**Seleção de Atributos Relevantes para
Aprendizado de Máquina Utilizando
a Abordagem de *Rough Sets****

Adriano Donizete Pila

Orientação:

Profa. Dra. Maria Carolina Monard

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo — ICMC/USP, como parte dos requisitos para a obtenção do título de Mestre em Ciências — Área de Computação e Matemática Computacional.

**USP – São Carlos
Abril de 2001**

*Trabalho realizado com o apoio da FAPESP (98/16172-3) — <http://www.fapesp.br>

*Aos meus pais, Antonio e Ivete,
pelo grande incentivo.*

*À minha esposa, Cássia,
pelo apoio nos momentos difíceis.*

Esta dissertação foi preparada com o formatador de textos L^AT_EX. O sistema de citações de referências bibliográficas utiliza o padrão *Apalike* do sistema B_IB_TE_X.

© Copyright 2001 por Adriano Donizete Pila
Todos os Direitos Reservados

*“Aquele que se apóia em uma vontade firme,
vive num mundo a seu gosto.” — Goethe*

Agradecimentos

À Profa. Maria Carolina Monard pelo grande incentivo, paciência e sabedoria extrema na condução deste trabalho. A cada nova dissertação ou tese finalizados sob sua orientação a certeza de um trabalho cuidadosamente revisado inúmeras vezes, muito próximo à desejada perfeição. Obrigado pelo crescimento profissional, intelectual e moral em tão pouco tempo. Posso dizer que realmente valeu a pena!!!

Ao Prof. Ricardo Luís de Freitas, antigo mestre e amigo que sempre me incentivou e ajudou a caminhar adiante em busca de meus objetivos, mesmo nos momentos de atribulação.

Aos Profs. Solange Oliveira Rezende, Roseli Aparecida Francelin Romero e Gilberto Shigueo Nakamiti pela amizade e incentivo.

Aos amigos do LABIC pelos momentos de descontração, mesmo nas horas de fúria. Embora o trabalho seja individual, de alguma forma há sempre a colaboração de todos. Para citar alguns, Chandler, Walter, Gustavo, Augusto, Betovs, Ernesto, Huei, Valéria, Claudinha, Paty, Teresinha, Cris, Cláudia e especialmente a Jaque pela ajuda na revisão final.

Aos funcionários do ICMC, Marília, Laura, Beth e Adriana pelo profissionalismo.

Aos antigos colegas de graduação espalhados pelo mundo.

À minha família pelo apoio incondicional, meus sogros e especialmente meus pais pela dedicação e incentivo desde o início.

À minha esposa Cássia, pelas constantes e incansáveis palavras de incentivo e apoio, mesmo perante os obstáculos aparentemente intransponíveis.

Obrigado a todos. Esta é certamente uma conquista NOSSA!

Resumo

No Aprendizado de Máquina Supervisionado — AM — o algoritmo de indução trabalha com um conjunto de exemplos de treinamento, no qual cada exemplo é constituído de um vetor com os valores dos atributos e as classes, e tem como tarefa induzir um classificador capaz de prever a qual classe pertence um novo exemplo. Em geral, os algoritmos de indução baseiam-se nos exemplos de treinamento para a construção do classificador, sendo que uma representação inadequada desses exemplos, bem como inconsistências nos mesmos podem tornar a tarefa de aprendizado difícil.

Um dos problemas centrais de AM é a Seleção de um Subconjunto de Atributos — SSA — cujo objetivo é diminuir o número de atributos utilizados na representação dos exemplos. São três as principais razões para a realização de SSA. A primeira razão é que a maioria dos algoritmos de AM, computacionalmente viáveis, não trabalham bem na presença de vários atributos. A segunda razão é que, com um número menor de atributos, o conceito induzido através do classificador pode ser melhor compreendido. E, a terceira razão é o alto custo para coletar e processar grande quantidade de informações. Basicamente, são três as abordagens para a SSA: *embedded*, *filtro* e *wrapper*.

A Teoria de Rough Sets — RS — é uma abordagem matemática criada no início da década de 80, cuja principal funcionalidade são os redutos, e será tratada neste trabalho. Segundo essa abordagem, os redutos são subconjuntos mínimos de atributos que possuem a propriedade de preservar o poder de descrição do conceito relacionado ao conjunto de todos os atributos.

Neste trabalho o enfoque está na abordagem filtro para a realização da SSA utilizando como filtro os redutos calculados através de RS. São descritos vários experimentos sobre nove conjuntos de dados naturais utilizando redutos, bem como outros filtros para SSA. Feito isso, os atributos selecionados foram submetidos a dois algoritmos simbólicos de AM. Para cada conjunto de dados e indutor, foram realizadas várias medidas, tais como número de atributos selecionados, precisão e números de regras induzidas. Também, é descrito um estudo de caso sobre um conjunto de dados do mundo real proveniente da área médica. O objetivo desse estudo pode ser dividido em dois focos: comparar a precisão dos algoritmos de indução e avaliar o conhecimento extraído com a ajuda do especialista. Embora o conhecimento extraído não apresente surpresa, pôde-se confirmar algumas hipóteses feitas anteriormente pelo especialista utilizando outros métodos. Isso mostra que o Aprendizado de Máquina também pode ser visto como uma contribuição para outros campos científicos.

Abstract

In Supervised Machine Learning — ML — an induction algorithm is typically presented with a set of training examples, where each example is described by a vector of feature values and a class label. The task of the induction algorithm is to induce a classifier that will be useful in classifying new cases.

In general, the inductive-learning algorithms rely on existing provided data to build their classifiers. Inadequate representation of the examples through the description language as well as inconsistencies in the training examples can make the learning task hard.

One of the main problems in ML is the Feature Subset Selection — FSS — problem, *i.e.* the learning algorithm is faced with the problem of selecting some subset of feature upon which to focus its attention, while ignoring the rest. There are three main reasons that justify doing FSS. The first reason is that most ML algorithms, that are computationally feasible, do not work well in the presence of many features. The second reason is that FSS may improve comprehensibility, when using less features to induce symbolic concepts. And, the third reason for doing FSS is the high cost in some domains for collecting data. Basically, there are three approaches in ML for FSS: *embedded*, *filter* and *wrapper*.

The Rough Sets Theory — RS — is a mathematical approach developed in the early 1980's whose main functionality are the reducts, and will be treated in this work. According to this approach, the reducts are minimal subsets of features capable to preserve the same concept description related to the entire set of features.

In this work we focus on the filter approach for FSS using as filter the reducts obtained through the RS approach. We describe a series of FSS experiments on nine natural datasets using RS reducts as well as other filters. Afterwards we submit the selected features to two symbolic ML algorithms. For each dataset, various measures are taken to compare inducers performance, such as number of selected features, accuracy and number of induced rules. We also present a case study on a real world dataset from the medical area. The aim of this case study is twofold: comparing the induction algorithms performance as well as evaluating the extracted knowledge with the aid of the specialist. Although the induced knowledge lacks surprising, it allows us to confirm some hypothesis already made by the specialist using other methods. This shows that Machine Learning can also be viewed as a contribution to other scientific fields.

Sumário

1	Introdução	1
2	Inteligência Artificial e Aprendizado	6
2.1	Considerações Iniciais	6
2.2	Paradigmas de Inteligência Artificial	6
2.3	Aprendizado de Máquina	8
2.4	Um Modelo Genérico para Aprendizado	9
2.4.1	Escolha do Conjunto de Treinamento	10
2.4.2	Escolha da Função Objetivo	11
2.4.3	Escolha da Representação para a Função Objetivo	12
2.4.4	Escolha do Algoritmo de Aproximação de Funções	12
2.4.5	O Modelo Final	13
2.5	Uma Abordagem Geral sobre Engenharia de Conhecimento	15
2.6	Engenharia de Conhecimento e Rough Sets	18
2.7	Considerações Finais	19
3	Rough Sets: Uma Visão Geral	20
3.1	Considerações Iniciais	20
3.2	Introdução	20
3.3	Conceitos Básicos	21
3.4	Áreas de Aplicação	27
3.5	Rough Sets e Aprendizado de Máquina	28
3.6	Rough Sets e Descoberta de Conhecimento	29
3.7	Considerações Finais	29

4	Rough Sets: Teoria	31
4.1	Considerações Iniciais	31
4.2	Sistemas de Informação	32
4.3	Distinguindo Objetos	33
4.3.1	Relação de Não-Discernimento	33
4.3.2	Matriz de Discernimento	35
4.3.3	Função de Discernimento	36
4.3.4	Redução da Representação	37
4.3.5	Aproximação de Conjuntos	39
4.3.6	Qualidade das Aproximações	41
4.4	De Redutos para Regras	42
4.5	Função de Pertinência Rough	43
4.6	Regras Certas e Regras Possíveis	45
4.7	Fator de Credibilidade	46
4.8	Dependência entre Atributos	46
4.9	Extensões	47
4.10	Considerações Finais	48
5	Ferramentas e Indutores	49
5.1	Considerações Iniciais	49
5.2	Grobian	49
5.3	Rough Enough	50
5.4	Rough Sets Library	51
5.5	Column Importance	51
5.6	Rosetta	51
5.6.1	Tratamento de Dados Incompletos	54
5.6.2	Discretização	56
5.6.3	Redutores	59
5.6.4	Geração de Regras	61
5.6.5	Classificação	61
5.6.6	Automação	64

5.7	A Biblioteca $\mathcal{MLC}++$	66
5.7.1	ID3	67
5.7.2	$\mathcal{C4.5}$	68
5.7.3	$\mathcal{C4.5}$ -rules	68
5.7.4	$\mathcal{CN}^2$	69
5.8	Considerações Finais	69
6	Redutos na Abordagem Filtro para Seleção de Atributos	71
6.1	Considerações Iniciais	71
6.2	Abordagens para a Seleção de um Subconjunto de Atributos	72
6.3	Conjuntos de Dados	73
6.3.1	Descrição Geral	74
6.3.2	Características dos Conjuntos de Dados	75
6.4	Organização dos Experimentos	76
6.5	Resultados Experimentais	78
6.5.1	Descrição das Tabelas	78
6.5.2	TA	79
6.5.3	Bupa	80
6.5.4	Pima	81
6.5.5	Breast Cancer2	81
6.5.6	Cmc	82
6.5.7	Breast Cancer	82
6.5.8	Smoke	83
6.5.9	Hungarian	84
6.5.10	Hepatitis	84
6.6	Análise dos Resultados	85
6.6.1	Número de Atributos Selecionados	85
6.6.2	Comparação sem Seleção de Atributos e com Seleção de Atributos utilizando Filtro	86
6.6.3	Outros Resultados para SSA Utilizando Filtros	91
6.6.4	Número de Regras Induzidas	96

6.7	Considerações Finais	98
7	Talassemia α — Um Estudo de Caso	99
7.1	Considerações Iniciais	99
7.2	Fases do Estudo de Caso	99
7.3	Definição e Compreensão do Domínio	100
7.3.1	Dados sobre Hematologia — Talassemia α	101
7.3.2	Objetivos	102
7.4	Limpeza e Pré-Processamento dos Dados	102
7.5	Extração de Conhecimento: Experimentos e Resultados	105
7.5.1	Descrição dos Experimentos	106
7.5.2	Resultados Obtidos	107
7.6	Avaliação do Conhecimento	113
7.7	Considerações Finais	115
8	Conclusões	117
	Referências	121
	Apêndices	126
A	Teoria da Evidência de Dempster-Shafer	126
A.1	Considerações Iniciais	126
A.2	Raciocínio Evidencial	127
A.3	Intervalo de Crença	129
A.4	Aplicação da Teoria de Dempster-Shafer	130
A.5	Interpretação dos Resultados	133
A.6	Considerações Finais	134
B	Teoria de Fuzzy Sets	135
B.1	Considerações Iniciais	135
B.2	Fuzzy Sets	135
B.3	Operações e Propriedades	139

B.4	Medidas de Ambigüidade	142
B.5	Considerações Finais	144

Lista de Figuras

1.1	Sistema de AM para Classificação	3
2.1	Possíveis aproximações de funções (b), (c) e (d) para os elementos do conjunto de treinamento representados em (a)	13
2.2	Um Modelo de Sistema de Aprendizado	14
2.3	Diferentes caminhos para mapear o espaço do domínio do problema no espaço de solução do problema (Kasabov, 1996)	16
2.4	Aplicabilidade dos diferentes métodos para solução de problemas, dependendo da disponibilidade de dados e teoria a respeito do problema (Kasabov, 1996)	17
2.5	Rough Sets e a interação com as outras áreas de Engenharia de Conhecimento para a solução de problemas	18
2.6	Rough Sets e a aplicabilidade dos diferentes métodos para solução de problemas	19
3.1	Aproximações Superior e Inferior para o conceito X	26
4.1	Dois redutos definidos pelos atributos $\{a,b\}$ e $\{c,d\}$. Ambos definem a mesma relação de não-discernimento definida pelos atributos $\{a,b,c,d\}$	39
4.2	Aproximações Superior e Inferior para o conjunto de elementos que formam o mapa do Brasil	44
5.1	Opções que a ferramenta Rosetta oferece para auxiliar na Extração de Conhecimento	52
5.2	Estrutura criada após a aplicação de alguns métodos no processo de Extração de Conhecimento	63
6.1	Dimensão dos Conjuntos de Dados	76
6.2	Passos do Experimento	77

6.3	$CN2$ – Diferença no Desvio Padrão dos Erros	88
6.4	$C4.5$ -rules – Diferença no Desvio Padrão dos Erros	90
6.5	Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Con- junto de Dados Ta	92
6.6	Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Con- junto de Dados Bupa	92
6.7	Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Con- junto de Dados Pima	93
6.8	Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Con- junto de Dados Breast Cancer2	93
6.9	Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Con- junto de Dados Cmc	94
6.10	Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Con- junto de Dados Breast Cancer	94
6.11	Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Con- junto de Dados Smoke	95
6.12	Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Con- junto de Dados Hungarian	95
6.13	Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Con- junto de Dados Hepatitis	96
7.1	Fases da Extração de Conhecimento	100
7.2	Experimentos Realizados sobre o Conjunto de Dados Talassemia α . .	107
7.3	<i>Boxplot</i> para o Atributo RDW	111
B.1	Função de pertinência representando a variável “altura”	137
B.2	Representação de conjuntos crisp e fuzzy como subconjuntos do domínio (universo) U	138
B.3	Representação de conjuntos crisp e fuzzy do ponto de vista da existência de bordas bem definidas	139
B.4	Suporte do conjunto fuzzy X	140
B.5	α -cut de um conjunto fuzzy	141
B.6	Cinco operações com dois conjuntos fuzzy A e B	141
B.7	A região em preto representa quantitativamente as medidas de simi- laridade (a) e distância (b) entre conjuntos fuzzy.	143

