet, e a evolução das necessidades de informação fazem com que qualquer esforço para facilitar a manipulação e o entendimento humano em relação a esse grande leque de informações ganhe grande importância. Nesse sentido, as organizações têm visto com crescente interesse o desenvolvimento de tecnologias automatizadas para extração de conhecimento de grandes volumes de dados. Estas tecnologias têm sido referenciada na literatura como Extração de Conhecimento de Bases de Dados ou Mineração de Dados (MD).

Um dos objetivos do processo de Extração de Conhecimento de Bases de Dados é que seus usuários finais possam analisar, compreender e utilizar o conhecimento extraído em um processo de tomada de decisão. Entretanto, do ponto de vista desses usuários, um problema encontrado no final do processo de Extração de Conhecimento é que muitos dos algoritmos utilizados geram uma enorme quantidade de padrões, dificultando consideravelmente sua análise. Outro problema freqüentemente identificado é a dificuldade na compreensão dos modelos extraídos dos dados. Nesse sentido, diversos trabalhos têm sido desenvolvidos para apoiar os usuários na avaliação e interpretação do conhecimento extraído a partir dos dados.

A existência de um Ambiente para Exploração de Regras em que as regras encontradas em um processo de Extração de Conhecimento estejam armazenadas, disponibilizadas aos usuários e que apresente diversos tipos de métodos para sua avaliação pode representar um auxílio efetivo para os usuários na compreensão, identificação das regras interessante e utilização das mesmas. Nesta dissertação, são apresentadas algumas características importantes de um Ambiente para Exploração de Regras, uma arquitetura geral e a implementação de um ambiente deste tipo, o Ambiente *RulEE*. É também apresentado um estudo de caso realizado com objetivo de mostrar essas características na análise e disponibilização de regras extraídas a partir de um conjunto de dados financeiros.

Abstract

Recently, the development of the data collection and storage technologies, the popularization of the Internet, and the evolution of the necessities of information give great importance to any effort to facilitate the manipulation and the human comprehension of this great amount of information. In this way, the organizations have seen, with increasing interest, the development of automatized technologies for knowledge extraction from great volumes of data. These technologies have been referenced in literature as Knowledge Discovery in Databases or Data Mining (DM).

One of the objectives of the Knowledge Discovery in Database process is to make its final users able to analyze, understand and use the knowledge in a decision making process. However, from the point of view of these users, a problem found in the end of the Knowledge Discovery process is that many of the used algorithms generate an enormous amount of patterns, making difficult its analysis. Another problem frequently identified is the difficulty to understand the models extracted from the data. In this way, several works have been developed to help the users in the evaluation and interpretation of the knowledge extracted from the data.

The existence of a Rule Exploration Environment where the rules found in a Knowledge Discovery in Databases process are stored, made available to the users and that presents several types of methods for its evaluation can represent an effective aid for the users in the understanding, identification and use of the interesting rules. In this work, it is presented some important characteristics of a Rule Exploration Environment, a general architecture, and an implementation of an environment of this type, the *RulEE* environment. A case study is also presented showing these characteristics in the exploration of rules extracted from a financial data set.

Sumário

Lista de Figuras v

Lista de Tabelas vii

1 Introdução 1

2 Visão Geral do Processo de Extração de Conhecimento de Bases de Dados 5

2.1 Considerações Iniciais 5

2.2 O Processo de Extração de Conhecimento de Bases de Dados 6

2.2.1 Conhecimento do Domínio e Identificação do Problema 7

2.2.2 Pré-Processamento 8

2.2.3 Extração de Padrões 12

2.2.4 Pós-Processamento 15

2.3 Ferramentas de Mineração de Dados 16

2.3.1 Protótipos de Pesquisa 16

2.3.2 Produtos Comerciais 17

2.4 Considerações Finais 21

3 Avaliação de Regras 23

3.1 Considerações Iniciais 23

3.2 Interessabilidade de Regras 24

3.3 Aspectos Objetivos para Avaliação de Regras 26

3.3.1 Medidas de Surpresa 26

3.3.2 Seleção de Regras não Redundantes 29

3.3.3 Medidas Derivadas da Tabela de Contingência 30

3.3.4 Medida *RI* (*Rule-Interest*) 34

3.4 Aspectos Subjetivos para Avaliação de Regras 34
3.4.1 Sistema de Crenças 35
3.4.2 Modelos de Regras 36
3.4.3 Aplicativo AIAS 37
3.5 Sumário das Medidas Apresentadas 40
3.6 Considerações Finais 41

4 Ambiente para Exploração de Regras **43**
4.1 Considerações Iniciais 43
4.2 O Ambiente Discover 44
4.2.1 Sintaxe Padrão para Representação de Dados 46
4.2.2 Sintaxe Padrão para Representação de Regras de Classificação 47
4.2.3 Sintaxe Padrão para Representação de Regras de Regressão 48
4.2.4 Sintaxe Padrão para Representação de Regras de Associação 48
4.3 Requisitos de um Ambiente para Exploração de Regras 49
4.4 O Ambiente *RulEE* 54
4.4.1 Descrição Geral 54
4.4.2 Funcionalidades do Ambiente *RulEE* 57
4.5 Considerações Finais 57

5 Implementação do Ambiente *RulEE* **61**
5.1 Considerações Iniciais 61
5.2 Implementação 61
5.2.1 Informações Gerais 66
5.2.2 Implementação da Base de Dados 68
5.2.3 Classes 73
5.2.4 Obtenção e Cálculo de Valores de Medidas 74
5.2.5 Descrição da Interface 78
5.2.6 Utilização de Consultas 89
5.3 Medidas Disponíveis 92
5.4 Considerações Finais 92

6 Estudo de Caso **95**

6.1 Considerações Iniciais 95

6.2 A Extração de Regras em Dados Financeiros 95

6.2.1 Conhecimento do Domínio e Identificação do Problema 96

6.2.2 Pré-Processamento 98

6.2.3 Extração de Padrões 100

6.2.4 Pós-Processamento 101

6.3 Exploração das Regras Utilizando o *RulEE* 101

6.3.1 Utilização do Ambiente por *Scripts* 101

6.3.2 Utilização do Ambiente pela Interface 104

6.4 Considerações Finais 109

7 Conclusões e Trabalhos Futuros **111**

Referências Bibliográficas **117**

Lista de Figuras

2.1 Etapas do Processo de Extração de Conhecimento 7

4.1 Estrutura do Ambiente Discover 44
4.2 Exemplo de Arquivo *names* na Sintaxe Padrão 46
4.3 Exemplo de Arquivo *data* na Sintaxe Padrão 47
4.4 Exemplo de Arquivo de Regras de Classificação na Sintaxe Padrão 47
4.5 Exemplo de Arquivo de Regras de Regressão na Sintaxe Padrão 49
4.6 Exemplo de Arquivo de Regras de Associação na Sintaxe Padrão 49
4.7 Metodologia para Obtenção de Regras nas Sintaxes Padrão 55

5.1 Arquitetura Geral do Ambiente *RulEE* 62
5.2 Diagrama Físico da Base de Dados do Ambiente *RulEE* 72
5.3 Relacionamentos entre as Classes do Módulo de Gerenciamento 74
5.4 Herança, Atributos e Métodos das Classes do Módulo de Gerenciamento . . . 75
5.5 *Login* no Ambiente 80
5.6 *Menu* Principal do Ambiente 80
5.7 Inserção de Conjuntos de Regras 81
5.8 Listagem de Conjuntos de Regras 81
5.9 Visualização de um Conjunto de Regras 83
5.10 *Download* ou Visualização do Arquivo de Regras Original 83
5.11 Inserção de um Novo Comentário 84
5.12 Visualização de Informações sobre uma Regra 84
5.13 Listagem de Consultas 85
5.14 Visualização de Consulta 85
5.15 Arquivos de Descrição e de Dados Referentes à Consulta 86
5.16 Edição de Consultas - Modo Simples 87

5.17 Edição de Consultas - Modo Avançado 88
5.18 Listagem de Medidas 90
5.19 Informações sobre Medidas 90

6.1 Diagrama da Base de Dados Financeira 96
6.2 Exemplo de *Script* para Criação de Sessões no Ambiente *RulEE* 102
6.3 Execução do *Script* de Exemplo para Criação de Sessões 102
6.4 Exemplo de *Script* para Inserção do Conjunto de Regras Extraído de *Financial_Data_1* 102
6.5 Resultado da Execução do *Script* de Exemplo para Inserção do Conjunto de Regras 103
6.6 Exemplo de *Script* para Execução de Consulta 103
6.7 Resultado do *Script* de Exemplo para Execução de Consulta 104
6.8 Tela para Inserção do Conjunto de Regras Extraído de *Financial_Data_2* . . . 104
6.9 Regras que Classificam os Clientes como *BAD* ou *GOOD* 106
6.10 Quantidade de Regras que Classificam como *BAD* e *GOOD* 106
6.11 Precisão das Regras 107
6.12 Precisão Relativa 107
6.13 Soma da Cobertura das 4 Regras com Maior Cobertura 108
6.14 Valor da Medida Novidade para as Regras 109
6.15 Valor da Medida Especificidade para as Regras 109

Lista de Tabelas

2.1 Conjunto de Dados no Formato Atributo-Valor 9

3.1 Tabela de Contingência 31

3.2 Tabela de Contingência com Freqüências Relativas 31

3.3 Classificação das Medidas de Avaliação de Conhecimento 40

4.1 Síntese de Funcionalidades de Ambientes para Análise de Regras 58

5.1 Subdiretórios do Ambiente *RulEE* 67

5.2 Descrição das Tabelas da Base de Dados do Ambiente *RulEE* 71

5.3 Descrição das Classes do Módulo de Gerenciamento 73

6.1 Descrição das Tabelas da Base de Dados Financeira 97

6.2 Transformação de Valores de Atributos da Base de Dados Financeira 99

6.3 Novos Atributos da Base de Dados Financeira 99

Capítulo 1

Introdução

O desenvolvimento da tecnologia para armazenamento de dados possibilitou às organizações coletarem e armazenarem informações de diversas fontes a uma taxa alguns anos atrás considerada inimaginável. Em (Zuffo 2002) é feita uma estimativa de quanta informação é produzida anualmente. Nesse trabalho, os pesquisadores estimaram que o volume de dados produzido anualmente está em torno de 1,5 *exabyte* (1,5 bilhão de *gigabytes*), incluindo mídia impressa, filme e meios óptico e magnético. Desse total, 99,997% está disponível em meio digital. Em (Lyman & Varian 2000) é destacado que a capacidade de armazenamento em meio magnético tem dobrado a cada ano, permitindo que um volume de dados cada vez maior esteja disponível.

Em uma economia cada vez mais competitiva, as organizações têm tido uma maior preocupação em adquirir novas tecnologias para apoiar o entendimento das informações e do conhecimento embutido em suas bases de dados. Os dados são vistos pelas organizações como uma fonte potencial de informação valiosa que pode prover vantagem competitiva. Esses dados podem conter, por exemplo, um conhecimento extremamente importante com relação ao comportamento de seus clientes, fornecedores e mesmo dos concorrentes que, possivelmente, não poderia ser descoberto sem uma análise mais profunda dos mesmos.

O método tradicional de transformar dados em conhecimento é baseado em análise e interpretação manual. Para muitas aplicações, essa busca manual é cara, lenta e altamente subjetiva. Além disso, a busca manual tem se tornado impraticável nos domínios em que a quantidade de dados tem crescido exponencialmente (Monard, Caulkins, Baranauskas, Oliveira, & Rezende 1999). Sem a existência de ferramentas para explorar o volume crescente de dados, o armazenamento desses dados, apesar de sua potencial utilidade, pode se tornar inútil (Keim 2001).

Técnicas de análise de dados normalmente não extrapolam a realização de consultas SQL (*Structured Query Language*) simples, utilização de ferramentas OLAP (*Online Analytical Processing*) ou mecanismos de visualização de dados. Com essa forma de análise de dados, algumas questões importantes para tomada de decisão não podem ser expressas, como:

- Quais são os usuários potenciais para praticar fraude?
- Quais clientes gostariam de comprar o novo produto X?

Diante da deficiência para analisar e compreender grandes volumes de dados, diversos estudos têm sido direcionados ao desenvolvimento de tecnologias de extração automática de conhecimento de bases de dados. Este campo de pesquisa é chamado de Extração de Conhecimento de Bases de Dados, normalmente referenciado na literatura como Mineração de Dados (MD).

O processo de Extração de Conhecimento tem o objetivo de encontrar conhecimento a partir de um conjunto de dados para ser utilizado em um processo de tomada de decisão. Portanto, um requisito importante é que esse conhecimento descoberto seja compreensível, além de útil e interessante para os usuários finais do processo, fornecendo a esses usuários apoio no processo de tomada de decisão.

Inicialmente, muitos estudos sobre o processo de Extração de Conhecimento estiveram focados na etapa de Extração de Padrões. Entretanto, o desenvolvimento da tecnologia e sua utilização para resolver problemas reais tornou claro que a Extração de Padrões não é o ponto final do processo. Desta maneira, diversos trabalhos têm sido desenvolvidos no sentido de analisar ou processar o conhecimento extraído com objetivo de auxiliar o usuário na sua interpretação e utilização.

Do ponto de vista do usuário, um dos problemas encontrados no final do processo de Extração de Conhecimento é que muitos dos algoritmos de extração geram uma enorme quantidade de padrões (Padmanabhan & Tuzhilin 2000). Por exemplo, algoritmos de extração de regras de associação geralmente produzem milhares ou dezenas de milhares de regras, especialmente quando os atributos da base de dados são altamente correlacionados. Assim, maior atenção têm sido direcionada a apoiar os usuários na identificação do conhecimento interessante (Liu, Hsu, & Ma 2001; Liu, Hsu, & Ma 1999; Lavrac, Flach, & Zupan 1999; Klemettinen, Mannila, & Toivonen 1997; Silberschatz & Tuzhilin 1996; Piatetsky-Shapiro & Matheus 1994).

Um outro problema identificado é a dificuldade na compreensão dos modelos extraídos dos dados. Muitas vezes, esses modelos podem ser muito complexos ou não fazerem sentido para os usuários especialistas no domínio da aplicação em que processo de Extração de Conhecimento está sendo aplicado (Pazzani, Mani, & Shankle 1997).

Dada a necessidade de se pesquisar técnicas para auxiliar o usuário na compreensão e utilização do conhecimento descoberto em um processo de Extração de Conhecimento, o objetivo principal deste projeto de mestrado é a proposta e implementação de um Ambiente para Exploração de Regras (*RulEE – Rule Exploration Environment*) que apresente características de apoio ao usuário na compreensão e identificação do conhecimento interessante.

Uma característica importante para o Ambiente *RulEE* é facilitar o acesso ao conhecimento descoberto. Isto é feito por uma interface de disponibilização do conhecimento baseada na WWW (*World Wide Web*). Dessa maneira, algumas características importantes da WWW são incorporadas ao ambiente, como a independência de plataforma, facilidade de utilização pelos usuários independente de sua localização física e da ferramenta de extração, além da utilização de toda uma infra-estrutura de rede existente.

Uma vez que o conhecimento descoberto tenha sido armazenado no ambiente, pode-se disponibilizar ao usuário não somente as regras, mas também métodos para que ele possa avaliá-las. O Ambiente *RulEE* apresenta um conjunto de medidas de avaliação que pode ser utilizado pelo usuário para ordenação e seleção de regras para auxiliá-lo na identificação do conhecimento mais interessante. O Ambiente *RulEE* apresenta também uma modelagem que visa facilitar a incorporação de novas medidas de avaliação pelos pesquisadores, sem necessidade de alteração do código fonte do ambiente. Desta maneira, o ambiente auxilia os pesquisadores na realização de experimentos com relação a novas medidas de avaliação.

A arquitetura proposta e implementada do Ambiente para Exploração de Regras é composta por: um Repositório de Regras e Medidas, em que são armazenadas as regras e os respectivos valores das medidas disponibilizadas; um Repositório de Dados, em que são armazenados os dados utilizados na extração e teste dos conjuntos de regras; um Repositório de Informação de Usuários, contendo os dados referentes aos usuários do ambiente; um Módulo de Gerenciamento, no qual estão implementadas todas as funções disponíveis no ambiente; um conjunto de Módulos de Pós-Processamento, que possui os métodos independentes do ambiente para o cálculo dos valores de algumas medidas; e a Interface, baseada na WWW, para disponibilização do conhecimento.

Para apresentar a revisão bibliográfica e o Ambiente *RulEE*, esta dissertação está organizada como segue:

No Capítulo 2 é descrito de forma geral o processo de Extração de Conhecimento de Bases de Dados, apresentando as etapas que o compõem e algumas ferramentas disponíveis no mercado para descoberta de conhecimento.

No Capítulo 3 são apresentados alguns aspectos sobre avaliação de regras no processo de Extração de Conhecimento, detalhando algumas medidas objetivas e subjetivas de avaliação.

No Capítulo 4 são apresentados de maneira geral os requisitos de um Ambiente para Exploração de Regras e o Ambiente Discover, que é um projeto de maior porte ao qual o Ambiente *RulEE* está integrado.

No Capítulo 5 é detalhada a implementação do Ambiente *RulEE*, apresentando sua arquitetura geral, a base de dados que contém o conhecimento, a biblioteca de classes e a interface.

No Capítulo 6 é apresentado um estudo de caso desenvolvido no domínio de dados financeiros cujo objetivo foi mostrar a viabilidade do Ambiente *RulEE* na análise das regras descobertas em um processo de Mineração de Dados.

No Capítulo 7 são apresentadas as conclusões deste trabalho e, finalizando, as Referências Bibliográficas.

Capítulo 2

Visão Geral do Processo de Extração de Conhecimento de Bases de Dados

2.1 Considerações Iniciais

Os métodos tradicionais de análise de dados têm se tornado impraticáveis com o aumento do volume de dados disponíveis. O processo de Extração de Conhecimento de Bases de Dados se apresenta como uma tecnologia que automatiza a análise de grandes volumes de dados buscando conhecimento e tem sido visto com elevado interesse pelas organizações.

Apesar de automatizar a análise de dados, o processo de Extração de Conhecimento de Bases de Dados é bastante complexo, envolvendo a execução de diversas etapas, iteração entre essas etapas e interação com os usuários. Além disso, o processo de Extração de Conhecimento é uma área multidisciplinar, envolvendo campos de pesquisa como Base de Dados, Aprendizado de Máquina (AM), Reconhecimento de Padrões, Estatística, Visualização de Dados. Dessa maneira, não se pode esperar a descoberta de conhecimento útil apenas submetendo um conjunto de dados a uma caixa preta.

Neste capítulo é apresentado de maneira geral o processo de Extração de Conhecimento de Bases de Dados, descrevendo suas etapas e algumas ferramentas disponíveis no mercado para apoio a este processo.

2.2 O Processo de Extração de Conhecimento de Bases de Dados

O processo de Extração de Conhecimento de Bases de Dados foi definido em (Fayyad, Piatetsky-Shapiro, & Smyth 1996) como: "Processo de identificar padrões nos dados que sejam válidos, novos, potencialmente úteis e compreensíveis a seres humanos". A Extração de Conhecimento é definida como sendo um processo pois envolve a execução de diversas etapas. O conhecimento extraído dos dados deve ser válido para um novo conjunto de dados com um certo grau de certeza, além de ser compreensível, pelo menos após algum Pós-processamento, para que possa ser utilizado por seres humanos, e trazer algo de novo para eles, sendo portanto, potencialmente útil.

Existem diversas abordagens para a divisão das etapas do processo de Extração de Conhecimento. Inicialmente, foi proposta em (Fayyad, Piatetsky-Shapiro, & Smyth 1996) uma divisão do processo em nove etapas. Já em (Weiss & Indurkhya 1998), essa divisão é composta por apenas quatro etapas. No entanto, neste trabalho, assim como definido em (Baranauskas 2001), considera-se a divisão do processo em três grandes etapas: Pré-processamento, Extração de Padrões e Pós-processamento. Nesta divisão, foram incluídas uma etapa anterior à execução do processo, correspondente ao conhecimento do domínio e identificação do problema, e uma etapa posterior, correspondente à utilização do conhecimento obtido. Essas etapas estão ilustradas na Figura 2.1.

Um aspecto importante a ser observado na figura é que a Extração de Conhecimento de Bases de Dados é um processo iterativo, envolvendo experimentação, iteração entre suas fases e muitas decisões e adaptações, de forma a ajustar o processo para o melhor desempenho ou outras características desejáveis, como, por exemplo, simplicidade da solução. Além disso, a Extração de Conhecimento é um processo centrado na interação entre usuários, e o sucesso do mesmo depende, em parte, da interação entre as diversas classes de usuários. Os usuários do processo são normalmente divididos em três classes (Rocha 1999):

Especialista do domínio Usuário que deve possuir amplo conhecimento do domínio da aplicação e deve fornecer apoio para a execução do processo;

Analista Usuário especialista no processo de Extração de Conhecimento e responsável por sua execução. Este usuário deve conhecer profundamente as etapas que compõem o processo;

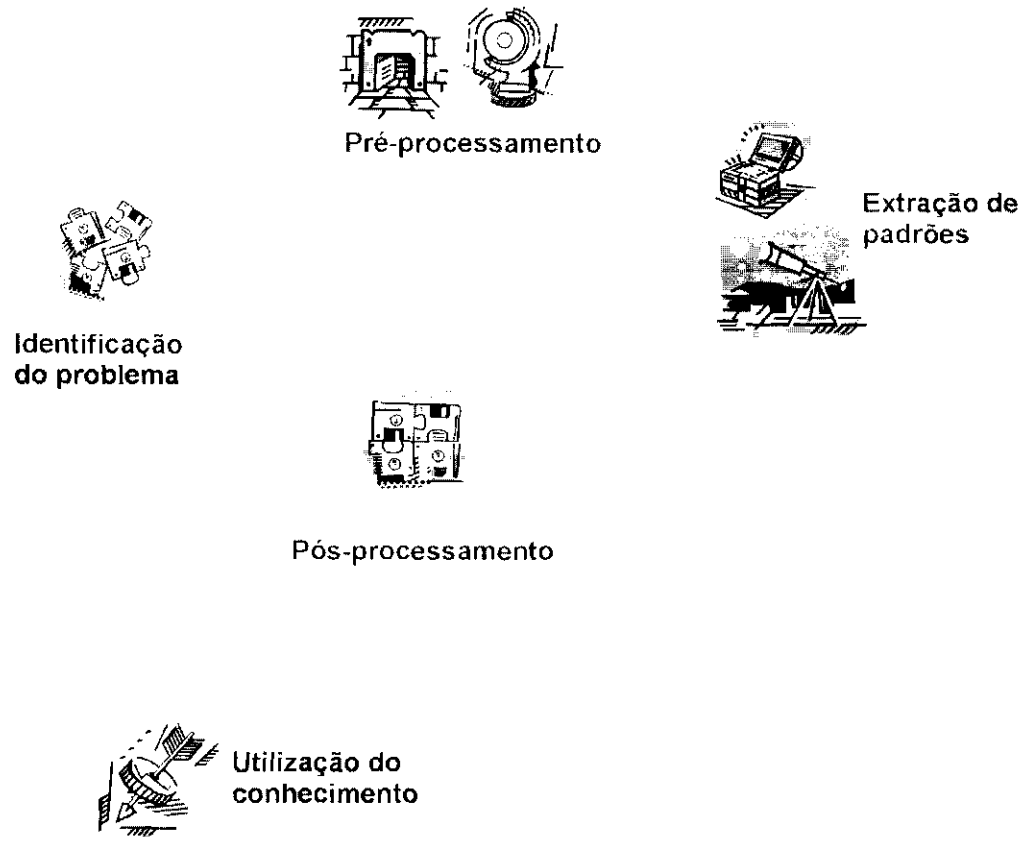

Figura 2.1: Etapas do Processo de Extração de Conhecimento (Rezende, Pugliesi, Melanda, & Paula 2003)

Usuário final Representa a classe de usuário que utiliza o conhecimento extraído no processo para auxiliá-lo em um processo de tomada de decisão. Não é necessário que esse usuário possua conhecimento profundo do domínio da aplicação.

Vale ressaltar que pode haver situações em que o especialista do domínio também é o usuário final, ou que este auxilie ou execute funções pertinentes ao analista. Entretanto, é pouco provável que o analista encontre conhecimento útil a partir dos dados sem o aval do especialista sobre o que seria interessante para um domínio específico (Rocha 1999).

Nas próximas seções são descritas as etapas que compõem o processo de Extração de Conhecimento.

2.2.1 Conhecimento do Domínio e Identificação do Problema

O sucesso do processo de Extração de Conhecimento depende, em parte, da participação dos especialistas do domínio da aplicação em fornecerem um conhecimento sobre o domí-

nio e apoiar os analistas em sua tarefa de encontrar os padrões. Dessa maneira, antes do início das tarefas do processo é importante a realização de um estudo a fim de se adquirir um conhecimento inicial do domínio.

Algumas questões importantes devem ser respondidas nesta fase de aquisição de conhecimento do domínio (Fayyad 1996), como:

- Quais são as principais metas do processo?
- Quais critérios de performance são importantes?
- O conhecimento extraído deve ser compreensível a seres humanos ou um modelo do tipo caixa preta é apropriado?
- Qual deve ser a relação entre simplicidade e precisão do conhecimento extraído?

Além dessa análise inicial para definição das principais metas e restrições, o conhecimento sobre o domínio fornece um subsídio para todas as etapas do processo de Extração de Conhecimento (Baranauskas & Monard 2000; Baranauskas 2001). Mais especificamente, na etapa de Pré-processamento, esse conhecimento pode ajudar os analistas na escolha do melhor conjunto de dados para se realizar a Extração de Padrões, saber quais valores são válidos para os atributos, critérios de preferência entre os possíveis atributos, restrições de relacionamento ou informações para geração de novos atributos.

Na etapa de Extração de Padrões, o conhecimento sobre o domínio pode ajudar os analistas na escolha de um critério de preferência entre os modelos gerados, no ajuste dos parâmetros do processo de indução, ou mesmo na geração de um conhecimento inicial a ser fornecido como entrada do algoritmo de Mineração de Dados para aumentar a eficiência no aprendizado dos conceitos, melhorar a precisão ou compreensibilidade do modelo final.

Na etapa de Pós-processamento, o conhecimento extraído pelos algoritmos de extração de padrões deve ser avaliado. Alguns critérios de avaliação utilizam o conhecimento do especialista para saber, por exemplo, se o conhecimento extraído é interessante ao usuário (Liu & Hsu 1996; Piatetsky-Shapiro & Matheus 1994).

2.2.2 Pré-Processamento

Normalmente, os dados disponíveis para análise não estão em um formato adequado para a Extração de Conhecimento. Além disso, devido a limitações de memória ou tempo de processamento, muitas vezes não é possível a aplicação direta dos algoritmos de extração de padrões sobre os dados. Desta maneira, torna-se necessária a aplicação de métodos

para tratamento, limpeza e redução do volume de dados antes de se iniciar a etapa de Extração de Padrões. É importante salientar que a execução das transformações deve ser guiada pelos objetivos do processo de extração a fim de que o conjunto de dados gerado apresente as características necessárias para que eles sejam cumpridos.

Diversas transformações nos dados podem ser executadas na etapa de Pré-processamento, dentre elas:

Extração e Integração Os dados disponíveis podem ser encontrados em diferentes fontes, como arquivos texto, arquivos no formato MS EXCEL, base de dados ou *Data Warehouse*. Dessa maneira, é necessária a extração desses dados e sua integração, formando uma única fonte de dados no formato atributo-valor, como ilustrado na Tabela 2.1. Esses dados no formato atributo-valor serão utilizados como entrada do algoritmo de extração de padrões.

A Tabela 2.1 representa um conjunto de dados com n exemplos de k atributos. Uma linha E_i representa o i-ésimo exemplo (i = 1, 2, ..., n) e uma coluna A_j representa o j-ésimo atributo (j = 1, 2, ..., k). Um valor $v_{i,j}$ corresponde ao valor do atributo A_j do exemplo E_i. A coluna C é um atributo especial, denominado classe ou meta. O valor desse atributo deve ser predito para novos exemplos pelo modelo encontrado, caso se esteja realizando uma tarefa de predição. Dependendo da função escolhida no processo de extração, não é necessária a existência desse atributo meta. Mais detalhes sobre a função desejada pela solução a ser encontrada e sobre o atributo especial C são encontrados na Seção 2.2.3.

	A_1	A_2	$\cdots$	A_k	C
E_1	$v_{1,1}$	$v_{1,2}$	$\cdots$	$v_{1,k}$	c_1
E_2	$v_{2,1}$	$v_{2,2}$	$\cdots$	$v_{2,k}$	c_2
E_3	$v_{3,1}$	$v_{3,2}$	$\cdots$	$v_{3,k}$	c_3
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
E_n	$v_{n,1}$	$v_{n,2}$	$\cdots$	$v_{n,k}$	c_n

Tabela 2.1: Conjunto de Dados no Formato Atributo-Valor

Transformação Após a extração e integração dos dados, estes devem ser adequados para serem utilizados pelos algoritmos de extração de padrões. 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 tipo, por exemplo quando um atributo do tipo data é transformado em um outro tipo para

que o algoritmo de extração de padrões possa utilizá-lo de forma mais adequada, e normalização de atributos contínuos, colocando seus valores em intervalos definidos como, por exemplo, entre 0 e 1.

Limpeza Os dados disponíveis para aplicação dos algoritmos de extração de padrões podem apresentar problemas advindos do processo de coleta. Estes problemas podem corresponder a erros de digitação ou erros na leitura dos dados por sensores. Como o resultado do processo de extração possivelmente será utilizado em um processo de tomada de decisão, a qualidade dos dados é um fator extremamente importante. Desta maneira, técnicas de limpeza devem ser aplicadas aos dados a fim de garantir sua qualidade.

A limpeza dos dados pode ser realizada utilizando o conhecimento do domínio. Por exemplo, pode-se encontrar registros com valor inválido em algum atributo. Pode-se também efetuar limpeza independente de domínio, como decisão da estratégia de tratamento de atributos incompletos, remoção de ruído ou tratamento de conjunto de exemplos não balanceados (Batista, Carvalho, & Monard 2000).

Redução de Dados Devido a restrições de espaço em memória ou tempo de processamento, o número de exemplos e de atributos disponíveis para análise pode inviabilizar a utilização de algoritmos de extração de padrões. Como solução para esse problema, pode ser necessária a aplicação de métodos para redução dos dados antes de se iniciar a busca pelos padrões. Esta redução pode ser feita de três maneiras (Weiss & Indurkhya 1998):

- Redução do número de exemplos;
- Redução do número de atributos;
- Redução do número de valores de um atributo.

A redução do número de exemplos deve ser feita de forma a manter as características do conjunto de dados original, isto é, através da geração de amostras representativas dos dados (Glymour, Madigan, Pregibon, & Smyth 1997). A principal abordagem para redução do número de exemplos é realizar amostragem aleatória (Weiss & Indurkhya 1998), pois este método tende a produzir amostras representativas.

É importante salientar que se a amostra não for representativa, ou se a quantidade de exemplos for insuficiente para caracterizar os padrões embutidos nos dados, os

modelos encontrados podem não representar o comportamento dos dados originais, não tendo, portanto, valor.

A redução do número de atributos também pode ser utilizada como forma de se reduzir o espaço de busca pela solução. O objetivo é selecionar um subconjunto dos atributos existentes de forma que isto não tenha grande impacto na qualidade da solução final. Esta redução pode ser realizada com o apoio do especialista do domínio, uma vez que ao se remover um atributo potencialmente útil para o modelo final, a qualidade do conhecimento extraído pode diminuir consideravelmente. Além disso, por não se saber inicialmente quais atributos serão importantes para atingir os objetivos, deve-se remover somente aqueles atributos que sabe-se com certeza não ter nenhuma importância para o modelo final.

Também como forma de redução do número de atributos, pode-se utilizar indução construtiva, em que um novo atributo é criado a partir do valor de outros. Caso os atributos originais utilizados na construção do novo atributo não estejam presentes em um novo modelo, eles podem ser descartados, reduzindo assim o número de atributos. A utilização de indução construtiva pode aumentar consideravelmente a qualidade do conhecimento extraído (Lee 2000).

A terceira forma de redução dos dados consiste na redução do número de valores de um atributo. Isso é feito, normalmente, por discretização ou suavização dos valores de um atributo contínuo.

Discretização de um atributo consiste na substituição de um atributo contínuo (inteiro ou real) por um atributo discreto pelo agrupamento de seus valores. Em essência, um algoritmo de discretização aceita como entrada os valores de um atributo contínuo e gera como saída uma pequena lista de intervalos ordenados. Cada intervalo é representado na forma $[V_{inferior} : V_{superior}]$, de tal forma que $V_{inferior}$ e $V_{superior}$ são os limites inferiores e superiores do intervalo. Os métodos de discretização podem ser classificados em supervisionados ou não supervisionados, locais ou globais, e parametrizados ou não parametrizados (Félix, Rezende, Monard, & Caulkins 2000).

Na suavização dos valores de um atributo, o objetivo é diminuir o número de valores do mesmo sem discretizá-lo. Nesse método, os valores de um determinado atributo são agrupados, mas ao contrário da discretização, cada grupo de valores é substituído por um valor numérico que o represente. Esse novo valor pode ser a média, a mediana ou mesmo os valores de borda de cada grupo (Weiss & Indurkhya 1998).

A aplicação das transformações apresentadas nesta seção deve ser realizada de forma bastante cuidadosa. É importante garantir que as informações presentes nos dados brutos continuem presentes nas amostras geradas de forma que os modelos finais representem a realidade expressa nos dados brutos.

A etapa de Pré-processamento é realizada antes da Extração de Padrões. Mas, como o processo de Extração de Conhecimento é iterativo, algumas atividades de Pré-processamento podem ser realizadas novamente ao se analisar os padrões encontrados na etapa de Extração de Padrões. Por exemplo, pode-se desejar acrescentar algum atributo, reduzir ou aumentar o volume de dados, fazer uma outra transformação no tipo de algum atributo, a fim de que o indutor possa utilizá-lo de forma mais eficiente e melhorar a qualidade do conhecimento extraído.

2.2.3 Extração de Padrões

O processo de Extração de Conhecimento é direcionado a cumprir um determinado objetivo. A definição desse objetivo depende das necessidades dos usuários finais do processo e da disponibilidade de dados. Com esses objetivos e os dados preparados, a etapa de Extração de Padrões trata da escolha, configuração e execução de um ou mais algoritmos visando cumprir os objetivos determinados.

Por ser um processo iterativo, esta etapa do processo de Extração de Conhecimento pode ser executada diversas vezes para ajustar todos os parâmetros a fim de se obter um resultado mais adequado aos objetivos, como por exemplo, quanto à precisão ou compreensibilidade do conhecimento extraído.

Esta etapa pode ser subdividida em três sub-etapas (Fayyad, Piatetsky-Shapiro, & Smyth 1996; Félix 1998):

Escolha da função A escolha da função é feita de acordo com os objetivos desejáveis para a solução a ser encontrada pelo algoritmo. As funções possíveis de um algoritmo de extração de padrões podem ser agrupadas em duas grandes categorias: predição e descrição.

Funções relacionadas à predição, ou Mineração de Dados preditivo, consistem na generalização de exemplos com suas respectivas classes conhecidas em uma linguagem capaz de reconhecer a classe de um novo exemplo. Os dois principais tipos de problemas de predição são classificação e regressão. Problemas de classificação consistem na predição de um valor categórico como, por exemplo, predizer se o cliente é bom ou mau pagador. Nos problemas de regressão, o atributo a ser predito consiste em

um valor contínuo como, por exemplo, predizer o lucro em um empréstimo (Weiss & Indurkhya 1998).

Funções relacionadas à descrição, ou Mineração de Dados descritivo, consistem na identificação de comportamentos no conjunto de dados, sendo que estes dados não possuem um atributo-meta especificado, como acontece na predição. Alguns dos problemas de descrição são a geração de *clusters*, geração de regras de associação e sumarização.

Escolha do Algoritmo Com a função desejada escolhida, existe uma variedade de algoritmos para realizá-la. Nesta etapa deve ser escolhido o algoritmo a ser utilizado, bem como a configuração de seus parâmetros.