B.8	Representação gráfica do cálculo da similaridade S entre dois conjuntos fuzzy X e Y com base na possibilidade P e na necessidade N	144
-----	--	-----

Lista de Tabelas

3.1	Tabela de Informação	22
3.2	Tabela Reduzida de Informação	24
3.3	Tabela de Informação Inconsistente	25
4.1	Sistema de Informação	32
4.2	Sistema de Decisão	33
4.3	Classes para $B=\{Estudos, Educação, Trabalha\}$	35
4.4	Matriz de Discernimento	35
4.5	Redutos Relativos	38
6.1	Características dos Conjuntos de Dados	76
6.5.1	TA – Descrição dos Atributos	80
6.5.2	TA – Atributos Seleccionados, Erros na Classificação e Número de Regras	80
6.5.3	Bupa – Descrição dos Atributos	80
6.5.4	Bupa – Atributos Seleccionados, Erros na Classificação e Número de Regras	80
6.5.5	Pima – Descrição dos Atributos	81
6.5.6	Pima – Atributos Seleccionados, Erros na Classificação e Número de Regras	81
6.5.7	Breast Cancer2 – Descrição dos Atributos	81
6.5.8	Breast Cancer2 – Atributos Seleccionados, Erros na Classificação e Número de Regras	82
6.5.9	Cmc – Descrição dos Atributos	82
6.5.10	Cmc – Atributos Seleccionados, Erros na Classificação e Número de Regras	82

6.5.11Breast Cancer – Descrição dos Atributos	83
6.5.12Breast Cancer – Atributos Seleccionados, Erros na Classificação e Número de Regras	83
6.5.13Smoke – Descrição dos Atributos	83
6.5.14Smoke – Atributos Seleccionados, Erros na Classificação e Número de Regras	83
6.5.15Hungarian – Descrição dos Atributos	84
6.5.16Hungarian – Atributos Seleccionados, Erros na Classificação e Número de Regras	84
6.5.17Hepatitis – Descrição dos Atributos	85
6.5.18Hepatitis – Atributos Seleccionados, Erros na Classificação e Número de Regras	85
6.2 Número de Atributos Seleccionados	86
6.3 Diferença no Desvio Padrão dos Erros	88
6.4 Mudança na Precisão com Nível Significativo	89
6.5 Número de Regras Induzidas pelo $\mathcal{C}4.5$ -rules	98
6.6 Número de Regras Induzidas pelo $\mathcal{CN}2$	98
7.1 Característica do Conjuntos de Dados Talassemia α	104
7.2 Descrição dos Atributos do Conjuntos de Dados Talassemia α	105
7.3 Estatísticas dos Atributos do Conjunto de Dados Talassemia α	105
7.4 Conjunto de Dados – Atributos Seleccionados	108
7.5 Índice de Correlação entre os Atributos	109
7.6 Erros na Classificação do Conjunto de Dados	110
7.7 Erros na Classificação Excluindo os Valores Aberrantes em Três Etapas	112
7.8 Número de Regras Induzidas	113
7.9 Matriz de Confusão para as Regras Induzidas pelo $\mathcal{C}4.5$ -rules	114
7.10 Matriz de Confusão para as Regras Induzidas pelo Rosetta	115
A.1 Hipóteses e as probabilidades de cada hipótese estar correta	131
A.2 Combinando as duas primeiras evidências	131
A.3 Resultado da inserção da terceira evidência	132
A.4 Resultado da inserção da quarta evidência	132

A.5	Resultados da normalização	132
-----	--------------------------------------	-----

Lista de Abreviaturas

AM	Aprendizado de Máquina
CF	Credibility Factor
CI	Column Importance
CV	Cross Validation
D-S	Dempster-Shafer
FS	Fuzzy Sets
FSS	Feature Subset Selection
IA	Inteligência Artificial
IND	Relação de Não-Discernimento
KDD	Knowledge Discovery in Database
ODBC	Open Database Connectivity
RB	Raciocínio Booleano
RNA	Redes Neurais Artificiais
ROC	Receiver Operator Characteristic
RS	Rough Sets
RSES	Rough Sets Expert System
RSL	Rough Sets Library
SA	Seleção de Atributos
SD	Sistema de Decisão
SI	Sistema de Informação
SSA	Seleção de um Subconjunto de Atributos
UCI	University of California Irvine

Capítulo 1

Introdução

COM a expansão tecnológica e o envolvimento da computação com diversas áreas do conhecimento surge uma categoria de problemas aparentemente sem solução. Esses problemas dependem fortemente do conhecimento do domínio no qual estão inseridos. A solução desses problemas torna-se possível através dos sistemas de Inteligência Artificial — IA — nos quais o aprendizado através da manipulação coerente de informação e conhecimento do domínio resulta em um sistema capaz de exibir inteligência.

Para que os computadores sejam capazes de exibir inteligência, o conhecimento do domínio do problema deve ser adquirido e incorporado a eles. A aquisição do conhecimento pode ser feita de forma explícita ou implícita. A aquisição explícita do conhecimento está relacionada ao contato direto do especialista do domínio do problema com o engenheiro de conhecimento, o qual adquire o conhecimento utilizando algumas das técnicas de aquisição de conhecimento explícito, tais como entrevistas, questionários, discussões em grupo, entre outras (Rezende & Pugliesi, 1998). A aquisição de conhecimento implícito é mais complexa, pois envolve conhecimento que não seja fornecido explicitamente pelo especialista do domínio. A realização da aquisição de conhecimento implícito é tarefa de uma área da Inteligência Artificial chamada Aprendizado de Máquina — AM.

Aprendizado de Máquina pesquisa métodos computacionais relacionados à aquisição automática de novos conhecimentos, novas habilidades e novas formas de organizar o conhecimento já existente (Mitchell, 1997).

Um sistema de AM é, então, um programa de computador que toma decisões baseadas em experiências acumuladas contidas em exemplos (ou casos) previamente re-

solvidos com sucesso (Mitchell, 1997). O aprendizado pode ser não-supervisionado ou supervisionado. No aprendizado não-supervisionado o objetivo é estabelecer agrupamentos de padrões que são similares e identificar potenciais classes nos exemplos¹. Já no aprendizado supervisionado o objetivo é extrair conhecimento de exemplos previamente rotulados com classes, a fim de prever a qual classe pertencem novos exemplos.

No Aprendizado de Máquina supervisionado os exemplos são compostos de características chamadas atributos, para os quais existem valores associados. Então, um sistema de AM supervisionado pode ser descrito como:

dados exemplos compostos por n atributos $\mathbf{x} = (x_1, x_2, \dots, x_n)$ e a classe associada a cada exemplo, y , a tarefa é encontrar um mapeamento f tal que $y=f(\mathbf{x})$.

Em AM supervisionado os exemplos estão rotulados com valores contínuos ou categóricos. Quando os exemplos estão rotulados com valores contínuos a tarefa de aprendizado é chamada de regressão e quando os exemplos estão rotulados com valores categóricos a tarefa de aprendizado é chamada de classificação.

Um sistema de AM para classificação pode ser dividido em duas fases — Figura 1.1. Na primeira fase os exemplos rotulados (exemplos de treinamento) são fornecidos ao sistema de AM, que geralmente é um algoritmo de indução (ou simplesmente indutor) capaz de extrair conhecimento desses exemplos rotulados e gerar um classificador representado em uma estrutura interna. Na segunda fase o classificador gerado pelo sistema de AM é utilizado para rotular novos exemplos (exemplos de teste).

Então, um sistema de AM para classificação é um programa capaz de extrair o conhecimento implícito nos exemplos e gerar um classificador cuja estrutura interna relaciona os valores dos atributos à classe (Weiss & Kulikowski, 1990). Esse classificador tem a habilidade de prever tanto a classe dos exemplos utilizados em sua construção, quanto a classe de novos exemplos. Este trabalho está centrado nos sistemas de AM para classificação.

A precisão com que o classificador prevê a classe dos exemplos está relacionada a sua representação interna utilizando os atributos. Se os atributos não são capazes de representar o conhecimento implícito nos exemplos, a precisão do classificador pode

¹O aprendizado não-supervisionado também é conhecido como *clusterização*, termo este adaptado da tarefa de encontrar agrupamentos (*clusters*).

Figura 1.1: Sistema de AM para Classificação

ser baixa, ou seja, novos exemplos submetidos ao classificador terão grande chance da classe ser predita incorretamente. Assim, quanto mais significativos forem os atributos utilizados para descrever os exemplos, mais confiável será a classificação.

Em outras palavras, se os atributos representam as propriedades essenciais dos exemplos, cabe ao sistema de AM representá-los em uma estrutura capaz de generalizar o conhecimento implicitamente representado nos exemplos. Essa estrutura é uma síntese de todos os exemplos e o sucesso da predição utilizando essa estrutura está diretamente relacionado ao poder de representação dos atributos.

Outro fator importante que influencia os sistemas de AM é o número de atributos utilizados para representar os exemplos (Kira & Rendell, 1992). Geralmente, os sistemas de AM computacionalmente viáveis não trabalham bem na presença de grande quantidade de atributos. Porém, determinar quais atributos são relevantes é uma tarefa complexa.

Este trabalho focaliza-se em métodos capazes de encontrar atributos importantes segundo algum critério de relevância (Caruana & Freitag, 1994). Métodos centralizados neste tipo de problema são chamados de métodos para seleção de atributos. A principal abordagem para a seleção de atributos é chamada Seleção de um Subcon-

junto de Atributos — SSA — cuja finalidade é encontrar um subconjunto de atributos possivelmente mínimo e ótimo do ponto de vista da representação do conhecimento. Outra abordagem para a seleção de atributos é a Indução Construtiva (Lee, 2000; Bloedorn & Michalski, 1998).

Os métodos que se enquadram na abordagem SSA funcionam de forma que se dado um conjunto de n atributos, após seu processamento será obtido um subconjunto de m atributos, tal que $m \leq n$. No caso da Indução Construtiva $m > n$. Em outras palavras, SSA reduz o espaço de representação dos exemplos enquanto que a Indução Construtiva amplia o espaço de representação dos exemplos.

A Teoria de Rough Sets — RS — é uma abordagem matemática introduzida por Zdzislaw Pawlak no início da década de 80 (Pawlak, 1982). A principal característica dessa abordagem são os redutos. Os redutos são subconjunto de atributos capazes de representar o poder de discernir os exemplos uns dos outros mantendo o mesmo poder de representação de quando utilizam-se todos os atributos. Assim, os redutos podem ser eficazes para seleção de atributos e seu uso para SSA é investigado neste trabalho.

Como não existe uma metodologia matemática capaz de avaliar a representação do classificador, o estudo dos sistemas de AM tornam-se uma ciência experimental, na qual a validação dos classificadores é feita através de sucessivos testes sobre o conjunto de exemplos. Da mesma forma, a determinação da relevância de determinados atributos é feita através de métodos empíricos, a fim de avaliar quais métodos de SSA são mais apropriados ou apresentam melhor desempenho, para indutores e conjuntos de exemplos específicos.

Assim, neste trabalho, além de apresentar a Teoria de Rough Sets e as principais formas de realizar Seleção de um Subconjunto de Atributos são apresentadas uma série de avaliações experimentais sobre conjuntos de exemplos naturais, bem como um estudo de casos envolvendo exemplos do mundo real. Essa distinção em relação a conjuntos de exemplos vem sendo feita nos últimos anos pela comunidade de AM, e pode ser dividida em três classes (Batista, 2000):

1. *reais*, cujos exemplos são obtidos diretamente de domínios do mundo real, como por exemplo, indústrias, área médica, bancos e outros;
2. *naturais*, obtidos diretamente de repositórios de dados como o Repositório da UCI (Blake et al., 1998);
3. *artificiais*, cujos exemplos são gerados por programas de computador com o

objetivo de testar capacidades específicas dos algoritmos de AM, uma vez que a função verdadeira f e os possíveis exemplos são previamente conhecidos.

Este trabalho está organizado da seguinte forma.

No **Capítulo 2** serão apresentados os principais paradigmas de IA, um modelo genérico para Aprendizado de Máquina, bem como uma abordagem geral sobre Engenharia de Conhecimento e onde a Teoria de Rough Sets pode ser visualizada dentre as demais abordagens.

No **Capítulo 3** serão apresentados e exemplificados os conceitos de Rough Sets sem levar em consideração a base teórica e matemática que a fundamentam. Serão apresentadas ainda algumas aplicações dessa teoria em diferentes áreas, com uma breve abordagem em Aprendizado de Máquina e Descoberta de Conhecimento em Bancos de Dados — KDD (Knowledge Discovery in Databases) (KDD 95, 1995; KDD 96, 1996).

As definições de sistemas de informação e de decisão, bem como a teoria originalmente proposta por (Pawlak, 1982) e algumas extensões são apresentadas no **Capítulo 4**.

Para analisar a aplicabilidade e eficiência da abordagem de Rough Sets, existem algumas ferramentas desenvolvidas por pesquisadores da área, as quais serão brevemente apresentadas no **Capítulo 5**.

No **Capítulo 6** são apresentados os resultados experimentais e comparativos a respeito da abordagem de Rough Sets utilizada como filtro na seleção de atributos relevantes, bem como o número de regras geradas utilizando esses subconjuntos de atributos selecionados. As comparações são feitas em relação a outros filtros utilizando alguns indutores conhecidos da área de AM.

Foi realizado um estudo de casos utilizando um conjunto de dados do mundo real. As características do conjunto de dados, os experimentos realizados e resultados obtidos estão relatados no **Capítulo 7**. Finalmente, no **Capítulo 8** são apresentadas as conclusões e futuras direções de pesquisas relacionadas com este trabalho.

O **Apêndice A** apresenta uma breve introdução sobre a Teoria de Dempster-Shafer e sua forma de tratar evidências conflitantes.

No **Apêndice B** a Teoria de Fuzzy Sets é brevemente apresentada, bem como as noções de conjuntos nebulosos e raciocínio aproximado.

Capítulo 2

Inteligência Artificial e Aprendizado

2.1 Considerações Iniciais

Inteligência Artificial compreende métodos, ferramentas, e sistemas para a modelagem de situações que normalmente requerem inteligência humana¹. Como existem várias formas de se definir o que é inteligência e conhecimento, várias abordagens surgiram em decorrência dessas diferentes interpretações e, conseqüentemente, diferentes modelos têm sido propostos para enquadrar estes conceitos. Uma das abordagens mais difundidas é a área de Aprendizado de Máquina. As abordagens híbridas também são de grande importância por tentar combinar o que há de melhor em cada abordagem. Neste capítulo serão apresentados os principais paradigmas de IA, um modelo genérico para Aprendizado de Máquina, bem como uma abordagem geral sobre Engenharia de Conhecimento e onde a Teoria de Rough Sets pode ser visualizada dentro desse contexto.

2.2 Paradigmas de Inteligência Artificial

Inteligência pode ser definida como a habilidade de aprender, compreender, reagir adaptativamente, tomar decisões corretas, comunicar-se através da língua natural ou de imagens de forma sofisticada (Kasabov, 1996). O principal objetivo da Inteligência Artificial é o desenvolvimento de métodos e sistemas para a execução de tarefas, as quais normalmente são executadas através da atividade intelectual humana. Por

¹De acordo com (Russel & Norvig, 1995) essa é apenas uma das formas de se definir IA, pois diferentes definições podem ser elaboradas com base na forma de agir e pensar dos seres humanos.

exemplo, reconhecimento de imagens, processamento de língua natural e de discurso, planejamento e predição, incrementando assim os sistemas de informação computacionais. Ainda, desenvolvem-se modelos que simulam organismos vivos e o cérebro humano em particular, aumentando nosso entendimento de como o cérebro humano funciona.

Inteligência Artificial tenta modelar a inteligência humana. Para alcançar esse objetivo, existem dois paradigmas principais adotados em IA: paradigma simbólico e paradigma sub-simbólico.

A principal diferença entre esses dois paradigmas é que o primeiro é baseado na manipulação de símbolos. Segundo (Newell & Simon, 1972) o paradigma simbólico é baseado na teoria dos sistemas simbólicos físicos. Um sistema simbólico é formado por dois conjuntos:

1. um conjunto de elementos (ou símbolos) os quais podem ser utilizados para construir elementos ou estruturas mais complicadas, possuindo conotação semântica, que representam conceitos ou objetos;
2. um conjunto de processos e regras que, quando aplicados a estruturas e símbolos, produzem novas estruturas.

Sistemas simbólicos de IA são aplicados no processamento de língua natural, sistemas especialistas, Aprendizado de Máquina, modelagem de processos cognitivos, entre outros. Infelizmente, eles não funcionam da melhor forma em todos os casos quando submetidos a informações inexatas, inexistentes ou incertas, quando somente dados brutos² estão disponíveis e uma aquisição de conhecimento deve ser realizada, ou quando soluções paralelas necessitam ser elaboradas. Deve ser observado que apesar dessas tarefas exigirem um grande esforço computacional, elas podem ser facilmente solucionadas por seres humanos.

Conforme (Smolenski, 1990), o paradigma sub-simbólico alega que o comportamento inteligente é executado no cérebro em um nível maior que o neurológico, mas de uma forma diferente da simbólica. Nessa abordagem, o processamento do conhecimento é feito através da mudança de estados numa rede formada por pequenos

²Dado é diferente de informação, uma vez que o primeiro pode não possuir semântica. Por exemplo, 23 é um dado sem significado, no entanto *Idade=23* é uma informação. Conhecimento é quando infere-se algo utilizando informação e inteligência é quando por um processo de inferência ou dedução chega-se a outro conhecimento. *Se Idade=23 então o risco de morte por infarto é pequena* é um conhecimento. No entanto pode-se utilizar de inteligência para chegar-se à conclusão de que uma pessoa com *Idade=22* possui o mesmo risco de morte por infarto.

elementos chamados neurônios, fazendo uma analogia com neurônios reais. Um neurônio, ou um conjunto de neurônios, podem representar uma pequena característica de um conceito, ou mesmo um objeto. Tem sido mostrado que é possível elaborar um sistema inteligente que seja capaz de atingir a meta global (generalização), embora todos os componentes do sistema sejam simples e operem somente com informações puramente locais (específicas).

No paradigma sub-simbólico é possível a utilização de todos os resultados significativos conquistados nos últimos 20 anos (por exemplo, reconhecimento de padrões e imagens, e processamento de fala) na área de redes neurais artificiais, e também o uso de modelos conexionistas para o processamento do conhecimento.

Existem várias formas nas quais os paradigmas simbólico e sub-simbólico podem interagir no processamento do conhecimento:

1. Eles podem ser desenvolvidos e utilizados separadamente e alternativamente.
2. Sistemas híbridos que incorporem ambos sistemas simbólico e sub-simbólico podem ser desenvolvidos.
3. Sistemas sub-simbólicos podem ser utilizados para modelar sistemas puramente simbólicos.

Assim, existe um terceiro paradigma — uma *agregação dos sistemas simbólicos e sub-simbólicos*, o qual tem-se mostrado como a melhor solução para alguns casos que envolvem problemas complexos de IA. Por exemplo, sistemas fuzzy que representam o conhecimento de forma simbólica, mas usam representação numérica similar aos sistemas sub-simbólicos.

Neste trabalho nos concentraremos no paradigma simbólico, mais especificamente, em uma nova abordagem simbólica para tratamento de incerteza em Aprendizado de Máquina.

2.3 Aprendizado de Máquina

Desde a invenção dos computadores tem-se questionado quando eles poderão ser construídos para aprender. Se fosse possível entender como programá-los para aprender — melhorar automaticamente com experiência — o impacto seria dramático. Seria possível que, por exemplo, os computadores aprendessem através de informações

qual o tratamento mais eficiente para novas doenças; casas aprendessem a otimizar o custo da energia com base nos padrões de uso de seus ocupantes; ou ainda, os programas assistentes aprendessem os interesses pessoais dos indivíduos e assim colocaria em destaque as informações mais relevantes de jornais *online* de acordo com os interesses de cada pessoa. Um entendimento bem-sucedido de como fazer os computadores aprenderem, abriria novas formas de utilização destes, bem como novos níveis de interação homem-máquina.

Não se sabe como fazer os computadores aprenderem da mesma forma que as pessoas aprendem, embora algoritmos são constantemente inventados para a aplicação em certos tipos de tarefas de aprendizado. Muitos programas computacionais são desenvolvidos com o objetivo de aprender sobre algum domínio específico. Para problemas como reconhecimento de fala, os algoritmos baseados em Aprendizado de Máquina têm obtido um desempenho superior a outras abordagens conhecidas (Mitchell, 1997). Na área conhecida como *data mining*, os algoritmos de AM são largamente utilizados para descoberta de conhecimento válido e previamente desconhecido, através da avaliação de bancos de dados comerciais contendo informações sobre transações financeiras, dados médicos, bolsa de valores, entre outras.

No entanto, para que os algoritmos de AM possam ser aplicados, é necessário um modelo do conhecimento a ser aprendido. Esse modelo deve conter alguns componentes bem definidos — não sendo necessariamente disjuntos — para que o sistema de aprendizado seja eficiente.

2.4 Um Modelo Genérico para Aprendizado

Neste trabalho, aprendizado será considerado de forma a incluir qualquer programa computacional capaz de incrementar seu desempenho através da experiência sobre a tarefa a ser executada.

Definição 2.4.1 (Aprendizado) *Diz-se que um programa computacional aprende através da experiência E com respeito a alguma classe de tarefas T e medida de desempenho P , se seu desempenho nas tarefas T , tal como medida por P , incrementa com experiência E .*

Por exemplo, um programa computacional que aprende a jogar xadrez deve melhorar seu desempenho como sendo sua *habilidade de vencer* na classe de tarefas envolvendo *jogar xadrez*, através da experiência *obtida jogando xadrez contra si próprio*

— conhecido como *conjunto de treinamento*. Em geral, para termos um modelo de aprendizado, deve-se identificar essas três características:

1. classe de tarefas T ;
2. medida de desempenho P a ser incrementada;
3. experiência E .

Por exemplo, no modelo de aprendizado para o jogo de xadrez, teria-se as seguintes características:

1. tarefa T : jogar xadrez;
2. medida de desempenho P : porcentagem dos jogos ganhos contra os oponentes;
3. conjunto experiência E (treinamento): jogar xadrez contra si próprio.

Podem ser especificados muitos modelos de aprendizado utilizando essas características, tais como reconhecimento de palavras escritas à mão ou dirigir um robô autônomo automaticamente. Identificar essas características não é tarefa fácil, pois influenciam diretamente no desempenho do modelo adotado para a solução da tarefa.

Para realizar uma boa modelagem do problema a ser resolvido, cada uma das características discutidas nas próximas seções deve ser considerada.

2.4.1 Escolha do Conjunto de Treinamento

A escolha do conjunto de treinamento pode ter um impacto decisivo no sucesso ou fracasso do sistema de aprendizado. Para tanto, recomenda-se levar em consideração três características — nem sempre consideradas na vida real, por serem às vezes inviáveis ou mesmo desnecessárias:

1. tipo do conjunto de treinamento;
2. grau de controle da seqüência de exemplos de treinamento;
3. quão bem o conjunto de treinamento representa a realidade.

O tipo do conjunto de treinamento diz respeito ao fornecimento ou não de uma resposta para cada elemento do conjunto sendo utilizado. Essas respostas fornecidas no conjunto de treinamento fazem o papel de um professor, informando ao sistema se a resposta obtida a partir de um elemento do conjunto é válida ou não (aprendizado supervisionado). Há casos em que essa resposta não existe e o sistema deve ser capaz de chegar às suas próprias conclusões sem o auxílio do professor (aprendizado não-supervisionado). No exemplo do jogo de xadrez, o professor seria uma característica do conjunto de treinamento que indicasse que determinada sequência de mudanças das peças levaria à vitória ou não.

O grau de controle está relacionado à forma com que o sistema de aprendizado controla a sequência de treinamento, anulando ou não características que são inválidas para o domínio de aplicação. No exemplo do jogo de xadrez, se o sistema de aprendizado possuir um bom grau de controle sobre os exemplos de treinamento, então o sistema deveria ignorar exemplos que possuam sequências de movimentação inválida no xadrez.

Outra característica importante na escolha do conjunto de treinamento é o quão bem o conjunto de treinamento representa o domínio real. No jogo de xadrez, se o sistema de aprendizado utilizar um conjunto de treinamento formado por partidas jogadas contra ele próprio, então o sistema estará ignorando movimentos efetuados por outros especialistas no assunto. Esse é um ponto crucial, porque se o sistema tiver um bom desempenho P , para um conjunto E , mas esse conjunto for restrito, não se pode garantir que o sistema terá o mesmo desempenho P para qualquer outro conjunto de exemplos.

2.4.2 Escolha da Função Objetivo

A escolha do tipo de conhecimento a ser aprendido e como ele será utilizado para medir o desempenho do algoritmo de aprendizado é outro ponto importante. No exemplo do jogo de xadrez, o programa de aprendizado pode gerar qualquer sequência válida de movimentos, mas o interessante é escolher apenas os melhores movimentos com o objetivo de otimizar o desempenho P . Esse tipo de tarefa representa uma grande classe de tarefas para as quais conhecemos o domínio do espaço de busca do problema, mas não sabemos qual é a melhor forma de percorrer esse espaço em busca da melhor solução do problema. Muitos programas de otimização pertencem a essa classe, tais como os de controle de manufaturas, para os quais os passos são bem conhecidos, mas não a melhor estratégia de fabricação.

No exemplo do jogo de xadrez, deve-se escolher uma função V que faça a mudança do estado A para o estado B da melhor forma possível. O estado A representa as disposições das peças no tabuleiro antes de passar para o estado B , o qual representa outra disposição de peças no tabuleiro após um movimento gerado pela função V . Assim, a tarefa de aprendizado é reduzida ao problema de se descobrir a *função ideal* V . Pode ser muito difícil definir uma função desse tipo a fim de que todos os jogos sejam ganhos com a máxima eficiência. Na realidade, os algoritmos de AM procuram por alguma *aproximação* dessa função objetivo, e por isso o processo de aprendizado dessa função é normalmente chamado de *aproximação de função*. Será utilizado $\hat{V}$ para se referir à aproximação gerada pelo sistema de aprendizado, para diferenciá-la da função ideal V .

2.4.3 Escolha da Representação para a Função Objetivo

Tendo a função ideal V , é necessário escolher uma representação que o programa de aprendizado utilizará para $\hat{V}$. Existem muitas representações possíveis para $\hat{V}$. Uma forma poderia ser uma tabela que tivesse todos os mapeamentos possíveis do argumento da função, ou tabela *look-up*. Essa representação é visivelmente inviável, devido a problemas de espaço e falta de flexibilidade, uma vez que a função somente reconhece argumentos previamente colocados na tabela. Formas de representação incluem regras, funções polinomiais, redes neurais, etc. O maior desafio é escolher a função $\hat{V}$ menos complexa que melhor se aproxime da função ideal V , pois isso torna o processo de aprendizado mais eficiente. Na Figura 2.1 tem-se cinco exemplos de treinamento, e as possíveis funções aproximadas $\hat{V}$ para a representação desse conjunto.

2.4.4 Escolha do Algoritmo de Aproximação de Funções

Para conseguir uma boa função $\hat{V}$ que represente o conjunto de treinamento — diz-se então que $\hat{V}$ aprendeu os conceitos presentes no conjunto de treinamento — é necessário um conjunto de treinamento que seja um par ordenado representado na forma $\langle A, V_{\text{treinamento}}(A) \rangle$, sendo que A representa os valores dos argumentos da função e $V_{\text{treinamento}}(A)$ representa o valor que assume a função V para esses valores de argumentos. No exemplo do jogo de xadrez, A representa as características que indicam a disposição das peças num dado instante do jogo, enquanto que $V_{\text{treinamento}}(A)$ indica o melhor movimento de uma peça a ser realizado nessa situação (Mitchell,

Figura 2.1: Possíveis aproximações de funções (b), (c) e (d) para os elementos do conjunto de treinamento representados em (a)

1997).

Em outras palavras, o conjunto de treinamento é formado por experiências — reais ou não — relativas ao domínio do problema, e tem o objetivo de modelar uma função que represente os conceitos que estão implícitos nos exemplos de treinamento. Para o jogo de xadrez, o conjunto de treinamento poderia ser formado por todos os movimentos das peças efetuados por Garry Kasparov³, o que tornaria o sistema de aprendizado um especialista em xadrez. Fazendo uso desse conjunto de treinamento, além de aprender o conhecimento presente nos dados, o sistema deve ser capaz de generalizar o conhecimento adquirido, sendo assim capaz de competir contra qualquer oponente.

2.4.5 O Modelo Final

O modelo final do sistema de aprendizado pode ser dividido em quatro módulos distintos, que representam os componentes centrais em muitos sistemas de aprendizado (Mitchell, 1997), tal como mostrado na Figura 2.2:

1. O **sistema de desempenho** é o módulo que faz a medida de desempenho da

³Melhor jogador de xadrez de todos os tempos, sendo superado apenas por *Deep Blue* — um computador criado pela IBM com o propósito de jogar xadrez. Maiores informações em <http://www.research.ibm.com/deepblue>

Figura 2.2: Um Modelo de Sistema de Aprendizado

tarefa. No exemplo considerado, ele deve “jogar” xadrez utilizando a função objetivo aprendida. Ele toma como entrada uma nova instância do problema (novo jogo) e acha uma solução para o problema (histórico do jogo).