A escolha do algoritmo está relacionada com a linguagem de representação dos padrões encontrados. Por exemplo, se o objetivo é realizar uma classificação, tem-se a possibilidade de utilizar algoritmos indutores de árvores de decisão, regras de produção, entre outros. Tipos de representações comuns são: árvores de decisão, regras, modelos lineares, modelos não lineares (Redes Neurais), modelos baseados em exemplos (KNN – *K-Nearest Neighbour*, Raciocínio Baseado em Casos) e modelos de dependência probabilística (Redes Bayesianas).

Um aspecto que merece atenção é a complexidade da solução encontrada pelo algoritmo de extração. A complexidade da solução está diretamente relacionada com a capacidade de representação do conceito embutido nos dados. O problema é que quando os parâmetros do algoritmo estão ajustados para encontrar soluções mais complexas que o conceito efetivamente existente nos dados, o modelo resultante poderá ter uma precisão boa para o conjunto de treinamento, mas um desempenho ruim para novos exemplos. Diz-se então que o modelo é específico para o conjunto de treinamento, ou seja, é pouco genérico. Neste caso, ocorreu *overfitting*.

Por outro lado, se a complexidade da solução que está sendo buscada não for suficiente para adequar o conceito representado nos dados, por exemplo, ao se definir um número insuficiente de neurônios para uma Rede Neural ou um fator de poda muito alto na geração de uma Árvore de Decisão, pode-se não conseguir induzir um modelo representativo (Baranauskas & Monard 2000). Isto é o que se chama de *underfitting*. Neste caso, o modelo encontrado pode não ter uma boa performance tanto nos exemplos disponíveis para o treinamento quanto para novos exemplos.

Desta maneira, a configuração dos parâmetros do algoritmo deve ser feita de forma bastante cuidadosa. Em (Kearns & Vazirani 1994) é sugerido que, na escolha de uma determinada solução, o modelo mais apropriado é aquele mais simples que seja consistente com todas as observações.

Em (Kohavi, Sommerfield, & Dougherty 1997) é mostrado experimentalmente que não existe um único bom algoritmo para todas as tarefas. Dessa maneira, a escolha de vários algoritmos para realizar a tarefa desejada pode ser feita. Isto levaria à obtenção de diversos modelos que, na etapa de Pós-processamento, seriam tratados a fim de fornecer o conjunto de padrões mais adequado para o usuário final.

Extração de Padrões A etapa de Extração de Padrões do processo de Extração de Conhecimento de Bases de Dados consiste na aplicação dos algoritmos de mineração escolhidos para a extração dos padrões embutidos nos dados. É importante ressaltar que, geralmente, essa etapa consiste de diversas execuções dos algoritmos de extração de padrões.

Técnicas de combinação de preditores têm sido pesquisadas com o objetivo de construir um preditor mais preciso pela combinação de vários outros. O resultado dessa combinação é chamado de *ensemble*. A utilização de *ensembles* tem obtido melhores resultados que a utilização de um único preditor (Dietterich 2000; Breiman 2000).

Um outro aspecto importante é o fato de que, com o objetivo de se fazer uma avaliação mais precisa da taxa de erro de um classificador, métodos de *resampling* têm sido normalmente utilizados na Extração de Padrões. *Resampling* consiste na realização de vários experimentos em que o classificador é treinado e testado com diferentes conjuntos de treinamento e teste disjuntos. Como resultado de cada um desses experimentos, é obtida uma taxa de erro. A média dessas taxas de erro é considerada uma boa aproximação para a taxa de erro do classificador para exemplos futuros.

Em um dos métodos de *resampling*, *k-fold cross-validation*, o conjunto de exemplos é dividido em k partições mutuamente exclusivas. Em cada iteração do método, k-1 partições são fornecidas ao algoritmo de aprendizado e a restante é utilizada para calcular a taxa de erro da iteração. A taxa de erro final estimada pelo método é a média das taxas de erro de cada uma das k iterações.

Após a etapa de Extração de Padrões, o conjunto de padrões descoberto deve ser analisado ou processado na etapa de Pós-processamento para ser então fornecido ao usuário.

2.2.4 Pós-Processamento

O processo de Extração de Conhecimento de Bases de Dados não termina após a descoberta dos padrões existentes nos dados. Uma vez que o conhecimento extraído tem sido utilizado para resolver problemas da vida real, sendo utilizado em um sistema inteligente ou por um ser humano em um processo de tomada de decisão, concorda-se que a obtenção do conhecimento não é o passo final do processo. Além disso, o fato de se utilizar técnicas de Extração de Conhecimento de Bases de Dados não significa que os usuários não possuem nenhum conhecimento prévio sobre os dados. Portanto, é importante que algumas questões sejam respondidas aos usuários (Liu & Hsu 1996):

- O conhecimento extraído representa o conhecimento do especialista?
- De que maneira o conhecimento do especialista difere do conhecimento extraído?
- Em que parte o conhecimento do especialista está correto?

Outro aspecto importante é que os algoritmos de extração de padrões podem gerar uma quantidade enorme de padrões, muitos dos quais podem não ser importantes, relevantes ou interessantes para o usuário, o que pode tornar inviável a tarefa realizada pelos usuários de encontrar o conhecimento interessante. Dessa maneira, muitos estudos têm sido direcionados ao desenvolvimento de técnicas para apoiar os usuários na seleção dos padrões mais interessantes (Silberschatz & Tuzhilin 1995). Uma descrição mais detalhada do problema da interessabilidade do conhecimento extraído é apresentada na Seção 3.2.

Um dos objetivos principais do processo de Extração de Conhecimento é que o usuário possa compreender e utilizar o conhecimento descoberto. Entretanto, podem ocorrer casos em que os modelos são muito complexos ou não fazem sentido para os especialistas (Pazzani 2000; Pazzani, Mani, & Shankle 1997). Assim, a compreensibilidade do conhecimento extraído é um outro aspecto bastante importante para o processo de Extração de Conhecimento de Bases de Dados.

A compreensibilidade de um dado conjunto de regras está relacionada com a facilidade de interpretação dessas regras por um ser humano. A compreensibilidade de um modelo pode ser estimada, por exemplo, pelo número de regras e o número de condições por regra. Nesse caso, quanto menor a quantidade de regras de um dado modelo e menor o número de condições por regra, maior será a compreensibilidade das regras descobertas (Fertig, Freitas, Arruda, & Kaestner 1999). Em (Pazzani 2000; Pazzani, Mani, & Shankle 1997) é discutido que outros fatores, além do tamanho do modelo, são importantes na determi-

nação da compreensibilidade de um conhecimento. Um fator citado é que os especialistas possuem tendência a compreender melhor modelos que não contradizem seu conhecimento prévio.

Após a análise do conhecimento nessa etapa de Pós-processamento, caso o conhecimento não seja de interesse do usuário final ou não cumpra com os objetivos propostos, o processo de extração pode ser repetido ajustando seus parâmetros ou melhorando o processo de escolha dos dados para a obtenção de resultados melhores numa próxima iteração.

No final do processo de Extração de Conhecimento, é interessante que todo o conhecimento adquirido seja disponibilizado em um ambiente adequado de forma a facilitar sua validação, interpretação e utilização pelo usuário final. Uma descrição geral de um Ambiente para Exploração de Regras é apresentada no Capítulo 4.

2.3 Ferramentas de Mineração de Dados

O desenvolvimento da tecnologia de Extração de Conhecimento e sua utilização nos mais variados domínios e pelas mais diversas organizações tem motivado o desenvolvimento de várias ferramentas para apoiar o processo e a elaboração de muitos protótipos de pesquisa. Esta seção apresenta uma visão geral de alguns protótipos de pesquisa de ferramentas de Mineração de Dados que possuem funcionalidades inovadoras e de alguns produtos comerciais disponíveis no mercado.

2.3.1 Protótipos de Pesquisa

Nos últimos anos, diversos protótipos de ferramentas de Mineração de Dados têm sido desenvolvidos em inúmeras universidades e laboratórios de pesquisa. Com o objetivo de ilustrar as principais áreas que têm recebido maior atenção dos pesquisadores do processo de Extração de Conhecimento, os protótipos apresentados nesta seção foram agrupados nas seguintes categorias (Thuraisingham 1999):

Novos Modelos Funcionais As ferramentas pertencentes a esta categoria são aquelas que visam desenvolver novos modelos funcionais para Mineração de Dados. Essencialmente, na maioria destes projetos, o objetivo é integrar métodos de Mineração de Dados com o gerenciamento de base de dados. A idéia principal desses trabalhos é desenvolver novas técnicas para otimizar as consultas e permitir um melhor suporte aos métodos de Mineração de Dados. Um exemplo de protótipo pertencente a esta categoria é o projeto *Queryflocks*, desenvolvido em conjunto pela Universida-

de de Stanford e as organizações MITRE e Hitachi (Tsur, Ullman, Abiteboul, Clifton, Motwani, Nestorov, & A. 1998).

Tratamento de Novos Tipos de Dados O tratamento de diferentes tipos de dados, como dados multimídia, é o objetivo dos protótipos pertencentes a esta categoria. Também pertencem a esta categoria as ferramentas que apóiam o processo de *Text Mining*. O protótipo *Queryflocks* pode ser utilizado para encontrar associações em bases textuais, pertencendo também a esta categoria.

Alguns protótipos são voltados à extração de conhecimento a partir de imagens (*Image Mining*). O protótipo SKICAT (*Sky Image Cataloging and Analysis Tool*), desenvolvido pelo JPL (*Jet Propulsion Laboratory*), visa detectar objetos desconhecidos a partir de imagens espaciais (Fayyad, Haussler, & Stolors 1996).

Escalabilidade Um problema identificado ao se utilizar muitos dos métodos de extração de conhecimento é que estes não foram projetados para tratar grandes volumes de dados. Por outro lado, com o desenvolvimento da tecnologia de armazenamento, o volume de dados disponível para análise tem crescido exponencialmente. Desta maneira, diversos projetos de pesquisa têm sido desenvolvidos com objetivo de tratar grandes bases de dados. Alguns trabalhos que tratam da escalabilidade dos algoritmos de extração de conhecimento têm sido desenvolvidos pela *Thinking Machines Corporation* e pela *Silicon Graphics*.

Compreensibilidade dos Resultados Um outro problema identificado no final do processo de Extração de Conhecimento de Bases de Dados é a compreensibilidade do resultado obtido. Os padrões extraídos pelos algoritmos podem ser muito complexos ou não fazerem sentido para os usuários do processo de Extração de Conhecimento, dificultando sua compreensão e possível utilização (Pazzani 2000; Pazzani, Mani, & Shankle 1997). Assim, muitos projetos têm sido voltados a tratar este aspecto. A visualização dos padrões extraídos tem sido utilizada nesses projetos como principal técnica para auxiliar a compreensão dos mesmos (Thearling, Becker, DeCoste, Mawby, Pilote, & Sommerfield 2002).

2.3.2 Produtos Comerciais

O desenvolvimento de ferramentas comerciais de Mineração de Dados tem como objetivo principal fornecer aos tomadores de decisão das organizações, usuários normalmente não especialistas em Mineração de Dados, ferramentas intuitivas e amigáveis (Freitas &

Lavington 1998). Nesta seção é fornecida uma visão geral de alguns produtos comerciais de Mineração de Dados disponíveis no mercado.

Clementine

Clementine[1] é uma ferramenta integrada que implementa basicamente indução de árvores de decisão (C5.0), algoritmos de Redes Neurais (*Multi-layer Perceptron*-MLP e redes *Kohonen*), *k-means*, regras de associação e regressão linear (Freitas & Lavington 1998).

Clementine oferece uma interface gráfica baseada em programação visual. Isto permite ao usuário especificar operações de Mineração de Dados, incluindo de Pré-processamento, Extração de Padrões e Pós-processamento, para serem executadas "arrastando" ícones e objetos relacionados facilitando a criação de uma seqüência de operações.

Uma outra característica da ferramenta Clementine é que ela pode fazer uma configuração automática dos parâmetros dos algoritmos de AM. A ferramenta apresenta três níveis de configuração dos parâmetros. No nível mais autônomo, todos os parâmetros são determinados automaticamente, baseando-se nos dados a serem minerados. No nível intermediário, a ferramenta permite ao usuário especificar os parâmetros de mais alto nível, como por exemplo o fator de generalidade das regras descobertas. No nível menos autônomo, a ferramenta permite ao usuário a configuração de todos os parâmetros (Freitas & Lavington 1998).

DARWIN

A ferramenta DARWIN[2] foi projetada para minerar grandes bases de dados utilizando paralelismo. A ferramenta é constituída de vários módulos (Freitas & Lavington 1998; Tamayo, Berlin, Dayanand, Drescher, Mani, & Wang 1997).

StarData Módulo que provê todas as funcionalidades de manipulação e transformação de dados utilizadas pelos outros módulos da ferramenta DARWIN;

StarTree Módulo baseado no algoritmo CART (*Classification and Regression Tree*) (Breiman, Friedman, Olshen, & Stone 1984), podendo realizar tanto tarefas de classificação quanto de regressão. O algoritmo CART paralelo implementado nessa ferramenta constrói árvores de decisão idênticas às construídas pelo algoritmo seqüencial;

StarNet Módulo baseado em algoritmos de Redes Neurais. Esse módulo oferece várias funções de ativação para serem utilizadas nas redes, além de um procedimento espe-

[1] http://www.spss.com/Clementine (último acesso: 21/01/2003)
[2] http://otn.oracle.com/products/datamining/content.html (último acesso: 21/01/2003)

cial para seleção automática do tamanho da rede. Esse módulo permite a adição de novas funções de ativação, medidas de erro e algoritmos de treinamento da rede;

StarMatch Módulo baseado no algoritmo *K-nearest neighbor* com atributos ponderados. Esse módulo permite a especificação manual ou automática dos pesos dos atributos e utiliza estruturas de dados do tipo *KDTrees* para aumentar a velocidade de busca dos exemplos armazenados mais similares a um novo exemplo fornecido;

StarGene Módulo baseado em Algoritmos Genéticos cuja função principal é otimizar os parâmetros usados por outros módulos da ferramenta. Por exemplo, o módulo *StarGene* pode ser usado para otimizar os pesos de interconexão de uma rede neural no módulo *StarNet*.

Mineset

O Mineset^{TM} (Motroni & Vanderberg 2000) consiste de um software produzido pela *Silicon Graphics* para mineração de dados. O Mineset^{TM} possui algumas ferramentas de Pré-processamento dos dados, um conjunto de ferramentas para Extração de Padrões e ferramentas de visualização de dados e dos modelos encontrados pelos algoritmos de Mineração de Dados.

O Mineset^{TM} utiliza uma Biblioteca de Algoritmos de Aprendizado de Máquina implementada em C^{++}, chamada de MLC^{++}, em que estão implementados os algoritmos de Aprendizado de Máquina utilizados pela ferramenta. Essa biblioteca possui um papel duplo por apresentar os principais algoritmos de aprendizado de máquina implementados e prontos para serem utilizados e por fornecer um conjunto de classes para auxiliar os desenvolvedores na construção de novos algoritmos (Félix, Rezende, Doi, Paula, & Romanato 1998).

A ferramenta Mineset^{TM} está dividida nos seguintes módulos:

Módulo de Controle Centralizado Módulo constituído de uma interface gráfica chamada *Tool Manager* e de um processo chamado *Data Mover*. A *Tool Manager* deve ser executada em uma estação cliente e é responsável pela maioria das interações com os componentes do Mineset^{TM}. O *Data Mover* é executado no servidor do Mineset^{TM} e é responsável principalmente pelo acesso e pela transformação dos dados utilizados na ferramenta;

Módulo de Ferramentas de Mineração de Dados Módulo constituído de um conjunto de algoritmos para extração de modelos e padrões dos dados. Os algoritmos implemen-

tados são Indutores de Árvores de Decisão e de Regressão, Cálculo da Importância de Colunas, Gerador de Regras de Associação, Gerador de *Clusters*, Indutor de Evidência, Discretizador Automático e Indutor de Tabelas de Decisão.

Módulo de Visualização Módulo constituido de um conjunto de ferramentas para visualização de dados e dos modelos ou padrões extraídos pelos algoritmos existentes no módulo de Ferramentas de Mineração de Dados. As ferramentas de visualização são Visualizador de Árvores, Visualizador de Mapas, Visualizador de Regras, Visualizador de Evidência, Visualizador de Registros, Visualizador de Estatísticas, Visualizador de *Clusters*, Visualizador de Tabelas de Decisão, *Scatter Visualizer* e *Splat Visualizer*.

Intelligent Miner

A Ferramenta Intelligent Miner[3] incorpora algumas pesquisas que tem sido desenvolvidas pela IBM. Intelligent Miner é uma ferramenta de Mineração de Dados que pode executar muitas tarefas, incluindo descoberta de regras de associação e padrões seqüenciais, classificação, clusterização e modelagem de dependência. A ferramenta também trata o problema de dados incompletos, detecção de ruídos e categorização de dados contínuos (Freitas & Lavington 1998; Thuraisingham 1999).

Com relação à classificação, o sistema inclui algoritmos de indução de árvores de decisão e de Redes Neurais. Dentre os algoritmos utilizados em Redes Neurais, encontram-se: *backpropagation*, *radial basis functions*, *adaptive resonance networks* e mapas de *kohonen* (Freitas & Lavington 1998). Alguns desses algoritmos podem ser utilizados para realizar *clustering*.

DataMind

Em um projeto conjunto entre a corporação Red Brick, especializada em *Data Warehousing*, e a corporação DataMind foi desenvolvido um produto chamado DataMind que integra produtos de *Data Warehousing* e Mineração de Dados (Thuraisingham 1999).

A ferramenta de *Data Warehouse* da Red Brick é uma base de dados relacional especialmente projetado para atender o mercado de *Data Warehouse*, apresentando algumas características importantes para este tipo de aplicação. As funcionalidades de Mineração de Dados do produto da Red Brick é integrada ao produto de *Data Warehousing* como uma extensão dos comandos SQL.

[3]http://www-3.ibm.com/software/data/iminer (último acesso: 21/01/2003)

WEKA (Waikato Environment for Knowledge Analysis)

WEKA[4] (Witten & Frank 1999) consiste de uma biblioteca contendo implementações na linguagem Java de alguns algoritmos de AM. Como na MLC^{++}, os algoritmos implementados na WEKA podem ser utilizados diretamente pelos usuários ou tem-se a possibilidade de utilizar as classes existentes na elaboração de variantes dos algoritmos já implementados ou na construção de novos algoritmos. Alguns dos algoritmos atualmente implementados são J48 (versão Java do C4.5), ID3, Naive Bayes, Regressão Linear e M5.

A WEKA também possui algumas ferramentas que executam tarefas de preparação de dados, como normalização, discretização, redução do número de valores e redução do número de exemplos.

2.4 Considerações Finais

Recentemente, com o desenvolvimento da tecnologia de armazenamento, a quantidade de dados disponível nas bases de dados das organizações tem crescido demasiadamente. Este crescimento inviabilizou a utilização de técnicas manuais para análise desses dados. Ao mesmo tempo, as organizações têm visto a possibilidade de utilizar seus dados a fim de obter conhecimento sobre seus clientes, produtos e parceiros, e assim adquirir vantagem competitiva perante seus concorrentes.

Nesse contexto, os pesquisadores da área de Inteligência Artificial têm direcionado estudos para o processo de Extração de Conhecimento de Bases de Dados, que visa automatizar a tarefa de extrair conhecimento útil a partir de grandes volumes de dados.

Este capítulo teve o objetivo de apresentar o processo de Extração de Conhecimento de Bases de Dados, detalhando as atividades que devem ser realizadas desde a compreensão do domínio, preparação dos dados, até a avaliação do conhecimento extraído para verificar se este é de interesse do usuário final, além de apresentar algumas ferramentas disponíveis no mercado para apoiar o processo de Extração de Conhecimento.

Conforme apresentado, pode-se observar que o processo de Extração de Conhecimento é bastante complexo e trabalhoso, pois envolve a execução de muitas tarefas, configuração de diversos parâmetros e grande interação com os usuários. Contudo, o sucesso do processo pode trazer uma recompensa valiosa para as organizações.

[4] http://www.cs.waikato.ac.nz/~ml/weka/index.html (último acesso: 21/01/2003)

O próximo capítulo é focado em alguns aspectos relativos ao Pós-processamento de conhecimento, mais especificamente sobre a avaliação de regras, apresentando algumas medidas objctivas e subjetivas para avaliação de regras.

Capítulo 3

Avaliação de Regras

3.1 Considerações Iniciais

A Extração de Conhecimento de Bases de Dados é um processo voltado a atender às necessidades de seus usuários finais. O objetivo final do processo é que esses usuários obtenham algum tipo de vantagem ao utilizar o conhecimento encontrado a partir dos dados. Contudo, a utilização efetiva desse conhecimento depende de fatores como a identificação do conhecimento mais interessante ou relevante e a facilidade de sua interpretação.

No contexto da disponibilização de conhecimento, são tratados modelos que possam ser compreendidos, interpretados e utilizados pelos usuários do processo, ou seja, são considerados principalmente os modelos simbólicos de Aprendizado de Máquina que utilizam regras como linguagem para representação das hipóteses. O objetivo é que os usuários possam olhar "dentro" desses modelos, isto é, os usuários podem analisar as regras existentes dentro de um modelo e utilizar esse conhecimento para apoiar um processo de tomada de decisão. Além de modelos simbólicos de classificação, também são considerados modelos simbólicos para regressão e conjuntos de regras de associação.

Análises mais detalhadas dos conjuntos de regras são necessárias para efetivamente apoiar os usuários na avaliação do conhecimento, sendo necessária a utilização de métodos mais sofisticados do que simplesmente utilizar a precisão de um modelo. Assim, maior atenção tem sido dada pelos pesquisadores do processo de Mineração de Dados a problemas relacionados à utilização e compreensão do conhecimento, e diversos trabalhos têm sido desenvolvidos, tendo sido propostas diversas medidas para avaliação de regras. Essas medidas permitem avaliar tanto o desempenho quanto a qualidade das regras.

O desempenho de uma regra está associado à fidelidade com que esta representa os dados. Por exemplo, uma medida de desempenho é a precisão da regra. Quanto à avaliação de qualidade, conceitos como compreensibilidade e interessabilidade são utilizados. A compreensibilidade de um dado conjunto de regras está relacionada à sua facilidade de interpretação pelos usuários, podendo ser estimada, por exemplo, pelo número de regras e o número de condições por regra, conforme apresentado no Capítulo 2. A interessabilidade, por sua vez, é uma avaliação realizada a partir de estimativas da quantidade de conhecimento interessante ao usuário está presente no conjunto de regras. Medidas de interessabilidade podem estar relacionadas, por exemplo, à utilidade ou novidade do conjunto de regras.

Nesse sentido, esse capítulo trata de alguns desses problemas, apresentando em mais detalhes o problema da interessabilidade do conhecimento extraído, assim como algumas medidas de avaliação quanto a desempenho e qualidade das regras.

3.2 Interessabilidade de Regras

Um problema que tem sido identificado no processo de Extração de Conhecimento é que, normalmente, o número de padrões encontrados a partir dos dados é muito grande. Devido a esse grande número de padrões, sua análise é certamente bastante difícil de ser executada manualmente. Normalmente, os usuários esperam receber somente uma pequena coleção de padrões que sejam úteis, novos e interessantes (Liu, Hsu, & Ma 2001; Liu, Hsu, & Ma 1999; Klemettinen, Mannila, & Toivonen 1997; Liu, Hsu, Mun, & Lee 1996). Alguns fatores que levam um conhecimento a não ser de interesse do usuário são (Padmanabhan & Tuzhilin 2000; Klemettinen, Mannila, Ronkainen, Toivonen, & Verkamo 1994):

- Uma regra pode corresponder a um conhecimento prévio;
- Uma regra pode referenciar atributos ou uma combinação de atributos não interessantes;
- Regras podem ser redundantes.

Desta maneira, é extremamente importante para o processo de Extração de Conhecimento o desenvolvimento de técnicas para seleção de regras interessantes. Essas técnicas são freqüentemente referenciadas na literatura como Interessabilidade de Regras.

Muitas pesquisas têm mostrado que a identificação e remoção dos padrões redundantes ou não interessantes para os usuários pode ser realizada por um processo automático.

Como resultado dessas pesquisas, diversas medidas de interessabilidade e ferramentas de apoio à seleção de padrões interessantes têm sido desenvolvidas pelos pesquisadores do processo de Extração de Conhecimento para auxiliar os usuários nessa tarefa de identificação dos padrões interessantes (Liu, Hsu, & Ma 2001; Liu, Hsu, & Ma 1999; Lavrac, Flach, & Zupan 1999; Klemettinen, Mannila, & Toivonen 1997; Silberschatz & Tuzhilin 1996; Piatetsky-Shapiro & Matheus 1994).

A interessabilidade se refere ao grau que um dado padrão descoberto é de interesse para um usuário final do processo de Extração de Conhecimento de Bases de Dados e está relacionada a fatores como novidade, utilidade, relevância e significância estatística. Um sistema automático de descoberta de conhecimento que seja interessante para o usuário requer que fatores de interesse sejam especificados e combinados numa medida que reflita como o especialista no domínio julga os padrões, para que seja possível a análise automática da interessabilidade dos padrões descobertos (Piatetsky-Shapiro & Matheus 1994). A identificação desses aspectos que permitem a determinação do conhecimento interessante não é simples e é extremamente dependente do domínio e do usuário.

Em (Liu, Hsu, Mun, & Lee 1996), a interessabilidade foi definida como: Seja B o conjunto de todos os padrões descobertos a partir de uma base de dados D pela aplicação de uma técnica Q, o problema da interessabilidade consiste em determinar o conjunto I de padrões interessantes e ordenar esses padrões de acordo com o grau de interesse do usuário em um determinado instante de tempo.

Alguns aspectos importantes sobre essa definição devem ser ressaltados:

- O conjunto I pode não ser o conjunto completo de todos os padrões que foram descobertos a partir de D. Geralmente, I é apenas um subconjunto dos padrões contidos em B;

- Nem todos os padrões de I são igualmente interessantes. Dois padrões diferentes pertencentes ao conjunto I podem ter diferentes graus de interesse para o usuário;

- I é um conjunto dinâmico, pois é dependente do usuário e do tempo. Diferentes usuários podem possuir interesse em diferentes subconjuntos do conjunto I, ou ainda, um único usuário pode ter interesse em diferentes subconjuntos em diferentes momentos.

É extremamente importante que seja possível projetar técnicas gerais de seleção e ordenação de padrões de acordo com sua interessabilidade. Diversas pesquisas têm sido

realizadas no sentido de desenvolver medidas para avaliação de interessabilidade independentes do usuário e do domínio (Freitas 1998b). Essas medidas mais gerais são chamadas de medidas objetivas para avaliação de conhecimento.

As medidas objetivas para avaliação de padrões dependem somente da estrutura dos padrões e dos dados que os originaram. Algumas dessas medidas estão detalhadas na Seção 3.3. Essas medidas são úteis em muitos aspectos, mas, por não levarem em consideração fatores específicos do usuário, elas são insuficientes para selecionar precisamente o foco de interesse do usuário, devido às características do problema de interessabilidade.

Dessa maneira, foram desenvolvidas medidas que levam em consideração aspectos relacionados ao usuário e ao domínio da aplicação. Por esse motivo, essas medidas são normalmente chamadas de medidas subjetivas para avaliação de conhecimento. Na Seção 3.4 algumas dessas medidas são detalhadas.

3.3 Aspectos Objetivos para Avaliação de Regras

Como as medidas objetivas dependem somente dos dados e da estrutura dos padrões, elas são, por definição, mais genéricas que medidas subjetivas. Isso motivou o desenvolvimento de muitas medidas objetivas para avaliação de regras, sendo algumas delas apresentadas nesta seção.

3.3.1 Medidas de Surpresa

Um conhecimento pode ser considerado surpreendente para o usuário se ele for previamente desconhecido ou contrariar seu conhecimento. Por ser bastante dependente do conhecimento que o usuário possui, a surpresa de regras é, normalmente, considerada um aspecto subjetivo. Entretanto, em (Freitas 1998b) é mostrado que é possível definir algumas medidas objetivas para identificação da surpresa de regras.

Surpresa dos Pequenos Disjuntos

Um conjunto de regras pode ser referenciado como uma disjunção de regras, e uma regra individual pode ser chamada de disjunto. O tamanho do disjunto representa o número de exemplos do conjunto de dados em que a parte condicional da regra é verdadeira, isto é, o número de exemplos cobertos pela regra. Pode-se então definir um pequeno disjunto como uma regra que cobre poucos exemplos do conjunto de dados. A determinação de quais regras são pequenos disjuntos é feita de acordo com algum critério específico, como por exemplo, um limite máximo para o número de exemplos cobertos pela regra. Nesse caso,

uma regra será considerada um pequeno disjunto se o número de exemplos cobertos por ela não ultrapassar o limite estipulado.

Normalmente, os algoritmos de indução de regras ou árvores possuem um *bias* favorável à descoberta de modelos mais genéricos, ou seja, favorável à descoberta de grandes disjuntos. Nesse caso, os pequenos disjuntos são normalmente removidos dos modelos para garantir generalidade. Entretanto, um pequeno disjunto pode representar um padrão de comportamento específico dos dados ou simplesmente um ruído. No primeiro caso, o disjunto deve ser mantido, podendo representar algum aspecto do conjunto de dados que pode ser de interesse do usuário. No segundo caso, o disjunto representa um erro dos dados e deve ser descartado. Descobrir a qual desses casos pertence um pequeno disjunto é extremamente complicado, baseando-se somente nos dados (Freitas 1998a). Portanto, os pequenos disjuntos devem ser tratados de maneira mais cuidadosa, evitando assim a eliminação de um conhecimento potencialmente interessante.

Em (Freitas 1998b) é proposto que um pequeno disjunto é surpreendente se ele prediz uma classe diferente das classes preditas pelas suas menores generalizações. As menores generalizações de um pequeno disjunto são definidas como segue.

Um disjunto é composto por conjunções de m condições, na forma $cond_1\ AND\ cond_2\ AND\ ...\ AND\ cond_m$, em que cada elemento $cond_k$, $k = 1, ..., m$, possui a forma $< Atributo\ op\ Valor >$, sendo op um elemento do conjunto $\{<, >, \leq, =\}$.

Um disjunto possui um total de m menores generalizações, em que cada generalização k é relacionada à condição $cond_k$ da regra. No caso do elemento $Atributo$ do termo $cond_k$ ser um atributo discreto, a k-ésima generalização é formada pela remoção da $cond_k$ da regra. Caso contrário, quando o elemento $Atributo$ de $cond_k$ possui valor contínuo, a k-ésima generalização pode ser definida de duas formas: (a) removendo a condição $cond_k$, ou (b) adicionando um pequeno valor α ao elemento $Valor$ de $cond_k$ se op é um elemento do conjunto $\{<, \leq\}$ ou removendo-se α de $Valor$ se op é um elemento do conjunto $\{>, \geq\}$. Em todos os casos, o número de exemplos cobertos por cada generalização de um disjunto é maior ou igual ao número de exemplos cobertos pelo disjunto original. Como resultado, a distribuição do atributo meta em cada generalização pode ser bastante diferente da distribuição do atributo meta no conjunto de exemplos coberto pelo pequeno disjunto original.

A medida de surpresa dos pequenos disjuntos é calculada como segue. Seja C a classe predita pelo pequeno disjunto original, C_k a classe predita pela k-ésima menor generalização do disjunto, e n o número de menores generalizações em que $C \neq C_k$, $k = 1, ..., m$. Então, a medida objetiva de surpresa do pequeno disjunto é definida pela Equação $SurpDisj =$

n/m. Como n é um número inteiro no intervalo $0..m$, essa medida de surpresa possui seus valores no intervalo $[0..1]$. Quanto maior o valor de $SurpDisj$, maior será a surpresa do pequeno disjunto original (Freitas 1998b).

Surpresa dos Atributos Individuais das Regras

Normalmente, as medidas de avaliação consideram o antecedente das regras de forma única, não se importando com os atributos presentes nas condições das regras. Entretanto, duas regras com um mesmo valor de alguma medida de avaliação podem possuir interessabilidade diferente para um determinado usuário, dependendo dos atributos que aparecem no antecedente das regras (Freitas 1998b).

Um atributo com alto valor do ganho de informação é um bom preditor para a classe quando considerado individualmente. Porém, devido a esse fato, é possível que já seja de conhecimento do usuário que esse atributo seja um bom preditor. Assim, o usuário pode ter uma tendência a se surpreender mais se a regra possui atributos com valores de ganho de informação baixos, isto é, regras que possuem atributos que individualmente não são bons preditores para a classe. A surpresa advém do fato de que, apesar dos atributos individualmente não serem bons preditores, sua combinação pode ter um alto valor preditivo. Isso pode ser bastante interessante ou surpreendente para o usuário.

O cálculo dessa medida de surpresa de atributos é feito como segue. Seja A_i o i-ésimo atributo da parte condicional da regra a ser analisada, A_{ik} o k-ésimo valor do atributo A_i, C o atributo meta da classificação, C_j o j-ésimo valor do atributo meta C, $P(X)$ representa a probabilidade de X e $P(X|Y)$ representa a probabilidade condicional de X dado Y. Nas Equações 3.1 e 3.2, m representa o número de valores possíveis do atributo A_i e n representa o número de possíveis valores do atributo meta. O valor do ganho de informação de um atributo A_i é calculado pela Equação 3.3.

A medida de surpresa dos atributos individuais das regras é calculada pela Equação 3.4. Ao analisar a Equação 3.4, observa-se que quanto menor a média do valor da medida de ganho de informação dos atributos que compõem o antecedente da regra, maior será o valor de $SurpAtr$ e mais surpreendente para o usuário será a regra, segundo este critério.

$$Info(C) = \sum_{j=1}^{n} P(C_j)\, log P(C_j) \tag{3.1}$$

$$Info(C|A_i) \sum_{k=1}^{m} P(A_{ik})(-\sum_{j=1}^{n} P(C_j|A_{ik})\, log P(C_j|A_{ik})) \tag{3.2}$$

$$GanhoInfo(A_i) = Info(C) - Info(C|A_i) \tag{3.3}$$

$$SurpAtr = 1/(\sum_{i=1}^{\#atr} GanhoInfo(A_i)/\#atr) \tag{3.4}$$

3.3.2 Seleção de Regras não Redundantes

Como apresentado anteriormente, um problema dos algoritmos de extração de padrões é a grande quantidade de padrões descobertos. Diversos trabalhos têm mostrado que muitos desses padrões são redundantes, já que muitos dos padrões podem ser derivados de outros (Padmanabhan & Tuzhilin 2000).

Não é uma boa técnica simplesmente fornecer um subconjunto arbitrário das regras descobertas para o usuário quando o número de regras geradas é muito grande. Isso advém do fato de que, dessa maneira, muitas regras possivelmente interessantes poderão ser descartadas. Assim, técnicas mais elaboradas para poda dessas regras redundantes têm sido propostas no sentido de reduzir o conjunto de regras e fornecer ao usuário somente aquelas realmente significativas.

Uma dessas técnicas para poda de regras pode ser encontrada em (Liu, Hsu, & Ma 1999). Esse trabalho é voltado ao tratamento de regras de associação, pois normalmente algoritmos para extração de regras de associação geram grandes conjuntos com muitas regras redundantes.

Essa técnica é composta por duas etapas. A primeira executa uma poda nas regras de associação descobertas para remover os padrões irrelevantes. Entretanto, o conjunto de regras resultantes dessa primeira etapa ainda pode ser grande, sendo necessária a execução de uma segunda etapa da técnica para uma redução ainda maior no número de regras. Nessa segunda etapa, esse conjunto é analisado com o intuito de encontrar um subconjunto especial de regras, formando um sumário das regras encontradas, chamado conjunto *DS* (*Direction Setting*) (Liu, Hsu, Ma, & Chen 1999; Liu, Hsu, & Ma 1999).

O conjunto de regras *DS* fornece um sumário do comportamento das regras de associação descobertas, representando os relacionamentos principais do domínio. O conjunto das regras que não pertencem ao conjunto *DS*, as regras não-*DS*, simplesmente fornecem detalhes adicionais que podem não ser importantes para o usuário.

Diversos experimentos mostraram que esse conjunto de regras *DS* é normalmente um conjunto bastante reduzido, permitindo que humanos possam analisá-lo manualmente e

assim focar seu interesse nas regras interessantes, de acordo com seu conhecimento (Ma, Liu, & Wong 2000). È importante ressaltar que essa técnica não identifica os padrões interessantes, ela apenas gera um conjunto reduzido de regras não redundantes e sumarizadas que pode ser mais facilmente analisado pelo usuário.