2. O módulo de **crítica** toma como entrada o histórico do jogo e produz como saída um conjunto de treinamento da função objetivo. Cada exemplo de treinamento corresponde a algum estado do jogo ao longo do histórico.
3. O **generalizador** toma como entrada o conjunto de treinamento e produz como saída uma hipótese que é uma estimativa da função objetivo. Ele tenta generalizar através do uso da hipótese e do conjunto de treinamento, a fim de conseguir uma função mais geral que cubra os exemplos de treinamento e outros casos alheios a esse conjunto.
4. O **gerador de experiências** toma como entrada a hipótese corrente (a função sendo aprendida) e produz um novo problema para o sistema de desempenho. Esse ciclo de aprendizado envolve uma estratégia muito simples, a qual pode ser modificada objetivando a exploração de regiões particulares do espaço de busca do domínio do problema.

Embora o modelo apresentado possa ser utilizado em várias tarefas de aprendizado, poderia ter sido proposto vários outros modelos alternativos. Por exemplo, para o

problema do jogo de xadrez, poderia se gerar um grande número de esquemas de jogos, combinar esses esquemas dois-a-dois e então escolher aquele que tivesse um maior número de vitórias. Outra alternativa seria ter uma grande quantidade de esquemas de jogo e seus respectivos movimentos, e a cada movimento requisitado no treinamento seria escolhido desse conjunto de movimentos o “mais próximo” do requisitado. Assim, pode ser observado que existem diversas formas de modelar um sistema de Aprendizado de Máquina.

2.5 Uma Abordagem Geral sobre Engenharia de Conhecimento

O objetivo principal do desenvolvimento de sistemas inteligentes é representar da forma mais adequada possível o conhecimento existente relativo ao domínio do problema, a fim de encontrar uma boa aproximação para a solução do problema. A solução de problemas pode ser representada como um processo de mapeamento do espaço do domínio do problema no espaço de solução do problema, utilizando conhecimento a respeito do problema, *i.e.* regras heurísticas ou dados, ou ambos (Kasabov, 1996).

Diferentes métodos podem ser utilizados para alcançar esse objetivo. A Figura 2.3 representa diferentes métodos para a solução de problemas e as relações existentes entre eles. Dependendo do tipo do problema e do conhecimento disponível a respeito do problema, diferentes métodos podem ser empregados. Seguem algumas considerações de quando utilizar cada um desses métodos na engenharia de conhecimento e solução de problemas:

- *Métodos estatísticos* podem ser utilizados quando dados possíveis de serem representados em formato estatístico estiverem disponíveis e o tipo da função meta for conhecido.
- *Sistemas de IA simbólicos baseados em regra* podem ser utilizados quando o problema a ser solucionado tiver um formato bem definido; a adaptação das regras é difícil de ser implementada.
- *Sistemas fuzzy* podem ser utilizados quando o problema inclui regras heurísticas, mas elas são vagas, mal definidas, aproximadas, e possivelmente contraditórias.

Figura 2.3: Diferentes caminhos para mapear o espaço do domínio do problema no espaço de solução do problema (Kasabov, 1996)

- *Redes neurais* podem ser utilizadas quando o problema possui dados contínuos e um modelo matemático de predição que não necessite ser representado de forma inteligível aos humanos seja satisfatório.
- *Algoritmos genéticos* não necessitam de dados nem de regras heurísticas, mas um simples critério de seleção para inicializar; eles são muito eficientes quando existe pouca informação para se inicializar o processo.

Diferentes *caminhos*, os quais mapeiam o espaço do domínio no espaço da solução do problema, são mostrados na Figura 2.3. Pode-se distinguir entre *caminhos de paradigmas simples* e *caminhos multi-paradigmas*. Os caminhos multi-paradigmas são aqueles que contém mais de um caminho utilizado no mapeamento do espaço do domínio do problema para o espaço de solução do problema. Por exemplo:

- Uma rede neural pode ser utilizada para aprender regras fuzzy, as quais são implementadas num sistema de inferência fuzzy.
- Métodos de aprendizado de máquina simbólicos podem representar as regras

aprendidas numa máquina de raciocínio simbólico.

- Regras simbólicas podem ser combinadas com redes neurais em sistemas híbridos.
- Algoritmos genéticos podem ser utilizados para definir valores para alguns parâmetros de aprendizado em redes neurais.
- Regras fuzzy podem ser implementadas tanto numa arquitetura conexionista ou combinadas com redes neurais em sistemas híbridos.

Na Figura 2.3 são também mostrados dois aspectos genéricos, representação e aprendizado, que são considerados pelos diversos métodos:

1. *representação*, quando regras heurísticas estiverem disponíveis, o que implica que existe conhecimento prévio;
2. *aprendizado*, quando dados estiverem disponíveis.

Diferentes métodos consideram esses dois aspectos em diferentes graus, tal como mostrado na Figura 2.4. Por exemplo, métodos simbólicos de IA são aplicáveis quando o problema é rico em teoria e pobre em dados, visto que o objetivo principal dos sistemas simbólicos é alcançar a generalização partindo-se de alguns exemplos específicos.

Figura 2.4: Aplicabilidade dos diferentes métodos para solução de problemas, dependendo da disponibilidade de dados e teoria a respeito do problema (Kasabov, 1996)

2.6 Engenharia de Conhecimento e Rough Sets

A Teoria de Rough Sets, descrita no Capítulo 4, possui um forte embasamento matemático e apresenta um novo formalismo para o tratamento de incerteza e inconsistência. Por se tratar de uma nova visão para a manipulação de conjuntos, os quais são a base da matemática e conseqüentemente de toda a computação, Rough Sets pode interagir com diferentes áreas da Engenharia de Conhecimento. Neste trabalho, a Figura 2.3 proposta inicialmente por (Kasabov, 1996) foi por nós reavaliada com o objetivo de inserir a abordagem de Rough Sets dentre as demais abordagens de Engenharia de Conhecimento. A Figura 2.5 mostra a nossa visão da interação de Rough Sets com as outras abordagens.

Figura 2.5: Rough Sets e a interação com as outras áreas de Engenharia de Conhecimento para a solução de problemas

Deve-se notar que Rough Sets pode ser um caminho de paradigma único para mapear o espaço de domínio do problema no espaço de solução do problema. Porém, existe a possibilidade de Rough Sets ser utilizado como um caminho adicional nas soluções híbridas que requerem caminhos de múltiplos paradigmas. A interação é possível com quase todas as áreas da Engenharia de Conhecimento.

Na Figura 2.4 proposta por (Kasabov, 1996) fica claro a aplicabilidade dos diferentes métodos para a solução de problemas dependendo da riqueza de teoria e dados referente ao problema. Devido as características de Rough Sets, pode-se dizer que essa abordagem é aplicável na presença de riqueza regular de teoria e dados. Essa aplicabilidade, segundo a nossa visão, está graficamente mostrada na Figura 2.6.

Figura 2.6: Rough Sets e a aplicabilidade dos diferentes métodos para solução de problemas

2.7 Considerações Finais

A partir do momento que existe a necessidade da extração de conhecimento de algum tipo estruturado de informação é necessário decidir qual abordagem melhor se adapta à solução do problema. Para decidir qual abordagem escolher deve-se levar em consideração os pontos fortes de cada uma, e em seguida fazer um modelo do sistema de aprendizado — o qual irá extrair conhecimento das informações — com o objetivo de delinear as etapas desse processo. Como nem toda abordagem é auto-suficiente, existem problemas para os quais a melhor solução é a aplicação de múltiplas abordagens, o que caracteriza uma solução híbrida. Rough Sets, objetivo de estudo deste trabalho, é mais uma dessas abordagens e será apresentada nos capítulos seguintes. No próximo capítulo a Teoria de Rough Sets será apresentada intuitivamente, *i.e.* sem levar em conta o formalismo dessa abordagem, mas as idéias utilizadas para tratamento de incerteza e imprecisão, bem como para seleção de atributos.

Capítulo 3

Rough Sets: Uma Visão Geral

3.1 Considerações Iniciais

A Teoria de Rough Sets é relativamente nova, e constantemente é confundida com outras teorias. Com base no trabalho de (Pawlak et al., 1995), neste capítulo serão apresentados e exemplificados os conceitos de RS sem levar em consideração a base teórica e matemática que a fundamentam. Serão apresentadas ainda algumas aplicações dessa teoria em diferentes áreas, com uma breve abordagem em AM e KDD.

3.2 Introdução

Como visto anteriormente, a Teoria de *Rough Sets*¹ é uma abordagem matemática para manipular incerteza e imprecisão, introduzida por Zdzislaw Pawlak no início da década de 80 (Pawlak, 1982). Essa abordagem pode potencialmente ser aplicada em diversas áreas de Inteligência Artificial (Pawlak et al., 1995).

O conceito de RS relaciona-se, de alguma maneira, com outras teorias matemáticas desenvolvidas para manipular incerteza e imprecisão, particularmente com a *Teoria da Evidência* de Dempster-Shafer². A principal diferença é que a teoria proposta por Dempster-Shafer utiliza a *função de crença* como ferramenta principal, enquanto que a teoria de RS faz uso de conjuntos — aproximações inferior e superior. Existe

¹Vários poderiam ser os significados do termo *rough* — áspero, desigual, tosco, rude, aproximado, grosseiro — sendo *aproximado* o significado mais apropriado. Neste trabalho será utilizado o termo em inglês, por ser amplamente aceito pela comunidade.

²Maiores detalhes podem ser encontrados no **Apêndice A**.

também uma relação entre a teoria de RS e a *Teoria de Fuzzy Sets*³ (Yao, 1998) — FS — as quais são freqüentemente comparadas e até mesmo confundidas. Ambas tratam do conhecimento imperfeito, enquanto FS trata da incerteza das informações, RS trata da imprecisão e ambigüidade dos dados.

Utilizando um exemplo clássico da área de processamento de imagens (Szladow & Ziarko, 1993), tem-se que FS trata da existência de mais de um nível de cinza nos *pixels*⁴, enquanto que RS trata do tamanho desses pixels. Fuzzy sets trata da relação entre intensidades de objetos dentro da mesma classe, enquanto que RS trata da relação entre grupos de objetos em diferentes classes. Entretanto, a teoria de RS não compete com a teoria de FS, mas a complementa. Na realidade, a teoria de RS e a teoria de FS são duas abordagens independentes para o tratamento de conhecimento imperfeito. Além disso, algumas outras relações podem ser encontradas entre a teoria de RS e análise discriminante, métodos de raciocínio booleano e análise de decisão.

Uma das principais vantagens da Teoria de Rough Sets é que ela não necessita de informações preliminares ou adicionais sobre os dados, tais como a distribuição de probabilidade em estatística, atribuição de probabilidades básicas na teoria de Dempster-Shafer, ou mesmo os graus de pertinência na teoria dos conjuntos fuzzy. Embora a teoria de RS pareça muito vantajosa em relação à outras abordagens, há quem questione sua aplicabilidade em problemas reais, nos quais atributos discretos não são suficientes para caracterizar o domínio (Koczkodaj et al., 1998). Na verdade, nenhuma abordagem é suficiente para resolver qualquer problema, por isso, freqüentemente, as abordagens de maior sucesso são híbridas, tal como as Redes Neurais Rough (Lingras, 1998).

3.3 Conceitos Básicos

Neste trabalho, são consideradas que as informações a respeito do mundo real são dadas na forma de uma *tabela de informação* — muitas vezes chamada de *tabela de decisão*⁵. Uma tabela de informação representa dados provenientes de qualquer domínio, tal como medicina, finanças, militar, etc. A Tabela 3.1 apresenta um exemplo clássico de uma tabela de informação, freqüentemente utilizado na literatura de RS, a qual será utilizada para ilustrar os conceitos e as definições.

³Maiores detalhes podem ser encontrados no **Apêndice B**.

⁴menor unidade luminosa presente num monitor de computador

⁵Na terminologia da área de aprendizado de máquina, essa informação é simplesmente denominada de *conjunto de dados* no formato atributo-valor.

Exemplos	Atributos			Decisão
	<i>Dor_de_Cabeça</i>	<i>Dor_Muscular</i>	<i>Temperatura</i>	
e_1	sim	sim	normal	não
e_2	sim	sim	alta	sim
e_3	sim	sim	muito_alta	sim
e_4	não	sim	normal	não
e_5	não	não	alta	não
e_6	não	sim	muito_alta	sim

Tabela 3.1: Tabela de Informação

As linhas e_1 , e_2 , e_3 , e_4 , e_5 e e_6 na Tabela 3.1 são chamadas de *exemplos* (objetos, entidades, casos). As propriedades dos exemplos são determinadas através dos valores atribuídos às variáveis envolvidas. As variáveis envolvidas no exemplo considerado são *Dor_de_Cabeça*, *Dor_Muscular*, *Temperatura* e *Gripe*. Existem dois tipos básicos de variáveis que devem ser distinguidas numa tabela de informação:

1. *atributos*, caracterizam os exemplos, normalmente atribuindo um valor a cada um desses atributos;
2. *decisão*, informa a decisão a ser tomada levando em consideração os valores dos atributos.

Por exemplo, se a tabela de informação descreve dados relativos a um hospital, então os exemplos podem ser dados de pacientes. Nesse caso, os atributos poderiam ser sintomas e exames referentes aos pacientes e a decisão poderia ser a doença determinada através dos valores desses atributos. Assim, cada paciente é caracterizado pelos resultados dos testes e sintomas e são classificados por médicos — no caso os *especialistas* do domínio — como possuindo a doença com algum grau de severidade. Por outro lado, se a tabela de informação descrevesse um processo industrial, os atributos poderiam ser características que descrevessem o processo, enquanto que a decisão poderia ser alguma ação tomada pelo operador (especialista).

O principal conceito envolvido em RS é a *relação de não-discernimento* (*indiscernibility relation*), a qual normalmente está associada a um conjunto de atributos. Por exemplo, para o conjunto que consiste dos atributos *Dor_de_Cabeça* e *Dor_Muscular* na Tabela 3.1, os exemplos e_1 e e_2 são caracterizados pelos mesmos valores para ambos atributos: para o atributo *Dor_de_Cabeça* os exemplos e_1 e e_2 possuem o mesmo valor *sim*, e para o atributo *Dor_Muscular* os exemplos e_1 e e_2 possuem o mesmo valor *sim*. Além disso, o exemplo e_3 é indiscernível dos exemplos e_1 e e_2 . Os exemplos e_4 e e_6 são também indiscerníveis entre si. Pode ser notado que essa relação de não-discernimento é uma relação de equivalência.

Os conjuntos que são indiscerníveis são chamados de *conjuntos elementares*. Assim, os atributos *Dor_de_Cabeça* e *Dor_Muscular* da Tabela 3.1, definem os seguintes três conjuntos elementares:

1. $\{e_1, e_2, e_3\} \Leftarrow \text{Dor_de_Cabeça}=\text{sim} \text{ e } \text{Dor_Muscular}=\text{sim}$
2. $\{e_4, e_6\} \Leftarrow \text{Dor_de_Cabeça}=\text{não} \text{ e } \text{Dor_Muscular}=\text{sim}$
3. $\{e_5\} \Leftarrow \text{Dor_de_Cabeça}=\text{não} \text{ e } \text{Dor_Muscular}=\text{não}$

Qualquer união finita de conjuntos elementares é chamada de *conjunto definível*. Na Tabela 3.1, o conjunto $\{e_1, e_2, e_3, e_5\}$ é definível através dos atributos *Dor_de_Cabeça* e *Dor_Muscular*, pois pode-se definir esse conjunto dizendo que qualquer membro desse conjunto é caracterizado pelos atributos *Dor_de_Cabeça*=sim e *Dor_Muscular*=sim, ou pelos atributos *Dor_de_Cabeça* = não e *Dor_Muscular*=não.

Através do conceito da relação de não-discernimento é simples definir atributos redundantes ou dispensáveis. Se um conjunto de atributos e seu sobreconjunto (*superset*) definem a mesma relação de não-discernimento (ou seja, os conjuntos elementares de ambas as relações são idênticos) então, qualquer atributo que pertence ao sobreconjunto e não pertence ao conjunto de atributos é *redundante*.

Como exemplo, seja o conjunto de atributos $\{\text{Dor_de_Cabeça}, \text{Temperatura}\}$, e $\{\text{Dor_de_Cabeça}, \text{Dor_Muscular}, \text{Temperatura}\}$ o sobreconjunto que contém todos os atributos da Tabela 3.1. Os conjuntos elementares da relação de não-discernimento definida pelo conjunto $\{\text{Dor_de_Cabeça}, \text{Temperatura}\}$, são os *conjuntos unitários* $\{e_1\}$, $\{e_2\}$, $\{e_3\}$, $\{e_4\}$, $\{e_5\}$ e $\{e_6\}$, que também são os conjuntos elementares definidos pela relação de não-discernimento do sobreconjunto de todos os atributos. Portanto, o atributo *Dor_Muscular* é redundante. Por outro lado, o conjunto $\{\text{Dor_de_Cabeça}, \text{Temperatura}\}$ não contém nenhum atributo redundante, pois os conjuntos elementares para os conjuntos de atributos $\{\text{Dor_de_Cabeça}\}$ e $\{\text{Temperatura}\}$ não são conjuntos unitários.

Um conjunto de atributos que não contém atributos redundantes é chamado de *conjunto mínimo* ou independente. Um conjunto P de atributos é um *reduto* (*reduct*) de um outro conjunto Q de atributos, se P é mínimo e as relações de não-discernimento, definidas por P e Q são as mesmas. Ou seja, os conjuntos elementares determinados pelas relações de não-discernimento definidas por P e Q são idênticos.

No exemplo da Tabela 3.1, o conjunto de atributos $\{\text{Dor_de_Cabeça}, \text{Temperatura}\}$ é um reduto do conjunto original de atributos $\{\text{Dor_de_Cabeça}, \text{Dor_Muscular}, \text{Temperatura}\}$.

atura}. Eliminando o atributo redundante, tem-se uma nova tabela de informação, baseada nesse reduto, representada pela Tabela 3.2.

Exemplos	Atributos		Decisão
	<i>Dor_de_Cabeça</i>	<i>Temperatura</i>	
e ₁	sim	normal	não
e ₂	sim	alta	sim
e ₃	sim	muito_alta	sim
e ₄	não	normal	não
e ₅	não	alta	não
e ₆	não	muito_alta	sim

Tabela 3.2: Tabela Reduzida de Informação

Analogamente aos conjuntos elementares associados somente aos atributos, define-se conjuntos elementares associados à decisão. Os conjuntos elementares associados à decisão são os subconjuntos do conjunto de todos os exemplos que possuem o mesmo valor de decisão. Esses subconjuntos são denominados de *conceitos*. Nas Tabelas 3.1 e 3.2, os conceitos são $\{e_1, e_4, e_5\}$ e $\{e_2, e_3, e_6\}$. O primeiro conceito corresponde ao conjunto de todos os pacientes que não tem gripe (*Gripe*=não), enquanto que o segundo o dos pacientes que tem gripe (*Gripe*=sim).

Analisando os conceitos e considerando os valores dos atributos da Tabela 3.2, uma questão importante surge:

Quando é possível dizer que determinado paciente tem gripe ou não, com base nos valores dos atributos?

Para responder essa questão, deve-se observar que em termos da Teoria de RS, a decisão *Gripe* depende dos atributos *Dor_de_Cabeça* e *Temperatura*, pois todos os conjuntos elementares formados da relação de não-discernimento associados com $\{Dor_de_Cabeça, Temperatura\}$ são subconjuntos de algum conceito. Na realidade, é possível induzir da Tabela 3.2 as seguintes regras:

Se	<i>Temperatura</i> = normal	então <i>Gripe</i> = não
Se <i>Dor_de_Cabeça</i> = não e	<i>Temperatura</i> = alta	então <i>Gripe</i> = não
Se <i>Dor_de_Cabeça</i> = sim e	<i>Temperatura</i> = alta	então <i>Gripe</i> = sim
Se	<i>Temperatura</i> = muito_alta	então <i>Gripe</i> = sim

Para exemplificar como a abordagem Rough Sets trata inconsistência, considere a Tabela 3.3 originada da Tabela 3.2 com a adição dos exemplos $\{e_7\}$ e $\{e_8\}$.

Os conjuntos elementares da relação de não-discernimento definida pelos atributos *Dor_de_Cabeça* e *Temperatura* são $\{e_1\}$, $\{e_2\}$, $\{e_3\}$, $\{e_4\}$, $\{e_5, e_7\}$ e $\{e_6, e_8\}$, enquanto

Exemplos	Atributos		Decisão
	<i>Dor_de_Cabeça</i>	<i>Temperatura</i>	
e ₁	sim	normal	não
e ₂	sim	alta	sim
e ₃	sim	muito_alta	sim
e ₄	não	normal	não
e ₅	não	alta	não
e ₆	não	muito_alta	sim
e ₇	não	alta	sim
e ₈	não	muito_alta	não

Tabela 3.3: Tabela de Informação Inconsistente

que os conceitos definidos pela decisão *Gripe* são os conjuntos $\{e_1, e_4, e_5, e_8\}$ e $\{e_2, e_3, e_6, e_7\}$.

Na Tabela 3.3 a decisão *Gripe* não depende dos atributos *Dor_de_Cabeça* e *Temperatura*, pois $\{e_5, e_7\}$ e $\{e_6, e_8\}$ não são subconjuntos de qualquer conceito. Em outras palavras, nenhum conceito pode ser definido pelo conjunto de atributos $\{Dor_de_Cabeça, Temperatura\}$. Nesse caso, diz-se que a Tabela 3.3 é *inconsistente* porque os exemplos e_5 e e_7 são *conflitantes* (ou são inconsistentes) — para ambos os exemplos os valores dos atributos são os mesmos, porém o valor da decisão é diferente. O mesmo ocorre com os exemplos e_6 e e_8 .

Nessa situação, RS oferece uma ferramenta simples para lidar com inconsistência. Para cada conceito X , o maior conjunto definível (*i.e.*, união finita de conjuntos elementares) contido em X e o menor conjunto definível que contém X são computados. O primeiro conjunto é chamado de *aproximação superior* de X , enquanto que o segundo conjunto é chamado de *aproximação inferior* de X . Na Tabela 3.3, para o conceito $X = \{e_2, e_3, e_6, e_7\}$, que descreve pessoas com gripe, a aproximação inferior é definida pelo conjunto $\{e_2, e_3\}$, e a aproximação superior é definida pelo conjunto $\{e_2, e_3, e_5, e_6, e_7, e_8\}$, tal como esquematizado na Figura 3.1.

Analogamente, para o conceito $X = \{e_1, e_4, e_5, e_8\}$, a aproximação inferior é $\{e_1, e_4\}$ e a aproximação superior é $\{e_1, e_4, e_5, e_6, e_7, e_8\}$. Qualquer um desses dois conceitos é um exemplo clássico de Rough Sets, ou seja, um conjunto que não pode ser claramente definido utilizando os atributos fornecidos, neste caso o conjunto de atributos $\{Dor_de_Cabeça, Temperatura\}$.

Para o conceito $X = \{e_2, e_3, e_6, e_7\}$, o conjunto $\{e_5, e_6, e_7, e_8\}$, o qual contém elementos da aproximação superior de X que não são elementos da aproximação inferior de X , é chamado de *região de borda*. Os elementos da região de borda não podem ser classificados como membros do conjunto X . Com isso, uma definição alternativa para

Figura 3.1: Aproximações Superior e Inferior para o conceito X

Rough Sets seria dizer que são conjuntos que não possuem região de borda vazia.

Para qualquer conceito, regras induzidas utilizando sua aproximação inferior são certamente válidas — chamadas de *regras certas* — enquanto que regras induzidas pela aproximação superior do conceito são possivelmente válidas — chamadas *regras possíveis*. Para a Tabela 3.3, as regras certas são:

Se	<i>Temperatura</i> = normal	então <i>Gripe</i> = não
Se <i>Dor_de_Cabeça</i> = sim e	<i>Temperatura</i> = alta	então <i>Gripe</i> = sim
Se <i>Dor_de_Cabeça</i> = sim e	<i>Temperatura</i> = muito_alta	então <i>Gripe</i> = sim

e as regras possíveis são:

Se <i>Dor_de_Cabeça</i>	= não	então <i>Gripe</i> = não
Se <i>Temperatura</i>	= normal	então <i>Gripe</i> = não
Se <i>Temperatura</i>	= alta	então <i>Gripe</i> = sim
Se <i>Temperatura</i>	= muito_alta	então <i>Gripe</i> = sim

Algumas medidas de incerteza foram desenvolvidas dentro da Teoria de Rough Sets com o objetivo de medir a qualidade das aproximações realizadas. As mais frequentemente utilizadas são:

- a qualidade da aproximação inferior, e
- a qualidade da aproximação superior.

Para um dado conjunto X de exemplos, não necessariamente definíveis pelo conjunto P de atributos, a qualidade da aproximação inferior é o coeficiente resultante do número de elementos na aproximação inferior de X , dividido pelo número total de exemplos. Similarmente, a qualidade da aproximação superior é o coeficiente resultante do número de elementos na aproximação superior de X , dividido pelo número total de exemplos. No exemplo da Tabela 3.3, para o conceito $X = \{e_2, e_3, e_6, e_7\}$, a qualidade da aproximação inferior é 0.25 e a qualidade da aproximação superior é 0.75, como mostrado a seguir.

$$\text{qualidade(Aproximação Inferior)} = \frac{|\{e_2, e_3\}|}{|\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}|} = \frac{2}{8} = 0.25$$

$$\text{qualidade(Aproximação Superior)} = \frac{|\{e_2, e_3, e_5, e_6, e_7, e_8\}|}{|\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}|} = \frac{6}{8} = 0.75$$

A qualidade da aproximação inferior pode ser interpretada como o coeficiente do número total de exemplos certamente classificados pelos atributos em P como pertencentes a X , dividido pelo número total de exemplos na tabela de informação. Caracteriza-se assim, um tipo de frequência relativa. Além disso, a qualidade da aproximação inferior é uma *função de crença (belief function)* de acordo com a teoria proposta por Dempster-Shafer. Também, a qualidade da aproximação superior pode ser interpretada como o coeficiente do número total de exemplos possivelmente classificados pelo conjunto de atributos P como pertencentes a X , dividido pelo número total de exemplos na tabela de informação. Segundo o ponto de vista da teoria proposta por Dempster-Shafer, a qualidade da aproximação superior é uma *função de possibilidade (plausibility function)*. A teoria envolvida em RS é objetiva, ou seja, para uma dada tabela de informação, as qualidades das correspondentes aproximações são calculadas. No entanto, a teoria de Dempster-Shafer é subjetiva, pois assume-se que os valores de crença (ou possibilidade) são atribuídos por um especialista (Pawlak et al., 1995).

3.4 Áreas de Aplicação

A Teoria de RS tem-se mostrado muito útil quando aplicada a problemas do mundo real. A sua aplicabilidade depende da disponibilidade de informações a respeito do

domínio. Os maiores problemas abordados utilizando a Teoria de Rough Sets incluem, entre outros:

- redução de dados (eliminação de dados supérfluos);
- descoberta de dependência entre os dados;
- descoberta de similaridade ou diferença entre os dados;
- descoberta de padrões nos dados;
- descoberta de relações de causa-efeito.

Em particular, a abordagem de RS tem sido aplicada em medicina (Komorowski & Øhrn, 1999) e (Tsomoto, 1998), farmacologia, negócios (Beaubouef & Lang, 1998), pesquisa de mercado, engenharia, meteorologia, análise de conflitos (Pawlak, 1998), problemas de controle, processamento de imagens, reconhecimento de voz, análise de sistemas concorrentes, reconhecimento de caracteres, entre outros campos do conhecimento (Pawlak et al., 1995).

3.5 Rough Sets e Aprendizado de Máquina

O conhecimento na forma de regras, induzido pelo aprendizado através de exemplos, pode ser utilizados em sistemas especialistas. Essas regras são mais gerais que as informações contidas na tabela de informação, já que novos exemplos diferentes dos contidos na tabela de informação inicial podem ser corretamente classificados por essas regras.

Um sistema baseado no aprendizado por exemplos chamado de Learning from Examples based on Rough Sets — LERS⁶ — foi desenvolvido na Universidade de Kansas, o qual consiste de duas opções de Aprendizado de Máquina por exemplos, e duas opções de aquisição de conhecimento (Pawlak et al., 1995). É interessante observar que nesse sistema, as opções de Aprendizado de Máquina produzem um número suficiente de regras para cobrir os exemplos utilizados no treinamento, enquanto que as opções de aquisição de conhecimento produzem uma quantidade de regras muito maior devido à combinação das entradas. Manipular incerteza presente na tabela de informação é um problema importante na área de AM. Uma das opções de aprendizado de máquina do LERS inclui o tratamento de incerteza utilizando Rough Sets.

⁶Não disponível para avaliação.