Em (Ma, Liu, & Wong 2000) é apresentada uma ferramenta que faz a utilização da sumarização de regras de associação em regras *DS* para auxiliar os usuários na análise de conjuntos de regras de associação. Esta ferramenta foi chamada de DS-WEB. Na análise de um conjunto de regras, a ferramenta faz a seleção das regras não redundantes e as organiza em uma estrutura hierárquica visando possibilitar ao usuário a navegação pelo conjunto de regras. Esta estrutura é então publicada por um conjunto de páginas WWW para facilitar o acesso ao conjunto de regras e o trabalho colaborativo entre os diversos usuários do ambiente.

3.3.3 Medidas Derivadas da Tabela de Contingência

Normalmente, em tarefas de classificação, a principal medida utilizada para avaliação do conhecimento é a precisão. Entretanto, quando o número de regras encontradas é muito grande, novas medidas para avaliação desse conhecimento são necessárias. Essas medidas têm como objetivo auxiliar a filtragem e ordenação desse grande conjunto de regras.

Em (Lavrac, Flach, & Zupan 1999) é apresentado um conjunto de medidas para avaliação de conhecimento. Essas medidas de avaliação não estão restritas ao conhecimento extraído por algoritmos de classificação, podendo ser aplicadas a qualquer conjunto de regras no formato $\underbrace{if < complexo >}_{Body\ ou\ B} then \underbrace{< complexo >}_{Head\ ou\ H}$, ou mais simplesmente $B \rightarrow H$, em que H pode possuir condições sobre mais que um atributo do conjunto de dados. Além disso, essas medidas são geradas a partir da tabela de contingência, representada na Tabela 3.1.

A definição dos elementos da tabela de contingência de uma regra R a partir de um conjunto de n exemplos é dada como segue. Na Tabela 3.1, b corresponde ao número de exemplos em que o corpo da regra B é verdadeiro e $\overline{b}$ o número de exemplos em que $\overline{B}$ é verdadeiro, isto é, em que B é falso. Da mesma maneira, h e $\overline{h}$ correspondem ao número de exemplos em que a cabeça da regra H é verdadeira e falsa, respectivamente. bh representa o número de exemplos em que ambos B e H são verdadeiros. Similarmente, são definidos $\overline{b}h$, $b\overline{h}$ e $\overline{b}\overline{h}$ (Prati, Baranauskas, & Monard 2001a).

A tabela de contingência também pode ser representada utilizando freqüências relativas, em que os valores contidos na tabela são divididos pelo número total de exemplos, obtendo as freqüências relativas $f_x = \frac{x}{n}$. Para o cálculo das medidas apresentadas em (Lavrac, Fla-

ch. & Zupan 1999), a freqüência relativa do evento x é utilizada como uma estimativa para a probabilidade $P(x)$. A tabela de contingência com freqüências relativas está representada na Tabela 3.2.

	H	$\overline{H}$	
B	bh	$b\overline{h}$	b
$\overline{B}$	$\overline{b}h$	$\overline{b}\overline{h}$	$\overline{b}$
	h	$\overline{h}$	n

Tabela 3.1: Tabela de Contingência

	H	$\overline{H}$	
B	f_{bh}	$f_{b\overline{h}}$	f_b
$\overline{B}$	$f_{\overline{b}h}$	$f_{\overline{b}\overline{h}}$	$f_{\overline{b}}$
	f_h	$f_{\overline{h}}$	1

Tabela 3.2: Tabela de Contingência com Freqüências Relativas

Algumas medidas derivadas da tabela de contingência são (Lavrac, Flach, & Zupan 1999; Prati, Baranauskas, & Monard 2001a):

Precisão Medida que indica o grau de confiabilidade da regra. Quanto maior seu valor, mais precisa será a regra.

$$Acc = P(H|B) = \frac{P(BH)}{P(B)} = \frac{f_{bh}}{f_b} \tag{3.5}$$

Erro Medida definida como $1 - Acc$. Quanto maior seu valor, menos precisa será a regra.

$$Err = P(\overline{H}|B) = \frac{f_{b\overline{h}}}{f_b} \tag{3.6}$$

Confiança Negativa Medida correspondente à precisão, mas para os exemplos que não são cobertos pela regra.

$$NegRel = P(\overline{H} \mid \overline{B}) = \frac{P(\overline{B}\overline{H})}{P(\overline{B})} = \frac{f_{\overline{b}\overline{h}}}{f_{\overline{b}}} \tag{3.7}$$

Sensitividade Medida que corresponde ao número relativo de exemplos cobertos pela regra em que a condição H é verdade.

$$Sens = P(B \mid H) = \frac{P(BH)}{P(H)} = \frac{f_{bh}}{f_h} \tag{3.8}$$

Especificidade Medida correspondente à Sensitividade, mas para os exemplos que não são cobertos pela regra.

$$Spec = P(\overline{B} \mid \overline{H}) = \frac{P(\overline{BH})}{P(\overline{H})} = \frac{f_{\overline{bh}}}{f_{\overline{h}}} \tag{3.9}$$

Cobertura Medida do número relativo de exemplos cobertos pela regra.

$$Cov = P(B) = f_b \tag{3.10}$$

Suporte Medida do número relativo de exemplos cobertos corretamente pela regra.

$$Sup = P(BH) = f_{bh} \tag{3.11}$$

Novidade Medida que quantifica a correlação estatística entre B e H. Pode-se demonstrar que $-0,25 \leq Nov \leq 0,25$. Quanto maior o valor de Nov, maior é a correlação entre B e H. Quanto menor seu valor, maior é a correlação entre B e $\overline{H}$. Se $Nov = 0$, então B e H são variáveis estatisticamente independentes.

$$Nov = P(BH) - P(H)P(B) = f_{bh} - f_h f_b \tag{3.12}$$

Satisfação Medida que indica o aumento relativo na precisão entre a regra $B \rightarrow verdade$ e a regra $B \rightarrow H$.

$$Sat = \frac{P(\overline{H}) - P(\overline{H} \mid B)}{P(\overline{H})} = \frac{f_{\overline{h}} - \frac{f_{b\overline{h}}}{f_b}}{f_{\overline{h}}} \tag{3.13}$$

Precisão Relativa Medida relativa que indica o ganho de precisão em relação à regra $verdade \rightarrow H$ pela consideração da condição B.

$$RAcc = P(H|B) - P(H) = \frac{f_{bh}}{f_b} - f_h \tag{3.14}$$

Confiança Negativa Relativa Medida correspondente à Precisão Relativa para os exemplos que não são cobertos pela regra.

$$RNegRel = P(\overline{H}|\overline{B}) - P(\overline{H}) = \frac{f_{\overline{bh}}}{f_{\overline{b}}} - f_{\overline{h}} \tag{3.15}$$

Sensitividade Relativa Medida que mede o ganho de sensitividade obtido em relação à sensitividade de uma regra $B \rightarrow verdade$.

$$RSens = P(B|H) - P(B) = \frac{f_{bh}}{f_h} - f_b \tag{3.16}$$

Especificidade Relativa Medida correspondente à Sensitividade Relativa para os exemplos que não são cobertos pela regra.

$$RSpec = P(\overline{B}\,\overline{H}) - P(B) = \frac{f_{\overline{bh}}}{f_{\overline{h}}} - f_{\overline{b}} \tag{3.17}$$

Com objetivo de garantir um balanceamento entre generalidade e relatividade para as regra, foi proposta em (Lavrac, Flach, & Zupan 1999) a atribuição de um peso para as medidas relativas, obtendo-se as medidas relativas ponderadas.

Precisão Relativa Ponderada

$$WRAcc = P(B)(P(H|B) - P(H)) = f_b\Big(\frac{f_{bh}}{f_b} - f_h\Big) \tag{3.18}$$

Confiança Negativa Relativa Ponderada

$$WRNegRel = P(\overline{B})(P(\overline{H}|\overline{B}) - P(\overline{H})) = f_{\overline{b}}\Big(\frac{f_{\overline{bh}}}{f_b} - f_{\overline{h}}\Big) \tag{3.19}$$

Sensitividade Relativa Ponderada

$$WRSens = P(H)(P(B|H) - P(B)) = f_h\Big(\frac{f_{bh}}{f_h} - f_b\Big) \tag{3.20}$$

Especificidade Relativa Ponderada

$$WRSpec = P(\overline{H})(P(\overline{B}\,\overline{H}) - P(\overline{B})) = f_{\overline{h}}\Big(\frac{f_{\overline{bh}}}{f_{\overline{h}}} - f_{\overline{b}}\Big) \tag{3.21}$$

Um aspecto importante sobre as medidas relativas (Equações 3.14, 3.15, 3.16 e 3.17) é que elas trazem mais informação sobre a utilidade de uma regra do que suas respectivas medidas absolutas (Equações 3.5, 3.7, 3.8 e 3.9). Por exemplo, se a precisão de uma regra é menor que a freqüência relativa da classe que ela prediz, então a regra não tem um desempenho muito bom, independente de sua precisão absoluta.

No entanto, é fácil obter uma alta precisão relativa para regras muito específicas. Para solucionar esse problema, são propostas as medidas relativas ponderadas (Equações 3.18, 3.19, 3.20 e 3.21). Em (Lavrac, Flach, & Zupan 1999) é demonstrado que as medidas relativas ponderadas e a medida Novidade (Equação 3.12) são equivalentes.

Uma biblioteca de ferramentas para cálculo das informações referentes à tabela de contingência de cada regra de um conjunto de regras de classificação foi proposta e implementada por Prati, Baranauskas, & Monard (2001a). A partir dessas informações, o cálculo de muitas medidas de avaliação, incluindo as propostas em (Lavrac, Flach, & Zupan 1999), pode ser facilmente realizado.

3.3.4 Medida *RI (Rule-Interest)*

A medida RI (*Rule-Interest*), proposta em (Piatetsky-Shapiro 1991), é usada para quantificar a correlação entre a cabeça e o corpo de regras do tipo $B \cdot H$.

A medida RI é dada pela Equação 3.22. Nesta equação, n é o número total de exemplos, $|B|$ e $|H|$ representam o número de exemplos que satisfazem as condições B e H, respectivamente, e $|B \cap H|$ representa o número de exemplos em que B e H são satisfeitos. O valor de $\frac{|B||H|}{n}$ corresponde ao número de exemplos esperado se B e H fossem variáveis estatisticamente independentes. O valor desta medida também pode ser obtido utilizando-se os valores da tabela de contingência, como pode ser observado na Equação 3.22.

$$RI = |B \cap H| - \frac{|B||H|}{n} = P(BH)n - \frac{P(B)nP(H)n}{n} = n(P(BH) - P(B)P(H)) = n(f_{bh} - f_b f_h) \quad (3.22)$$

Se $RI = 0$, então H e B são independentes e a regra não é considerada interessante. Quando $RI > 0$, H é positivamente correlacionado com B, ou seja, a probabilidade de H ocorrer aumenta quando a condição B é considerada. Quando $RI < 0$, H é negativamente correlacionado com B, ou seja, a probabilidade de H ocorrer diminui quando a condição B é considerada. Quanto maior o valor absoluto de RI, mais interessante será a regra.

É importante observar que a medida Novidade, descrita na Seção 3.3.3, é equivalente à medida RI.

3.4 Aspectos Subjetivos para Avaliação de Regras

Medidas objetivas não consideram fatores específicos do domínio e do usuário e, dessa maneira, muitas regras que possivelmente não são de interesse para o usuário ainda são

apresentadas pela utilização desses tipos de medidas na filtragem de regras. Apesar de úteis, essas medidas objetivas não são suficientes para solucionar o problema da interessabilidade de regras do processo de Extração de Conhecimento de Bases de Dados.

Medidas subjetivas de interessabilidade têm sido desenvolvidas com o objetivo de selecionar entre o conjunto de regras aquelas que são realmente interessantes para o usuário (Silberschatz & Tuzhilin 1996; Klemettinen, Mannila, & Toivonen 1996; Klemettinen, Mannila, Ronkainen, Toivonen, & Verkamo 1994; Piatetsky-Shapiro & Matheus 1994; Piatetsky-Shapiro 1991). Essas medidas subjetivas não consideram somente a estrutura de uma regra ou os dados utilizados no processo de descoberta, mas também o conhecimento que o usuário possui e seu interesse no momento da análise dos padrões.

Nesta seção são apresentadas algumas medidas subjetivas para seleção e ordenação de regras interessantes.

3.4.1 Sistema de Crenças

Em (Silberschatz & Tuzhilin 1996; Silberschatz & Tuzhilin 1995) é proposto o uso de um sistema de crenças (*beliefs*) para identificar o grau de interessabilidade de um conhecimento. Crenças são definidas como fórmulas de predicados arbitrários expressos em lógica de primeira ordem. Nos artigos, são definidos dois tipos de crenças: *Hard* e *Soft*.

Crenças *Soft* representam conceitos que o usuário acredita serem verdadeiros com algum grau de certeza. Caso o usuário encontre novos padrões que prevêem uma nova evidência, crenças desse tipo podem ser alteradas ou podem ter seus graus de certeza alterados. O grau de certeza/confiança do usuário em seu conceito é chamado de grau de confidência da crença *Soft*.

Já as crenças *Hard* representam conhecimentos rígidos e fixos sobre os quais o usuário não possui dúvidas. Essas crenças não podem ser alteradas, mesmo que seja encontrada uma nova evidência que a contrarie. Se uma evidência contradiz uma crença do tipo *Hard*, deve ter ocorrido algum erro no processo de coleta dos dados que produziram a evidência. Para uma crença *Hard* não é associado nenhum grau de confidência, pois ela representa um conhecimento fixo, rígido que não pode ser alterado. É importante ressaltar que as crenças *Soft* e *Hard* são subjetivas e podem variar de usuário para usuário.

Um padrão é considerado interessante em um sistema de crenças se ele afeta esse sistema. Além disso, quanto mais o padrão afeta o sistema, maior seu grau de interessabilidade. Nessa avaliação da interessabilidade são identificadas duas razões pelas quais um padrão pode ser interessante para o usuário:

Inesperabilidade Se um padrão é inesperado para o usuário, então ele é interessante. Os padrões inesperados contradizem a expectativa ou crença do usuário.

Utilidade Um padrão é interessante para o usuário se ele pode obter algum benefício ao utilizá-lo.

Segundo Silberschatz & Tuzhilin (1996), esses dois tipos de interessabilidade são independentes entre si. Apesar disso, é discutido que a maioria dos padrões são inesperados e úteis ao mesmo tempo e que, além disso, o conjunto dos padrões inesperados é uma boa aproximação do conjunto dos úteis e vice-versa. Somando-se a isso o fato de que a medida de utilidade de um padrão não pode ser facilmente calculada, foi proposta a utilização da medida de inesperabilidade de um padrão como forma de prever sua interessabilidade.

Uma das maneiras apontadas para a identificação do grau de Inesperabilidade de um padrão é pela utilização da teoria Bayesiana. Essa abordagem assume que se pode acreditar em algo de maneira parcial, ou seja, é associado um grau, medida ou fator de confidência à crença, como especificado anteriormente. Esse fator de confidência da crença é associado à probabilidade condicional que a crença possui dada uma evidência prévia (o conhecimento prévio do usuário). Dessa maneira, pode ser calculado o grau em que uma crença é afetada dado um novo padrão descoberto a partir dos dados.

Ao se descobrir um padrão e realizar a avaliação de quanto ele afeta o sistema de crenças, algumas ações podem ser realizadas com relação às crenças e aos dados de onde o padrão foi extraído.

No caso de uma crença *Hard*, se um padrão a contradiz, então esse padrão sempre é interessante para o usuário. Porém, essa contradição não implica que alterações devem ser feitas na crença. Isso indica que deve existir algum erro no processo de coleta do conjunto de dados do qual o padrão foi obtido, devendo-se então checar tal processo para tentar identificar e corrigir a falha.

Caso um padrão contrarie uma crença *Soft*, os dados devem ser checados. Nesse caso, a realização de alguma ação com relação à crença depende do resultado da análise dos dados.

3.4.2 Modelos de Regras

Modelos de Regras (*Rule Templates*) foram propostos por Klemettinen, Mannila, Ronkainen, Toivonen, & Verkamo (1994) como uma abordagem subjetiva para seleção de regras interessantes para o usuário. Modelos de Regras também foram abordados nos trabalhos

(Klemettinen, Mannila, & Toivonen 1996; Klemettinen, Mannila, & Toivonen 1997). Esses trabalhos foram voltados para a seleção de regras de associação devido ao fato de que algoritmos para geração dessas regras normalmente extraem um conjunto muito grande de regras. Apesar disso, esses modelos também podem ser aplicados a regras de classificação.

Nesses trabalhos é proposto que parte do problema de se fornecer regras interessantes pode ser resolvido pela possibilidade do usuário fornecer classes de regras interessantes e não interessantes, que são especificadas através dos Modelos de Regras. Esses modelos especificam quais atributos devem e quais não devem aparecer no antecedente e conseqüente da regra, de forma a selecionar possíveis regras interessantes e excluir as que não interessam ao usuário.

Um Modelo de Regra é uma expressão na forma $A_1, ..., A_k \rightarrow A_{k+1}, ..., A_l$, em que cada A_i representa o nome de um atributo ou o nome de uma classe de atributos, ou mesmo uma expressão C^+ ou C^*. Nas expressões C^+ e C^*, C representa o nome de uma classe de atributos, e $^+$ ou * correspondem a um ou mais, zero ou mais, respectivamente, instanciações de elementos da classe C. Uma regra $B_1, ..., B_m \rightarrow B_{m+1}, ..., B_n$ "casa" com um Modelo de Regra A se ela pode ser considerada uma instanciação de A.

Modelos de Regras podem ser seletivos ou restritivos. Em um ambiente que utiliza essa abordagem para seleção de regras, o usuário pode fornecer um conjunto de modelos seletivos e um conjunto de modelos restritivos. No conjunto de modelos seletivos, o usuário deve especificar o seu foco de interesse, isto é, os tipos de regras nas quais possui interesse. No conjunto restritivo, o usuário deve especificar tipos de regras nas quais ele não possui qualquer interesse. Para que uma regra seja considerada interessante e apresentada ao usuário, ela deve "casar" com pelo menos um dos modelos do conjunto seletivo e não "casar" com nenhum dos modelos do conjunto restritivo.

3.4.3 Aplicativo AIAS

Em (Liu, Hsu, Chen, & Ma 2000; Liu, Hsu, Ma, & Chen 1999) é apresentado o aplicativo *Association Interestingness Analysis System* (AIAS) que visa auxiliar os usuários na identificação das regras interessantes a partir de um conjunto de regras de associação pela utilização de medidas subjetivas. De acordo com a técnica utilizada no aplicativo para o Pós-processamento das regras de associação, primeiramente, o AIAS solicita que o usuário especifique seu conhecimento sobre o domínio ou seu foco de interesse atual. O aplicativo então utiliza esse conhecimento para analisar o conjunto de regras de acordo com vários

critérios de interesse. Então, elas são apresentadas ao usuário por uma ferramenta de visualização que permite identificar visualmente as regras mais interessantes.

Desta maneira, os três componentes principais da ferramenta AIAS, bem como as possíveis linguagens de representação que podem ser utilizadas pelos usuários e os tipos de análises que podem ser feitas, são apresentados em mais detalhes a seguir.

A linguagem de especificação O AIAS possui uma linguagem de especificação simples que permite ao usuário expressar seu conhecimento em diferentes graus de precisão. Essas linguagens são: Impressões Gerais, Conhecimentos Imprecisos e Conhecimentos Precisos. Os primeiros dois tipos representam conhecimentos que o usuário supõe ser verdadeiro, enquanto que o último tipo representa seu conhecimento preciso sobre o domínio (Liu, Hsu, Chen, & Ma 2000). Estas linguagens de especificação são apresentadas a seguir.

Impressões Gerais (IG) As Impressões Gerais possibilitam aos especialistas informarem de maneira bastante simples conjuntos de atributos entre os quais acreditam que exista alguma relação. Uma IG é uma expressão na forma:

$IG(< S_1, S_2, ..., S_m >)$ *[suporte, confidência]*

em que:

- Cada S_i representa um item da base de dados, uma classe ou conjunto de itens, ou uma expressão na forma C^+ e C^*, em que C representa uma classe ou conjunto de itens e C^+ e C^* representam uma ou mais, zero ou mais, instanciações de algum elemento de C, respectivamente;
- Uma regra $a_1, ..., a_n \rightarrow b_1, ..., b_k$ está em conforme com uma IG se $a_1, ..., a_n, b_1, ..., b_k$ pode ser considerada uma instanciação de $< S_1, S_2, ..., S_m >$;
- O suporte e a confidência são opcionais. O usuário pode especificar suporte e confidência mínimos para que as regras sejam apresentadas no resultado.

Conhecimentos Imprecisos (CI) O conhecimento impreciso possibilita informar um conhecimento que o especialista supõe ser verdadeiro. Um CI é uma expressão na forma:

$CI(< S_1, S_2, ..., S_m \rightarrow V_1, V_2, ..., V_q >)$ *[suporte, confidência]*

em que:

- Cada S_i ou V_j são equivalentes aos elementos S_k de uma IG;

- Uma regra $a_1, ..., a_n \rightarrow b_1, ... b_k$ está em conforme com um CI se ela pode ser considerada uma instanciação do CI;
- O suporte e a confidência são opcionais, como na especificação das Impressões Gerais.

Conhecimentos Precisos (CP) Deve ser especificado um CP quando o usuário sabe da existência de uma associação na base de dados. CPs são expressos na forma:

$CP(< S_1, S_2, ..., S_m \rightarrow V_1, V_2, ..., V_g >)$ *[suporte, confidência]*

em que:

- Cada S_i ou V_j é um item da base de dados;
- Uma regra $a_1, ..., a_n \rightarrow b_1, ..., b_k$*[suporte, confidência]* é equivalente a um CP se a regra é equivalente a $S_1, S_2, ..., S_m \rightarrow V_1, V_2, ..., V_g$. Uma regra está em conforme ou é inesperada com relação a um CP dependendo do valor do suporte e da confidência;
- O suporte e a confidência são obrigatórios.

Analisando o conjunto de regras No processo de avaliação das regras, o aplicativo AIAS realiza as análises de Identificação de Conformidade, Conseqüente Inesperado, Antecedente Inesperado, e Antecedente e Conseqüente Inesperados, apresentadas a seguir (Liu, Hsu, Ma, & Chen 1999):

Identificação de Conformidade Uma regra está em conformidade com o conhecimento fornecido se o antecedente e o conseqüente da regra "casam" com um subconjunto do conhecimento fornecido. Este tipo de análise pode ser utilizado para verificar quais regras estão totalmente ou parcialmente em conformidade com o conhecimento fornecido pelo usuário.

Conseqüente Inesperado Neste tipo de análise, são apresentadas as regras em que o antecedente "casa" com um subconjunto do conhecimento fornecido pelo usuário, mas o conseqüente não. Nesse critério, são apresentadas as regras que são contrárias ao conhecimento do usuário.

Antecedente Inesperado São apresentadas as regras cujo conseqüente "casa" com algum subconjunto do conhecimento fornecido, mas o antecedente não. Nesse critério, é mostrado ao usuário condições não previstas por ele que podem levar a uma determinada condição no conseqüente da regra.

Antecedente e Conseqüente Inesperados Neste tipo de análise, são identificadas as regras em que ambos o antecedente e conseqüente não "casam" com nenhum subconjunto do conhecimento fornecido pelo usuário. Neste critério, são apresentadas regras cujo antecedente e conseqüente são desconhecidos pelo usuário.

Componente de Visualização Após os conjuntos de regras terem sido analisados, as regras são apresentadas em uma ferramenta de visualização que visa facilitar a identificação das regras mais interessantes ao usuário (Liu, Hsu, Ma, & Chen 1999).

Deve-se observar que a técnica utilizada na ferramenta AIAS é iterativa e interativa, de maneira que após ter visualizado o resultado da análise na ferramenta de visualização, é permitido ao usuário adicionar conhecimentos ou revisar o conhecimento fornecido previamente para então realizar uma nova análise.

3.5 Sumário das Medidas Apresentadas

Nesta seção, é apresentado um sumário das medidas de avaliação de regras apresentadas nas seções anteriores. Essas medida são classificadas de acordo com dois critérios, como pode ser observado na Tabela 3.3. A coluna *Problema* apresenta o tipo de problema tratado pelas regras utilizadas por cada medida e a coluna *Tipo* indica se a medida é objetiva ou subjetiva. A coluna *Referência* cita uma publicação relevante em que a medida é especificada.

Trabalho	Problema	Tipo	Referência
Surpresa dos Pequenos Disjuntos	Classificação	Objetiva	(Freitas 1998b)
Surpresa dos Atributos Individuais das Regras	Classificação	Objetiva	(Freitas 1998b)
Seleção de Regras não Redundantes	Associação	Objetiva	(Liu, Hsu, & Ma 1999)
Medidas Derivadas da Tabela de Contingência	Classificação Associação	Objetiva	(Lavrac, Flach, & Zupan 1999)
Medida *RI*	Classificação	Objetiva	(Piatetsky-Shapiro 1991)
Aplicativo AIAS	Associação	Subjetiva	(Liu, Hsu, Chen, & Ma 2000)
Sistema de Crenças	Classificação Associação	Subjetiva	(Silberschatz & Tuzhilin 1996)
Modelos de Regras	Associação	Subjetiva	(Klemettinen, Mannila, & Toivonen 1997)

Tabela 3.3: Classificação das Medidas de Avaliação de Conhecimento

Pode-se observar na Tabela 3.3 que tanto para os tipos de medidas objetivas quanto para as subjetivas existem medidas para tratamento de regras de classificação e de associação.

Entretanto, caso alguma medida adequada para somente um tipo de regra tenha alguma característica interessante, esta possivelmente pode ser modificada para adequar-se também para um outro tipo de regra. Pode ser observado também na tabela que não foram apresentadas medidas para tratamento de regras de regressão. Isto pode ser contornado pela adequação de alguma dessas medidas para permitir o tratamento deste tipo de regra (Pugliesi, Dosualdo, & Rezende 2003).

3.6 Considerações Finais

Em diversas atividades do processo de Extração de Conhecimento de Bases de Dados, a quantidade de padrões extraídos pode ser bastante grande. Esse volume pode sobrecarregar o usuário, dificultando a tarefa de identificação do conhecimento interessante. Dessa maneira, diversos trabalhos têm sido desenvolvidos no sentido de auxiliar o usuário na identificação e interpretação do conhecimento interessante.

A interessabilidade de um padrão se refere ao grau de interesse que o usuário possui sobre o mesmo. Na tentativa de determinar o grau de interesse de um padrão para o usuário, diversas medidas de avaliação de padrões foram apresentadas. Essas medidas podem ser objetivas, quando dependem somente dos dados e da estrutura do padrão, ou subjetivas, quando consideram também aspectos relacionados ao conhecimento do usuário para identificar os padrões interessantes. Para que se possa fazer uma avaliação de interessabilidade realmente válida, diversos tipos de medidas devem estar disponíveis para o usuário. A utilização conjunta de medidas objetivas e subjetivas pode permitir que o conjunto dos padrões a ser analisado manualmente pelo usuário seja bastante reduzido.

Assim, a utilização das medidas objetivas e subjetivas em um Ambiente para Exploração de Regras pode representar um excelente auxílio na identificação das regras interessantes. As medidas objetivas podem ser utilizadas para filtrar o conjunto inicial de regras a ser analisado e as medidas subjetivas podem ser utilizadas para permitir aos usuários focarem seu interesse em um pequeno conjunto de regras. Nesse sentido, no próximo capítulo é apresentado o Ambiente para Descoberta de Conhecimento Discover, alguns requisitos importantes de um Ambiente para Exploração de Regras, assim como é apresentado de maneira geral o Ambiente *RulEE* a ser integrado ao Discover.

Capítulo 4

Ambiente para Exploração de Regras

4.1 Considerações Iniciais

Por ser um ramo de pesquisa relativamente recente e complexo, existem poucas ferramentas que suportam o processo de Extração de Conhecimento de Bases de Dados como um todo, isto é, que automatizam o processo do início ao fim e que possuam uma variedade de técnicas disponíveis para utilização em cada uma de suas etapas. Esse fato é agravado pela utilização, por parte das ferramentas, de algoritmos proprietários, limitando ainda mais o trabalho dos pesquisadores. Dessa maneira, é comum aos pesquisadores desenvolverem aplicativos que os auxiliem na realização de seus experimentos científicos.

Dessa maneira, os pesquisadores do Laboratório de Inteligência Computacional (LABIC) do Instituto de Ciências Matemáticas e de Computação (ICMC) da USP/São Carlos que trabalham com Aprendizado de Máquina e Mineração de Dados sentiram-se motivados em desenvolver um ambiente que integre um conjunto de ferramentas para apoiar o processo de Extração de Conhecimento de Dados e Textos como um todo, abrangendo os diversos trabalhos que vêm sendo desenvolvidos no laboratório, a fim de satisfazer suas necessidades relativas às pesquisas referentes ao processo. Este ambiente foi intitulado Discover e tem como objetivo dar suporte às fases do processo de Extração de Conhecimento, desde o Pré-processamento dos dados até o Pós-processamento do Conhecimento.

Recentemente, tem surgido a necessidade do desenvolvimento de um ambiente que permita o Pós-processamento em um ambiente colaborativo para auxiliar os usuários do processo de Extração de Conhecimento na compreensão, identificação do conhecimento interessante, além de possibilitar o acesso remoto a esse conhecimento. Nesse sentido, este capítulo fornece uma breve descrição de um Ambiente para Exploração de Regras, focando

em seus principais requisitos, bem como apresenta o Ambiente Discover ao qual o Ambiente para Exploração de Regras deve estar integrado.

4.2 O Ambiente Discover

Com o objetivo de fornecer um ambiente integrado para apoiar as etapas do processo de Extração de Conhecimento de Dados e Textos, foi proposto o Ambiente Discover (Baranauskas & Batista 2000). Nesse ambiente são utilizados algoritmos de AM implementados pela comunidade, bem como módulos com finalidades específicas desenvolvidos pelos pesquisadores do LABIC. Entre as funcionalidades desses módulos estão: Pré-processamento de dados, Pré-processamento de textos e Pós-processamento de conhecimento.

Como ilustrado na Figura 4.1, o Ambiente Discover é capaz de acessar os dados na forma atributo-valor ou no formato texto, devendo conseqüentemente estar disponível um conjunto de métodos para Pré-processamento de dados e de textos. Caso a quantidade de dados exceda a capacidade dos algoritmos de extração de padrões, amostras representativas devem ser geradas para serem fornecidas aos algoritmos. Os resultados dos algoritmos de extração de padrões ou modelos podem ser avaliados através de diversas medidas, entre elas precisão, compreensibilidade e interessabilidade.

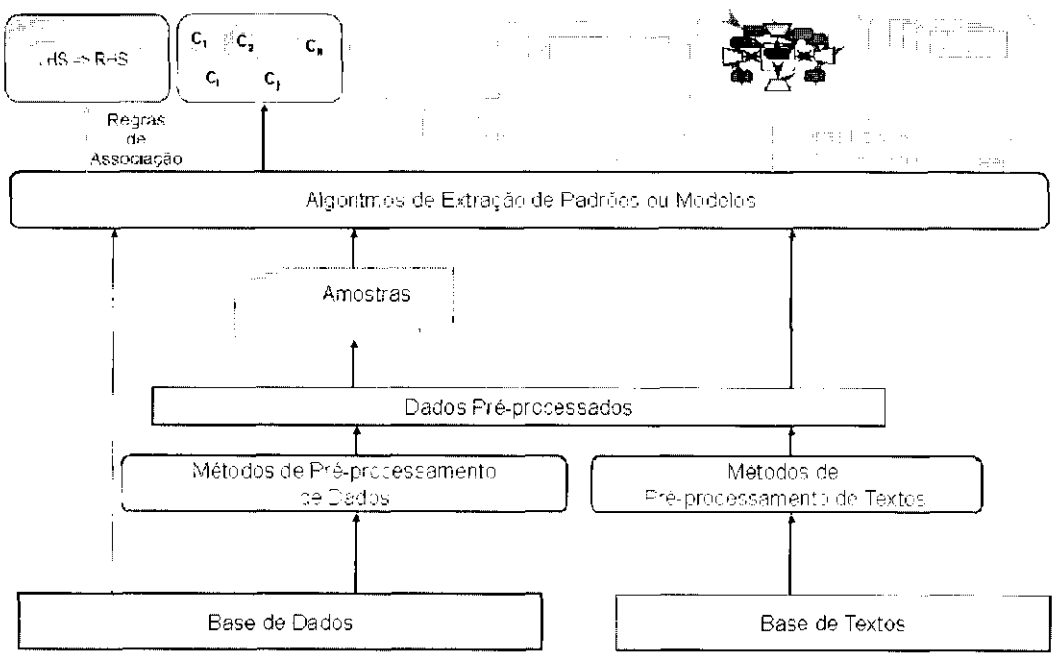

Figura 4.1: Estrutura do Ambiente Discover

Como apresentado anteriormente, o Ambiente Discover integra trabalhos dos diversos membros do LABIC. Isto é feito principalmente através da padronização do desenvolvimen-

to dos diversos trabalhos de forma que os componentes desenvolvidos por um pesquisador possam ser reutilizados por outros que necessitem de sua funcionalidade nas pesquisas ou experimentos. Assim, um ponto crucial é que o projeto do Ambiente Discover seja suficientemente rígido para garantir a integração dos trabalhos desenvolvidos, mas simultaneamente seja flexível para permitir que novas pesquisas, projetos e algoritmos que venham a ser desenvolvidos pelos membros do LABIC ou por outros pesquisadores do processo de Extração de Conhecimento também possam ser integrados ao ambiente.

A princípio, o Discover consistia apenas de um repositório de *scripts*, os quais seriam considerados apenas como pequenos programas para realizar tarefas atômicas. Entretanto, surgiu a proposta de se criar um ambiente integrado com uma interface gráfica bem definida que auxilie, visualmente, a composição do processo de Extração de Conhecimento (Geromini 2002). Desta maneira, a interface deve permitir que o pesquisador "monte" o processo de Extração de Conhecimento através de componentes visuais e da configuração dos atributos dos mesmos.

O Ambiente Discover pode ser entendido como um conjunto de métodos que são aplicados sobre dados e textos ou sobre o conhecimento extraído a partir dos mesmos. Dessa forma, é muito importante que o ambiente ofereça facilidades para manipular dados e conhecimento. Isto é obtido pela utilização de sintaxes padrão para a representação de dados e conhecimento, e por componentes que ofereçam um conjunto de funcionalidades básicas para manipulação de dados e conhecimento. Atualmente, existem definidas sintaxes padrão para representação de dados (Batista 2001; Prati, Baranauskas, & Monard 2001a), de regras de classificação (Baranauskas 2001; Prati, Baranauskas, & Monard 2001b), regras de regressão (Pugliesi, Dosualdo, & Rezende 2003) e regras de associação (Melanda & Rezende 2003), além de bibliotecas para manipulação e conversão dessas sintaxes padrão. Essas sintaxes padrão estão detalhadas respectivamente nas Seções 4.2.1, 4.2.2, 4.2.3 e 4.2.4.

Através de interfaces bem definidas entre os módulos e procedimentos para conversão de dados e de conhecimento para as sintaxes padrão, são utilizados no Discover os algoritmos já desenvolvidos, testados e consolidados pela comunidade científica, evitando assim sua reimplementação e garantindo maior confiabilidade aos experimentos realizados. O LABIC dispõe de diversos sistemas computacionais que implementam alguns dos métodos mais conhecidos e utilizados para extração de conhecimento, sendo necessária a criação das interfaces que permitem a utilização desses sistemas de forma combinada.

Dessa maneira, pela integração dos diversos trabalhos que vem sendo desenvolvidos e utilização dos mais diversos algoritmos, espera-se obter um ganho de produtividade dos pesquisadores através da redução da repetitividade em seus trabalhos e maior facilidade para realização de experimentos.

4.2.1 Sintaxe Padrão para Representação de Dados

A sintaxe padrão para representação de dados foi definida em (Batista 2001) como sendo uma extensão do formato utilizado como entrada do algoritmo C4.5. Esta extensão foi realizada para adequar o formato dos arquivos de dados para as necessidades do Ambiente Discover.

Um conjunto de exemplos na sintaxe padrão para representação de dados deve apresentar dois arquivos. Os nomes dos atributos do conjunto de dados, assim como seus respectivos domínios, devem estar especificados em um arquivo *.names*. Os valores dos atributos para os exemplos devem estar contidos em um arquivo *.data*. Esses dois arquivos devem possuir o mesmo nome, se diferenciando somente pela extensão.

Na Figura 4.2 é apresentado um exemplo de arquivo *.names*. Nesse arquivo, a primeira linha apresenta o nome do atributo correspondente ao atributo classe, caso ele exista. Na Figura 4.3 é apresentado um exemplo de arquivo *.data* correspondente ao arquivo *.names* apresentado na Figura 4.2.

```
loan_predict.

loan_amount: real.
loan_duration: integer.
loan_payments: real.
account_frequency: nominal(weekly,monthly,aftertransaction).
account_months_before_loan: integer.
client_age: real.
client_sex: nominal(M,F).
disposition_type: nominal(OWNER,DISPONENT).
credit_card_type: nominal(classic,junior,gold).
credit_card_months_before_loan: integer.
district_region: nominal(westBohemia,eastBohemia,...,northBohemia).
loan_predict: nominal(GOOD,BAD).
```

Figura 4.2: Exemplo de Arquivo *names* na Sintaxe Padrão

Uma biblioteca, chamada de *Dataset*, foi implementada para manipulação de conjunto de dados na sintaxe padrão. Esta biblioteca apresenta métodos para conversão de dados na sintaxe padrão para os formatos de entrada dos algoritmos C4.5, C4.5rules, C5.0, Cubist, CN2, Newid, RT, RETIS, CART, WEKA, SNNS e SVMTorch (Batista 2001; Pugliesi, Dosualdo, & Rezende 2003).

```
127060,60,2118,monthly,5,66,M,OWNER,?,?,eastBohemia,GOOD
87840,24,3660,monthly,4,17,M,DISPONENT,?,?,centralBohemia,GOOD
174744,24,7281,monthly,5,54,F,OWNER,?,?,westBohemia,BAD
117024,24,4876,monthly,5,67,M,OWNER,?,?,southBohemia,GOOD
21924,36,609,monthly,10,43,F,DISPONENT,?,?,southMoravia,GOOD
95400,36,2650,weekly,9,32,M,OWNER,?,?,centralBohemia,GOOD
86616,12,7218,monthly,4,41,M,OWNER,?,?,centralBohemia,GOOD
80952,24,3373,monthly,11,58,M,OWNER,?,?,Prague,GOOD
80952,24,3373,monthly,11,58,M,OWNER,?,?,Prague,GOOD
21072,24,878,monthly,4,38,F,OWNER,?,?,southBohemia,GOOD
108144,36,3004,monthly,8,46,M,DISPONENT,?,?,southMoravia,GOOD
197748,36,5493,monthly,5,32,F,OWNER,?,?,centralBohemia,GOOD
197748,36,5493,monthly,5,34,M,DISPONENT,?,?,centralBohemia,GOOD
300660,60,5011,monthly,10,53,M,OWNER,?,?,northBohemia,GOOD
```

Figura 4.3: Exemplo de Arquivo *data* na Sintaxe Padrão

4.2.2 Sintaxe Padrão para Representação de Regras de Classificação

No Ambiente Discover foi desenvolvida uma sintaxe padrão para representação de regras de classificação, bem como *scripts* para conversão de regras ou árvores de decisão extraídas por diversos algoritmos de Aprendizado de Máquina para esse formato padrão, além de *scripts* para cálculo da tabela de contingência para cada regra (Prati, Baranauskas, & Monard 2001b; Prati, Baranauskas, & Monard 2001a).

Na Figura 4.4 é apresentada uma pequena parte de um arquivo contendo regras de classificação no formato padrão. Na figura, os elementos entre colchetes correspondem aos valores da tabela de contingência de cada regra, apresentada na Seção 3.3.3, e referem-se respectivamente a f_{bh}, $f_{b\bar{h}}$, $f_{\bar{b}h}$, $f_{\bar{b}\bar{h}}$ e n. Quando existe algum valor desconhecido nos dados, os valores da tabela de contingência encontram-se após o caracter ?.

```
Standard Rules Conversor 1.2.1      Copyright (c) Ronaldo C. Prati
Inducer: c4.5     Input File: financial_data.rules
Date: Fri Nov  1 15:45:26 2002

Rules Evaluated as UNORDERED
Names File: financial_data.names    Data File: financial_data.test

R0001       IF disposition_type = DISPONENT
      THEN CLASS = GOOD [0.1993, 0.0000, 0.1014, 0.6993, 296]
                        ?[0.000, 0.000, 0.000, 0.000, 0]

R0002       IF disposition_type = OWNER
            AND loan_amount <= 244560
            AND account_months_before_loan <= 4
            AND account_frequency = weekly
      THEN CLASS = GOOD [0.0000, 0.0000, 0.1014, 0.8986, 296]
                        ?[0.000, 0.000, 0.000, 0.000, 0]
.
.
.
```

Figura 4.4: Exemplo de Arquivo de Regras de Classificação na Sintaxe Padrão

Existem implementados *scripts* para conversão da saída dos algoritmos ID3, C4.5, C4.5*rules*, C5.0/See5, CN2, OC1, *Ripper*, T2 e MC4 para o formato padrão de regras de classificação (Prati, Baranauskas, & Monard 2001b). Para o cálculo dos valores da tabela de contingência e geração dos arquivos no formato padrão estendido foram desenvolvidos *scripts* por Prati, Baranauskas, & Monard (2001a).

4.2.3 Sintaxe Padrão para Representação de Regras de Regressão

Foi desenvolvida também uma sintaxe padrão para representação de regras de regressão adequada para englobar o tipo de representação da hipótese induzida por vários algoritmos de regressão. Além da definição da sintaxe para regras de regressão, foram desenvolvidos *scripts* para conversão dos conjuntos de exemplos da sintaxe padrão do Discover para a sintaxe dos algoritmos M5, RT, Cubist, CART e RETIS, e conversão da saída desses algoritmos para a sintaxe padrão de regressão. Todo esse trabalho foi desenvolvido por Pugliesi, Dosualdo, & Rezende (2003).

Assim como nas sintaxes padrão para representação de regras de classificação e associação, é interessante que a sintaxe padrão para regras de regressão apresentasse a tabela de contingência para cada regra. Entretanto, regras de regressão não predizem um valor categórico e sim um valor numérico, dificultando o cálculo do número de exemplos falsos e verdadeiros quanto ao atributo meta, por este ser contínuo. Portanto, o cálculo dos valores relacionados à H e $\overline{H}$ não são triviais. Entretanto, foi definido por Pugliesi, Dosualdo, & Rezende (2003) um método para o cálculo da tabela de contingência para regras de regressão.

Na Figura 4.5 é apresentado um exemplo contendo regras de regressão escritas na sintaxe padrão juntamente com a freqüência relativa de cada um dos quatro valores da tabela de contingência e o número total de exemplos. Para cada regra, os elementos entre colchetes referem-se respectivamente a f_{bh}, $f_{b\overline{h}}$, $f_{\overline{b}h}$, $f_{\overline{b}\overline{h}}$, n.

4.2.4 Sintaxe Padrão para Representação de Regras de Associação

Conforme a padronização definida para classificação e regressão, também foi definida para o Discover uma sintaxe padrão para representação de regras de associação. Nesta sintaxe, para cada regra também estão presentes os valores da tabela de contingência. Uma regra de associação na sintaxe padrão obedece à formatação apresentada na Expressão 4.1.

$$itens\ do\ antecedente, itens\ do\ consequente, [f_{bh}, f_{b\overline{h}}, f_{\overline{b}h}, f_{\overline{b}\overline{h}}, n] \tag{4.1}$$

```
Standard Regression Rules Conversor Copyright (c) Daniel Gomes Dosualdo
Inducer: rt                    Input File:  abalone.out
Date: Mon Aug 19 20:23:19 2002      Output File: abalone.rules


[R0001]       IF shell_weight <=  0.03
                   AND whole_weight <=  0.02
              THEN rings = 3.33 +/- 0.428
              [ 0.009584664, 0.30919417, 0.009939652, 0.671281505, 2817 ]

[R0002]       IF shell_weight <=  0.02
                   AND whole_weight >  0.02
              THEN rings = 4.35 +/- 0.231
.
.
.
```

Figura 4.5: Exemplo de Arquivo de Regras de Regressão na Sintaxe Padrão

Um exemplo de um conjunto de regras de associação na sintaxe padrão é fornecido na Figura 4.6. Têm sido desenvolvidos *scripts* para conversão dos dados na sintaxe padrão de exemplos para o formato de entrada dos algoritmos de extração de regras de associação Apriori, MagnumOpus e do *software* Mineset, além de *scripts* para conversão da saída desses algoritmos para a sintaxe padrão de associação.

```
agua_p=(89.776...90.628],PEROCPPR=(67.09...69.8643],[0.017,0,0.875,0.108,27746]
agua_p=(89.776...90.628],DENSCONS=(269120...296722],[0.017,0,0.875,0.108,27746]
agua_p=(89.776...90.628],dpi_p=(...0.108933],[0.017,0,0.233,0.75,27746]
agua_p=(89.776...90.628],porpdimp=(...0.112647],[0.017,0,0.208,0.775,27746]
SINSTSAN=(3...4],PEROCPPR=(...50.4443],[0.017,0,0.95,0.033,27746]
SINSTSAN=(3...4],QV=(0.86529...],[0.017,0,0.95,0.033,27746]
SINSTSAN=(3...4],PORPOPAL=(...74.2307],[0.017,0,0.933,0.0500000000000001,27746]
SINSTSAN=(3...4],RENDCHEF=(104668...132103],[0.017,0,0.8,0.183,27746]
SINSTSAN=(3...4],PESS_DOM=(3...],[0.017,0,0.683,0.3,27746]
.
.
.
```

Figura 4.6: Exemplo de Arquivo de Regras de Associação na Sintaxe Padrão

4.3 Requisitos de um Ambiente para Exploração de Regras

Como já mencionado, muita pesquisa tem sido feita no sentido de desenvolver medidas para avaliação de regras, e muitos protótipos têm sido desenvolvidos. No entanto, percebeu-se que não é possível definir uma melhor medida de avaliação de conhecimento em todos os domínios (Freitas 1998a; Freitas 1998b). Somando-se a isso o fato de que todas essas medidas e metodologias normalmente são desenvolvidas independentemente, não é possível a utilização de um único sistema para avaliação de regras no qual diversos tipos de análises possam ser feitos.

Além disso, apesar do processo de Extração de Conhecimento ter sido definido como "processo de identificação de padrões nos dados que sejam válidos, novos, potencialmente

úteis e compreensíveis a seres humanos", muitos trabalhos têm tratado a validade do conhecimento, mas poucos têm tratado a novidade, utilidade e compreensibilidade (Pazzani 2000). Em (Pazzani, Mani, & Shankle 1997) é mostrado que o conhecimento descoberto por algoritmos de extração de regras de classificação pode não fazer sentido para os especialistas do domínio. Um conhecimento não compreensível, possivelmente, também não será útil ao usuário em um processo de tomada de decisão.

Nesse contexto, o desenvolvimento de um Ambiente para Exploração de Regras que auxilie os especialistas do domínio e usuários finais na compreensão e identificação do conhecimento interessante é de grande relevância para o sucesso do processo de Extração de Conhecimento. Além disso, o ambiente pode possibilitar que novas pesquisas relacionadas aos métodos para avaliação de regras e ao processo de MD como um todo possam ser feitas. O desenvolvimento de um Ambiente para Exploração de Regras vem suprir essas deficiências por disponibilizar o conhecimento para facilitar o acesso aos usuários, além de agregar métodos para avaliar as regras quanto a precisão, compreensibilidade ou interessabilidade, por exemplo.

Um aspecto importante do processo de Extração de Conhecimento é que normalmente as regras extraídas se encontram em arquivos texto, o que dificulta sua análise. Pela utilização de ferramentas comerciais mais completas, algumas funcionalidades adicionais podem estar disponíveis, como por exemplo ferramentas $3D$ para visualização dos modelos. Entretanto, existem poucos métodos implementados para auxiliar na análise, identificação e utilização do conhecimento interessante.

A implementação de um Ambiente para Exploração de Regras implica no desenvolvimento de uma arquitetura para facilitar o cálculo e armazenamento do valor de medidas de avaliação de regras, além de facilitar a incorporação de novos métodos para cálculo de novas medidas. É interessante que cada conjunto de regras disponibilizado apresente um conjunto de medidas tanto objetivas quanto subjetivas armazenadas visando o apoio ao usuário em sua análise. Entretanto, com o acúmulo desses dados sobre avaliação, uma nova base de dados valiosa para os pesquisadores do processo de Extração de Conhecimento também fica disponível. A partir desses dados, análises podem ser feitas sobre as medidas ou métodos de avaliação visando, por exemplo, identificar medidas mais efetivas para avaliação, relacionamentos entre as mesmas, entre outras. Somando-se a esses dados o *feedback* fornecido pelos usuários durante a consulta às regras disponibilizadas, algumas análises bastante interessantes podem ser feitas. Dentre essas análises utilizando o *feedback* do usuário, destaca-se a possibilidade da análise das funcionalidades mais

utilizadas com o objetivo de encontrar as medidas de avaliação mais efetivas e verificar a existência de alguma relação entre medidas objetivas e subjetivas.

Para auxiliar o usuário na compreensão e identificação do conhecimento interessante, algumas características desejáveis para um Ambiente para Exploração de Regras são:

Interatividade e Iteratividade Como citado anteriormente, muitos pesquisadores do processo de Extração de Conhecimento de Bases de Dados têm concentrado suas pesquisas na fase de descoberta do conhecimento. Entretanto, a inferência de padrões é somente uma pequena parte do problema de se obter conhecimento útil a partir de dados. Além disso, o processo de Extração de Conhecimento de Bases de Dados é descrito de maneira a enfatizar a importância da interação com o usuário e iteração entre suas fases, sendo que a principal iteração do processo é a repetição das fases de Pré-processamento e Extração de Padrões após a análise do conhecimento na etapa de Pós-processamento. A qualidade do conhecimento extraído pode ser melhorada em cada iteração ao reutilizar informações coletadas nas iterações anteriores (Klemettinen, Mannila, Ronkainen, Toivonen, & Verkamo 1994; Klemettinen, Mannila, & Toivonen 1996; Klemettinen, Mannila, & Toivonen 1997).

No entanto, a iteratividade e interatividade do processo de Extração de Conhecimento também podem ser realizadas na fase de disponibilização do conhecimento. A iteratividade dessa fase pode ser feita através da visualização do conjunto de regras, permitindo ao usuário escolher a cada momento um tipo de filtro ou medida para ordenação das regras e assim identificar as mais interessantes. Caso não seja obtido sucesso, outros filtros podem ser escolhidos. Desta maneira, um aspecto extremamente importante para um Ambiente para Exploração de Regras é permitir que o usuário possa explorar iterativamente e interativamente o conhecimento, especificando a cada instante o foco de interesse desejado.

Medidas de Avaliação Outro requisito importante para um Ambiente para Exploração de Regras é a disponibilização ao usuário de diversos métodos para poda e ordenação dos padrões. Além disso, tanto as medidas objetivas quanto as subjetivas devem estar disponíveis para seleção de regras. As medidas objetivas podem ser consideradas como um filtro inicial para seleção de regras, enquanto que medidas subjetivas podem ser utilizadas para focar o interesse do usuário em um conjunto de regras realmente interessante (Freitas 1998a).

Na disponibilização, as medidas de avaliação podem ser utilizadas para:

Poda do conjunto de regras O usuário pode escolher alguma medida de avaliação para selecionar um subconjunto das regras apresentadas. Para isso, ele pode definir algum limite de valor para a medida, sendo apresentadas somente aquelas regras em que o valor da medida é superior ou inferior ao limite estipulado, ou utilizar alguma medida que selecione somente regras interessantes segundo algum critério definido;

Ordenação O usuário pode escolher alguma medida para ordenar o conjunto de regras apresentado.

Visualização dos Modelos ou Regras Como em Mineração de Dados busca-se por padrões de comportamento dos dados que não são previamente conhecidos pelos usuários, a compreensão dos modelos encontrados pode ser difícil. Dessa maneira, a compreensão dos modelos é um tópico que precisa ser bastante pesquisado. Uma das maneiras utilizadas para auxiliar os usuários na compreensão e interpretação do conhecimento é através da utilização de ambientes em que os modelos possam ser visualizados de maneira adequada pelo usuário.

Diversas ferramentas de Mineração de Dados possuem módulos de visualização de modelos e de dados bastante completos e úteis. Entretanto, nessas ferramentas a visualização é realizada somente nas estações em que a ferramenta está instalada. Um aspecto adicional de um Ambiente para Exploração de Regras é a possibilidade de visualização do conhecimento através de uma interface baseada na WWW. Algumas vantagens obtidas pela visualização do conhecimento pela WWW são a independência de plataforma, facilidade de uso e, especialmente, a acessibilidade, permitindo a visualização do conhecimento pelos usuários do processo independente de sua localização física (Grinstein & Ward 2002; Ma, Liu, & Wong 2000).

Durante a visualização de um modelo, é importante que o usuário possa interagir com ele. Isso significa permitir que o usuário possa selecionar valores de atributos e ver os efeitos que isto pode causar. Esta interação provê a compreensão de causa-efeito, isto é, o usuário pode entender quanto cada atributo do conjunto de dados afeta a classificação final (Thearling, Becker, DeCoste, Mawby, Pilote, & Sommerfield 2002).

A visualização interativa dos modelos deve permitir que usuários com pouco conhecimento em Aprendizado de Máquina e Estatística sejam capazes de compreender

fatores que influenciam nas classificações e que eles também possam visualizar o modelo completo para compreenderem como os exemplos são tratados (Becker, Kohavi, & Sommerfield 2002).

Integração Uma outra forma de se auxiliar o usuário na compreensão do conhecimento é fornecer um contexto. Este contexto pode ser fornecido, por exemplo, ao mostrar a relação existente entre o conhecimento extraído e o conjunto de dados em que foi gerado. Isto é chamado de integração (Thearling, Becker, DeCoste, Mawby, Pilote, & Sommerfield 2002). Uma forma de integração é a possibilidade de se selecionar parte de um modelo e obter acesso aos dados originais a partir dos quais essa parte foi gerada.

Ambientes Colaborativos Outro requisito importante para um Ambiente para Exploração de Regras é que ele seja um ambiente colaborativo, propiciando a interação e integração entre os diversos usuários do ambiente (Foster & Gee 2002). No caso de um Ambiente para Exploração de Regras, o primeiro passo para fornecer um suporte eficiente a múltiplos usuários é armazenar as seções dos usuários em um sistema de *logs*.

O sistema de *logs* deve armazenar os trabalhos dos usuários, permitindo que eles possam repeti-los ou que outros usuários possam analisá-los (Foster & Gee 2002). Isto pode permitir um ganho geral de produtividade entre os usuários do ambiente, além de possibilitar que as informações contidas nos *logs* possam ser utilizadas pelos analistas do processo de Extração de Conhecimento para auxiliar na obtenção de melhores resultados em uma próxima iteração do processo.

A utilização da WWW como interface para disponibilização de regras pode facilitar também a implementação de um ambiente que permita o trabalho cooperativo dos usuários do processo de Extração de Conhecimento. Dessa maneira, os diversos usuários do processo podem analisar e compreender o conhecimento individualmente, fornecerem e compartilharem *feedback* para o ambiente na forma de comentários, idéias, dicas ou sugestões (Ma, Liu, & Wong 2000). Além disso, através dessa interface, os diversos usuários finais e especialistas do domínio poderão analisar os conjuntos de regras em suas estações de trabalho sem necessidade de instalar as ferramentas de Mineração de Dados.

A partir das informações contidas nos *logs* de utilização dos usuários e de seu *feedback*, algumas análises poderão ser feitas, entre elas:

- Permitir aos especialistas do processo de Extração de Conhecimento verificar o que é de interesse do usuário, fornecendo novos subsídios para apoiar uma próxima iteração de todo o processo e trazer resultados mais interessantes para o usuário final;
- Possibilitar uma análise das funcionalidades mais utilizadas com o objetivo de encontrar as medidas de avaliação mais efetivas;
- Verificar a existência de alguma relação entre medidas objetivas e subjetivas, dadas as consultas realizadas pelos usuários.

Na seção seguinte é apresentado o Ambiente para Exploração de Regras desenvolvido neste projeto de mestrado.

4.4 O Ambiente RulEE

Neste projeto de mestrado foi proposto um Ambiente para Exploração de Regras que viabilize tanto a análise quanto a disponibilização de regras. Com intuito de mostrar a viabilidade desse tipo de ambiente, foi desenvolvido o Ambiente *RulEE* (*Rule Exploration Environment*).

O *RulEE* apresenta alguns dos requisitos apresentados anteriormente e contem as funcionalidades necessárias para mostrar a vialibilidade do ambiente. Nas próximas seções, o Ambiente *RulEE* é apresentado, sendo especificadas quais funcionalidades foram implementadas. Também é apresentada uma comparação do Ambiente *RulEE* com ambientes com funcionalidades semelhantes existentes na literatura. No Capitulo 5, a implementação do Ambiente *RulEE* é apresentada em mais detalhes.

4.4.1 Descrição Geral

O Ambiente *RulEE* suporta análise de regras de classificação, regressão e associação. O ambiente tem como entrada arquivos com conjuntos de regras representados utilizando a sintaxe padrão definida no Discover. Por utilizar esses formatos de regras e pela utilização dos *scripts* de conversão dos diversos tipos de regras para a sintaxe padrão, também implementados no Discover, podem ser inseridas no ambiente as saídas de uma grande variedade de algoritmos para Extração de Conhecimento de Dados. Dentre esses algoritmos, encontram-se os mais utilizados na literatura, como o C4.5, CN2, *Cubist* e *Apriori*. A lista

completa dos algoritmos suportados pelos *scripts* de conversão para as sintaxes padrão foi apresentada nas Seções 4.2.1, 4.2.2, 4.2.3 e 4.2.4.

O processo de obtenção das regras a serem manipuladas pelo Ambiente *RulEE* está exemplificado na Figura 4.7. De acordo com a figura, inicialmente, os conjuntos de dados de origem são pré-processados e convertidos para a sintaxe padrão de dados do Discover. Após a preparação dos dados e definição do algoritmo a ser utilizado, os dados são convertidos para o formato de entrada do algoritmo selecionado. Deve-se ressaltar que, para cada tipo de conjunto de regras, diversos algoritmos podem ser utilizados. A conversão pode ser feita utilizando os métodos definidos na biblioteca para tratamento dos dados na sintaxe padrão (Batista 2001).

Após a execução dos algoritmos, seus resultados são convertidos para a sintaxe padrão do tipo de problema tratado (classificação, regressão ou associação). A partir das regras expressas na sintaxe padrão e de um conjunto de dados de teste, devem ser utilizados os *scripts* para cálculo dos valores da tabela de contingência (Prati, Baranauskas, & Monard 2001a; Pugliesi, Dosualdo, & Rezende 2003; Melanda & Rezende 2003). Os resultados desses *scripts* são utilizados como entrada no *RulEE*, não sendo portanto necessário o cálculo dessas informações internamente no ambiente.

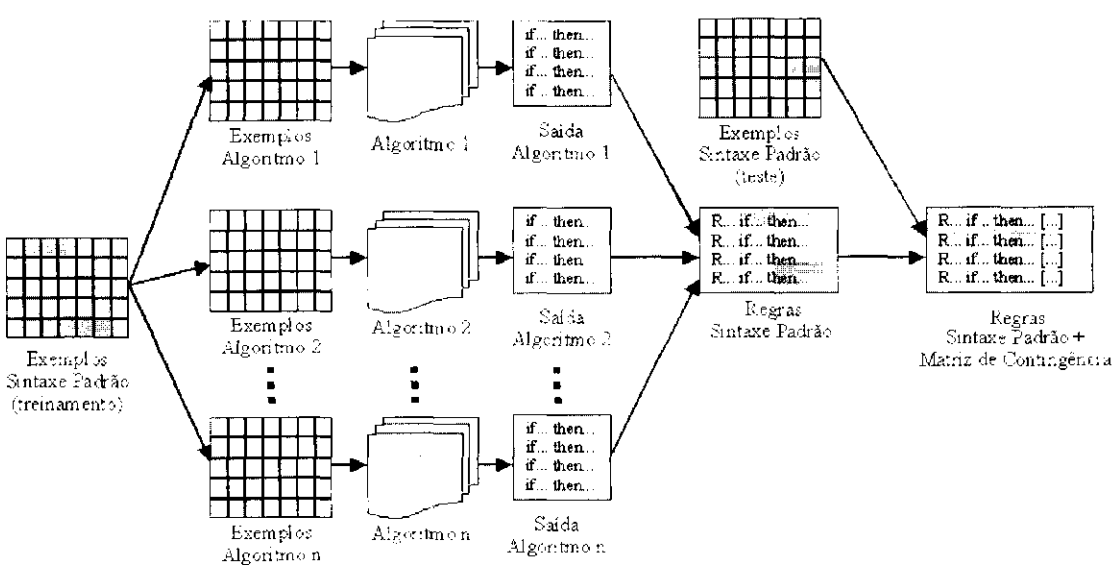

Figura 4.7: Metodologia para Obtenção de Regras nas Sintaxes Padrão

O *RulEE* apresenta um conjunto de medidas para avaliação de regras para classificação, regressão e associação de forma que diversos tipos de análise possam ser realizadas. Após um conjunto de regras ter sido inserido no ambiente, os valores das diversas medidas são calculados e ficam disponíveis para os usuários. O ambiente provê um fácil acesso

às regras e aos valores dessas medidas, e permite que várias medidas sejam utilizadas de forma combinada para a análise das regras pela aplicação de restrições ou filtros e utilização de medidas para a ordenação dos conjuntos de regras. Esse acesso às regras e aos valores das medidas é possibilitado pelo uso de consultas. Além disso, com o intuito de auxiliar a criação de novas medidas e a realização de experimentos, nesse ambiente é possível definir de maneira simples novas medidas cujo valor pode ser obtido a partir de uma expressão que utiliza o valor de outras medidas.

Deve-se ressaltar que vários trabalhos relacionados com a avaliação de conhecimento estão sendo desenvolvidos no Ambiente Discover. O Ambiente *RulEE* apresenta uma arquitetura que visa facilitar o armazenamento e disponibilização dos valores das medidas desenvolvidas permitindo uma fácil utilização conjunta de todas as medidas existentes. O cálculo do valor de novas medidas deve ser realizado pelos procedimentos desenvolvidos pelos próprios pesquisadores, bastando ao ambiente invocar esses procedimentos de maneira correta e buscar o valor desejado no arquivo contendo o resultado da execução do procedimento. A implementação do Ambiente *RulEE* permite que estas configurações possam ser feitas facilmente, sem necessidade de alteração de código fonte. Desta maneira, o Ambiente *RulEE* é independente dos procedimentos para Pós-processamento de conhecimento.

O *RulEE* provê duas formas para sua utilização. A primeira é através de uma biblioteca de classes que permite o acesso às diversas funções implementadas, como inserção, recuperação de regras e de valores de medidas, bem como acesso aos dados disponíveis na base de dados do ambiente. A utilização dessa biblioteca deve ser realizada através de programas escritos na linguagem *PERL*, sendo voltada a usuários analistas do processo de Extração de Conhecimento de Bases de Dados e que necessitam de recuperar as informações em seus *scripts* para Pós-processamento ou mesmo para definir alguma função desejada que não esteja implementada no ambiente. A segunda forma de interação é através da interface WWW. Esta interface possui a função principal de facilitar o acesso às informações armazenadas no ambiente e apresenta as mais diversas funções para acesso às regras e valores de medidas, além de facilitar a inserção de novos conjuntos de dados. A interface é voltada para usuários não especialistas ou usuários que não necessitem da flexibilidade da utilização da biblioteca de classes. O objetivo da existência dessa interface é facilitar o acesso e apresentação das regras descobertas através de uma interface mais intuitiva de maneira que os usuários finais ou especialistas do domínio possam acessá-

las sem necessidade de o fazer através da leitura de arquivos texto, propiciando assim a disponibilização desse conhecimento na WWW.

4.4.2 Funcionalidades do Ambiente RulEE

Na Seção 4.3 foram apresentadas características e requisitos importantes para um Ambiente para Exploração de Regras genérico. O *RulEE* foi implementado buscando-se atender à toda especificação apresentada. Entretanto, como um primeiro trabalho desenvolvido nesse sentido, o objetivo principal foi a construção da arquitetura básica do ambiente, de forma a possibilitar que todas as características propostas pudessem ser implementadas em trabalhos futuros. Dessa maneira, as principais características ou requisitos implementados no Ambiente *RulEE* são:

- As medidas objetivas para avaliação de regras descritas no *framework* apresentado em (Lavrac, Flach, & Zupan 1999) e a medida *RI* estão disponíveis para análise;
- Implementação de consultas para recuperação dos conjuntos de regras e dos respectivos valores de medidas;
- Implementação de uma interface simples baseada na WWW;
- Interface interativa e possibilidade de análise dos conjuntos de regras pela utilização de consultas de forma iterativa;
- Armazenamento de valores inseridos por usuários como indicativos da interessabilidade, compreensibilidade, utilidade e novidade das regras, além de armazenamento de comentários anexados a regras, conjuntos de regras e consultas.

4.5 Considerações Finais

Recentemente, ao observar que a Extração de Padrões não deve ser a etapa final do processo de Extração de Conhecimento de Bases de Dados, o Pós-processamento de conhecimento têm sido foco de muitas pesquisas. Nesse sentido, muitas ferramentas têm sido desenvolvidas para apoiar os usuários na avaliação do conhecimento encontrado. A integração dessas ferramentas em um Ambiente para Exploração de Regras pode representar um auxílio efetivo para os usuários do processo na compreensão, identificação do conhecimento interessante e utilização do mesmo. Neste capítulo foram apresentadas algumas características e requisitos importantes para um Ambiente para Exploração de Regras, assim como apresentado de maneira geral o protótipo desenvolvido neste projeto, o Ambiente *RulEE*.

Uma síntese de algumas funcionalidades existentes em ferramentas cujo objetivo é auxiliar a análise de conjuntos de regras é apresentada na Tabela 4.1. Nesta tabela estão presentes o Ambiente *RulEE* e as ferramentas AIAS, apresentada na Seção 3.4.3, e DS-WEB, apresentada na Seção 3.3.2.

Funcionalidade	**AIAS**	**DS-WEB**	***RulEE***
Tipo de Problema	Associação	Associação	Classificação, Regressão e Associação
Tipo de Análise	Identificação de Conformidade, Conseqüente Inesperado, Antecedente Inesperado, e Antecedente e Conseqüente Inesperados	Seleção de regras não redundantes	Utilização de medidas de avaliação
Medidas Subjetivas	Sim	Não	Não
Disponibilização dos Resultados na WWW	Não	Sim	Sim
Iteratividade	Sim	Não	Sim
Interatividade	Sim	Sim	Sim
Disponibilização das Análises	Não	Não	Sim

Tabela 4.1: Síntese de Funcionalidades de Ambientes para Análise de Regras

Cada linha da Tabela 4.1 corresponde a uma funcionalidade das ferramentas. Na linha correspondente à característica Tipo de Análise, não são especificadas medidas para o Ambiente *RulEE*, pois as análises realizadas no ambiente são independentes de medida ou procedimento para Pós-processamento, conforme apresentado anteriormente. Para a funcionalidade Medidas Subjetivas, o Ambiente *RulEE* apresenta o valor Não, pois ainda não estão sendo utilizadas no ambiente medidas desta natureza, apesar dele ter sido projetado para isto. Trabalhos futuros deverão estar relacionados com a utilização destes tipos de medidas no ambiente. A característica Iteratividade indica se a análise dos conjuntos de regras pode ser feita iterativamente, isto é, em diversos passos de análise, e a característica Interatividade indica se o usuário pode interagir com a ferramenta. Por fim, a característica Disponibilização das Análises indica se a ferramenta pode armazenar e disponibilizar para outros usuários algum tipo de análise realizada por algum usuário.

A disponibilização de regras deve estar integrada a um ambiente para apoiar as diversas etapas do processo de Extração de Conhecimento de Bases de Dados. O Ambiente para Exploração de Regras proposto neste projeto é integrado ao Discover, um ambiente que vem sendo desenvolvido em conjunto pelos pesquisadores do processo de Mineração de Dados e Aprendizado de Máquina do LABIC. Este ambiente deve apresentar uma interface integrada, com vários métodos para apoiar as diversas etapas do processo de Extração de

Conhecimento de Bases de Dados e Textos, desde o acesso aos dados até a avaliação do conhecimento.

No próximo capítulo é apresentada a implementação do Ambiente *RulEE*, sendo mostrada sua arquitetura, a base de dados, a biblioteca de classes, as formas de obtenção de medidas, as medidas disponíveis, a interface e as consultas que podem ser realizadas no repositório de regras e medidas.

Capítulo 5

Implementação do Ambiente RulEE

5.1 Considerações Iniciais

De maneira geral, o objetivo principal deste projeto de mestrado é a proposta e implementação de um Ambiente para Exploração de Regras que apresente os conceitos discutidos no Capítulo 4. Conforme a proposta discutida, esse ambiente deve apoiar tanto os usuários finais, por fornecer facilidades e flexibilidade para avaliação e compreensão das regras disponibilizadas, quanto os pesquisadores do processo de Extração de Conhecimento, por auxiliar a criação e alteração de medidas de avaliação. Neste capítulo é apresentada a implementação do Ambiente *RulEE* (*Rule Exploration Environment*), que apresenta as características de um Ambiente para Exploração de Regras proposto neste projeto de mestrado, descrevendo primeiramente a sua arquitetura e então detalhando seus principais elementos.

5.2 Implementação

A implementação do Ambiente *RulEE* está baseada em três elementos:

1. Biblioteca de classes, na qual estão implementadas as funções disponíveis pelo ambiente;

2. Base de dados, na qual estão armazenados o conhecimento no formato de regras, as medidas de avaliação das mesmas, os dados referentes aos usuários e os utilizados na extração e teste das regras;

3. Interface, construída baseada na WWW visando facilitar o acesso aos conjuntos de regras.

A arquitetura geral do Ambiente *RulEE* pode ser visualizada na Figura 5.1. No contexto do *RulEE*, o processo de análise e disponibilização de regras está baseado na idéia de se utilizar um Repositório de Regras e Medidas, que é a entidade principal do ambiente. Neste Repositório devem ser armazenados os conjuntos de regras, os valores das medidas e as consultas realizadas pelos usuários sobre o repositório de regras e medidas.

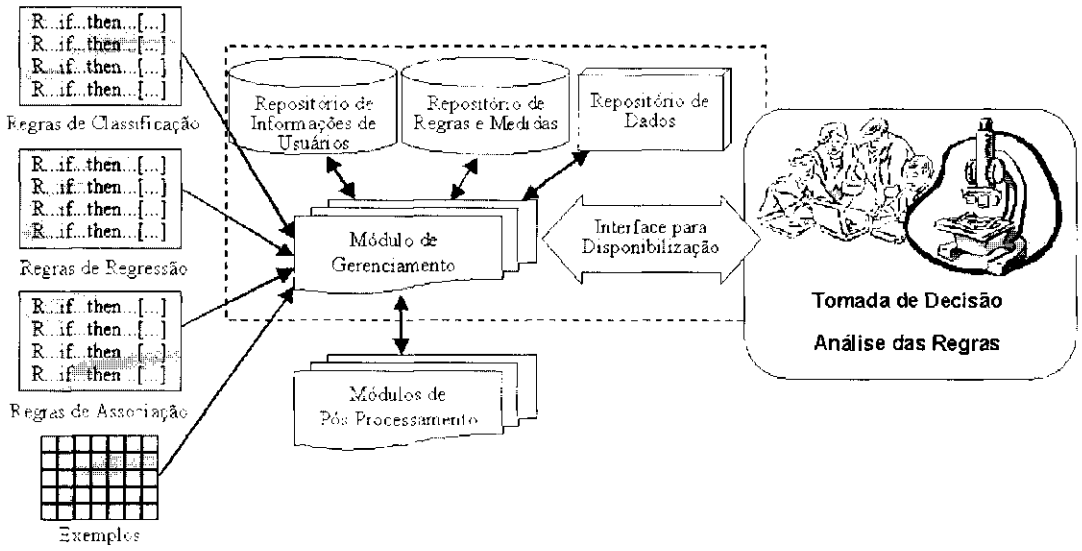

Figura 5.1: Arquitetura Geral do Ambiente *RulEE*

Conforme a Figura 5.1, a disponibilização das regras deve ser iniciada pela inserção no Repositório de Regras dos conjuntos de regras representados na sintaxe padrão. Esta atividade é realizada por métodos existentes no Módulo de Gerenciamento. Após estarem disponíveis no Repositório, as regras podem ser acessadas pela interface WWW ou pela biblioteca de classes, podendo ser analisadas pelos usuários e, possivelmente, utilizadas em um processo de tomada de decisão. Os elementos da figura que estão no retângulo pontilhado correspondem ao Ambiente *RulEE*.