O sistema LERS foi utilizado durante dois anos no Centro Espacial Johnson na NASA, como uma ferramenta para o desenvolvimento de sistemas especialistas para o auxílio de tomada de decisões médicas a bordo da estação espacial Freedom. Outra aplicação do LERS foi feita para prever o nascimento pré-maturo de seres humanos, conseguindo uma taxa de acerto entre 68%–90%, contra 17%–38% dos métodos convencionais presentes nos manuais médicos.

3.6 Rough Sets e Descoberta de Conhecimento

Conforme citado anteriormente, a metodologia de RS cobre uma gama muito grande de problemas. Uma importante aplicação é a Descoberta de Conhecimento em Bases de Dados (Knowledge Discovery in Databases) (Fayyad et al., 1996). Essa é uma área nova dentro de IA que tenta extrair conhecimento novo e não-trivial de grandes massas de dados. Uma das principais tarefas é a descoberta e caracterização das relações existentes entre as informações dos bancos de dados, as quais, na maior parte dos casos, estão estruturadas da mesma forma que as tabelas de informação. Por exemplo, as relações existentes entre os sintomas e as doenças no domínio médico. A descoberta dessas relações ajuda os especialistas do domínio a compreender os fenômenos presentes nas informações, e até mesmo na previsão de novos casos. Outro aspecto utilizando a abordagem RS é a descoberta de comportamentos ou padrões anormais nos dados, detectando assim fraudes e intrusões.

Essa abordagem tem sido utilizada para o propósito de KDD, particularmente com o surgimento de programas como o *Datalogic*⁷, que tornaram essa tecnologia acessível para diferentes setores da ciência e indústria (Pawlak et al., 1995).

3.7 Considerações Finais

Tem-se mostrado que a Teoria de Rough Sets é útil em muitas aplicações do mundo real e oferece métodos efetivos que são aplicáveis em muitas áreas de IA (Sistemas Especialistas, AM, KDD, entre outras). Uma das vantagens dessa teoria é que programas que implementam essa metodologia podem facilmente ser executados em máquinas paralelas.

Contudo, muitos problemas continuam sem solução. Embora essa teoria seja de-

⁷<http://www.reduct.com>

senrolvida sobre sólidas fundamenteções matemáticas, muitos problemas teóricos ainda precisam ser solucionados. *Rough logic* — uma lógica para raciocínio impreciso baseada na filosofia de Rough Sets parece ser o tópicó mais importante entre os problemas em aberto. O desenvolvimento de métodos baseados na Teoria de Rough Sets para redes neurais e algoritmos genéticos, bem como *Controladores Rough*, i.e., controladores baseados na Teoria de Rough Sets seguindo a linha de aplicação de conjuntos fuzzy, mostram-se temas promissores para serem investigados (Pawlak et al., 1995).

No próximo capítulo será apresentada a base teórica da Teoria de Rough Sets, bem como as metodologias para se avaliar a qualidade das aproximações obtidas e do conhecimento extraído.

Capítulo 4

Rough Sets: Teoria

4.1 Considerações Iniciais

A Teoria Rough Sets é motivada pela necessidade prática de interpretar, caracterizar, representar e processar o não-discernimento entre elementos. Em outras palavras, RS manipulam a incerteza presente nos dados e nos conceitos (classes). A premissa central na filosofia de Rough Sets é que o conhecimento consiste na habilidade de classificar objetos (Slowinski, 1995). Por exemplo, se um grupo de pacientes é descrito utilizando vários sintomas, então muitos pacientes compartilham os mesmos sintomas e, portanto, são indistinguíveis em relação a esses sintomas. Rough Sets fornecem um método sistemático para representar e processar conceitos vagos causados pela falta de discernimento em situações com informações incompletas ou falta de conhecimento. Embora a teoria seja poderosa o suficiente para manipular a incerteza presente nos dados, algumas extensões foram feitas a essa teoria. A Teoria de Rough Sets é aplicada a sistemas de informação bidimensionais formados por exemplos no formato atributo-valor. Um tipo especial de sistema de informação é o sistema de decisão, no qual os exemplos são agrupados em classes devido a suas características semelhantes. Neste capítulo serão apresentadas as definições de sistemas de informação e de decisão, bem como a teoria originalmente proposta por (Pawlak, 1982) e algumas extensões.

4.2 Sistemas de Informação

A forma mais comum para representação dos dados na abordagem RS é um *sistema de informação*, o qual contém um conjunto de *objetos*. Cada objeto tem um número de *atributos* com *valores* relacionados a cada um desses atributos. Os atributos são os mesmos para todos os objetos, mas os valores dos atributos podem diferir. Portanto, um sistema de informação é semelhante a um banco de dados relacional.

Definição 4.2.1 (Sistema de Informação) Um Sistema de Informação — SI — é um par ordenado $\mathcal{A} = (U, A)$ onde U é um conjunto finito e não-vazio de objetos chamado de Universo, e A é um conjunto finito e não-vazio de elementos chamado de Atributos. Os elementos do Universo serão referenciados como objetos. Cada atributo $a \in A$ é uma função total $a : U \rightarrow V_a$, onde V_a é o conjunto dos valores permitidos para o atributo a (sua faixa de valores).

A Tabela 4.1 mostra um exemplo de Sistema de Informação.

Exemplos	Atributos		
	Estudos	Educação	Trabalha
e_1	não	boa	sim
e_2	não	boa	sim
e_3	sim	boa	sim
e_4	não	pobre	não
e_5	não	pobre	não

Tabela 4.1: Sistema de Informação

Em muitas aplicações é possível classificar esses objetos utilizando conhecimento *a posteriori* expresso por um atributo especial denominado *atributo de decisão*, sendo esse processo conhecido como aprendizado supervisionado. Sistemas de informação nessa categoria são denominados *Sistemas de Decisão*.

Definição 4.2.2 (Sistema de Decisão) Um Sistema de Decisão — SD — é qualquer SI da forma $\mathcal{A} = (U, A \cup \{d\})$, onde $d \notin A$ é o atributo de decisão. Os elementos de A são chamados de atributos condicionais ou simplesmente condições.

Um exemplo de sistema de decisão relacionado ao sistema de informação da Tabela 4.1 é mostrado na Tabela 4.2. Tal como esperado, é uma tabela bidimensional. As linhas representam os objetos, enquanto que as colunas representam os valores dos atributos desses objetos.

Exemplos	Atributos			Decisão
	<i>Estudos</i>	<i>Educação</i>	<i>Trabalha</i>	<i>Renda</i>
e ₁	não	boa	sim	alta
e ₂	não	boa	sim	alta
e ₃	sim	boa	sim	nenhuma
e ₄	não	pobre	não	baixa
e ₅	não	pobre	não	média

Tabela 4.2: Sistema de Decisão

Nesse SD existem 5 pessoas (objetos) com atributos refletindo características da vida de cada pessoa. O atributo *Renda* é o atributo de decisão (atributo dependente). Os demais atributos, *Estudos*, *Educação* e *Trabalha* são os atributos de condição (atributos independentes). A intenção é descobrir regras para a predição da renda esperada de uma pessoa com base nos valores dos atributos condicionais que descrevem essa pessoa.

4.3 Distinguindo Objetos

A próxima definição introduz o conceito de *relação de não-discernimento*. Se tal relação existe entre dois objetos, isso significa que todos os valores de seus atributos são idênticos com respeito aos atributos sendo considerados, portanto não podem ser discernidos (distinguidos) entre si considerando esses atributos.

4.3.1 Relação de Não-Discernimento

Um sistema de decisão (*i.e.*, uma tabela de decisão) expressa todo o conhecimento sobre o modelo¹. Essa tabela pode ser desnecessariamente grande, em parte porque ela é redundante pelo menos de duas formas:

1. O mesmo objeto ou os objetos indistinguíveis podem estar representados várias vezes;
2. Alguns dos atributos podem ser supérfluos.

Uma relação binária $R \subseteq X \times X$, a qual é reflexiva (*i.e.*, um objeto está relacionado com ele próprio xRx), simétrica (se xRy então yRx) e transitiva (se xRy e yRx então

¹Na Teoria de Rough Sets admi-ti-se que um SD expressa todo o conhecimento do domínio, embora saiba-se que nem sempre isso ocorra nas tabelas de decisão contendo dados reais. Este é um problema clássico de amostragem e balanceamento dos dados (Batista, 2000)

xRz), é chamada de *relação de equivalência*. A *classe de equivalência* de um elemento $x \in X$ consiste de todos os objetos $y \in X$ para os quais xRy .

Definição 4.3.1 (Relação de Não-Discernimento) Para cada subconjunto de atributos $B \subseteq A$ no $SI A = (U, A)$, uma relação de equivalência $IND_A(B)$ é associada, chamada de Relação de Não-Discernimento, e é definida como segue:

$$IND_A(B) = \{(x, y) \in U^2 \mid \forall a \in B, a(x) = a(y)\}$$

na qual $IND_A(B)$ é chamado de relação de não-discernimento- B^2 . O conjunto de todas as classes de equivalência na relação $IND_A(B)$ é denotado por $U/IND_A(B)$.

O subscrito A na relação de não-discernimento é usualmente omitido no caso de não haver dúvida sobre qual o sistema de informação que está sendo referenciado.

Para o sistema de informação representado na Tabela 4.1, os possíveis subconjuntos não-vazios dos atributos condicionais são: $\{Estudos\}$, $\{Educação\}$, $\{Trabalha\}$, $\{Estudos, Educação\}$, $\{Estudos, Trabalha\}$, $\{Educação, Trabalha\}$ e $\{Estudos, Educação, Trabalha\}$. Considerando por exemplo o subconjunto $\{Educação\}$, os objetos e_1, e_2 e e_3 estão na mesma classe de equivalência e são não-discerníveis, assim como os objetos e_4 e e_5 . Assim, $U/IND(B)$ para cada um dos sete possíveis subconjuntos $B \subseteq A$ é:

$$\begin{aligned} U/IND(\{Estudos\}) &= \{\{e_1, e_2, e_4, e_5\}, \{e_3\}\} \\ U/IND(\{Educação\}) &= \{\{e_1, e_2, e_3\}, \{e_4, e_5\}\} \\ U/IND(\{Trabalha\}) &= \{\{e_1, e_2, e_3\}, \{e_4, e_5\}\} \\ U/IND(\{Estudos, Educação\}) &= \{\{e_1, e_2\}, \{e_3\}, \{e_4, e_5\}\} \\ U/IND(\{Estudos, Trabalha\}) &= \{\{e_1, e_2\}, \{e_3\}, \{e_4, e_5\}\} \\ U/IND(\{Educação, Trabalha\}) &= \{\{e_1, e_2, e_3\}, \{e_4, e_5\}\} \\ U/IND(\{Estudos, Educação, Trabalha\}) &= \{\{e_1, e_2\}, \{e_3\}, \{e_4, e_5\}\} \end{aligned}$$

Pode-se notar que para cada subconjunto de atributos os objetos são agrupados e os grupos consistem de objetos que não podem ser discernidos entre si quando utilizado esse subconjunto de atributos. Segundo a Teoria de RS, cada um desses grupos é uma classe. Por exemplo, as classes para o subconjunto $\{Estudos, Educação, Trabalha\}$ estão representadas na Tabela 4.3. A classe E_1 originou-se dos objetos e_1 e e_2 , a classe E_2 originou-se do objeto e_3 e a classe E_3 originou-se dos objetos e_4 e e_5 . Note ainda que a classe E_3 possui dois objetos com diferentes valores no atributo de decisão.

² B -indiscernibility relation.

Classes	Atributos		
	<i>Estudos</i>	<i>Educação</i>	<i>Trabalha</i>
E_1	não	boa	sim
E_2	sim	boa	sim
E_3	não	pobre	não

Tabela 4.3: Classes para $B = \{\textit{Estudos}, \textit{Educação}, \textit{Trabalha}\}$

4.3.2 Matriz de Discernimento

Uma *Matriz de Discernimento* é uma matriz na qual as classes são índices e os atributos condicionais que podem ser utilizados para distinguir entre as classes são inseridos na linha e coluna correspondente às classes a serem discernidas.

Definição 4.3.2 (Matriz de Discernimento) Para um conjunto de atributos $B \subseteq A$ em $\mathcal{A} = (U, A)$, a Matriz de Discernimento é dada por $M_D(B) = \{m_D(i, j)\}_{n \times n}$, $1 \leq i, j \leq n$, com $n = |U/IND(B)|$, onde

$$m_D(i, j) = \{a \in B \mid a(E_i) \neq a(E_j)\} \text{ para } i, j = 1, 2, \dots, n$$

O elemento $m_D(i, j)$ na matriz de discernimento é o conjunto de atributos de B que *discerne (distingue)* as classes de objetos $E_i, E_j \in U/IND(B)$.

Para a Tabela 4.3, pode-se observar que o único atributo com valor diferente para as classes E_1 e E_2 é *Estudos*. Esse atributo é então colocado na posição correta (1,2) na matriz. No caso das classes E_1 e E_3 são dois os atributos com valores diferentes, *i.e.* $\{\textit{Educação}, \textit{Trabalha}\}$, e ambos devem ser colocados na posição (1,3) da matriz de discernimento. Naturalmente, a matriz será simétrica pelo fato dos atributos que diferem em valor para os objetos a e b , também diferem da mesma maneira para os objetos b e a . A Tabela 4.4 mostra a matriz de discernimento correspondente às classes na Tabela 4.3³.

	E_1	E_2	E_3
E_1	—	<i>Estudos</i>	<i>Educação, Trabalha</i>
E_2	<i>Estudos</i>	—	<i>Estudos, Educação, Trabalha</i>
E_3	<i>Educação, Trabalha</i>	<i>Estudos, Educação, Trabalha</i>	—

Tabela 4.4: Matriz de Discernimento

Se alguma das classes tiver o mesmo valor de decisão, pode-se decidir não discernir entre essas classes. Fazendo isso, os atributos relacionados às classes que possuam

³Ainda que os elementos da matriz de discernimento são conjuntos, a notação utilizada na bibliografia de Rough Sets é a da Tabela 4.4.

o mesmo valor de decisão não serão adicionados à matriz. Isso pode resultar em regras mais simples se alguma classe tiver o mesmo valor de decisão. No exemplo apresentado essa opção não é válida, pois todas as classes possuem valores de decisão diferentes entre si.

4.3.3 Função de Discernimento

Definição 4.3.3 (Função de Discernimento) A Função de Discernimento $f(B)$ de um conjunto de atributos $B \subseteq A$ de um sistema de informação é a função booleana

$$f(B) = \bigwedge_{i,j \in \{1, \dots, n\}} \bigvee \overline{m}_D(E_i, E_j)$$

onde $n = |U/IND(B)|$, e $\bigvee \overline{m}_D(E_i, E_j)$ é a disjunção sobre o conjunto de variáveis booleanas $\overline{m}_D(E_i, E_j)$ que correspondem ao elemento $m_D(i, j)$ da matriz de discernimento.