Uma descrição mais detalhada dos elementos representados na Figura 5.1 é fornecida a seguir:

Módulos de Processamento São os programas escritos para processamento de dados, regras e informações dos usuários. Seguindo o padrão de desenvolvimento do Ambiente Discover, esses programas foram implementados na linguagem *PERL*. Os módulos de processamento encontrados no ambiente são:

Módulo de Gerenciamento Responsável pelo gerenciamento das regras disponibilizadas no ambiente. Este módulo recebe requisições da Interface do ambiente,

busca as regras correspondentes no Repositório de Regras, invoca os módulos de Pós-processamento ou recupera algum conjunto de dados no Repositório de Dados, para então retornar o resultado da requisição para a Interface. Este módulo é responsável também pela obtenção ou cálculo dos valores de medidas para as regras.

O Módulo de Gerenciamento possui também um conjunto de métodos que recebem como entrada conjuntos de regras nos formatos padrão de classificação, associação ou regressão, e fazem a inserção desses conjuntos de regras no Repositório de Regras e Medidas. Foram implementados métodos independentes responsáveis pela inserção das regras de classificação, associação e regressão.

O Módulo de Gerenciamento está implementado como uma biblioteca de classes que realiza todo esse processamento. Uma breve descrição das classes disponíveis é apresentada na Seção 5.2.3.

Módulos de Pós-processamento São os módulos responsáveis pelo cálculo das medidas de avaliação das regras disponibilizadas. Esses módulos contém procedimentos para cálculo das medidas e apresentam métodos distintos para tratamento de regras de classificação, associação e regressão. É importante destacar que no Ambiente *RulEE* são utilizados os módulos de Pós-processamento implementados no Ambiente Discover. Portanto, a implementação desses módulos de Pós-processamento não faz parte do escopo do *RulEE*, sendo definida somente uma interface para invocar os métodos de Pós-processamento quando necessário e para a obtenção dos valores das medidas para as regras nos arquivos contendo o resultado do Pós-processamento. Apesar de estar projetada nas tabelas e nas classes do ambiente, a utilização desses procedimentos não está completamente finalizada, ainda sendo necessária a implementação de alguns métodos. Deve-se ressaltar que, para a incorporação de medidas subjetivas no ambiente, faz-se necessária a utilização desses procedimentos.

Na implementação atual, as medidas existentes no Repositório têm seus valores obtidos a partir da tabela de contingência das regras, que é calculada antes da inserção das regras no ambiente.

Uma descrição mais detalhada sobre como os procedimentos são executados e como os valores das medidas são obtidos pode ser encontrada na Seção 5.2.4.

Interface Por meio da Interface, o usuário pode acessar todas as informações disponibilizadas pelo Ambiente *RulEE*. A Interface é baseada na WWW, a fim de aproveitar algumas de suas vantagens, como independência de plataforma e, especialmente, facilidade de utilização pelos usuários do processo independente de sua localização física, conforme apresentado na Seção 4.3.

Pela Interface de disponibilização, o usuário interativamente seleciona as regras que julgar mais interessantes pela utilização das medidas para então interpretá-las e utilizá-las. Uma outra função da Interface é coletar, armazenar e disponibilizar informações dos usuários. A implementação da Interface foi realizada utilizando o pacote *EmbPerl* para facilitar a utilização das classes *PERL* do ambiente na geração das páginas. Todas as funções disponíveis na Interface estão implementadas na biblioteca de classes correspondentes ao Módulo de Gerenciamento, de forma que não exista acesso direto à base de dados do ambiente. A Interface é abordada em mais detalhes na Seção 5.2.5.

Repositórios Correspondem à base de dados, desenvolvida utilizando o gerenciador MySql, ou aos arquivos nos quais são armazenadas as informações disponibilizadas. No ambiente, são encontrados três tipos de repositórios:

Repositório de Informações de Usuário Conjunto de tabelas MySql nas quais são armazenadas as informações relacionadas aos usuários do ambiente. Algumas informações armazenadas são: cadastro de usuários, comentários e votações em regras.

Repositório de Regras e Medidas Conjunto de tabelas para armazenamento das regras e dos valores das medidas a serem disponibilizadas no ambiente. Um aspecto importante foi a definição do modelo deste repositório, de forma a facilitar o armazenamento e recuperação das regras, o cadastro e inserção de novas medidas para avaliação, e o cálculo e recuperação dos valores das medidas para as regras.

Repositório de Dados Arquivos nos quais são armazenados os dados referentes aos conjuntos de regras disponibilizados. Estes dados podem ser utilizados para auxiliar o usuário na compreensão das regras, assim como podem ser utilizados por algum módulo de Pós-processamento. A fim de facilitar a utilização dos métodos de Pós-processamento implementados no Ambiente Discover, estes arquivos de dados devem seguir a sintaxe padrão para representação de dados proposta em

(Batista 2001). Estes arquivos são armazenados em um subdiretório do ambiente, sendo referenciados em sua base de dados.

A base de dados do ambiente é melhor detalhada na Seção 5.2.2.

Conjuntos de Regras Representam as regras extraídas e padronizadas para suas respectivas sintaxes padrão. Como citado anteriormente, o Ambiente *RulEE* trata regras expressas no formato padrão, permitindo que o resultado de uma grande variedade de algoritmos possa ser tratado pela utilização das bibliotecas de conversão. No ambiente, são tratados os seguintes tipos de regras:

Regras de Classificação Conjuntos de regras de classificação extraídos pelos mais diversos algoritmos de indução de regras e árvores de decisão e expressos na sintaxe padrão de regras de classificação definida em (Baranauskas 2001; Prati, Baranauskas, & Monard 2001b). Atualmente existem implementados procedimentos para conversão para esse formato padrão dos conjuntos de regras extraídos pelos algoritmos ID3, C4.5, C4.5*rules*, CN2, OC1, *Ripper*, C5.0 *See*5, T2 e MC4.

Regras de Regressão Os conjuntos de regras extraídos por diversos algoritmos de regressão e convertidos para o formato padrão de regras de regressão também são tratados pelo Ambiente *RulEE*. Foi desenvolvida a sintaxe padrão para regras de regressão, bem como uma biblioteca que converte as saídas dos algoritmos M5, RT, Cubist, CART e RETIS para esta sintaxe (Pugliesi, Dosualdo, & Rezende 2003).

Regras de Associação O terceiro tipo de regras tratado pelo Ambiente *RulEE* são as regras de associação. Um formato padrão de regras de associação bem como uma biblioteca para conversão dos resultados dos algoritmos Apriori, MagnumOpus e do *software* Mineset para este formato têm sido definidos por Melanda & Rezende (2003).

Uma breve descrição das sintaxes padrão para representação de dados, de regras de classificação, regressão e associação é apresentada nas Seções 4.2.1, 4.2.2, 4.2.3 e 4.2.4, respectivamente.

5.2.1 Informações Gerais

Nesta seção são apresentadas algumas informações gerais relacionadas à implementação do Ambiente *RulEE* e a sua utilização pela biblioteca de classes ou pela interface.

Variável de Ambiente Para permitir o acesso ao ambiente pela utilização de *scripts*, é necessário que os usuários LINUX apresentem dentre suas variáveis de ambiente a variável RULEE_DATABASE_NAME, a qual deve armazenar o nome da base de dados do Ambiente *RulEE* existente no servidor MySql em que o ambiente está instalado. Na implementação realizada, o nome da base de dados do ambiente é *rulee*. Caso o ambiente não encontre a variável RULEE_DATABASE_NAME, o acesso à base de dados fica impossibilitado, e ocorre o erro *Database name not found* (nome da base de dados não encontrado). É importante ressaltar que, para obter acesso à base de dados, o usuário necessita de direitos para acesso no servidor de base de dados MySql.

Direitos de Acesso à Base de Dados Conforme apresentado no item anterior, além de apresentar a variável de ambiente indicando o nome da base de dados do Ambiente *RulEE*, cada usuário que desejar utilizar os *scripts* deve possuir acesso a todos os objetos da base de dados. Assim, para esses usuário, deve ser executado o comando na base de dados `grant all on rulee.* to mfpaula@localhost;`. Neste comando, `mfpaula` deve ser substituido pelo *username* do usuário que desejar utilizar os *scripts*.

Estrutura de Diretórios O Ambiente *RulEE* é composto por um conjunto de classes *PERL*, *scripts* SQL, conjuntos de regras, conjuntos de dados, programas para Pós-processamento, entre outros. Para organizar todos esses elementos, foi definida uma estrutura de diretórios para o ambiente.

Na implementação do Ambiente *RulEE*, o diretório raiz é */home/mfpaula/rulee* e está localizado no servidor em que ele está instalado. Seus subdiretórios, bem como seus respectivos conteúdos, estão apresentados na Tabela 5.1.

Bibliotecas Necessárias Para o processamento e acesso aos conjuntos de dados de treinamento e teste dos conjuntos de regras cadastrados no ambiente, é utilizada a biblioteca *Dataset* definida por Batista (2001) para o Ambiente Discover. Além disso, para facilitar a leitura das regras existentes nos arquivos escritos na sintaxe padrão de classificação e sua inserção no ambiente, foi utilizada a classe *Rule* implementada

Subdiretório	Conteúdo
/db	Diretório que contém os arquivos com a definição dos objetos da base de dados.
/documentation	Diretório que contém a documentação disponível.
/source_code	Diretório que contém os códigos fonte das classes *PERL* correspondentes ao Módulo de Gerenciamento.
/interface/scripts	Diretório que contém os *scripts EmbPerl* correspondentes à interface do ambiente (arquivos *.html*).
/interface/images	Diretório que contém as imagens utilizadas na interface do ambiente.
/rule_repository	Diretório que contém os arquivos que representam os conjuntos de regras armazenados no ambiente. Os nomes desses arquivos são definidos pelos padrões *ruleset_ID.classification.model*, *ruleset_ID.regression.model* ou *ruleset_ID.association.ruleset*, nos quais *ID* representa o identificador do conjunto de regras.
/data_repository	Diretório que contém os arquivos de dados. Os nomes dos arquivos são: *ruleset_ID.data* para os arquivos de dados de treinamento, *ruleset_ID.test* para os arquivos de testes e *ruleset_ID.names* para os arquivos com a descrição dos dados.
/html_files	Diretório que contém descrições de conjuntos de regras, projetos, procedimentos ou medidas. Os nomes dos arquivos existentes neste diretório são definidos pelos padrões: *ruleset_ID.html*, *proj_PROJECT_CODE.html*, *proc_PROCEDURE.html* e *measure_MEASURE.html*, nos quais *PROJECT_CODE* representa o código do projeto, *PROCEDURE* representa o nome do procedimento e *MEASURE* representa o nome da medida descritos pelos arquivos.
/pos_proc_scripts	Diretório destinado para armazenamento dos *scripts* responsáveis pelo Pós-processamento. O armazenamento dos arquivos neste diretório é opcional.
/tmp	Diretório destinado para armazenamento de arquivos temporários.

Tabela 5.1: Subdiretórios do Ambiente *RulEE*

por Prati, Baranauskas, & Monard (2001a). No ambiente, classes semelhantes foram criadas para a inserção de regras de regressão e associação.

Pacotes *PERL* Necessários Para a execução do Ambiente *RulEE*, é necessária a instalação de alguns pacotes *PERL*. Dentre os pacotes necessários, estão:

- Pacote *DBI*, versão 1.18-1, para o acesso à base de dados;
- Pacote *Exception*, versão 1.4, para o tratamento de erros;
- Pacotes *Date-Calc*, versão 5.0, *Math-CDF*, versão 0.1, e *Math-Round*, versão 0.02, necessários para a utilização da biblioteca *Dataset*;
- Pacotes *EmbPerl*, versão 1.3.4, e *mod_perl*, versão 1.27, utilizados na implementação da interface do Ambiente *RulEE*.

O pacote *DBI* é normalmente obtido na distribuição LINUX utilizada. Os outros pacotes necessários podem ser obtidos em `www.cpan.org`.

Exceções O Ambiente *RulEE* apresenta um conjunto de exceções que podem ocorrer durante a execução dos algoritmos. Quando ocorre um erro, é criado um objeto da classe *Exception* e gerada uma exceção que finaliza a execução do *script*. Se o usuário estiver utilizando as classes do ambiente em algum *script*, é possível identificar se foi algum erro por alguma condição inválida do ambiente, ou se foi algum outro tipo de erro, como uma permissão requerida para acessar algum arquivo, um erro de conexão com a base de dados, entre outros. Caso o ambiente esteja sendo utilizado pela interface WWW e alguma exceção ocorrer, é apresentado ao usuário a mensagem completa do erro sem nenhum tratamento.

Criação e Acesso dos Usuários Do ponto de vista da forma de utilização do ambiente, existem dois tipos básicos de usuários: usuários que utilizam somente a interface WWW e usuários que utilizam a interface e as classes. Para permitir o acesso ao ambiente desses dois tipos de usuários, algumas configurações de acesso devem ser feitas:

Usuários que acessam a interface e as classes Para que os usuários possam utilizar o ambiente através da biblioteca de classes, é necessário que eles apresentem códigos de usuário e senha para utilização da estação de trabalho em que o ambiente está instalado. É necessário que ele possua também direito de acesso aos diretórios de *scripts* e dados, direito de acesso à base de dados, além de cadastro na base de dados do ambiente. Com essa configuração, é possível também o acesso direto ao conteúdo das tabelas existentes na base de dados do ambiente.

Usuários que acessam somente a interface Para acesso pela interface, não é necessário código de acesso e senha na estação onde o ambiente está instalado, pois a interação com a base de dados e com a biblioteca de classes é feita através da WWW. Entretanto, é necessário o cadastro do usuário na base para que seja possível o *login* no *RulEE*.

Na próxima seção será apresentada em mais detalhes a implementação da base de dados do Ambiente *RulEE*.

5.2.2 Implementação da Base de Dados

Conforme apresentado anteriormente, a base de dados do Ambiente *RulEE* foi implementada no gerenciador de base de dados MySql. Todas as tabelas do ambiente foram implementadas em uma única base de dados. Estas tabelas estão descritas na Tabela 5.2

e o diagrama físico da base de dados, contendo as colunas, chaves e restrições, está representado na Figura 5.2.

Tabela	**Descrição**
COMMENT	Tabela que armazena comentários diversos que os usuários do ambiente podem anexar em regras, conjuntos de regras ou seleções. Os comentários ficam disponíveis para todos os usuários que possuem acesso às mesmas regras, conjuntos de regras ou consultas.
MEASURE	Tabela em que estão cadastradas as medidas para avaliação de regras utilizadas pelo ambiente. Nesta tabela são especificadas informações inerentes às medidas, não apresentando detalhes de sua obtenção.
MEASURE_HOW_CALCULATED	Tabela em que estão armazenadas as informações de como as medidas cadastradas no ambiente têm seus valores obtidos. Para cada medida existente na tabela *MEASURE*, deve existir um registro nesta tabela para especificar como seu valor é obtido para cada um dos tipos de conjuntos de regras tratados pelo ambiente (classificação, regresssão e associação).
PARAMETER	Tabela que armazena os parâmetros de configuração do ambiente. Os parâmetros cadastrados nesta tabela são o tempo de *timeout* das sessões do ambiente e o diretório raiz do mesmo.
PROC	Tabela de cadastro dos procedimentos para Pós-processamento dos conjuntos de regras.
PROJECT	Tabela que armazena os projetos cadastrados no ambiente.
PROJECT_USER	Tabela que indica quais projetos os usuários possuem acesso.
RULE	Tabela que armazena as regras pertencentes aos conjuntos de regras cadastrados no ambiente. As regras são armazenadas nesta tabela em formato texto de acordo com a sintaxe padrão para o seu tipo (classificação, regressão ou associação).

continua na próxima página...

continuação

Tabela	Descrição
RULESET	Tabela que armazena os conjuntos de regras disponibilizados no ambiente. Após um conjunto de regras ter sido cadastrado no ambiente, é criado um arquivo no subdiretório *rule_repository* contendo a cópia do arquivo original que representa o conjunto de regras utilizado para inserção, de maneira que esse arquivo possa ser acessado por algum procedimento de Pós-processamento.
SELECTION	Tabela que armazena as consultas feitas pelos usuários no repositório de regras e medidas.
SESSION	Tabela que armazena as sessões de utilização do ambiente.
USER	Tabela que armazena o cadastro dos usuários do ambiente. O cadastro de um usuário nesta tabela não é condição suficiente para que ele possa utilizar o ambiente. É necessário também fornecer acesso para a base de dados, conforme apresentado na seção anterior. Além disso, é necessário especificar na tabela *PROJECT_USER* quais projetos o usuário possui acesso.
USER_MEASURE	Tabela que armazena os índices fornecidos pelos usuário para as regras como forma de indicativo para a interessabilidade, compreensibilidade, utilidade e novidade. Ao visualizar uma regra, o usuário pode verificar a média dos valores fornecidos pelos outros usuários do ambiente para a mesma.
USER_MEASURE_SUM	Tabela auxiliar que objetiva permitir a utilização das médias dos valores dos índices inseridos por usuários nas consultas do ambiente. Os valores dessa tabela são uma sumarização dos registros da tabela *USER_MEASURE* para cada regra e cada tipo de índice fornecido pelos usuários.

continua na próxima página...

continuação

Tabela	Descrição
VALUE_MEASURE_RULE	Tabela que armazena os valores das medidas das regras disponibilizadas. Sempre que o usuário requisita o valor de uma medida de uma determinada regra, o ambiente verifica nesta tabela se o valor já está calculado. Caso esteja, o valor é retornado. Caso o valor não esteja presente, ele é obtido, armazenado nesta tabela para futuros acessos e retornado ao usuário.
VALUE_MEASURE_RULESET	Tabela com os valores das medidas referentes a conjuntos de regras. Da mesma maneira que para a tabela *VALUE_MEASURE_RULE*, o ambiente armazena nesta tabela os valores das medidas que foram acessadas pelos usuários.

Tabela 5.2: Descrição das Tabelas da Base de Dados do Ambiente *RulEE*

Na Figura 5.2, são apresentadas as restrições de integridade existentes entre as tabelas do ambiente. É importante destacar que o gerenciador de base de dados MySql não implementa as chaves estrangeiras e restrições do tipo *check*. Apesar disso, nos *scripts* de definição dos objetos da base, as restrições foram todas especificadas de maneira a auxiliar possíveis desenvolvedores a compreender melhor o modelo e a facilitar a utilização de um gerenciador mais robusto que implemente essas restrições.

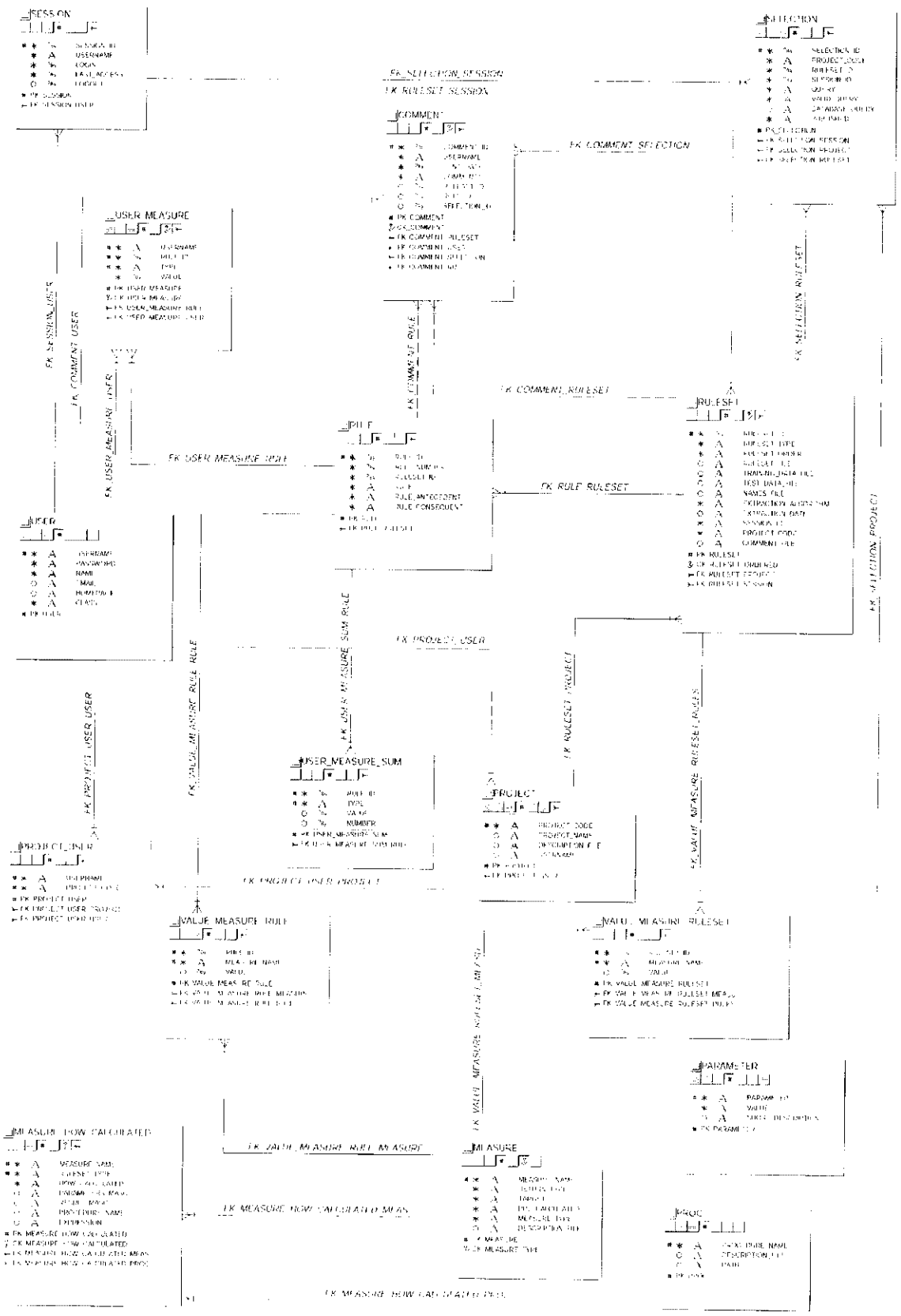

Figura 5.2: Diagrama Físico da Base de Dados do Ambiente *RulEE*

5.2.3 Classes

Nesta seção, são apresentadas as classes que compõem o Módulo de Gerenciamento do Ambiente *RulEE*. Na Tabela 5.3, as classes estão descritas de maneira bastante resumida e na Figura 5.3 é apresentado o diagrama de classes que apresenta os relacionamentos entre elas.

Classe	**Descrição**
Comment	Classe que representa um comentário que pode ser anexado por um usuário a uma regra, um conjunto de regras ou a uma consulta.
Measure	Classe que representa uma medida de avaliação.
Procedure	Classe que representa um procedimento do ambiente.
Project	Classe que representa um projeto existente no Ambiente *RulEE*. Essa classe é utilizada para retornar informações referentes a um determinado projeto, como sua descrição e seus conjuntos de regras.
Repository	Classe utilizada para retornar elementos referentes ao Repositório de Regras e Medidas como um todo, como as medidas cadastradas no ambiente.
Rule	Classe que representa uma regra cadastrada no ambiente.
Ruleset	Classe que encapsula um conjunto de regras de classificação, regressão ou associação.
Selection	Classe que representa uma consulta feita pelo usuário no Repositório de Regras e Medidas.
Session	Classe que representa uma sessão no ambiente.

Tabela 5.3: Descrição das Classes do Módulo de Gerenciamento

Na Figura 5.4 pode-se encontrar um outro diagrama que contém os relacionamentos de herança, os atributos e os métodos de cada classe do Módulo de Gerenciamento. Como pode ser observado na figura, todas as classes são subclasses de *Session*. Esta decisão foi tomada para permitir que todas as classes do ambiente possuíssem métodos para a conexão com a base de dados. Também na Figura 5.4, podem ser observados os métodos e atributos públicos e privados de cada classe. Os métodos ou atributos privados estão identificados com o ícone de uma chave anterior ao nome do atributo ou método. Apesar de não existir definição de métodos ou atributos privados em *PERL*, no projeto do Ambiente

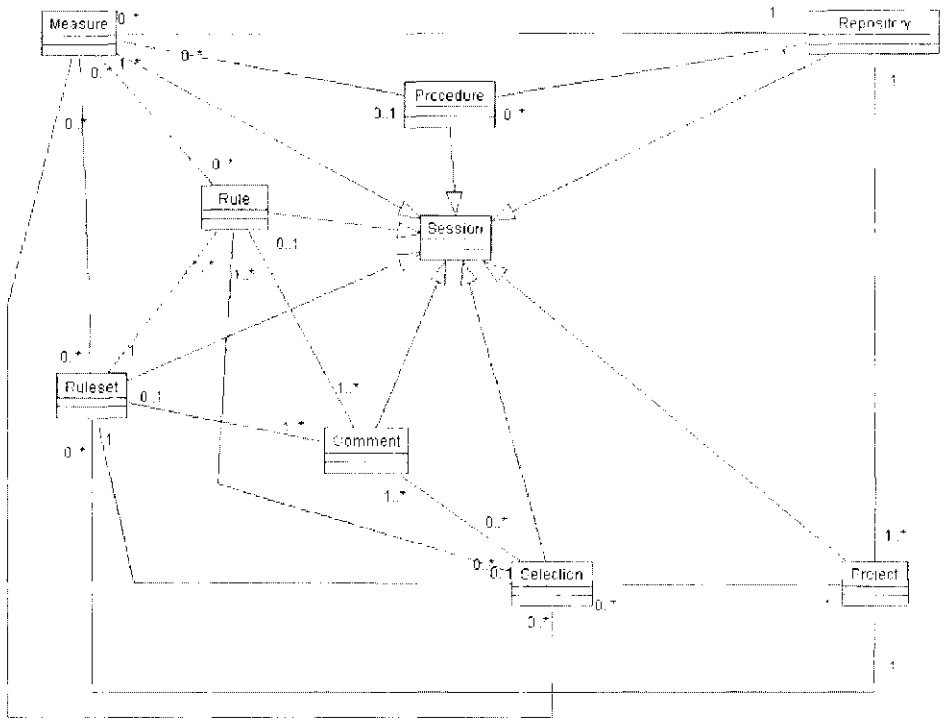

Figura 5.3: Relacionamentos entre as Classes do Módulo de Gerenciamento

RulEE alguns métodos e atributos foram assim definidos, pois não é recomendada sua utilização pelos usuários.

Como pode ser observado na Figura 5.4, todos os atributos das classes são privados. O acesso aos valores desses atributos deve ser realizado pelos métodos que retornam seus valores.

5.2.4 Obtenção e Cálculo de Valores de Medidas

Todas as medidas disponíveis no Ambiente *RulEE* devem estar cadastradas na tabela *MEASURE*. Entretanto, para cada tipo de conjunto de regras, deve estar especificado na tabela *MEASURE_HOW_CALCULATED* como o valor da medida é obtido. Por exemplo, a medida *Acc*, que corresponde à precisão de uma regra, deve estar cadastrada na tabela *MEASURE*. Mas, para que ela esteja disponível para regras de classificação, deve existir um registro na tabela *MEASURE_HOW_CALCULATED* indicando a maneira pela qual o valor de *Acc* é obtido para problemas de classificação. Ao se tentar utilizar essa medida em uma consulta em um tipo de conjunto de regras (classificação, regressão ou associação) não existente na tabela *MEASURE_HOW_CALCULATED*, é apresentado um erro ao usuário indicando que ela não é válida para este tipo de regra.

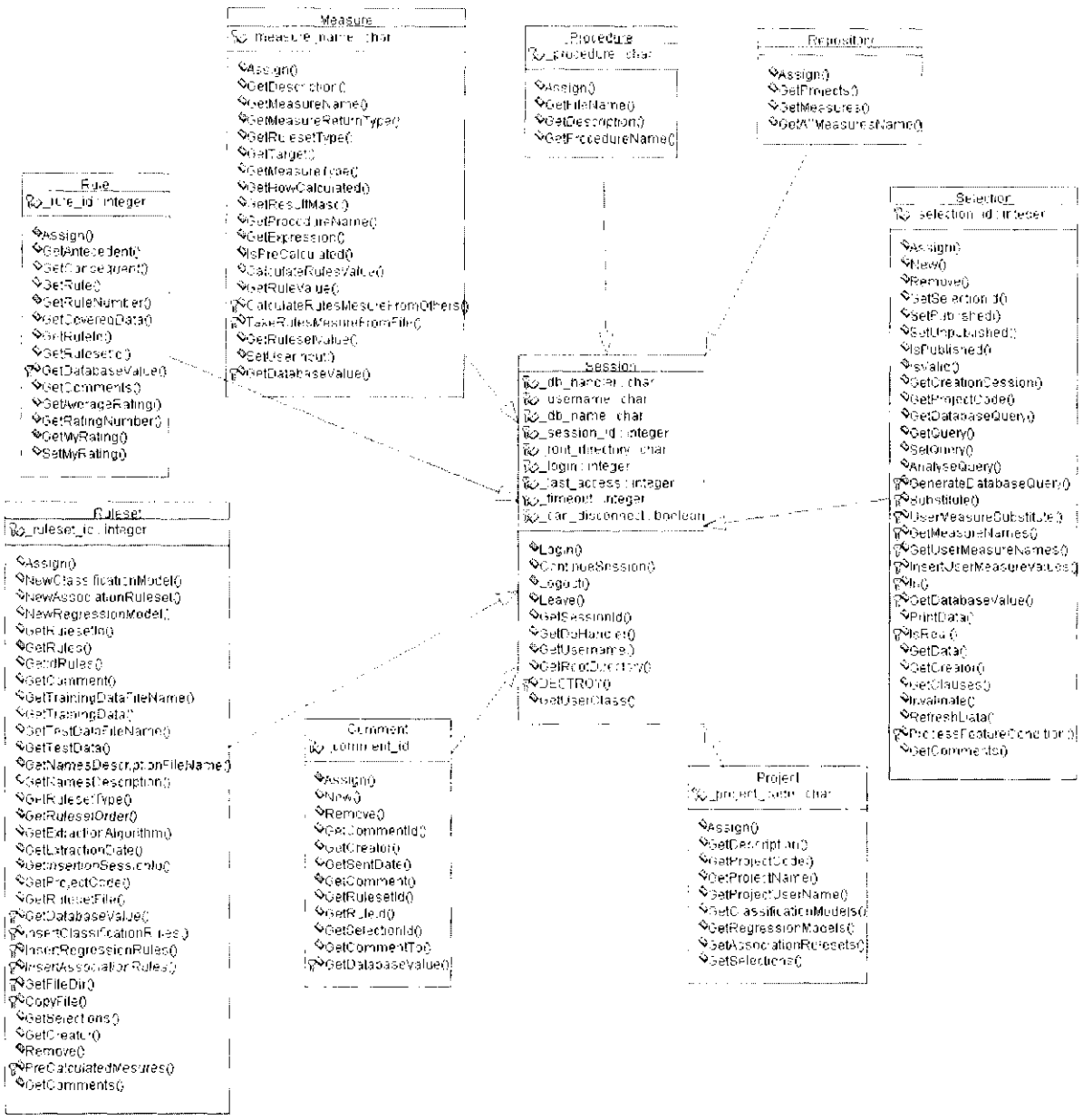

Figura 5.4: Herança, Atributos e Métodos das Classes do Módulo de Gerenciamento

Conforme descrito nas seções anteriores, a obtenção dos valores das medidas disponíveis no ambiente pode ser feita de três maneiras: leitura das medidas do arquivo que contém o conjunto de regras, execução de procedimentos e cálculo de medidas derivadas a partir do valor de outras medidas cadastradas no ambiente. Essas três possibilidades para obtenção dos valores das medidas para as regras estão descritas a seguir.

Obtenção pelo Arquivo de Regras Para a inserção de conjuntos de regras, devem ser fornecidos ao ambiente arquivos que contenham o conjunto de regras na sintaxe padrão com os valores da tabela de contingência para cada regra. Assim, os valores da tabela

de contingência são obtidos a partir destes arquivos, sendo as medidas derivadas da tabela de contingência calculadas pelo próprio ambiente pela utilização de medidas derivadas.

Desta maneira, os registros correspondentes aos valores da tabela de contingência da tabela *MEASURE_HOW_CALCULATED* devem apresentar a coluna *HOW_CALCULATED* com valor *RULESET_DESCRIPTION*, indicando que o ambiente deve buscar no arquivo de regras o valor dessas medidas. Além disso, a coluna *RESULT_MASC* deve apresentar uma máscara para permitir ao ambiente buscar no arquivo o valor da medida para cada regra do conjunto. Para estas medidas, as colunas *PARAMETERS_MASC*, *PROCEDURE_NAME* e *EXPRESSION* devem ficar com valores nulos.

Para que o conteúdo da coluna *RESULT_MASC* possa ser utilizado para identificar o valor de uma medida para cada regra do ambiente, ele deve corresponder a um padrão válido da linguagem *PERL* que, para cada regra do conjunto de regras, deve casar uma única vez com o arquivo de regras como um todo. Desta maneira, variáveis de substituição devem ser utilizadas nas máscaras para a identificação de cada regra.

Para exemplificar a obtenção de valores a partir dos arquivos de descrição dos conjuntos de regras, no padrão `$rule_antecedent,$rule_consequent,\[[ \t]*(-?[0-9] *\.?[0-9]*)[ \t]*`, as variáveis de substituição `$rule_antecedent` e `$rule_consequent` são utilizadas para identificar uma regra dentro de um arquivo de regras. Uma outra variável de substituição que pode ser utilizada nas máscaras é `$rule_number`, representado o número da regra no arquivo de regras. O padrão apresentado corresponde ao conteúdo da coluna *RESULT_MASC* para o valor da tabela de contingência f_{bb} para conjuntos de regras de associação. Neste exemplo, para cada regra de um conjunto de regras, `$rule_antecedent` e `$rule_consequent` são substituídos pelos respectivos antecedente e conseqüente da regra. Assim, esse padrão "casa" com o arquivo de regras uma única vez, e após isso, o valor de f_{bb} para uma determinada regra é obtido pela variável $1, isto é, o valor da medida fica instanciado com o primeiro sub-padrão entre parênteses do conteúdo de *RESULT_MASC*. No exemplo, o valor da medida é obtido pelo sub-padrão `(-?[0-9]*\.?[0-9]*)`.

Obtenção pela Execução de Procedimento É importante que o Ambiente *RulEE* permita, de maneira simples, a utilização de procedimentos para cálculo de medidas. Essa funcionalidade está completamente modelada na base de dados, mas está parcialmente

implementada no ambiente, pois alguns métodos necessários não estão implementados.

Como previsto na tabela *MEASURE_HOW_CALCULATED*, medidas que deveriam ter seu valor obtido a partir da execução de um procedimento devem apresentar a coluna *HOW_CALCULATED* com o valor *PROCEDURE* e valores válidos para as colunas *PARAMETERS_MASC*, *RESULT_MASC* e *PROCEDURE_NAME*. A coluna *EXPRESSION* deve ficar com valor nulo.

Para uma medida calculada por procedimento, a coluna *PROCEDURE_NAME* deve apresentar o nome de um procedimento cadastrado na tabela *PROC*. A coluna *PARAMETERS_MASC* deve apresentar uma máscara para obtenção dos parâmetros a serem utilizados na execução do procedimento. Por exemplo, a coluna *PARAMETERS_MASC* pode conter o valor `$ruleset_file_name $rule_number $output_file_name`, em que `$ruleset_file_name` é substituído pelo nome do arquivo que contém o conjunto de regras na sintaxe padrão, `$rule_number` é substituído pelo número da regra que se deseja processar e `$output_file_name` é substituído pelo nome do arquivo no qual o ambiente deve buscar o resultado. A coluna RESULT_MASC deve conter a máscara para obtenção do resultado para cada regra dentro do arquivo de saída criado pelo procedimento. Esta máscara deve ser semelhante às mascaras de obtenção de valores de medidas nos arquivos contendo os conjuntos de regras.