A Função Relativa de Discernimento $f(E, B)$ de uma classe de objetos E e atributos $B \subseteq A$ é a função booleana

$$f(E, B) = \bigwedge_{j \in \{1, \dots, n\}} \bigvee \overline{m}_D(E, E_j)$$

onde $n = |U/IND(B)|$.

Isso implica que a função de discernimento $f(B)$ computa o conjunto mínimo de atributos necessários para discernir qualquer classe de equivalência de todas as demais. Similarmente, a função relativa de discernimento $f(E, B)$ computa o conjunto mínimo de atributos necessários para discernir uma dada classe E das demais classes.

Para o exemplo considerado, com $B = \{\text{Estudos}, \text{Educação}, \text{Trabalha}\}$ e cuja matriz de discernimento é apresentada na Tabela 4.4, a função de discernimento é a função booleana

$$f(B) = \text{Estudos} \wedge (\text{Educação} \vee \text{Trabalha}) \wedge (\text{Estudos} \vee \text{Educação} \vee \text{Trabalha})$$

após simplificação

$$f(B) = \text{Estudos} \wedge (\text{Educação} \vee \text{Trabalha})$$

$$\begin{aligned}
f(E_1, B) &= Estudos \wedge (Educação \vee Trabalha) \\
f(E_2, B) &= Estudos \wedge (Estudos \vee Educação \vee Trabalha) \\
f(E_3, B) &= (Educação \vee Trabalha) \wedge (Estudos \vee Educação \vee Trabalha)
\end{aligned}$$

Com as seguintes funções relativas de discernimento:

Definição 4.3.4 (Dispensável) *Um atributo a é dispensável ou supérfluo ou redundante em $B \subseteq A$ se $IND(B) = IND(B - \{a\})$, caso contrário é indispensável em B . Se todos os atributos $a \in B$ são indispensáveis em B , então B é chamado ortogonal.*

Considerando o exemplo da Tabela 4.1, no qual $B = \{Estudos, Educação, Trabalha\}$, nota-se no conjunto de classes que os valores dos atributos *Educação* e *Trabalha* propagam-se juntos. Sempre que *Educação* é boa, *Trabalha* é sim e sempre que *Educação* é pobre, *Trabalha* é não. Portanto, $IND(B) = IND(B - \{Trabalha\}) = IND(B - \{Educação\})$. Nesse exemplo o único atributo indispensável é *Estudos*.

4.3.4 Redução da Representação

Os dados em um sistema de informação podem ser utilizados para discernir classes somente até um certo grau. Contudo, nem todos os atributos podem ser necessários para desempenhar essa tarefa. Em razão desse fato, a próxima definição é importante.

Definição 4.3.5 (Reduto, Reduto Relativo) *Um Reduto de B é um conjunto de atributos $B' \subseteq B$ tal que todos os atributos $a \in B - B'$ são dispensáveis e $IND(B') = IND(B)$. O termo $RED(B)$ é utilizado para denotar a família de redutos de B . O conjunto de primos implicantes (prime implicants)⁴ da função de discernimento $f(B)$ determina os redutos de B .*

O conjunto de primos implicantes da função relativa de discernimento $f(E, B)$ determina os redutos relativos de B . O termo $RED(E, B)$ denota a família de redutos relativos de B para uma classe de objetos E .

⁴Um implicante de uma função booleana f é uma conjunção de literais (variáveis ou suas negações) tal que se os valores desses literais são verdade sob uma avaliação arbitrária v de variáveis, então o valor da função f sob v também será verdade. Um primo implicante é o implicador mínimo. Neste trabalho o interesse está somente em implicantes de funções booleanas monotônicas, i.e., funções construídas sem negação.

Isso implica que um reduto relativo contém informação suficiente para discernir os objetos de uma classe em relação a todas as demais classes em um sistema de informação.

Para encontrar os redutos relativos do exemplo considerado, a função de discernimento é utilizada. Cada função é minimizada no formato de soma de produtos, como mostrado na Tabela 4.5. Isso resulta em redutos relativos, por exemplo, $RED(E_1, B) = \{\{Estudos, Educação\}, \{Estudos, Trabalha\}\}$. Os redutos relativos são mínimos porque cada função de discernimento foi minimizada. Um reduto relativo mínimo é portanto um reduto no qual nenhum dos atributos pode ser removido sem modificar as propriedades do reduto (Solheim & Øyvind Tuset Aasheim, 1996).

$ \begin{aligned} f(E_1, B) &= Estudos \wedge (Educação \vee Trabalha) \\ &= (Estudos \wedge Educação) \vee (Estudos \wedge Trabalha) \\ f(E_2, B) &= Estudos \wedge (Estudos \vee Educação \vee Trabalha) \\ &= Estudos \\ f(E_3, B) &= (Educação \vee Trabalha) \wedge (Estudos \vee Educação \vee Trabalha) \\ &= Educação \wedge Trabalha \end{aligned} $
--

Tabela 4.5: Redutos Relativos

O conceito de reduto pode ser melhor entendido através do exemplo na Figura 4.1. A relação de não-discernimento definida pelo conjunto de atributos $\{a, b, c, d\}$, pode ser definida como a união das relações de não-discernimento definidas pelos atributos $\{a\}$, $\{b\}$, $\{c\}$ e $\{d\}$ separadamente. No entanto, não há necessidade da inclusão de todos esses atributos para definir a relação por completo. A mesma relação de não-discernimento pode ser obtida por qualquer um dos conjuntos de atributos: $\{a, b\}$ ou $\{c, d\}$. Com isso, $IND(\{a, b, c, d\}) = IND(\{a, b\}) = IND(\{c, d\})$. Portanto, os conjuntos de atributos $\{a, b\}$ e $\{c, d\}$ são redutos do conjunto de atributos $\{a, b, c, d\}$. A intersecção de todos os redutos define o *núcleo* (*core*), o qual contém todos os atributos que são importantes para definir todas as relações de não-discernimento.

Deve ser observado que computar classes de equivalência é um processo simples. Entretanto, encontrar redutos mínimos, ou seja, redutos que têm cardinalidade mínima entre todos os redutos, é um problema *NP-hard*. Na realidade, o cálculo de redutos é considerado o maior problema na abordagem de Rough Sets. Afortunadamente, existem algumas heurísticas que permitem computar um número suficiente de redutos em tempo aceitável, sempre que o número de atributos não for muito grande (Komorowski et al., 1999).

Figura 4.1: Dois redutos definidos pelos atributos $\{a,b\}$ e $\{c,d\}$. Ambos definem a mesma relação de não-discernimento definida pelos atributos $\{a,b,c,d\}$

4.3.5 Aproximação de Conjuntos

Uma relação de equivalência induz um particionamento do universo (o conjunto de casos no exemplo considerado). Essas partições podem ser utilizadas para construir novos subconjuntos do universo. Os subconjuntos que são de maior interesse são aqueles que têm o mesmo valor do atributo de decisão. No entanto, pode acontecer que um conceito como *Renda* não possa ser definido de uma maneira *crisp*⁵. Por exemplo, o conjunto de pessoas com *Renda* baixa não pode ser definido de maneira crisp utilizando os atributos presentes na Tabela 4.2. Os objetos problemáticos são e_4 e e_5 , por não permitir uma descrição precisa sobre a decisão utilizando os atributos condicionais. São nesses casos que a noção de Rough Sets emerge. Embora não seja possível definir esses objetos de uma forma crisp, em geral, pode-se definir objetos que certamente possuem ou não uma dada propriedade e objetos que pertencem à região de borda entre os casos corretos. Se essa borda é não-vazia, o conjunto é rough. Essas noções são formalmente expressadas a seguir.

Definição 4.3.6 (Aproximação Inferior, Aproximação Superior) A Aproximação Inferior $\underline{B}X$ e a Aproximação Superior $\overline{B}X$ de um conjunto de objetos $X \subseteq U$ com respeito a um conjunto de atributos $B \subseteq A$ (definindo uma relação de equivalência em U) pode ser definido em termos das classes na relação de equivalência, da seguinte forma:

$$\underline{B}X = \bigcup \{E \in U/IND(B) \mid E \subseteq X\}$$

$$\overline{B}X = \bigcup \{E \in U/IND(B) \mid E \cap X \neq \emptyset\}$$

chamadas de aproximações B -inferior e B -superior de X , respectivamente. A região

⁵Os conjuntos *crisp* são os conjuntos convencionais que fundamental a matemática.

$BN_B(X) = \overline{B}X - \underline{B}X$ é chamada de *B-borda* de X ⁶.

A aproximação inferior de X é o conjunto de objetos que podem ser classificados com total certeza como membros do conjunto X utilizando o conjunto de atributos B . Similarmente, a aproximação superior de X são os objetos que podem ser classificados como membros do conjunto X utilizando o conjunto de atributos B . A região de borda possui os objetos que não podem ser classificados, com certeza, como pertencentes ou não a X utilizando o conjunto de atributos B .

Definição 4.3.7 (Rough) Um conjunto é chamado de rough se a região de borda é não-vazia.

Definição 4.3.8 (Crisp) Um conjunto é chamado de crisp se a região de borda é vazia.

Através dos conceitos de aproximação superior e inferior, algumas propriedades podem ser definidas conforme (Pawlak, 1996). Sejam quaisquer dois subconjuntos $X, Y \subseteq U$, então:

1. $\underline{B}(X) \subseteq X \subseteq \overline{B}(X)$
2. $\underline{B}(\emptyset) = \overline{B}(\emptyset) = \emptyset$
3. $\underline{B}(U) = \overline{B}(U) = U$
4. $\underline{B}(X \cap Y) = \underline{B}(X) \cap \underline{B}(Y)$
5. $\overline{B}(X \cap Y) \subseteq \overline{B}(X) \cap \overline{B}(Y)$
6. $\underline{B}(X \cup Y) \supseteq \underline{B}(X) \cup \underline{B}(Y)$
7. $\overline{B}(X \cup Y) = \overline{B}(X) \cup \overline{B}(Y)$
8. $\underline{B}(-X) = -\overline{B}(X)$
9. $\overline{B}(-X) = -\underline{B}(X)$
10. $\underline{B}(\underline{B}(X)) = \underline{B}(\overline{B}(X)) = \underline{B}(X)$
11. $\underline{B}(\overline{B}(X)) = \overline{B}(\overline{B}(X)) = \overline{B}(X)$

⁶A letra B refere-se ao subconjunto B de atributos de A . Se um outro subconjunto for escolhido, por exemplo $F \subseteq A$, os correspondentes nomes das relações são *aproximações F-inferior, F-superior e região de F-borda*.

$$12. X \subseteq Y \implies \underline{B}(X) \subseteq \underline{B}(Y) \text{ e } \overline{B}(X) \subseteq \overline{B}(Y)$$

onde $-X$ denota $U - X$ (complemento).

Pode-se definir as seguintes quatro classes básicas de Rough Sets, ou seja, quatro categorias de imprecisão (Komorowski et al., 1999):

1. X é *rough B-definível*, se e somente se $\underline{B}(X) \neq \emptyset$ e $\overline{B}(X) \neq U$;
2. X é *internamente B-indefinível*, se e somente se $\underline{B}(X) = \emptyset$ e $\overline{B}(X) \neq U$;
3. X é *externamente B-indefinível*, se e somente se $\underline{B}(X) \neq \emptyset$ e $\overline{B}(X) = U$;
4. X é *totalmente B-indefinível*, se e somente se $\underline{B}(X) = \emptyset$ e $\overline{B}(X) = U$.

O significado intuitivo dessas categorias é o seguinte:

1. Se X é *rough B-definível*, isso significa que é possível decidir para alguns elementos de U quando eles pertencem a X e para alguns elementos de U quando eles pertencem a $-X$, utilizando B ;
2. Se X é *internamente B-indefinível*, isso significa que é possível decidir para alguns elementos de U quando eles pertencem a $-X$, mas não é possível decidir para nenhum elemento de U quando ele pertence a X , utilizando B ;
3. Se X é *externamente B-indefinível*, isso significa que é possível decidir para alguns elementos de U quando eles pertencem a X , mas não é possível decidir para qualquer elemento de U quando ele pertence a $-X$, utilizando B ;
4. Se X é *totalmente B-indefinível*, isso significa que não é possível decidir para qualquer elemento de U quando ele pertence a X ou a $-X$, utilizando B .

4.3.6 Qualidade das Aproximações

As aproximações obtidas a partir das definições previamente feitas podem ter sua qualidade medida em termos dos próprios elementos que as definem. Três importantes medidas de qualidade são:

1. coeficiente de incerteza;
2. qualidade da aproximação superior;

3. qualidade da aproximação inferior.

Seja $0 \leq \alpha_B \leq 1$ o coeficiente para medir as qualidades anteriormente citadas, então elas podem ser definidas respectivamente como:

$$\alpha_B(X) = \frac{|\underline{B}(X)|}{|\overline{B}(X)|} \quad \alpha_B(\overline{B}(X)) = \frac{|\overline{B}(X)|}{|U|} \quad \alpha_B(\underline{B}(X)) = \frac{|\underline{B}(X)|}{|U|}$$

sendo $|Q|$ a cardinalidade do conjunto Q , $Q \neq \emptyset$. Se $\alpha_B(X) = 1$, o conjunto X é crisp com respeito a B . Por outro lado, se $\alpha_B(X) < 1$, o conjunto X é vago com respeito a B . Portanto, o coeficiente $\alpha_B(X)$ pode ser entendido como a qualidade da aproximação do conceito X . O coeficiente $\alpha_B(\overline{B}(X))$ pode ser interpretado como o percentual de todos os exemplos possivelmente classificados como pertencentes a X , sendo ainda similar à *função de possibilidade* de acordo com a teoria de Dempster-Shafer. O coeficiente $\alpha_B(\underline{B}(X))$ pode ser interpretado como o percentual de todos os elementos certamente pertencentes à X , sendo ainda similar a *função de crença* de acordo com a teoria de Dempster-Shafer.