Obtenção pelo Cálculo de Medidas Derivadas Tendo obtido o valor de alguma medida a partir do arquivo de descrição de regras ou pela execução de um procedimento, é possível a definição de medidas derivadas, que têm seu valor calculado a partir do valor dessas outras medidas. Com este tipo de medida, é possível o cálculo das várias medidas derivadas a partir dos valores da tabela de contingência.

Medidas derivadas são identificadas na tabela *MEASURE_HOW_CALCULATED* por conter o valor *DERIVED* na coluna *HOW_CALCULATED*. Um registro de medida derivada deve apresentar a expressão de como o valor da medida é obtido na coluna *EXPRESSION*, e as colunas *PARAMETERS_MASC*, *RESULT_MASC* e *PROCEDURE_NAME* devem ter valor nulo, pois não são necessárias para o cálculo do valor desse tipo de medida.

O valor da coluna *EXPRESSION* deve ser uma expressão aritmética válida em que são utilizados os nomes de outras medidas, como variáveis *PERL*, e operadores matemáticos para o cálculo do valor da medida referente ao registro. Por exemplo, a expressão

$\$fbh/\$fb$ é utilizada para calcular o valor da medida *Acc* de uma regra de classificação. Nessa expressão, fbh e fb devem corresponder a nomes de medidas válidas para o mesmo tipo de conjunto de regras que a medida *Acc*. Para calcular o valor de *Acc* para uma determinada regra, o ambiente identifica as medidas que são utilizadas na expressão, calcula seus valores para a regra em questão para então calcular o valor de *Acc*.

Na próxima seção, é apresentada a interface do Ambiente *RulEE*.

5.2.5 Descrição da Interface

Conforme os requisitos de um Ambiente para Exploração de Regras apresentados no Capítulo 4, é importante que esteja disponível uma interface WWW para permitir um acesso facilitado de seus usuários às regras disponibilizadas. O desenvolvimento da interface do Ambiente *RulEE* foi realizado com objetivo de facilitar a demonstração das funcionalidades e potencialidades do ambiente na análise e disponbilização das regras e medidas. Dessa maneira, foi desenvolvida uma interface bastante simples voltada à apresentação das informações, não havendo grande preocupação com a usabilidade da mesma. É importante destacar que o foco principal do trabalho foi a proposta e a implementação de uma arquitetura para exploração de regras, de maneira que trabalhos futuros possam se desenvolver utilizando esta arquitetura. Por ser de grande importância para um Ambiente para Exploração de Regras, a interface do Ambiente *RulEE* deve ser abordada em trabalhos futuros, conforme apresentado no Capítulo 7.

Para a construção da interface do Ambiente *RulEE*, foi utilizado o servidor *Apache* e os pacotes *EmbPerl* e *mod_perl* para a construção dinâmicas das páginas acessadas pelo usuário. Na construção destas páginas, foram utilizadas as classes existentes no Módulo de Gerenciamento, descritas na Sessão 5.2.3.

Para utilização da interface é necessário que o *browser* da estação cliente esteja com a opção *Java Script* habilitada. Caso contrário, muitas funcionalidades da interface não estarão disponíveis. Além disso, a interface do ambiente foi desenvolvida para que suas funções funcionem corretamente nos *browsers Mozilla* versão 0.9.9 no sistema operacional LINUX e no *Internet Explorer* versão 5.0 no sistema operacional *Windows*. O Ambiente *RulEE* pode ser utilizado pela interface no endereço `http://143.107.232.90/rulee/index.html`.

Nesta seção são apresentadas as principais funções disponíveis aos usuários pela interface WWW através da descrição das telas do ambiente. As funções disponíveis aos usuários pela interface do Ambiente *RulEE* são:

- *Login* no ambiente;
- Visualização de informações sobre o projeto;
- Inserção, visualização e remoção de conjuntos de regras;
- Criação, visualização, edição e remoção de consultas sobre conjuntos de regras e valores de medidas;
- Inserção, visualização e remoção de comentários sobre regras, conjuntos de regras ou consultas;
- Inserção, visualização e alteração de índices de Interessabilidade, Compreensibilidade, Utilidade e Novidade fornecidas por usuários sobre regras;
- Visualização das medidas disponíveis no ambiente.

A seguir, serão apresentadas as principais telas do ambiente, sendo descritas suas principais funcionalidades.

Login no Ambiente

Ao entrar na interface do ambiente, a tela de *login*, correspondente à Figura 5.5, é apresentada ao usuário. Para acessar o ambiente, é necessário que os usuários forneçam seu código de acesso e senha.

Menu Principal

Após fornecer o código de acesso e senha na tela de *login*, o ambiente apresenta a tela com o *menu* principal e as informações de um projeto que o usuário possui acesso, conforme pode ser observada na Figura 5.6. As principais funções disponíveis neste *menu* são:

Troca de Projeto Ao clicar no campo em que está presente o nome do projeto, contendo DadosFinanceiros na Figura 5.6, é apresentada ao usuário a lista dos projetos aos quais ele possui acesso. Ao escolher um projeto da lista, é aberta a tela para acesso das informações do projeto e todas as informações seguintes serão relativas ao projeto escolhido.

***Link* Todas as Consultas** *Link* utilizado para acessar a tela contendo todas as consultas do projeto atual.

Para cada um dos tipos de conjuntos de regras tratados pelo Ambiente *RulEE* (classificação, regressão e associação), estão presentes no *menu* os seguintes *links*:

***Link* Conjuntos de Regras** *Link* de acesso à tela com a listagem dos conjuntos de regras disponíveis no projeto atual do usuário.

***Link* Consultas** *Link* de acesso à tela com a listagem das consultas sobre regras e valores de medidas disponíveis no projeto atual do usuário.

***Link* Medidas** *Link* de acesso à tela com a listagem das medidas disponíveis no ambiente.

***Link* Novo Conjunto de Regras** Por este *link* o usuário tem acesso à tela de inserção de um novo conjunto de regras.

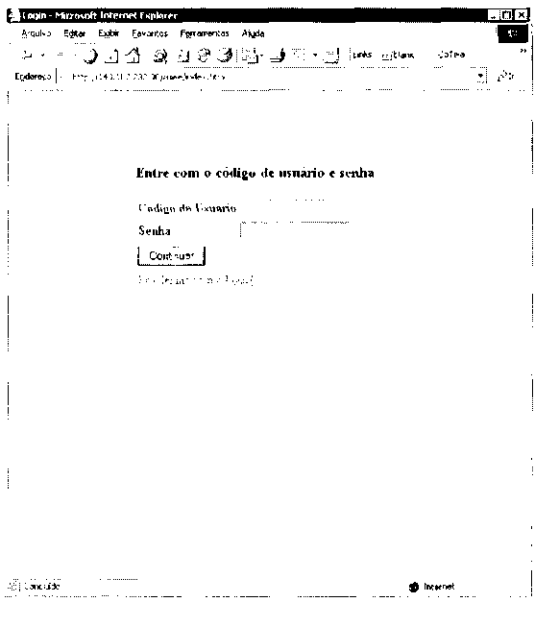

Figura 5.5: *Login* no Ambiente

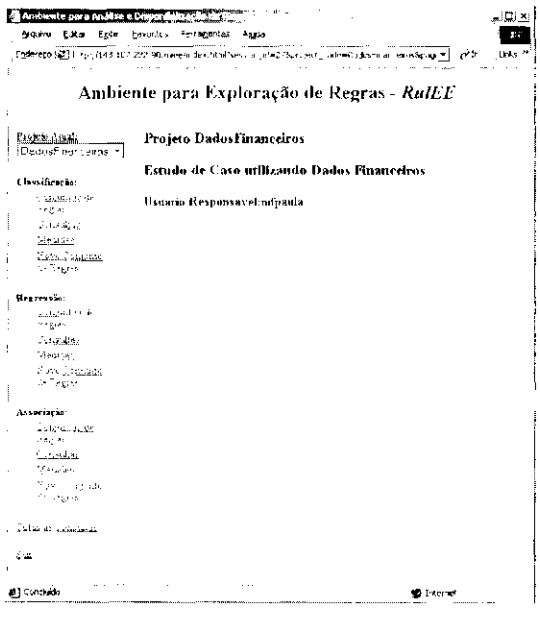

Figura 5.6: *Menu* Principal do Ambiente

Inserção de um Conjunto de Regras

Na Figura 5.7 é apresentada a tela utilizada para inserção de um conjunto de regras de classificação. A inserção de conjuntos de regras de regressão e associação é realizada por telas equivalentes. Para inserir o conjunto de regras de classificação no ambiente, o usuário deve selecionar, pelos botões Procurar, o arquivo que apresenta o conjunto de regras (arquivo *.rules*), os arquivos contendo a descrição dos dados (arquivo *.names*), arquivos com os dados de treinamento (arquivo *.data*), os dados de testes (arquivo *.test*) e o arquivo com algum comentário sobre o conjunto de regras. Após selecionar os arquivos, ao pressionar o botão Enviar, o conjunto de regras é inserido na base de dados do ambiente e

apresentado ao usuário. Caso ocorra algum erro em seu processamento, uma mensagem com a descrição do erro é apresentada ao usuário.

É importante destacar que é obrigatória a inserção do arquivo contendo o conjunto de regras. A inserção dos arquivos com os dados de treinamento, de teste, o arquivo *.names* e o arquivo com comentários sobre o conjunto de regras é opcional. É obrigatório também que o arquivo contendo o conjunto de regras esteja na sintaxe padrão, seguindo as sintaxes apresentadas nas Seções 4.2.2, 4.2.3 e 4.2.4. É importante também que os arquivos de dados estejam na sintaxe padrão de dados, também apresentada na Seção 4.2.1.

Listagem de Conjuntos de Regras

Na Figura 5.8 é apresentada a tela com a listagem dos conjuntos de regras de associação disponíveis no projeto atual do usuário. As telas de listagem dos conjuntos de regras de regressão e classificação são equivalentes a essa.

Visando facilitar a identificação de um conjunto de regras, para cada elemento da lista, estão presentes as seguintes informações: o identificador do conjunto de regras, a data de extração, o algoritmo utilizado, o usuário que cadastrou o conjunto de regras no ambiente e um comentário. No campo Comentário, é mostrada uma pequena parte do comentário inserido pelo usuário juntamente com o conjunto de regras.

O acesso a um determinado conjunto de regras pode ser feito pelo *link* existente no identificador do conjunto escolhido.

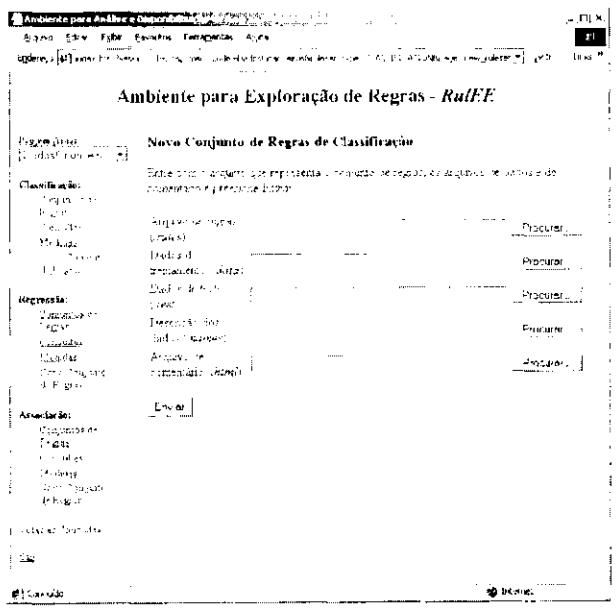

Figura 5.7: Inserção de Conjuntos de Regras

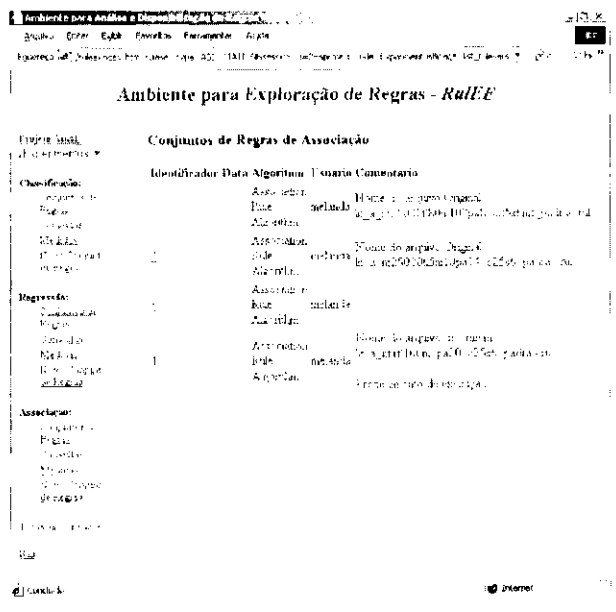

Figura 5.8: Listagem de Conjuntos de Regras

Visualização de um Conjunto de Regras

Na Figura 5.9 é apresentada a tela de visualização de conjuntos de regras. Nesta tela podem ser observadas as seguintes informações:

- Código identificador do conjunto de regras;
- Tipo do conjunto de regras (classificação, regressão ou associação);
- Tipo de ordenação entre as regras (não ordenadas, ordenadas entre classes e ordenadas);
- O algoritmo utilizado para extração das regras;
- Data de extração das regras;
- Algum comentário inserido pelo usuário juntamente com o arquivo que contém o conjunto de regras;
- As regras pertencentes ao conjunto de regras. Para cada regra é apresentado um indicativo dos valores das medidas de Interessabilidade, Compreensibilidade, Utilidade e Novidade inseridas pelos usuários. Estes indicativos obedecem a seguinte regra, sendo m a média dos valores inseridos para cada uma das quatro medidas por todos os usuários:

  $m \geq 0.5$: Seta para cima;

  $m \leq \ 0.5$: Seta para baixo;

  $-0.5 < m < 0.5$: Seta horizontal;

- Comentários inseridos por outros usuários do ambiente sobre o conjunto de regras;

Além das informações listadas anteriormente, a tela de apresentação de conjuntos de regras disponibiliza *links* para acesso a outras informações, listadas a seguir:

***Link* Arquivo com o Conjunto de Regras** Por este *link*, o usuário pode visualizar ou salvar o conjunto de regras original, utilizado para inserção das regras no ambiente. Ao clicar neste *link*, é mostrada ao usuário uma tela semelhante ao exemplo apresentado na Figura 5.10.

***Link* Arquivo Names e Arquivos de Dados de Treinamento e Teste** Caso os arquivos contendo a descrição dos dados (*.names*), os dados de treinamento (*.data*) e teste (*.test*) te-

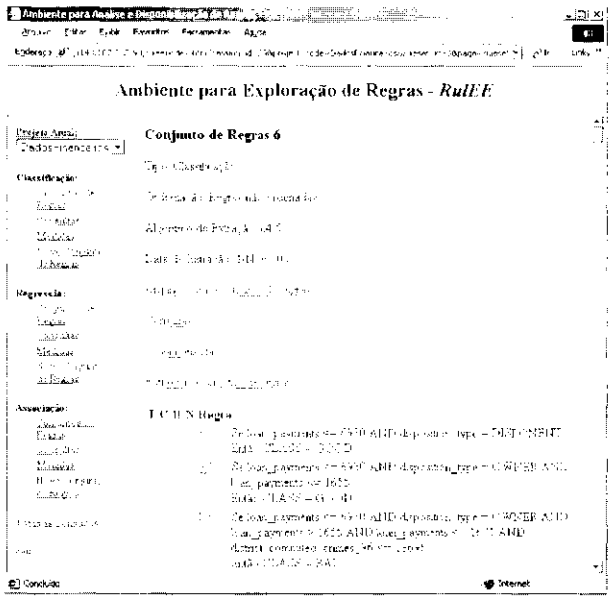

Figura 5.9: Visualização de um Conjunto de Regras

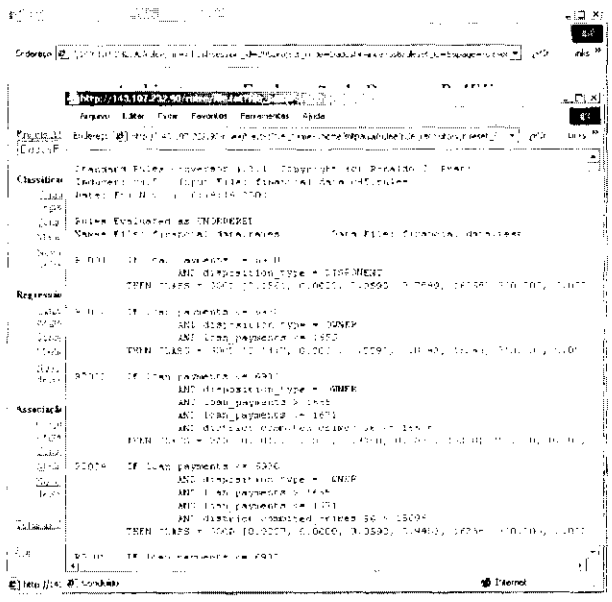

Figura 5.10: *Download* ou Visualização do Arquivo de Regras Original

nham sido inseridos no ambiente juntamente com o arquivo de regras, estes ficam disponíveis para visualização e *download*.

***Link* Consultas** Por este *link*, o usuário tem acesso à tela com a listagem das consultas realizadas sobre o conjunto de regras criadas por ele e aquelas criadas por outros usuários e que estão com *status* igual a Publicada.

***Link* Nova Consulta** *Link* para acesso à tela para edição de uma nova consulta sobre o conjunto de regras acessado.

***Link* Remover Conjunto de Regras** *Link* utilizado para remover o conjunto de regras apresentado. Após clicar neste *link*, é apresentada ao usuário uma tela requisitando uma confirmação da remoção.

***Link* Adicionar Comentário** Ao visualizar o conjunto de regras, ficam disponíveis ao usuário todos os comentários inseridos para esse conjunto de regras. Pelo *link* Adicionar Comentário, é apresentada ao usuário uma tela na qual ele pode adicionar algum comentário sobre o conjunto de regras. Na Figura 5.11 é apresentada a tela pela qual o usuário insere um novo comentário sobre um conjunto de regras.

***Link* Identificador da Regra** Para cada regra do conjunto de regras apresentado, é disponível um *link* no identificador da regra para acesso às informações referentes à mesma.

Visualização de Informações sobre uma Regra

Na Figura 5.12 pode ser observada a tela de apresentação de uma regra. Esta tela apresenta o identificador da regra, o identificador do conjunto de regras ao qual ela pertence, o número dela dentro do arquivo de dados e a própria regra. Além disso, é apresentado ao usuário uma tabela contendo indicativos fornecidos pelos usuários quanto a Interessabilidade, Compreensibilidade, Utilidade e Novidade para a regra apresentada. Nesta tabela, são apresentados o número de usuários que forneceram valores para cada índice da regra, o valor médio de cada índice e o valor inserido pelo usuário atual. Caso ele não tenha inscrido nenhum valor, então a última linha da tabela fica em branco.

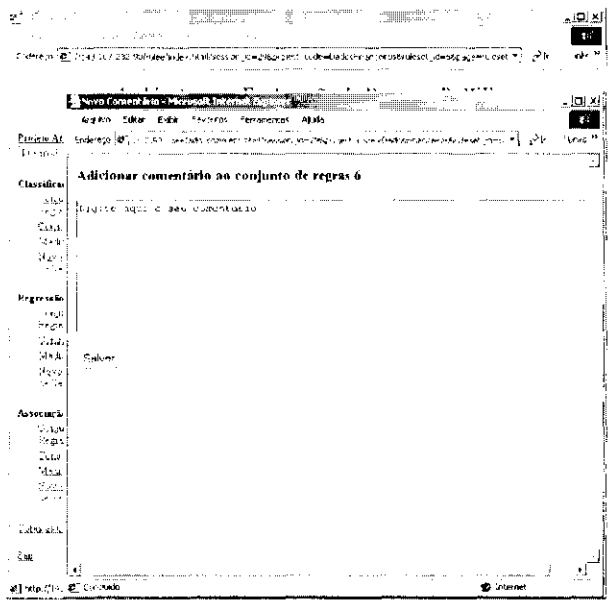

Figura 5.11: Inserção de um Novo Comentário

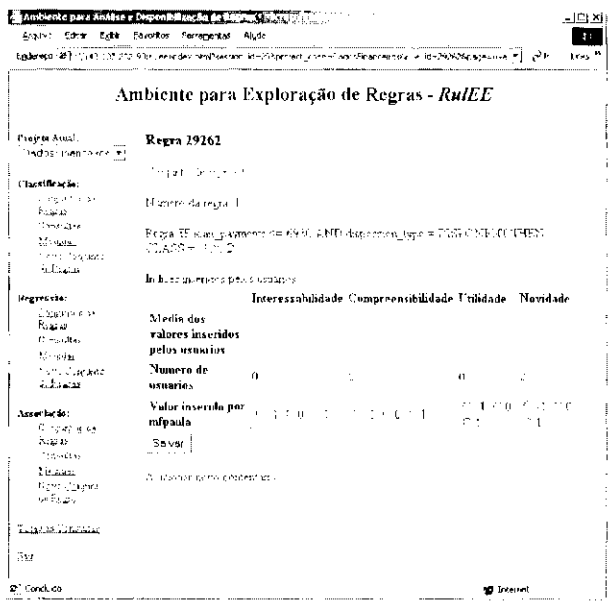

Figura 5.12: Visualização de Informações sobre uma Regra

Listagem de Consultas

Na Figura 5.13 é apresentada a tela com a listagem de todas as consultas criadas no projeto acessado pelo usuário. Deve-se destacar que esta tela apresenta todas as consultas do projeto, independente do tipo do conjunto de regras a que ela se refere e se ela apresenta ou não algum tipo de erro. Assim, são apresentadas também as consultas com *status* Inválida. Na apresentação das consultas específicas de cada tipo de conjunto de regras, são apresentadas aquelas em que o Módulo de Gerenciamento do ambiente pode identificar o conjunto de regras ao qual a consulta pertence. Em alguns casos, consultas com *status* Inválida são apresentadas somente na lista total de consultas.

Na listagem das consultas são apresentados o identificador da consulta, contendo um *link* para o acesso a informações referentes à consulta, o texto correspondente à consulta, o indicativo se a consulta está publicada, se está válida e o código do usuário que a inseriu. No final da lista é apresentado um *link* para a criação de uma nova consulta.

Visualização de Consulta

A tela de visualização de consultas está exemplificada na Figura 5.14. Como pode ser observado, esta tela apresenta o código identificador da consulta, a consulta inserida pelo usuário, indicativos se a consulta está válida e publicada, e os dados referentes à consulta em formato de tabela. Na tela de visualização de consultas estão disponíveis alguns *links* contendo funcionalidades adicionais, listadas a seguir:

Acesso aos arquivos de dados e descrição dos dados da consulta Pelos *links* Arquivo de Dados e Arquivo de Descrição dos Dados é possível a visualização e *download* dos dados referentes à consulta no formato padrão. Na Figura 5.15 é apresentado um exemplo contendo os dados da consulta e o arquivo de descrição.

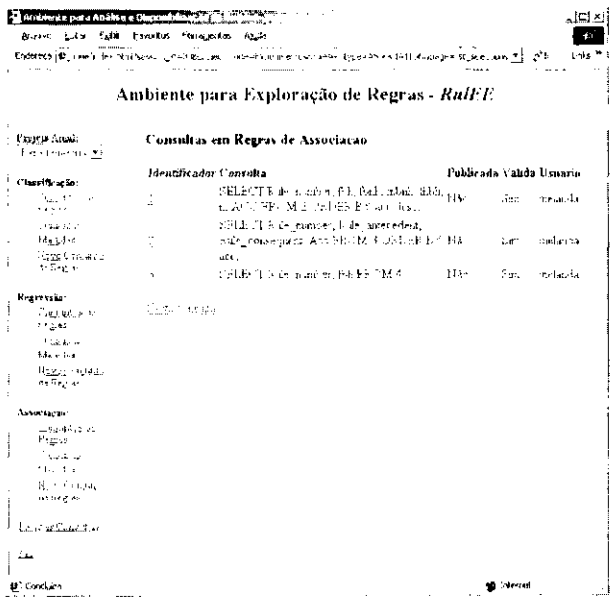

Figura 5.13: Listagem de Consultas

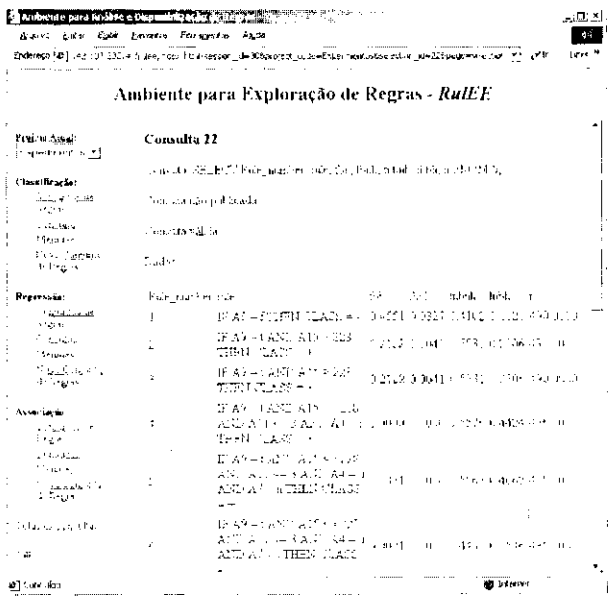

Figura 5.14: Visualização de Consulta

***Link* Atualizar Dados** Ao clicar nesse *link*, os valores das medidas para as regras do conjunto de regras utilizado na consulta são removidos da base de dados e toda a busca por esses valores das medidas é realizada novamente. Após isso, a consulta é apresentada novamente ao usuário contendo os dados atualizados. Como conseqüência

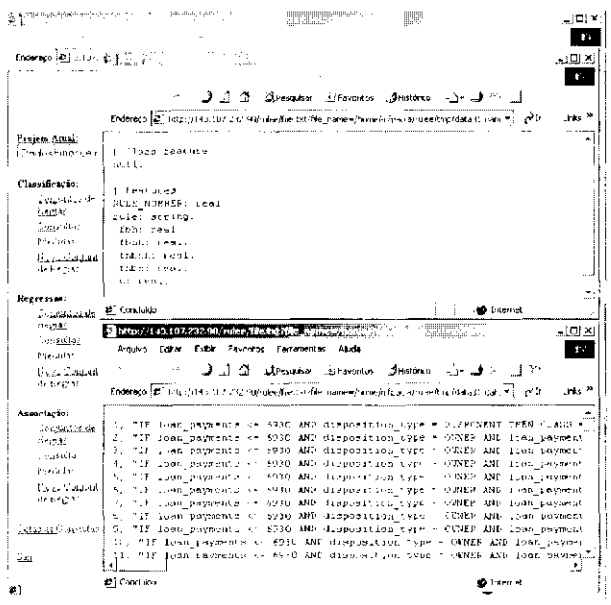

Figura 5.15: Arquivos de Descrição e de Dados Referentes à Consulta

da utilização dessa funcionalidade, todas as consultas referentes ao mesmo conjunto de regras utilizado pela consulta cujos dados foram atualizados se tornam inválidas.

***Link* Analisar Consulta** Quando uma consulta está com *status* Inválida, o *link* Analisar Consulta fica disponível ao usuário. Ao clicar neste *link*, a consulta é processada novamente para verificar sua validade. Caso nenhum erro ocorra, a consulta é apresentada ao usuário com *status* Válida. Conforme citado anteriormente, quando o usuário utiliza a função Atualizar Dados, muitas consultas podem se tornar inválidas, sendo necessário analisá-las novamente.

***Link* Editar Consulta** *Link* de acesso à tela para alteração da consulta atual.

***Link* Criar uma Cópia** Este *link* permite ao usuário criar uma cópia da consulta atual, permitindo realizar alguma alteração na mesma sem afetar a consulta original.

Edição de Consulta

A interface do Ambiente *RulEE* apresenta dois modos para criação e edição de consultas, detalhados a seguir:

Modo Simples A edição de consultas no modo simples foi desenvolvida visando facilitar a criação de consultas por usuários não experientes no ambiente. Por este modo de edição, não é possível a utilização de todas as funcionalidades disponíveis, descritas na Seção 5.2.6, mas a edição das consultas é consideravelmente facilitada. Conforme

pode ser observado na Figura 5.16, a tela para edição de consultas no modo simples apresenta alguns campos para a entrada das informações da consulta:

Colunas Neste campo devem ser inseridas as colunas a serem visualizadas nos dados finais referentes à consulta. Dentre as colunas que podem ser utilizadas neste campo, encontram-se *RULE*, referenciando a regra completa, *RULE_NUMBER*, referenciando o número da regra dentro do conjunto de regras, e os nomes das medidas disponíveis para o tipo do conjunto de regras, por exemplo, *Acc* referenciando a precisão da regra. Uma lista contendo as colunas disponíveis para utilização se encontra ao lado do campo Colunas.

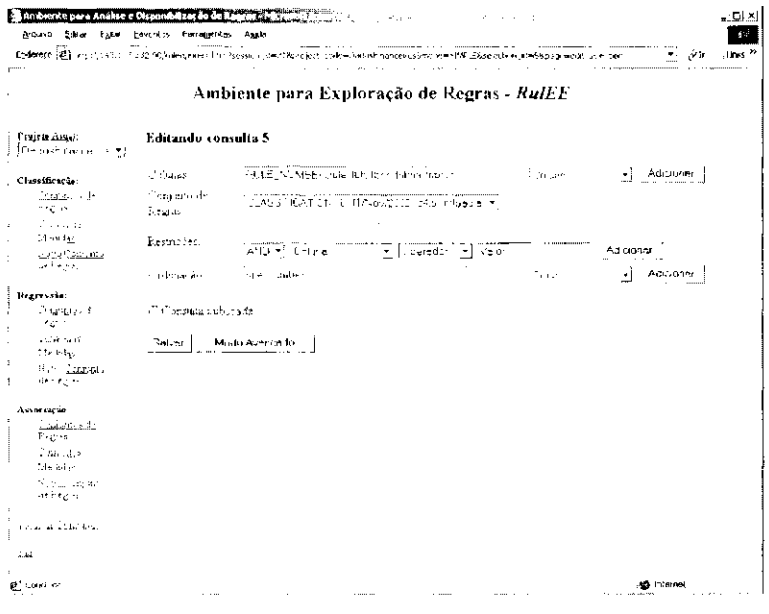

Figura 5.16: Edição de Consultas - Modo Simples

Conjunto de Regras Neste campo deve ser selecionado o conjunto de regras do qual a seleção fará a leitura das regras a serem apresentadas. Visando facilitar a identificação do conjunto de regras, cada elemento deste campo apresenta as seguintes informações: tipo do conjunto de regras, identificador, data de extração, algoritmo e usuário.

Restrições Neste campo devem ser inseridas as restrições a serem aplicadas às regras visando verificar quais delas devem ser selecionadas pela consulta. Neste campo podem ser utilizadas restrições sobre quaisquer medidas disponíveis para o conjunto de regras utilizado. Abaixo do campo Restrições existe alguns campos para auxiliar o usuário na elaboração das restrições. Por estes campos, o usuário

pode selecionar a coluna sobre a qual deseja aplicar a restrição, o operador a ser utilizado e o valor de teste.

Ordenação Neste campo, deve ser especificada a lista de colunas para ordenação das regras, podendo ser utilizadas quaisquer colunas disponíveis para o tipo do conjuntos de regras utilizado.

Consulta Publicada Caso seja do interesse do usuário que a seleção a ser criada esteja disponível para acesso por outros usuários, ele deve selecionar o campo Consulta Publicada.

Durante a edição da consulta, caso o usuário queira utilizar o modo avançado, ele pode clicar no botão Modo Avancado.

Modo Avançado Pela edição no modo avançado, o usuário deve especificar completamente a consulta a ser enviada ao ambiente, sem nenhum auxílio. A consulta dever ser especificada conforme a descrição da sintaxe das consultas fornecida na Seção 5.2.6. A Figura 5.17 apresenta um exemplo da tela para edição avançada de consultas. Como pode ser observado, existe um campo no qual o usuário deve fornecer a consulta, o campo do tipo *check* que indica se ela deve ficar disponível aos usuários e o botão para salvá-la no ambiente.

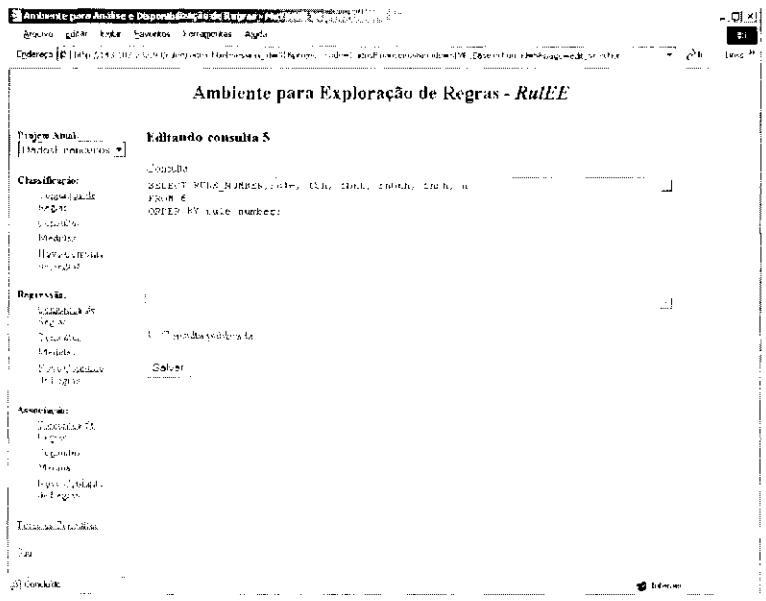

Figura 5.17: Edição de Consultas - Modo Avançado

Após editar a consulta, ao clicar no botão Salvar, o ambiente faz a análise da consulta inserida e, caso nenhum erro ocorra, é apresentada a tela para visualização da consulta

previamente editada. Caso algum erro ocorra, uma mensagem é mostrada ao usuário e a interface continua na tela de edição de consultas.

Mais detalhes sobre a utilização das consultas podem ser obtidos na Seção 5.2.6.

Listagem de Medidas

Para cada tipo de conjunto de regras tratado pelo ambiente existe disponível um conjunto de medidas cujos valores podem ser requisitados para as regras. Na Figura 5.18 é apresentada a tela com a listagem das medidas disponíveis para classificação. A listagem das medidas de regressão e associação é semelhante.

Na lista de medidas, são apresentadas as seguintes informações:

- Nome da medida;
- Tipo, contendo os possíveis valores:

  OBJECTIVE para medidas objetivas,

  SUBJECTIVE para medidas subjetivas e

  INFORMATION para medidas que representam informações sobre a regra, como o número de exemplos que ela cobre;
- Tipo de retorno (inteiro ou real);
- Escopo da medida, isto é, se ela é válida para regras ou para conjuntos de regras.

Para cada medida, existe um *link* para acesso a informações adicionais.

Informações sobre Medidas

Ao buscar por informações sobre uma determinada medida, o ambiente apresenta a tela exemplificada na Figura 5.19. As seguintes informações são apresentadas nesta tela: nome da medida, tipo da medida, tipo do conjunto de regras a que ela se refere, tipo de retorno, escopo, como ela é obtida no ambiente e uma descrição.

Na próxima seção é apresentada a sintaxe para a criação das consultas no Repositório de Regras e Medidas.

5.2.6 Utilização de Consultas

Com o objetivo de possibilitar um acesso facilitado ao conjunto de regras e valores de medidas, no Ambiente *RulEE* foram criadas as consultas sobre regras e medidas. A elaboração das consultas visa também o auxílio à análise dos conjuntos de regras, permitindo a

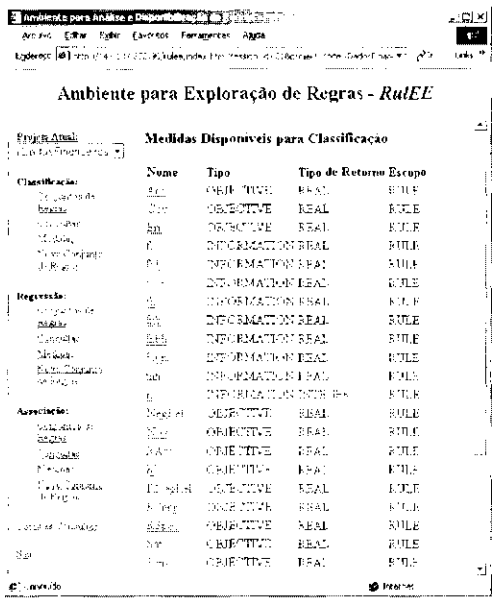

Figura 5.18: Listagem de Medidas

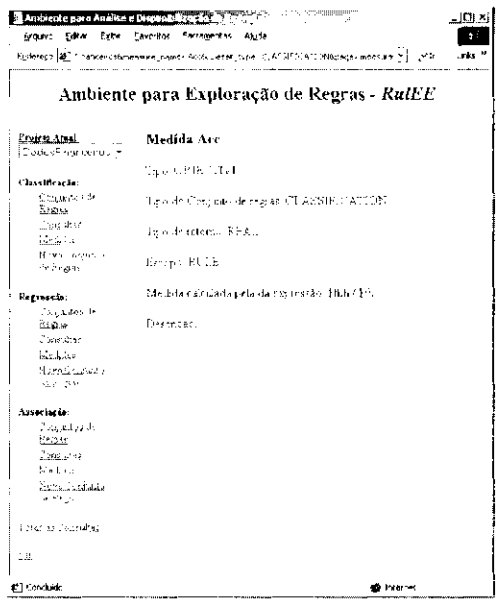

Figura 5.19: Informações sobre Medidas

recuperação das regras juntamente com os valores de medidas para o tipo de regra tratado, apresentando a possibilidade de se aplicar restrições ou filtros nas regras com relação aos valores das medidas e sobre os atributos presentes nas condições do antecedente, do conseqüente e na regra como um todo. As consultas possibilitam também a ordenação do conjunto de regras e o agrupamento do resultado, como uma cláusula SQL *GROUP BY*. A interface do Ambiente *RulEE* é iterativa para edição e visualização dos resultados das consultas.

Após criada uma consulta, ela recebe um identificador único e é armazenada no ambiente. Desta maneira, a consulta pode ser visualizada futuramente pelo usuário que a criou ou pelos outros usuários do projeto, caso ela esteja publicada. É possível também a alteração e remoção da consulta pelo usuário que a criou, e sua cópia por qualquer usuário que possa visualizá-la. Os resultados podem ser visualizados pela interface do ambiente e pode ser feito o *download* de um arquivo contendo o conjunto de dados resultante da consulta visando facilitar a utilização desses dados em outra ferramenta para sua análise.

A implementação das consultas na base de dados está baseada na classe *Selection* que implementa o processamento da *string* correspondente à consulta e faz a seleção das regras e dos valores das medidas na base de dados do ambiente. Neste processamento, é gerada uma nova consulta adequada a ser executada no Repositório de Regras e Medidas, de forma que retorne os resultados esperados pelo usuário. Conforme a arquitetura do ambiente, a

classe de seleção de regras é utilizada pela interface do ambiente para edição das consultas, podendo ser utilizada também em *scripts PERL* escritos pelos usuários.

As consultas a serem realizadas na base de dados foram projetas de forma semelhante às consultas SQL, sendo expressões na forma:

```
SELECT coluna[, coluna]*
 FROM identificador_do_conjunto_de_regras
[WHERE expressão_de_restrição]
[GROUP BY coluna[, coluna]*
[HAVING expressão_de_agrupamento]]
[ORDER BY coluna[, coluna]*];
```

Na cláusula *SELECT* devem ser especificadas as colunas a serem apresentadas no resultado da consulta, como em uma consulta SQL. Dentre as colunas que podem ser utilizadas, encontram-se o antecedente, o conseqüente, a regra completa e as medidas disponíveis para o conjunto de regras. Na cláusula *FROM* deve ser especificado o identificador do conjunto de regras do qual as regras serão selecionadas.

Na cláusula *WHERE* devem ser especificadas as restrições para a seleção das regras a serem apresentadas no resultado final. Assim, esta cláusula deve apresentar uma expressão booleana, como em uma cláusula *WHERE* SQL, utilizando na aplicação das restrições o antecedente, o conseqüente, a regra completa e as medidas disponíveis para o conjunto de regras selecionado.

Como em uma consulta SQL, o resultado de uma consulta no ambiente pode ser agrupado por alguma coluna. Isto permite ao usuário retornar sumarizações de valores de medidas, como o valor médio, mínimo ou máximo de uma determinada medida pela utilização de comandos *SUM*, *AVG*, *MAX*, *MIN* e *COUNT*. Devem ser selecionadas na cláusula *GROUP BY* as colunas pelas quais o resultado de uma consulta deve ser agrupado. Ao se utilizar uma cláusula *GROUP BY*, é possível aplicar restrições sobre o resultado do agrupamento. Por exemplo, ao se utilizar *HAVING count(*) > 2*, é retornado ao usuário somente os valores agrupados nos quais o número de registros em cada linha agrupada é maior que dois.

Na cláusula *ORDER BY* deve ser especificada a lista de colunas pela qual o resultado deve ser ordenado. O resultado da consulta pode ser ordenado pelo valor de qualquer medida de avaliação.

Na implementação atual do ambiente, o processamento das consultas não é realizado de maneira completa, isto é, não é feita uma análise léxica e sintática das consultas escritas pelos usuários. As consultas são processadas visando a geração de uma nova consulta a ser executada na base de dados do ambiente equivalente à fornecida pelo usuário.

Alguns exemplos de consultas realizadas no Ambiente *RulEE* podem ser encontradas no estudo de caso apresentado no Capítulo 6.

5.3 Medidas Disponíveis

Na implementação atual do Ambiente *RulEE*, algumas medidas estão disponíveis para utilização nas consultas. Para os conjuntos de regras de classificação, regressão e associação estão disponíveis para utilização os valores da tabela de contingência, que são obtidos nos arquivos de descrição dos conjuntos de regras, não sendo, portanto, calculados no ambiente. Os valores da tabela de contingência podem ser acessados utilizando os identificadores fbh, $fnbh$, $fbnh$, $fnbnh$, n, fb, fnb, fh, fnh referenciando os valores da matriz f_{bh}, $f_{\overline{b}h}$, $f_{b\overline{h}}$, $f_{\overline{b}\overline{h}}$, n, f_b, $f_{\overline{b}}$, f_h, $f_{\overline{h}}$, respectivamente.

É importante destacar que os valores da tabela de contingência de cada regra de um conjunto de regras de regressão são obtidos segundo métodos definidos por Pugliesi, Dosualdo, & Rezende (2003), conforme foi destacado na Seção 4.2.3.

Todas as outras medidas disponibilizadas no ambiente são calculadas a partir dos valores da tabela de contingência. Dentre as medidas disponibilizadas estão as definidas pelas Equações 3.5 a 3.22, apresentadas no Capítulo 3. Estas medidas podem ser acessadas no ambiente através do uso dos identificadores equivalentes aos nomes das medidas definidas em suas respectivas equações.

Atualmente, todas as medidas disponíveis no Ambiente *RulEE* são medidas objetivas. Trabalhos futuros referentes ao ambiente poderão estar relacionados com a disponibilização de medidas subjetivas.

5.4 Considerações Finais

Este capítulo abordou a implementação do Ambiente *RulEE*, que se apresenta como uma instanciação de um Ambiente para Exploração de Regras proposto no Capítulo 4. Como pode-se observar, o Ambiente *RulEE* apresenta algumas características importantes para facilitar a análise de regras de associação, regressão e classificação, como a disponibilidade de diversas medidas de avaliação e a possibilidade de se aplicar restrições às medidas para seleção e ordenação das regras.

Neste capítulo, primeiramente, foi apresentada a arquitetura geral do Ambiente *RulEE*, detalhando seus principais módulos e repositórios. Foram então apresentados alguns elementos do ambiente, como sua base de dados, a biblioteca de classes e sua interface. Também foram abordados alguns tópicos sobre como são obtidos os valores das medidas, a sintaxe das consultas no Repositório de Regras e Medidas e sobre quais medidas estão disponíveis para utilização pelos usuários. Mais detalhes sobre a implementação do Ambiente *RulEE* pode ser encontrada em (Paula & Rezende 2003).

No próximo capítulo é apresentado um estudo de caso utilizando dados financeiros em que são utilizadas as consultas do ambiente para mostrar algumas características importantes para a análise de regras. No próximo capítulo, a utilização de consultas pode ser melhor compreendida pelos exemplos utilizados.

Capítulo 6

Estudo de Caso

6.1 Considerações Iniciais

O projeto do Ambiente *RulEE* visa facilitar a exploração dos resultados obtidos no processo de Mineração de Dados. Com objetivo de apresentar algumas das funcionalidades deste ambiente, foi desenvolvido um estudo de caso no qual foi utilizado um conjunto de dados financeiros disponibilizado para o *Discovery Challenge*, realizado como parte do PKDD'99 (*European Conference on Principles and Practice of Knowledge Discovery in Databases*)[1].

Deve-se ressaltar que o objetivo principal deste estudo não é atingir todas as metas propostos pelo *Discovery Challenge*, mas a utilização de dados desse domínio para mostrar algumas funcionalidades do Ambiente *RulEE* na análise e disponibilização das regras descobertas, exemplificar a utilização das sintaxes padrão para dados e regras de classificação e a utilização dos *scripts* do Ambiente Discover para conversão das regras.

Para exemplificar a utilização do Ambiente *RulEE*, é apresentado, na Seção 6.2, todo o processo de Extração de Conhecimento e, na Seção 6.3, a disponibilização dos resultados utilizando o *RulEE*.

6.2 A Extração de Regras em Dados Financeiros

Como mostrado na Figura 2.1, o processo de Mineração de Dados é composto pelas etapas de Conhecimento do Domínio e Identificação do Problema, Pré-Processamento, Extração de Padrões e Pós-Processamento. Para uma melhor compreensão do domínio, do conjunto de dados e das regras encontradas, nesta seção é descrita a execução destas etapas com os dados do domínio especificado.

[1] http://lisp.vse.cz/pkdd99/Challenge/ (último acesso: 17/02/2003)

6.2.1 Conhecimento do Domínio e Identificação do Problema

Os dados disponibilizados para o *Discovery Challenge* foram fornecidos por alguma entidade financeira e o objeto principal da análise desses dados é encontrar novos conhecimentos visando melhorar a qualidade dos serviços oferecidos pela entidade aos seus clientes. Os gerentes desta entidade querem determinar quais as características dos clientes bons, a quem desejam oferecer serviços adicionais, e as características dos clientes ruins, os quais devem ser observados com cautela para evitar futuras perdas.

A base de dados fornecida é constituída de um conjunto de tabelas contendo informações sobre contas. Dentre essas informações encontram-se dados sobre empréstimos, pagamentos, transações, cartões de crédito, cliente e informações sobre a região onde o cliente reside. Na Figura 6.1 pode ser observado o diagrama físico da base de dados fornecida, sendo apresentadas as colunas e as chaves de cada tabela. Na Tabela 6.1, pode ser encontrada uma pequena descrição de cada tabela da base de dados, assim como o número de registros existentes em cada uma.

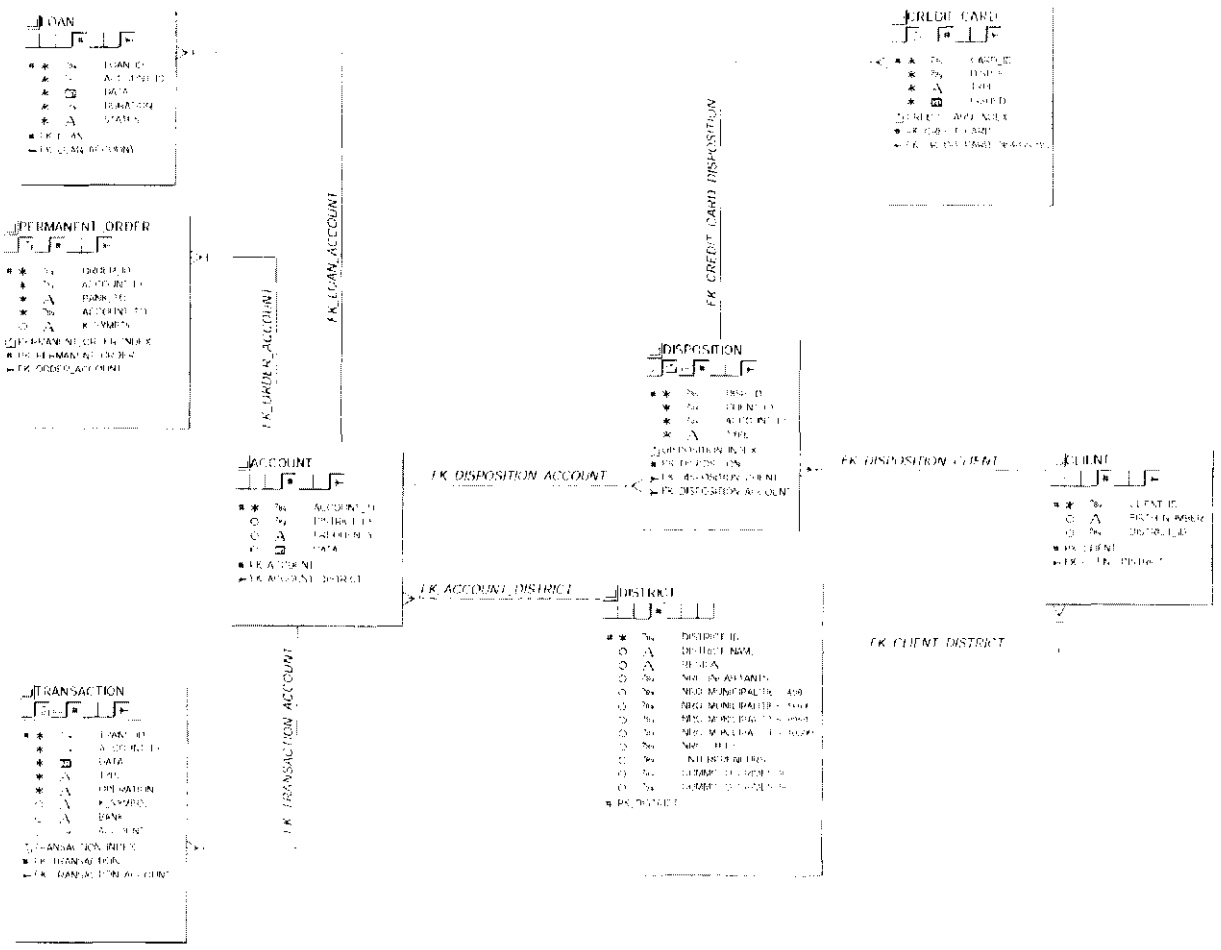

Figura 6.1: Diagrama da Base de Dados Financeira

Nome da Tabela	Número de Registros na Tabela	Descrição
account	4.500	Contém características estáticas de cada conta.
loan	682	Cada registro representa um empréstimo relacionado a uma determinada conta.
permanent_order	6.471	Os registros apresentam características das ordens de pagamentos.
transaction	1.056.320	Cada registro representa uma transação de uma determinada conta.
credit_card	892	Contém os cartões de créditos relacionados às contas.
disposition	5.369	Os registros apresentam os direitos de cada cliente para operar sua conta.
district	77	Cada registro apresenta informações demográficas das regiões em que os clientes residem.
client	5.369	Apresenta as características dos clientes.

Tabela 6.1: Descrição das Tabelas da Base de Dados Financeira

Cada conta de cliente existente na base de dados possui características estáticas, correspondentes às colunas da tabela *account*, como por exemplo sua data de criação, e características dinâmicas, como pagamentos debitados ou creditados e saldos, correspondentes aos registros das tabelas *permanent_order* e *transaction*. A tabela *client* descreve características dos clientes que podem manipular as contas. Cada cliente pode possuir muitas contas e uma conta pode ser manipulada por vários clientes. A listagem dos clientes e suas respectivas contas estão presentes na tabela *disposition*. As tabelas *loan* (empréstimo) e *credit_card* (cartão de crédito) armazenam informações referentes a serviços que o banco pode oferecer a seus clientes.

Existem várias abordagens ou objetivos específicos para se realizar a análise dos dados. Por exemplo, pode-se realizar uma análise descritiva, visando identificar relações não triviais ou não previstas, ou realizar uma análise preditiva, ao analisar os clientes que possuem ou não cartões de crédito, visando identificar possíveis novos clientes para oferecer o serviço. Neste estudo de caso, o objetivo da análise dos dados foi verificar os empréstimos oferecidos pelo banco visando identificar as características dos clientes para os quais futuros pedidos de empréstimos poderão ser aceitos e características dos clientes que apresentaram problemas no pagamento de algum empréstimo passado e, portanto, o pedido de um novo empréstimo deve ser analisado com mais cautela.

Com um conhecimento inicial sobre o problema e a especificação do objetivo da análise dos dados, pode-se iniciar o Pré-processamento.

6.2.2 Pré-Processamento

Os dados desse domínio foram fornecidos em formato texto juntamente com a descrição do modelo apresentado na Figura 6.1. Para facilitar seu Pré-processamento, foram criadas as tabelas especificadas na seção anterior em uma base de dados MySql.

Para que seja possível a utilização dos algoritmos de Aprendizado de Máquina, é necessário que os dados estejam em um formato atributo-valor, conforme definido na Seção 2.2.2. Desta maneira, foi necessária a junção dos dados fornecidos em uma única tabela. No Pré-processamento, as seguintes atividades foram realizadas:

Junção das Tabelas A definição utilizada para categorizar cada cliente como bom ou ruim foi realizada utilizando os dados existentes na coluna *status* da tabela *loan*. Nesta definição, um cliente é considerado bom se ele nunca teve problemas com o pagamento dos empréstimos, sendo considerado ruim caso contrário. Assim, para a extração de padrões foram utilizados somente os dados dos clientes que, em algum momento, obtiveram crédito, ou seja, foi realizada a junção das tabelas utilizando somente os dados de clientes e contas que apresentam algum registro na tabela *loan*.

Os atributos das tabelas correspondentes aos conjuntos de dados gerados pelas junções foram nomeados utilizando o seguinte padrão: *nome da tabela original_nome do atributo original*. Por exemplo, os dados do atributo *account_frequency* nas tabelas geradas correspondem aos dados do atributo *frequency* da tabela *account*.

Transformação de Valores de Atributos Visando facilitar a compreensão dos dados e dos resultados da execução dos algoritmos, alguns atributos tiveram seus valores transformados, conforme a descrição fornecida juntamente com os dados e apresentado na Tabela 6.2.

Criação de Novos Atributos No Pré-processamento, novos atributos foram criados com base nos dados originais.

Conforme especificado na etapa de Conhecimento do Domínio e Identificação do Problema, o objetivo da análise dos dados é prever se um cliente poderá apresentar problemas no pagamento de um empréstimo, foi necessária a criação da coluna *loan_predict* como sendo o atributo meta do conjunto de dados. O atributo *loan_predict* teve seu valor definido como segue: *loan_predict* = *GOOD* se *loan_status* = *A* (contrato finalizado sem problemas de pagamento) ou *loan_status* = *C* (contrato não finalizado e pagamento em dia) e *loan_predict* = *BAD* se loan_status = *B* (contrato finalizado e em-

Atributo	Valores Originais	Novos Valores
account_frequency	*POPLATEK MESICNE*	*monthly*
	POPLATEK TYDNE	*weekly*
	POPLATEK PO OBRATU	*after transaction*
permanent_order_ksymbol	*POJISTNE*	*insurrance*
	SIPO	*household*
	LEASING	*leasing*
	UVER	*loan*
transaction_type	*PRIJEM*	*credit*
	VYDAJ	*withdrawal*
transaction_operation	*VYBER KARTOU*	*credit card withdrawal*
	VKLAD	*credit in cash*
	PREVOD Z UCTU	*collection from another bank*
	VYBER	*withdrawal in cash*
	PREVOD NA UCET	*remittance to another bank*
transaction_k_symbol	*POJISTNE*	*insurrance payment*
	SLUZBY	*payment for statement*
	UROK	*interest credited*
	SANKC. UROK	*interest if negative balance*
	SIPO	*household*
	DUCHOD	*old-age pension*
	UVER	*loan payment*

Tabela 6.2: Transformação de Valores de Atributos da Base de Dados Financeira

préstimo não pago corretamente) ou loan_status = *D* (contrato não finalizado e cliente em débito).

Além do atributo meta *loan_predict*, foram também criados os atributos definidos na Tabela 6.3.

Atributo	Descrição
client_age	Idade do cliente em meses. Valor calculado a partir da coluna *client_birth_number*, cujo valor determina a data de nascimento e o sexo do cliente.
client_sex	Sexo do cliente. Valor também obtido a partir da coluna *client_birth_number*.
account_months_before_loan	Número de meses entre a criação da conta do cliente e a obtenção do empréstimo.
credit_card_months_before_loan	Número de meses entre a obtenção de um cartão de crédito e a obtenção do empréstimo.
transaction_months_before_loan	Número de meses entre a transação e a obtenção do empréstimo.

Tabela 6.3: Novos Atributos da Base de Dados Financeira

Geração dos Arquivos de Dados na Sintaxe Padrão No final do Pré-processamento dos dados, eles foram convertidos para a sintaxe padrão do Ambiente Discover. Para

ambos os conjuntos de exemplos criados, foram gerados arquivos de treinamento e de teste.

Após os dados terem sido preparados e formatados na sintaxe padrão, a próxima etapa do processo de Mineração de Dados pode ser iniciada.

6.2.3 Extração de Padrões

De acordo com o objetivo definido, este estudo de caso trata de um problema de classificação. Assim, para a extração de padrões foi utilizado o algoritmo de Aprendizado de Máquina C4.5, sendo necessária a conversão dos conjuntos de dados para o formato de entrada do algoritmo C4.5 utilizando os *scripts* de conversão disponíveis no Discover.

Neste estudo de caso foram utilizados dois conjuntos de dados. Em um primeiro instante, foi gerado o conjunto de dados *Financial_Data_1*. Nesse primeiro conjunto, foram utilizados os dados de todas as tabelas da base de dados na geração da tabela no formato atributo-valor. Após executada a junção, a tabela correspondente aos dados *Financial_Data_1* apresentou um total de 161.314 registros com 33 atributos. Para permitir uma execução mais rápida do algoritmo C4.5, foi criada uma amostra desse conjunto, contendo 48.382 registros, aproximadamente 30% dos dados originais. Após a divisão do conjunto *Financial_Data_1*, o arquivo de dados de treinamento ficou com 32.114 exemplos e 16.268 exemplos foram utilizados para teste. A árvore de decisão induzida a partir desses dados apresentou um total de 105 regras e uma taxa de erro estimada de 0.5%.

Ao verificar a árvore de decisão induzida, observou-se que não foram utilizados os atributos da tabela *transaction* para realizar a classificação. Assim, em uma segunda iteração do processo, foi gerado o segundo conjunto de dados (*Financial_Data_2*) no qual não foi utilizada a tabela *transaction* na junção dos dados para a formação da tabela no formato atributo valor.

No segundo conjunto, foram utilizadas as tabelas *account*, *loan*, *permanent_order*, *credit_card*, *disposition*, *district* e *client* para a geração da tabela no formato atributo-valor. Este conjunto de dados apresentou um total de 1.841 exemplos, sendo que 1.212 foram usados para treinamento e 629 para teste. Para este conjunto de dados, foram utilizados os mesmos atributos do conjunto *Financial_Data_1*, excluindo somente os atributos referentes à tabela *transaction*. A árvore de decisão induzida extraída de *Financial_Data_2* apresentou um total de 28 regras e uma taxa de erro estimada de 7.1%.

Após a Extração de Padrões, as árvores de decisão foram processadas e inseridas no Ambiente *RulEE* para uma análise mais detalhada dos resultados, na etapa de Pós-processamento.

6.2.4 Pós-Processamento

No Pós-processamento das árvores de decisão, foram utilizados os *scripts* do Discover para sua conversão em conjuntos de regras na sintaxe padrão de classificação (Prati, Baranauskas, & Monard 2001b). Posteriormente, foram utilizados os *scripts* para cálculo dos valores da tabela de contingência para cada regra a partir dos conjuntos de dados de testes (Prati, Baranauskas, & Monard 2001a). Desta maneira, os conjuntos de regras extraídos a partir de dois conjuntos de dados puderam ser explorados no Ambiente *RulEE*. Na seção seguinte é apresentada a utilização do ambiente para a análise e disponibilização das regras encontradas no processo.

6.3 Exploração das Regras Utilizando o RulEE

Como citado anteriormente, o Ambiente *RulEE* foi desenvolvido com objetivo de auxiliar a análise e disponibilização de regras, atividade pertencente à etapa de Pós-processamento do processo de Mineração de Dados. O ambiente possibilita a execução dessa análise de maneira bastante flexível através da utilização de suas funcionalidades em *scripts* de Pós-processamento desenvolvidos por especialistas do processo de Mineração de Dados. Por outro lado, essa exploração das regras pode ser realizada mais facilmente utilizando a interface definida para o ambiente. Este tipo de interação é voltada a usuários não especialistas. Estes dois tipos de interação são apresentados nas seções seguintes.

6.3.1 Utilização do Ambiente por Scripts

Nesta seção, é exemplificada a utilização do Ambiente *RulEE*, pelas classes do Módulo de Gerenciamento, para executar as seguintes atividades: criação de uma sessão, inserção do conjunto de regras extraído do conjunto de dados *Financial_Data_1* no *RulEE* e execução de uma consulta.

Primeiramente, para que o usuário possa utilizar as classes em *scripts*, ele deve possuir senha para *login* no servidor em que o Ambiente *RulEE* está instalado, além de direito de acesso às classes e à base de dados.

Para permitir a utilização do ambiente por *scripts* é necessária a criação de uma sessão. Esta criação pode ser realizada através do *script* apresentado na Figura 6.2. O identificador

da sessão criada deve ser armazenado para os acessos seguintes ao ambiente. Ao executar o *script* para criação de sessões, a saída obtida é semelhante à apresentada na Figura 6.3.

```
use lib '/home/mfpaula/rulee/source_code';
use Session;

my ($sessao);

$sessao = Session->Login("mfpaula","senha");
print "Identificador da sessao criada: ".$sessao->{_session_id}."\n";

$sessao->Leave();
```

Figura 6.2: Exemplo de *Script* para Criação de Sessões no Ambiente *RulEE*

```
[mfpaula@felicia cap6_scripts]$ perl cap6_cria_sessao.pl
Identificador da sessao criada: 21
[mfpaula@felicia cap6_scripts]$
```

Figura 6.3: Execução do *Script* de Exemplo para Criação de Sessões

Um conjunto de regras pode então ser inserido no ambiente por um *script* semelhante ao exemplo apresentado na Figura 6.4. O resultado da execução desse *script* pode ser observado na Figura 6.5, em que é fornecido o identificador do conjunto de regras referente à arvore extraída a partir dos dados *Financial_data_1*. Este identificador deve ser utilizado para a realização das consultas referentes a este conjunto de regras.

```
use lib '/home/mfpaula/rulee/source_code';

use Ruleset;
use Rule;

$ruleset = Ruleset->NewClassificationModel("financial_data_1.rules.info",
                                           "","","","","DadosFinanceiros",21);

print "\nIdentificador do conjunto de regras: ".$ruleset->GetRulesetId();

print "\n\nRegras:\n\n";

@regras = $ruleset->GetRules();
for ($i = 0; $i < @regras; $i++)
{
        $rule = $regras[$i];
        print "R ".$rule->GetRuleNumber()."\nIF ".$rule->GetAntecedent();
        print "\nTHEN\n".$rule->GetConsequent()."\n\n";
}

$ruleset->Leave();
```

Figura 6.4: Exemplo de *Script* para Inserção do Conjunto de Regras Extraído de *Financial_Data_1*

Para exemplificar a utilização da classe para execução de consultas, é utilizado o *script* apresentado na Figura 6.6 que retorna ao usuário os valores da tabela de contingência das

```
[mfpaula@felicia cap6_scripts]$ perl cap6_new_ruleset.pl |more

Identificador do conjunto de regras: 6

Regras:

R 1
IF loan_payments <= 6930 AND disposition_type = DISPONENT
THEN
CLASS = GOOD

R 2
IF loan_payments <= 6930 AND disposition_type = OWNER AND loan_payments <= 1655
THEN
CLASS = GOOD

.
.
.
```

Figura 6.5: Resultado da Execução do *Script* de Exemplo para Inserção do Conjunto de Regras

regras do conjunto de regras inserido anteriormente. Pela execução de tal *script*, obtém-se o resultado apresentado na Figura 6.7.

```
use lib '/home/mfpaula/rulee/source_code';
use Selection;

$selection = Selection->New("select RULE_NUMBER, fbh, fbnh, fnbnh, fnbh, n from 6;",
                                                              "DadosFinanceiros",21);
$selection->AnalyseQuery();

print "\nIdentificador da consulta: ".$selection->GetSelectionId()."\n";

if ($selection->IsValid())
{
    print "\nResultado da consulta:\n\n".$selection->PrintData();
}

$selection->Leave();
```

Figura 6.6: Exemplo de *Script* para Execução de Consulta

Deve-se ressaltar que a construção desses *scripts* pode ser realizada pelos usuários com conhecimento da biblioteca de classes e da base de dados do Ambiente *RulEE*. Uma descrição completa de todas as classes e métodos existentes na biblioteca de classes e das tabelas e atributos existentes na base de dados encontra-se em (Paula & Rezende 2003).

Na próxima seção é apresentada a utilização da interface do ambiente na análise das regras descobertas neste estudo de caso. Para isto, serão apresentadas algumas consultas realizadas.

```
[mfpaula@felicia cap6_scripts]$ perl cap6_new_selection.pl |more

Identificador da consulta: 5

Resultado da consulta:

RULE_NUMBER      fbh      fbnh     fnbnh    fnbh      n
1        0.1561   0.0000   0.0590   0.7849   16268.0000
2        0.1317   0.0000   0.0590   0.8093   16268.0000
3        0.0012   0.0030   0.9380   0.0578   16268.0000
4        0.0007   0.0000   0.0590   0.9403   16268.0000
5        0.1343   0.0013   0.0577   0.8067   16268.0000
6        0.0001   0.0000   0.9410   0.0590   16268.0000
.
.
.
```

Figura 6.7: Resultado do *Script* de Exemplo para Execução de Consulta

6.3.2 Utilização do Ambiente pela Interface

Para exemplificar a utilização do Ambiente *RulEE* pela interface, foi inserido o conjunto de regras extraído a partir do conjunto de dados *Financial_Data_2* já que o número de regras extraídas deste conjunto de dados foi menor.

Para a inserção de um conjunto de regras de classificação no ambiente através da interface, após estar conectado no mesmo deve-se clicar no *link* Novo Conjunto de Regras nos *links* referentes a classificação. Na Figura 6.8 é mostrada a inserção do conjunto de regras extraídas a partir de *Financial_Data_2*. Após inserido, o conjunto de regras é apresentado ao usuário, o qual deve verificar o identificador do conjunto para permitir futuros acessos.

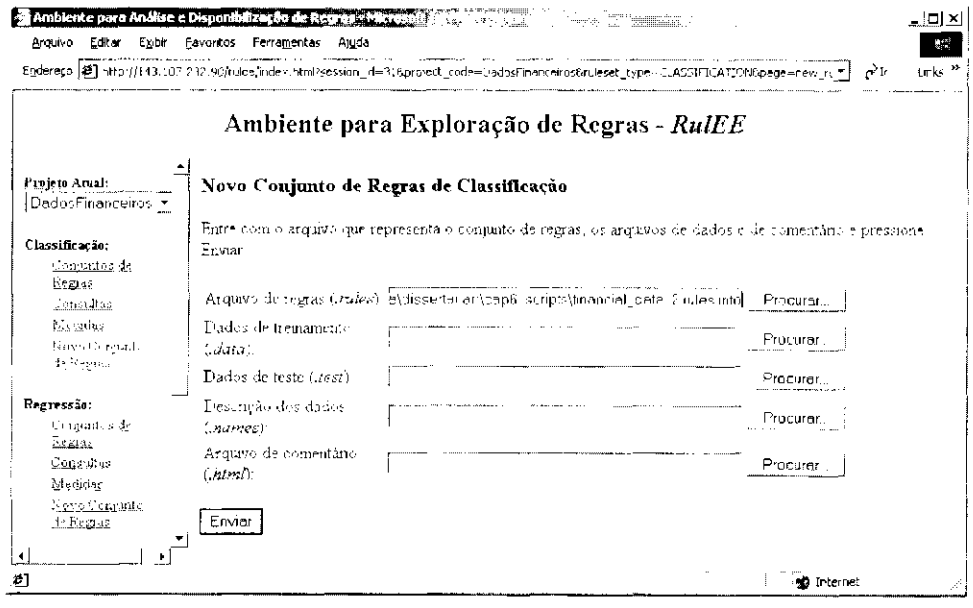

Figura 6.8: Tela para Inserção do Conjunto de Regras Extraído de *Financial_Data_2*

Pode-se observar nesta seção que a interface desenvolvida é bastante simples, uma vez que o foco principal deste trabalho foi a proposta e modelagem do Ambiente *RulEE*. Outro aspecto importante é que todas os valores das medidas utilizadas nesta seção são calculados automaticamente pelo ambiente, conforme apresentado na Seção 5.3.

A seguir são apresentadas algumas consultas para ilustrar a utilização das mesmas na exploração das regras descobertas. Nas consultas realizadas, é feita uma análise quantitativa baseada nos valores das medidas. O resultado de algumas consultas é apresentado pela interface do ambiente, enquanto que para outras consultas é apresentada somente a conclusão obtida ou a interpretação do resultado, por não ser necessária a apresentação de todas as telas. Para cada consulta, são apresentados seu objetivo ou descrição do resultado e a consulta realizada.

A consultas realizadas foram as seguintes:

1. Verificação das características dos cliente bons e ruins. Consulta realizada:

   ```
   SELECT rule_number, rule_antecedent, rule_consequent FROM 7 ORDER BY
   rule_consequent, rule_number;
   ```

   Resultado pode ser visualizado na Figura 6.9. Primeiramente são apresentadas as regras que classificam os clientes como ruins e posteriormente as que classificam os clientes como bons.

2. Dezoito regras classificam os cliente como *GOOD* e dez regras classificam os clientes como *BAD*. Consulta realizada:

   ```
   SELECT rule_consequent, count(*) FROM 7 group by rule_consequent;
   ```

   Resultado apresentado na Figura 6.10.

3. As regras apresentam uma precisão bastante alta. Consulta realizada:

   ```
   SELECT RULE_NUMBER, acc FROM 7 ORDER BY acc desc, rule_number asc;
   ```

   Parte do resultado desta consulta pode ser observado na Figura 6.11.

4. Dezessete das vinte e oito regras (aproximadamente 60%) apresentam precisão maior que 0,95. Consulta realizada:

   ```
   SELECT count(*) FROM 6 WHERE acc > 0.95;
   ```

rule_number	rule_antecedent	rule_consequent
4	loan_amount <= 249792 AND account_months_before_loan <= 6 AND account_frequency = weekly AND client_sex = F	CLASS = BAD
5	loan_amount <= 249792 AND account_months_before_loan <= 6 AND account_frequency = aftertransaction AND loan_amount <= 80340	CLASS = BAD
8	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration <= 36 AND district_enterpreneurs <= 109	CLASS = BAD
10	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration <= 36 AND district_enterpreneurs > 109 AND client_age > 48	CLASS = BAD
15	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration > 36 AND client_sex = F AND district_region = westBohemia AND district_nro_cities <= 3	CLASS = BAD
17	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration > 36 AND client_sex = F AND district_region = southBohemia AND account_months_before_loan <= 15	CLASS = BAD
19	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration > 36 AND client_sex = F AND district_region = southMoravia AND account_months_before_loan <= 10	CLASS = BAD
23	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration > 36 AND client_sex = F AND district_region = northMoravia AND district_committed_crimes_96 <= 4433 AND district_ratio_urban_inhabitants > 52	CLASS = BAD
26	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration > 36 AND client_sex = F AND district_region = Prague AND account_months_before_loan > 15 AND loan_amount > 402000	CLASS = BAD
28	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 > 3 AND account_months_before_loan <= 11	CLASS = BAD
1	loan_amount <= 249792 AND account_months_before_loan > 6	CLASS = GOOD
2	loan_amount <= 249792 AND account_months_before_loan <= 6 AND account_frequency = monthly	CLASS = GOOD
3	loan_amount <= 249792 AND account_months_before_loan <= 6 AND account_frequency = weekly AND client_sex = M	CLASS = GOOD
6	loan_amount <= 249792 AND account_months_before_loan <= 6 AND account_frequency = aftertransaction AND loan_amount > 80340	CLASS = GOOD
7	loan_amount > 249792 AND disposition_type = DISPONENT	CLASS = GOOD
9	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration <= 36 AND district_enterpreneurs > 109 AND client_age <= 48	CLASS = GOOD
11	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration > 36 AND client_sex = M	CLASS = GOOD
12	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration > 36 AND client_sex = F AND district_region = eastBoheme	CLASS = GOOD
13	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration > 36 AND client_sex = F AND district_region = centralBohemia	CLASS = GOOD
14	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration > 36 AND client_sex = F AND district_region = northBohemia	CLASS = GOOD
16	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration > 36 AND client_sex = F AND district_region = westBohemia AND district_nro_cities > 3	CLASS = GOOD
18	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration > 36 AND client_sex = F AND district_region = southBohemia AND account_months_before_loan > 15	CLASS = GOOD
21	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration > 36 AND client_sex = F AND district_region = southMoravia AND account_months_before_loan > 10	CLASS = GOOD
22	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration > 36 AND client_sex = F AND district_region = northMoravia AND district_committed_crimes_96 > 4433	CLASS = GOOD
24	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration > 36 AND client_sex = F AND district_region = northMoravia AND district_committed_crimes_96 <= 4433 AND district_ratio_urban_inhabitants <= 52	CLASS = GOOD
24	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration > 36 AND client_sex = F AND district_region = Prague AND account_months_before_loan <= 15	CLASS = GOOD
25	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 <= 3 AND loan_duration > 36 AND client_sex = F AND district_region = Prague AND account_months_before_loan > 15 AND loan_amount <= 402000	CLASS = GOOD
27	loan_amount > 249792 AND disposition_type = OWNER AND district_nro_municipalities_10000 > 3 AND account_months_before_loan <= 11	CLASS = GOOD

Figura 6.9: Regras que Classificam os Clientes como *BAD* ou *GOOD*

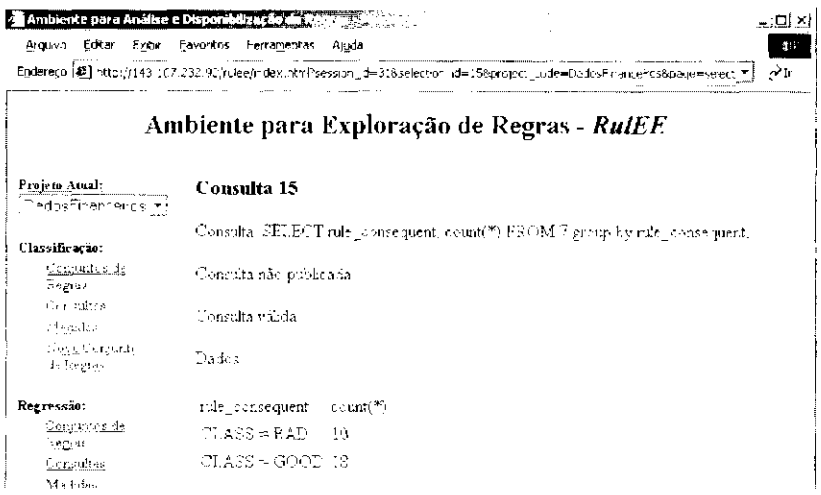

Figura 6.10: Quantidade de Regras que Classificam como *BAD* e *GOOD*

5. Pela verificação da precisão relativa, observa-se que as regras que classificam os clientes como *BAD* apresentam, em geral, uma precisão relativa alta. As regras que

classificam os clientes como *GOOD* apresentam, em geral, uma precisão relativa baixa. Consulta realizada:

```
SELECT RULE_NUMBER, racc, rule_consequent FROM 7 ORDER BY racc desc;
```

Resultado apresentado parcialmente na Figura 6.12.

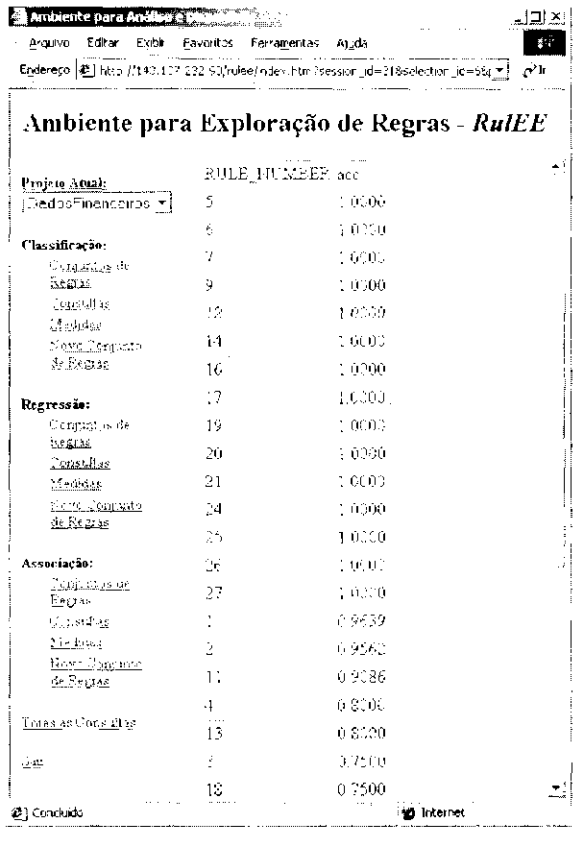

Figura 6.11: Precisão das Regras

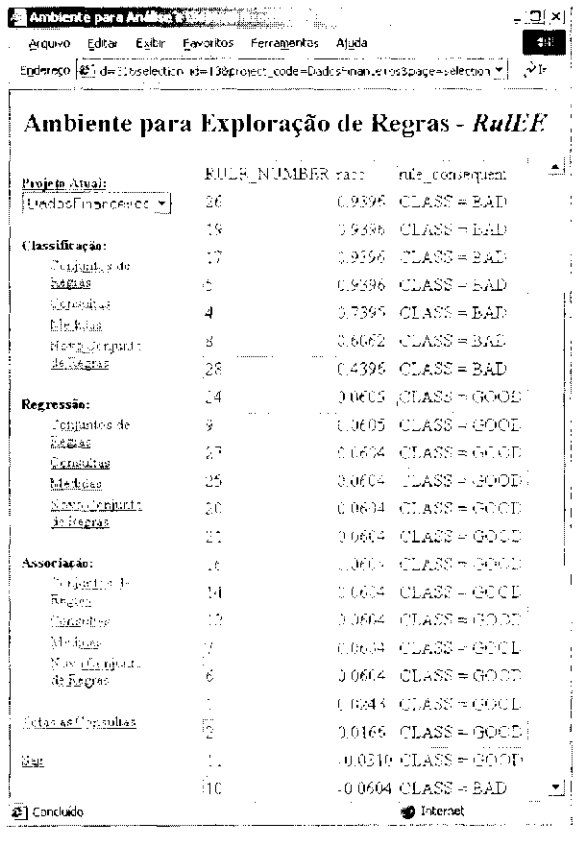

Figura 6.12: Precisão Relativa

6. A precisão relativa média das regras que classificam os clientes como *BAD* é 0.792.

   A precisão relativa média das regras que classificam os clientes como *GOOD* é 0.054. Neste resultado, foram consideradas somente as regras que apresentam precisão relativa maior que zero. Consulta realizada:

```
SELECT avg(racc), rule_consequent FROM 7 where racc > 0
group by rule_consequent;
```

7. Poucas regras cobrem uma grande quantidade de exemplos. As quatro regras com maior cobertura (aproximadamente 14% do total) cobrem aproximadamente 90% dos exemplos. Consulta realizada:

```
SELECT sum(cov), count(*) FROM 7 WHERE cov > 0.02;
```

O resultado desta consulta pode ser observado na Figura 6.13.

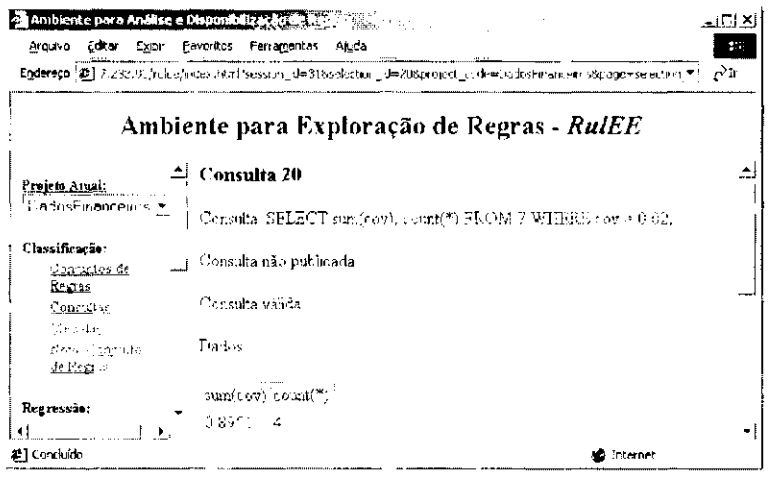

Figura 6.13: Soma da Cobertura das 4 Regras com Maior Cobertura

8. O número de exemplos classificados incorretamente é trinta e um. Consulta realizada:

```
SELECT sum(cov*n*err) FROM 7;
```

9. As dez regras que classificam os clientes como *BAD* cobrem 3% dos registros. As dezoito regras que classificam os cliente como *GOOD* cobrem 97% dos registros. Consulta Realizada:

```
SELECT rule_consequent, sum(cov), count(*) FROM 7
group by rule_consequent;
```

10. As regras selecionadas utilizando a medida Novidade são obtidas utilizando a seguinte consulta:

```
SELECT RULE_NUMBER, nov FROM 7 ORDER BY nov desc;
```

Parte do resultado desta consulta pode ser observado na Figura 6.14.

11. As regras selecionadas utilizando a medida Especificidade são obtidas utilizando a seguinte consulta:

```
SELECT RULE_NUMBER, spec FROM 7 ORDER BY spec desc;
```

Resultado apresentado parcialmente na Figura 6.15.

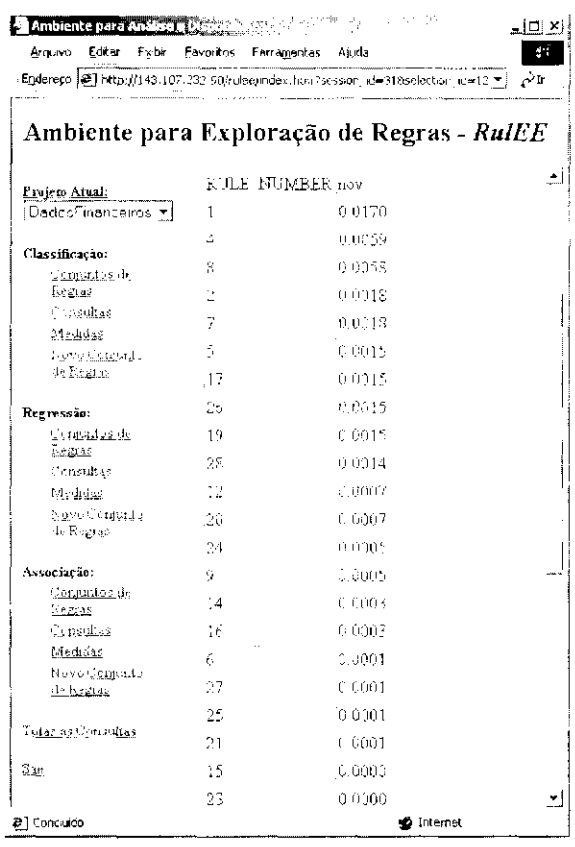

Figura 6.14: Valor da Medida Novidade para as Regras

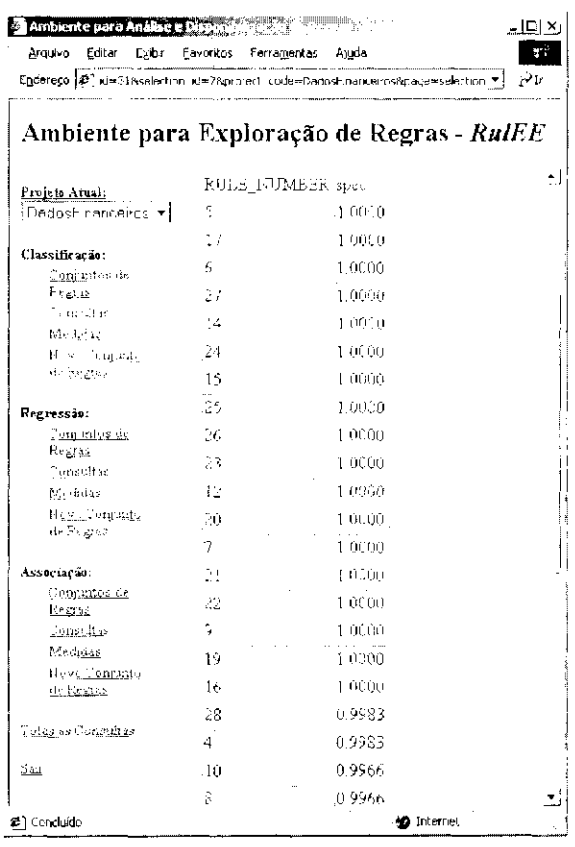

Figura 6.15: Valor da Medida Especificidade para as Regras

Essas 11 consultas foram apresentadas para exemplificar alguns tipos de análises que podem ser realizadas com o apoio do Ambiente *RulEE*.

6.4 Considerações Finais

Após a implementação do Ambiente *RulEE*, foi realizado um estudo de caso, apresentado neste capítulo, com objetivo de verificar a utilização deste ambiente inserido no contexto de um processo completo de Mineração de Dados. Desta maneira, foram apresentadas todas as etapas realizadas para a obtenção das regras, como deve-se proceder para a utilização dessas regras no ambiente e alguns exemplos de análises das mesmas.

O estudo de caso foi realizado no contexto de dados financeiros e o objetivo do processo de mineração foi encontrar regras que determinam se os clientes são bons, aos quais serviços adicionais devem ser oferecidos, ou ruins, que devem ser observados com mais cautela ao se oferecer serviços. Deve-se destacar, entretanto, que o objetivo principal deste capítulo foi mostrar algumas das funcionalidades do ambiente para exploração de regras.

Pode-se observar neste capítulo que a análise dos conjuntos de regras pode ser bastante facilitada pela utilização do Ambiente *RulEE*. Algumas características do ambiente que auxiliam na análise de regras estão a possibilidade da utilização de diversas medidas de forma conjunta, utilização de medidas para ordenação e seleção de regras além do acesso facilitado aos conjuntos de regras e aos valores das medidas por meio da interface WWW que viabiliza a disponibilização das regras.

Capítulo 7

Conclusões e Trabalhos Futuros

Com o desenvolvimento da tecnologia de Mineração de Dados e sua utilização para solucionar problemas reais em processos de tomada de decisão, observou-se que a extração de padrões não é a etapa final desse processo, tendo o Pós-processamento do conhecimento um papel fundamental no sucesso do mesmo. A utilização efetiva do conhecimento descoberto em um processo de apoio à tomada de decisão depende de fatores como a possibilidade do usuário compreender esse conhecimento, conseguir identificar o conjunto de regras mais interessantes, entre outros. Assim, a avaliação dos conjuntos de regras quanto a precisão, compreensibilidade, utilidade e interessabilidade, por exemplo, é de extrema relevância no auxílio ao usuário final para viabilizar a utilização do conhecimento descoberto. Nesse sentido, muitos trabalhos relacionados ao processo de Mineração de Dados têm sido desenvolvidos para avaliação do conhecimento descoberto.

Visando auxiliar os usuários na análise e utilização desses conjuntos de regras, foi proposto neste trabalho um Ambiente para Exploração de Regras que trata aspectos referentes à avaliação de conjuntos de regras de classificação, regressão e associação. Nesse sentido, a principal contribuição desta dissertação é o projeto e desenvolvimento de um Ambiente para Exploração de Regras denominado *RulEE* (*Rule Exploration Environment*).

O Ambiente *RulEE* apresenta uma arquitetura aberta para disponibilização de conjuntos de regras e valores de diversas medidas de avaliação. Este ambiente auxilia os usuários finais e especialistas do domínio na análise, avaliação e identificação de regras interessantes através da visualização dos conjuntos de regras e da realização de análises pela utilização de medidas. A interação com os usuários é realizada por interface baseada na WWW, visando facilitar o acesso aos conjuntos de regras. O ambiente apóia também os

usuários analistas do processo de Mineração de Dados na criação ou armazenamento de novas medidas em uma estrutura integrada, facilitando sua utilização conjunta.

O *RulEE* foi desenvolvido visando a criação de um ambiente que apresente um Repositório de Regras no qual, após sua extração, as regras são armazenadas e disponibilizadas para acesso pelos usuários do processo de Mineração de Dados. Além das regras, são armazenadas diversas medidas de avaliação para auxiliar o usuário na análise das mesmas. Identifica-se então o elemento principal do ambiente, o Repositório de Regras e Medidas.

O Repositório de Regras e Medidas armazena as regras em uma sintaxe padrão e permite que, associada a cada regra, sejam armazenadas as medidas de avaliação utilizadas para sua análise. Para cada regra é armazenada sua tabela de contingência, calculada utilizando *scripts* disponíveis no Ambiente Discover (Prati, Baranauskas, & Monard 2001b; Prati, Baranauskas, & Monard 2001a; Pugliesi, Dosualdo, & Rezende 2003; Melanda & Rezende 2003). A partir desses valores, são calculadas no Ambiente *RulEE* as medidas Precisão, Erro, Confiança Negativa, Sensitividade, Especificidade, Cobertura, Suporte, Novidade, Satisfação, Precisão Relativa, Confiança Negativa Relativa, Sensitividade Relativa, Especificidade Relativa, Precisão Relativa Ponderada, Confiança Negativa Relativa Ponderada, Sensitividade Relativa Ponderada e Especificidade Relativa Ponderada, apresentadas no *framework* (Lavrac, Flach, & Zupan 1999), e a medida *RI* (*Rule-Interest*), proposta em (Piatetsky-Shapiro 1991).

É importante destacar que o Repositório de Regras e Medidas do *RulEE* foi modelado de maneira que o ambiente seja independente das medidas de avaliação. O *RulEE* possui implementado o conjunto de medidas apresentadas anteriormente, mas os pesquisadores do processo podem acrescentar novas medidas pelo uso de medidas derivadas ou pela execução de procedimentos de Pós-processamento independentes do ambiente. Isto pode ser feito sem necessidade de alteração de código fonte, e qualquer nova medida pode ser utilizada de forma conjunta com as pré-existentes.

Neste trabalho foram identificados alguns requisitos importantes para um Ambiente para Exploração de Regras. Dentre eles, o Ambiente *RulEE* viabiliza a Interatividade e Iteratividade na disponibilização dos conjuntos de regras e apresenta um conjunto de medidas de avaliação que pode ser utilizado de forma integrada na seleção e ordenação das regras. O acesso aos conjuntos de regras é realizado pela WWW visando a utilização de alguns benefícios obtidos pela utilização deste tipo de interface, como a independência de plataforma, facilidade de uso e especialmente a acessibilidade. Também como requisito implementado

está a possibilidade dos usuários compartilharem suas análises no ambiente, assim como comentários, idéias e sugestões.

Com a implementação do Ambiente *RulEE*, uma base de dados valiosa referente a conjuntos de regras e valores de medidas tornou-se disponível. Somando-se outros tipos de dados fornecidos pelos usuários, tem-se um conjunto de dados em que pesquisas referentes ao processo de análise de regras podem ser feitas.

No LABIC vêm sendo desenvolvidos diversos trabalhos relacionados à avaliação de conhecimento. A fim de integrar esses trabalhos, bem como os relativos a outras etapas do processo de Mineração de Dados, está sendo desenvolvido o Ambiente Discover. No Discover foram desenvolvidas sintaxes padrão para representação de dados, regras de classificação, regressão e associação, assim como *scripts* para manipulação desses dados e conjuntos de regras. O Ambiente *RulEE* utiliza esses trabalhos permitindo que a saída de uma grande variedade de algoritmos utilizados em Mineração de Dados possa ser disponibilizada, além de se apresentar como uma interface para a disponibilização do conhecimento descoberto.

Visando mostrar algumas das funcionalidades existentes no Ambiente *RulEE*, foi realizado um estudo de caso utilizando uma base de dados financeira. Nesse estudo de caso foi realizada uma análise qualitativa das regras descobertas por meio de consultas no Repositório de Regras e Medidas. Pela realização dessas consultas, observou-se que a análise dos conjuntos de regras pode ser bastante facilitada. Entretanto, para que resultados satisfatórios sejam obtidos quanto ao objetivo final da análise dos dados do estudo de caso, seria necessária a participação dos especialistas do domínio e usuários finais do processo de Mineração de Dados em uma análise qualitativa dos conjuntos de regras, que não foi possível devido a não disponibilidade dos mesmos.

A seguir, são apresentadas algumas propostas de trabalhos futuros que complementam o desenvolvimento relatado nesta dissertação e podem trazer novas contribuições para o processo de Extração de Conhecimento de Bases de Dados:

- Como qualquer ferramenta cujo objetivo é sua utilização por usuários não especialistas em computação ou Mineração de Dados, é extremamente importante para a aceitação da mesma que esta apresente uma interface intuitiva e de fácil utilização. Entretanto, a interface do Ambiente *RulEE* foi desenvolvida visando a disponibilização dos resultados do ambiente, ou seja, o foco principal não foi sua usabilidade. Dessa maneira, diversos aspectos devem ser trabalhados nesta interface, dentre eles:

  - Desenvolvimento voltado a facilitar a navegação e a identificação das informações, ou seja, considerando aspectos relativos à usabilidade da interface;
  - Quanto à visualização dos conjuntos de regras pela interface, é interessante que no ambiente existissem várias maneiras para a visualização das regras. Por exemplo, poderia haver alguma ferramenta para apresentação não textual, na qual o usuário pudesse interagir com o conjunto de regras. Tendo disponível várias possibilidades para visualização, os usuários poderiam utilizar a que julgarem mais conveniente;
  - Realização de uma análise dos erros que podem ocorrer durante a execução dos *scripts* referentes ao Ambiente *RulEE*, visando apresentar alguma descrição mais detalhada do erro e, assim, auxiliar os usuários na compreensão do mesmo;

- Na implementação atual, a integração do Ambiente *RulEE* com o Discover é realizada através da utilização de conjuntos de regras representados nas sintaxes padrão e utilização de *scripts* de conversão de dados e regras, implementado para o Ambiente Discover. É interessante permitir que exista uma integração desses ambientes através de suas interfaces de forma que os conjuntos de regras extraídos pelo Discover pudessem ser disponibilizados no Ambiente *RulEE* de forma transparente;

- Um outro requisito importante de um ambiente cujo objetivo é auxiliar os usuários na análise de conjuntos de regras é que este viabilize a utilização de medidas subjetivas para avaliação de regras. Esta funcionalidade não está completamente implementada no ambiente. Nesse sentido, é de extrema importância para o ambiente o desenvolvimento de trabalhos visando a utilização de medidas subjetivas;

- Um outro ponto a ser melhorado no Ambiente *RulEE* é o processamento das consultas. Na implementação atual, as consultas inseridas pelos usuários são mapeadas em novas consultas adequadas a serem executadas no Repositório de Regras e Medidas, sem a existência de um *parsing* completo das consultas inseridas pelos usuários. Um problema encontrado nesta abordagem é que não existe uma verificação de erros desenvolvida especificamente para o tratamento das consultas fornecidas pelos usuários, dificultando sua identificação e correção pelos usuários;

- Ao se utilizar grandes conjuntos de regras, o Ambiente *RulEE* apresentou problemas de desempenho. Por exemplo, o ambiente apresenta tempos de resposta elevados ao se acessar conjuntos com mais de 2.000 regras. Devido ao grande número de regras

normalmente extraídas pelos algoritmos de regras de associação, este problema deve ser solucionado.

Referências Bibliográficas

Baranauskas, J. A. (2001). *Extração Automática de Conhecimento por Múltiplos Indutores.* Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação, São Carlos, SP - Brasil.

Baranauskas, J. A. & G. E. A. P. A. Batista (2000). O projeto Discover: Idéias iniciais (comunicação pessoal).

Baranauskas, J. A. & M. C. Monard (2000). Reviewing some machine learning concepts and methods. Relatório Técnico 102, Instituto de Ciências Matemáticas e de Computação, São Carlos, SP - Brasil.

Batista, G. E. A. P. A. (2001). Sintaxe padrão do arquivo de exemplos do projeto discover. http://www.icmc.sc.usp.br/~gbatista/Discover/SintaxePadraoFinal.htm (último acesso: 21/01/2003).

Batista, G. E. A. P. A., A. C. P. L. Carvalho, & M. C. Monard (2000). Applying one-sided selection to unbalanced datasets. In *Proceedings of the Mexican Congress on Artificial Inteligence MICAI, Lecture Notes in Artificial Inteligence*, pp. 315–325.

Becker, B., R. Kohavi, & D. Sommerfield (2002). Visualizing the simple bayesian classifier. In U. Fayyad, G. G. Grinstein, & A. Wierse (Eds.), *Information Visualization in Data Mining and Knowledge Discovery*, pp. 237–249. Morgan Kaufmann Publishers.

Breiman, L. (2000). Some infinity theory for predictor ensembles. Relatório Técnico 577, University of California, Berkeley.

Breiman, L., J. H. Friedman, R. A. Olshen, & C. J. Stone (1984). *Classification and Regression Trees*. Pacific Groves, CA: Wadsworth.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *J. Kittler and F. Roli, editors, Multiple Classier Systems, Lecture Notes in Computer Science*, Volume 1857, pp. 1–15.

Fayyad, U. (1996). Data mining and knowledge discovery: Making sense out of data. *IEEE Expert-Intelligent Systems & Their Applications 11*, 20–25.

Fayyad, U., D. Haussler, & P. Stolors (1996). Mining scientific data. *Communications of the ACM 39*(11), 51–57.

Fayyad, U., G. Piatetsky-Shapiro, & P. Smyth (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM 39*(11), 27–34.

Fertig, C. S., A. A. Freitas, L. V. R. Arruda, & C. Kaestner (1999). A fuzzy beam-search rule induction algorithm. In *Proceedings of the Third European Conference (PKDD-99) Lecture Notes in Artificial Intelligence 1704*, pp. 341–347.

Félix, L. C. M. (1998). Data mining no processo de extração de conhecimento de bases de dados. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação, São Carlos, SP - Brasil.

Félix, L. C. M., S. O. Rezende, C. Y. Doi, M. F. Paula, & M. J. Romanato (1998). MLC^{++}: Biblioteca de aprendizado de máquina em C^{++}. Relatório Técnico 72, Instituto de Ciências Matemáticas e de Computação, São Carlos, SP - Brasil.

Félix, L. C. M., S. O. Rezende, M. C. Monard, & C. W. Caulkins (2000). Transforming a regression problem into a classification problem using hybrid discretization. *Computación y Sistemas 4*, 44–52.

Foster, M. & A. G. Gee (2002). The data visualization environment. In U. Fayyad, G. G. Grinstein, & A. Wierse (Eds.), *Information Visualization in Data Mining and Knowledge Discovery*, pp. 87–93. Morgan Kaufmann Publishers.

Freitas, A. A. (1998a). A multi-criteria approach for the evaluation of rule interestingness. In *Proceedings of the International Conference on Data Mining*, pp. 7–20.

Freitas, A. A. (1998b). On objective measures of rule surprisingness. In *Proceedings of the 2nd European Conference on the Principles of Data Mining and Knowledge Discovery (PKDD-98)*, Volume 1510, pp. 1–9.

Freitas, A. A. & S. H. Lavington (1998). *Mining Very Large Databases With Parallel Processing*. Kluwer Academic Publishers.

Geromini, M. R. (2002). Projeto e desenvolvimento da interface gráfica do sistema discover. Exame de Qualificação de Mestrado, Instituto de Ciências Matemáticas e de Computação, São Carlos, SP - Brasil.

Glymour, C., D. Madigan, D. Pregibon, & P. Smyth (1997). Statistical themes and lessons for data mining. *Data Mining and Knowledge Discovery 1*, 11–28.

Grinstein, G. G. & M. O. Ward (2002). Introduction to data visualization. In U. Fayyad, G. G. Grinstein, & A. Wierse (Eds.), *Information Visualization in Data Mining and Knowledge Discovery*, pp. 21–45. Morgan Kaufmann Publishers.

Kearns, M. J. & U. V. Vazirani (1994). *An introduction to computational learning theory*. Ellis Horwood.

Keim, D. A. (2001). Visual exploration of large data sets. *Communications of the ACM 44*(8), 38–44.

Klemettinen, M., H. Mannila, P. Ronkainen, H. Toivonen, & A. I. Verkamo (1994). Finding interesting rules from large sets of discovered association rules. In *Proceedings of the Third International Conference on Information and Knowledge Management*, pp. 401 – 407.

Klemettinen, M., H. Mannila, & H. Toivonen (1996). Interactive exploration of discovered knowledge: A methodology for interaction, and usability studies. Relatório Técnico C-1996-3, University of Helsinki, Finland.

Klemettinen, M., H. Mannila, & H. Toivonen (1997). A data mining methodology and its application to semi-automatic knowledge acquisition. In *DEXA Workshop*, pp. 670–677.

Kohavi, R., D. Sommerfield, & J. Dougherty (1997). Data mining using MLC^{++}: A machine learning library in C^{++}. *International Journal on Artificial Intelligence Tools 6*(4), 537–566.

Lavrac, N., P. Flach, & B. Zupan (1999). Rule evaluatin measures: A unifying view. *Lecture Notes in Artificial Inteligence 1634*, 174–185.

Lee, H. D. (2000). Selcção e construção de features relevantes para o aprendizado de máquina. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação, São Carlos, SP - Brasil.

Liu, B. & W. Hsu (1996). Post-analysis of learned rules. *AAAI 1*, 828–834.

Liu, B., W. Hsu, S. Chen, & Y. Ma (2000). Analyzing the subjective interestingness of association rules. *IEEE Intelligent Systems & their Applications 15*(5), 47–55.

Liu, B., W. Hsu, & Y. Ma (1999). Pruning and summarizing the discovered associations. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD-99)*, pp. 125–134.

Liu, B., W. Hsu, & Y. Ma (2001). Identifying non-actionable association rules. In *Proceedings of the International Conference in Knowledge Discover and Data Mining (KDD-01)*, pp. 329–334.

Liu, B., W. Hsu, Y. Ma, & S. Chen (1999). Mining interesting knowledge using DM-II. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD-99)*, pp. 430–434.

Liu, B., W. Hsu, L. Mun, & H. Lee (1996). Finding interesting patterns using user expectations. Relatório Técnico TRA7/96, National University of Singapore.

Lyman, P. & H. R. Varian (2000). How much information? http://www.sims.berkeley.edu/research/projects/how-much-info/index.html (último acesso: 12/02/2003).

Ma, Y., B. Liu, & C. K. Wong (2000). Web for data mining: Organizing and interpreting the discovered rules using the web. *SIGKDD Explorations 2*(1), 16–23.

Melanda, E. A. & S. O. Rezende (2003). Pós-processamento de regras de associação. Relatório Técnico em preparação, Instituto de Ciências Matemáticas e de Computação, São Carlos, SP - Brasil.

Monard, M. C., C. W. Caulkins, J. A. Baranauskas, R. B. T. Oliveira, & S. O. Rezende (1999). Data preparation, reduction and prediction in the context of data mining. Relatório Técnico 81, Instituto de Ciências Matemáticas e de Computação, São Carlos, SP - Brasil.

Motroni, S. & H. Vanderberg (2000). Mineset 3.0 enterprise edition interface guide. Silicon Graphics Inc. http://techpubs.sgi.com/library/manuals/4000/007-4005-001/pdf/007-4005-001.pdf (último acesso: 21/01/2003).

Padmanabhan, B. & A. Tuzhilin (2000). Small is beautiful: Discovering the mininmal set of unexpected patterns. In *Proceedings of the International Conference in Knowledge Discover and Data Mining (KDD-00)*, pp. 54–63.

Paula, M. F. & S. O. Rezende (2003). Ambiente para exploração de regras *RulEE*. Relatório Técnico em preparação, Instituto de Ciências Matemáticas e de Computação, São Carlos, SP - Brasil.

Pazzani, M., S. Mani, & W. Shankle (1997). Comprehensible knowledge discovery in databases. In *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, *Lawrence Erlbaum*, pp. 596–601.

Pazzani, M. J. (2000). Knowledge discovery from data? *IEEE Intelligent Systems 15*(2), 10–13.

Piatetsky-Shapiro, G. (1991). Discovery, analysis and presentation of strong rules. In *Knowledge Discovery in Databases*, pp. 229–248.

Piatetsky-Shapiro, G. & C. J. Matheus (1994). The interestingness of deviations. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD-94)*, pp. 23–36.

Prati, R. C., J. A. Baranauskas, & M. C. Monard (2001a). Extração de informações padronizadas para a avaliação de regras induzidas por algoritmos de aprendizado de máquina simbólico. Relatório Técnico 145, Instituto de Ciências Matemáticas e de Computação, São Carlos, SP - Brasil.

Prati, R. C., J. A. Baranauskas, & M. C. Monard (2001b). Uma proposta de unificação da linguagem de representação de conceitos de algoritmos de aprendizado de máquina simbólicos. Relatório Técnico 137, Instituto de Ciências Matemáticas e de Computação, São Carlos, SP - Brasil.

Pugliesi, J. B., D. G. Dosualdo, & S. O. Rezende (2003). Pós-processamento de regras de regressão. Relatório Técnico em preparação, Instituto de Ciências Matemáticas e de Computação, São Carlos, SP - Brasil.

Rezende, S. O., J. B. Pugliesi, E. A. Melanda, & M. F. Paula (2003). Mineração de dados. In S. O. Rezende (Ed.), *Sistemas Inteligentes: Fundamentos e Aplicações*, pp. 307–335. Manole.

Rocha, C. (1999). Redes baysianas para extração de conhecimento de bases de dados considerando a incorporação de conhecimento de fundo e tratamento de dados incompletos. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação, São Carlos, SP - Brasil.

Silberschatz, A. & A. Tuzhilin (1995). On subjective measures of interestingness in knowledge discovery. *Proceedings of the First International Conference on Knowledge Discovery and Data Mining 1*, 275–281.

Silberschatz, A. & A. Tuzhilin (1996). What makes patterns interesting in knowledge discovery patterns. *IEEE Transactions on Knowledge and Data Engineering 8*(6), 970–974.

Tamayo, P., J. Berlin, N. Dayanand, G. Drescher, D. R. Mani, & C. Wang (1997). Darwin: A scalable integrated system for data mining. Thinking Machines Corporation. http://otn.oracle.com/products/datamining/pdf/darwin_arch.pdf (último acesso: 21/01/2003).

Thearling, K., B. Becker, D. DeCoste, W. D. Mawby, M. Pilote, & D. Sommerfield (2002). Visualizing data mining models. In U. Fayyad, G. G. Grinstein, & A. Wierse (Eds.), *Information Visualization in Data Mining and Knowledge Discovery*, pp. 205–222. Morgan Kaufmann Publishers.

Thuraisingham, B. (1999). *Data Mining: Technologies, Techniques, Tools, and Trends*. CRC Press.

Tsur, D., J. D. Ullman, S. Abiteboul, C. Clifton, R. Motwani, S. Nestorov, & R. A. (1998). Query flocks: A generalization of association rule mining. In *Proceedings of the ACM SIGMOD Conference*, pp. 1–12.

Weiss, S. M. & N. Indurkhya (1998). *Predictive Data Mining: A Practical Guide*. Morgan Kaufmann Publishers, Inc., San Francisco, CA.

Witten, I. H. & E. Frank (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers.

Zuffo, J. A. (2002). *A Sociedade e a Economia no Novo Milênio - Os empregos e as empresas no turbulento alvorecer do século XXI*. Manole.