4.4 De Redutos para Regras

Regras representam dependências no conjunto de dados e conhecimento extraído, as quais podem ser utilizadas para classificar novos objetos que não estavam presentes no sistema de decisão original. Quando os redutos são encontrados, o trabalho de se definir regras para os valores de decisão com base nos atributos condicionais está praticamente feito. Para transformar um reduto (relativo ou não) em regras, deve-se somente unir os valores dos atributos condicionais da classe de objetos da qual foi originado o reduto com os atributos correspondentes ao reduto. Então, para completar a regra, a decisão é adicionada ao final da regra. As regras para o exemplo da Tabela 4.5 são:

$$\begin{aligned} E_1 : & \quad \text{Estudos} = \text{não} \wedge \text{Educação} = \text{boa} \longrightarrow \text{Renda} = \text{alta} \\ & \quad \text{Estudos} = \text{não} \wedge \text{Trabalha} = \text{sim} \longrightarrow \text{Renda} = \text{alta} \\ E_2 : & \quad \text{Estudos} = \text{sim} \longrightarrow \text{Renda} = ? \\ E_3 : & \quad \text{Educação} = \text{pobre} \longrightarrow \text{Renda} = ? \\ & \quad \text{Trabalha} = \text{não} \longrightarrow \text{Renda} = ? \end{aligned}$$

As regras derivadas com base em E_3 não especificam o valor do atributo *Renda*, pois o valor desse atributo não é o mesmo para todos os objetos da classe. Ele pode ser

chamado de categoria imprecisa. Uma forma melhor de apresentar esse tipo de regra sem utilizar um sinal de interrogação é dizer que quando a *Educação* é pobre, existe uma chance de 50% de que a *Renda* seja baixa, e existe uma chance de 50% de que a *Renda* seja média.

Se um novo objeto for introduzido ao sistema de informação mas com o valor de decisão desconhecido, pode-se determinar esse valor através da aplicação das regras previamente geradas. Se somente uma regra é aplicável, então a classificação é direta. Isso implica que para os objetos contidos na região de borda de diferentes classes, nenhuma decisão consistente pode ser tomada.

4.5 Função de Pertinência Rough

Na teoria clássica de conjuntos, um elemento pertence ou não pertence a um conjunto. Assim, a função de pertinência correspondente possui, respectivamente, os valores 1 e 0. No caso de Rough Sets a noção de pertinência é diferente pois um conceito vago contém exemplos que pertencem à região de borda, isto é, elementos do universo que não podem ser — com certeza — classificados como elementos do conceito. Observando a Figura 4.2, fica claro que alguns elementos pertencem ao conjunto (conceito) com maior ou menor intensidade. Portanto, essa incerteza leva à questão da utilização de uma *função de pertinência* nos elementos do conjunto. Contudo, para se discutir a incerteza através da perspectiva de RS, é necessário definir a função de pertinência relacionada aos seus conceitos — a função de pertinência rough.

Definição 4.5.1 (Função de Pertinência Rough) Para $\mathcal{A} = (U, A)$, $x \in U$, $X \subseteq U$, atributos $B \subseteq A$, a Função de Pertinência Rough para a classe $E \in U/IND(B)$ é

$$\mu_B(E, X) = \frac{|E \cap X|}{|E|}, \quad 0 \leq \mu_B(E, X) \leq 1$$

Conforme (Pawlak, 1996), a função de pertinência rough possui as seguintes propriedades:

1. $\mu_B(E, X) = 1 \iff E \in \underline{B}(X)$
2. $\mu_B(E, X) = 0 \iff E \in U - \overline{B}(X)$
3. $0 < \mu_B(E, X) < 1 \iff E \in BN_B(X)$

Figura 4.2: Aproximações Superior e Inferior para o conjunto de elementos que formam o mapa do Brasil

4. $E_1, E_2 \in U/IND(B) \implies \mu_B(E_1, X) = \mu_B(E_2, X)$
5. $\mu_{A-B}(E, X) = 1 - \mu_B(E, X), \forall E \in U/IND(B)$
6. $\mu_{B \cup C}(E, X) \geq \max(\mu_B(E, X), \mu_C(E, X)), \forall E \in U/IND(B) \cup U/IND(C)$
7. $\mu_{B \cap C}(E, X) \leq \min(\mu_B(E, X), \mu_C(E, X)), \forall E \in U/IND(B) \cap U/IND(C)$

Vale ressaltar que os elementos que estão na mesma relação de não-discernimento possuem o mesmo grau de pertinência, conforme explicitado pela Propriedade 4.

Existe uma conexão entre incerteza e inconsistência na teoria de RS, a qual pode ser notada utilizando as definições anteriores. A inconsistência está relacionada aos conjuntos, enquanto que a incerteza está relacionada aos elementos do conjunto. Portanto, as aproximações são necessárias quando se fala sobre conceitos vagos e a função de pertinência rough é necessária quando dados incertos são considerados para definir um conjunto.

A função de pertinência rough pode ser interpretada como uma estimativa baseada na frequência de $Pr(x \in X \mid x, B)$, a probabilidade condicional que x pertença ao conjunto X , tendo conhecimento da informação de x com respeito aos atributos B .

A fórmula para as aproximações inferior e superior podem ser generalizadas para algum nível arbitrário de precisão $\pi \in (\frac{1}{2}, 1]$ em termos da função de pertinência rough. Essa forma parametrizada origina os *Rough Sets com precisão variável*, definidos a seguir:

$$\underline{B}_\pi X = \{x \mid \mu_B(E, X) \geq \pi\}$$

$$\overline{B}_\pi X = \{x \mid \mu_B(E, X) > 1 - \pi\}$$

Note que as aproximações inferior e superior originalmente formuladas são obtidas como um caso especial quando $\pi = 1.0$. Essa forma generalizada pode ser empregada como um método para estreitamento da região de borda, e pode ainda ser melhor generalizada com o objetivo de obter bordas assimétricas.

4.6 Regras Certas e Regras Possíveis

Cada tabela de decisão pode ser representada na forma de um algoritmo de decisão, consistindo de regras de decisão na forma “Se ... então ...”. Para a Tabela 3.3 pág. 25, os exemplos podem ser representados da seguinte forma:

R ₁ : Se Dor_de_Cabeça = sim e Temperatura = normal	então Gripe = não
R ₂ : Se Dor_de_Cabeça = sim e Temperatura = alta	então Gripe = sim
R ₃ : Se Dor_de_Cabeça = sim e Temperatura = muito_alta	então Gripe = sim
R ₄ : Se Dor_de_Cabeça = não e Temperatura = normal	então Gripe = não
R ₅ : Se Dor_de_Cabeça = não e Temperatura = alta	então Gripe = não
R ₆ : Se Dor_de_Cabeça = não e Temperatura = muito_alta	então Gripe = sim
R ₇ : Se Dor_de_Cabeça = não e Temperatura = alta	então Gripe = sim
R ₈ : Se Dor_de_Cabeça = não e Temperatura = muito_alta	então Gripe = não

As regras R₅ e R₇, bem como as regras R₆ e R₈ têm as mesmas condições, mas diferentes decisões. Portanto não se pode tomar uma decisão correta aplicando esse tipo de regra. Regras desse tipo são chamadas de *inconsistentes* (não-determinísticas ou conflitantes), enquanto que as regras R₁, R₂, R₃ e R₄ são chamadas de consistentes (determinísticas).

Na Teoria de Rough Sets, as regras que são consistentes são chamadas de *regras certas*, pois determinam decisões únicas a respeito de seus atributos de condição. Por

outro lado, as regras que são inconsistentes são chamadas de *regras possíveis*, pois podem levar a uma decisão utilizando os atributos de condição.

4.7 Fator de Credibilidade

A cada regra possível associa-se um *fator de credibilidade* (*credibility factor*) — CF — o qual indica a credibilidade da decisão associada à regra. Uma forma de definir esse coeficiente é utilizando a função de pertinência (Pawlak, 1996). Cada regra de decisão é gerada por uma linha da tabela de decisão, que são elementos que possuem um grau de pertinência associado. Portanto, esse grau de pertinência pode ser utilizado como a credibilidade da regra.

Seja $\delta(x)$ a regra de decisão associada ao elemento x da tabela de decisão. Então será dito que o elemento x suporta a regra $\delta(x)$. O fator de credibilidade dessa regra pode ser definido conforme a seguir:

$$CF(\delta(x)) = \begin{cases} 1, & \text{se } \mu_B(E, X) = 0 \text{ ou } 1 \\ \mu_B(E, X), & \text{se } 0 < \mu_B(E, X) < 1. \end{cases}$$

Com isso, qualquer regra consistente receberá um fator de credibilidade igual a 1, enquanto que as regras inconsistentes receberão um fator de credibilidade menor que 1, porém não igual a zero. Quanto mais próximo de 1 é o fator de credibilidade, maior é a credibilidade da regra.

4.8 Dependência entre Atributos

Uma outra característica importante na análise de informações é a descoberta de dependência entre atributos. Intuitivamente, um conjunto de atributos D depende totalmente de um conjunto de atributos C , denotado $C \Rightarrow D$, se todos os valores dos atributos de D são unicamente determinados pelos valores dos atributos de C . Em outras palavras, D depende totalmente de C , se existe uma dependência funcional entre valores de D e C . Formalmente, a dependência entre atributos pode ser definida da seguinte forma.

Definição 4.8.1 (Dependência entre Atributos) *Sejam D e C subconjuntos de A . Diz-*

se que D depende de C com grau k ($0 \leq k \leq 1$), denotado $C \Rightarrow_k D$, se

$$k = \gamma(C, D) = \frac{|POS_C(D)|}{|U|},$$

onde

$$POS_C(D) = \bigcup_{X \in U/D} \underline{C}(X),$$

chamada de região positiva da partição U/D com respeito a C , a qual é o conjunto de todos os elementos de U que podem ser unicamente classificados como blocos da partição U/D , com respeito a C .

É possível deduzir que,

$$\gamma(C, D) = \sum_{X \in U/D} \frac{|\underline{C}(X)|}{|U|}$$

Se $k=1$ diz-se que D depende totalmente de C , e se $k < 1$ diz-se que D depende parcialmente de C com grau k .

O coeficiente k expressa a taxa de todos os elementos do Universo que podem ser apropriadamente classificados como blocos da partição U/D , empregando o conjunto de atributos C , o qual será chamado de *grau de dependência*.

Pode-se notar que se D depende totalmente de C então $IND(C) \subseteq IND(D)$. Isso significa que a partição gerada por C é mais refinada que a partição gerada por D . Deve ser observado que esse conceito de dependência corresponde àquele considerado em bancos de dados relacionais.

Resumidamente, D é *totalmente* (parcialmente) dependente de C , se *todos* (alguns) elementos do universo U podem ser unicamente classificados como blocos da partição U/D empregando C .

4.9 Extensões

Desde o aparecimento da Teoria de RS no mundo acadêmico, ela recebeu algumas extensões e diferentes interpretações que dão margem à criação de novas ferramentas matemáticas baseadas nessa teoria. Algumas dessas extensões incluem:

1. Diferentes abordagens para as aproximações superior e inferior. Em (Bonikowski, 1998) é proposta a idéia de aproximação de atributos desconhecidos com base

em atributos conhecidos, utilizando os mesmos conceitos das aproximações superior e inferior. São propostas ainda algumas extensões que possibilitam a elaboração de relações e operações sobre conjuntos rough, similares às aquelas sobre conjuntos clássicos.

2. Medidas de incerteza baseadas em teoria da informação. Em (Beaubouef et al., 1998) são definidas algumas medidas de incerteza baseadas em entropia e em modelos de bancos de dados relacionais.
3. Raciocínio em sistemas de informação incompletos. Em (Kryszkiewicz, 1998) é proposta a redução da representação de forma a produzir regras com número mínimo de antecedentes e o mais determinísticas possível, embora as tabelas de decisão contenham muitos valores de atributos faltantes.
4. Aplicação da Teoria de Rough Sets em bases de dados relacionais (Lin & Cercone, 1997; Guan & Bell, 1998; Hu & Cercone, 1994; Hu, 1995).

4.10 Considerações Finais

Conforme apresentado neste capítulo, a noção de aproximação de conjuntos está diretamente ligada à relação de não-discernimento. Com ela pode-se definir a aproximação de um conjunto em termos dos elementos que o compõem. Como o conjunto é aproximado são necessárias algumas medidas para avaliar a qualidade das aproximações, bem como decidir a pertinência dos elementos em relação a esse conjunto aproximado.

Neste capítulo também foram mencionadas algumas extensões da teoria inicial, as quais facilitam o cálculo das aproximações dos conjuntos, bem como melhoram a precisão das aproximações obtidas.

No próximo capítulo serão apresentadas algumas ferramentas e indutores que servirão como suporte no estudo e análise dessa abordagem para tratamento de incerteza e seleção de atributos relevantes.

Capítulo 5

Ferramentas e Indutores

5.1 Considerações Iniciais

Como visto nos capítulos anteriores, o processo de análise das informações utilizando a abordagem de Rough Sets inclui várias etapas. Para analisar a aplicabilidade e eficiência dessa abordagem, existem ferramentas desenvolvidas por pesquisadores da área, algumas das quais serão brevemente discutidas neste capítulo. Inicialmente serão explanadas as funcionalidades de algumas ferramentas, sendo que a última ferramenta — *Rosetta* — será apresentada em maiores detalhes por ter sido a escolha no apoio aos nossos estudos por conter todas as funcionalidades requeridas nas análises. Alguns comentários sobre a ferramenta *Rosetta* encontram-se neste capítulo, os quais são complementados oportunamente no decorrer do trabalho. Ainda neste capítulo serão apresentadas outras ferramentas e indutores conhecidos pela comunidade de AM, os quais também serão utilizados no desenvolvimento deste trabalho.

5.2 Grobian

A ferramenta *Grobian*¹, desenvolvida por Ivo Düntsch da Universidade of Osnabrueck e Günther Gediga da Universidade of Ulster, possui várias funcionalidades para a análise de dados empregando Rough Sets. Utiliza para tanto a Rough Sets Library —

¹É um anacronismo de *Grobmengen Informations-Analysator*. Uma tradução adequada para o inglês seria *ROUGHIAN* — *Roughset Information Analyzer*. Esta ferramenta está disponível em <http://www.infj.ulst.ac.uk/~ccc23/grobian/grobian.html>.

RSL — que é uma biblioteca com funções desenvolvidas nos moldes da Teoria de Rough Sets, a qual esta atualmente na versão 2.0. Essa ferramenta suporta as três funcionalidades básicas para a análise, utilizando Rough Sets, *i.e.* encontrar os redutos, o núcleo e gerar regras. Outras funcionalidades adicionais são citadas a seguir:

- Análise de redutos e núcleos, inclusive na situação na qual o núcleo é vazio.
- Extração de regras, a fim de tornar o conhecimento adquirido inteligível para os seres humanos.
- Classificação baseada em regras.
- Análise estatística da qualidade das regras.
- Filtragem e transformação dos dados.
- Medidas de incerteza.
- Validação Jackknife.
- *Cross-validation*².

Existem outras três características que devem ser ressaltadas:

1. Testes aleatórios para medir a qualidade das regras extraídas.
2. Métodos para a filtragem de dados, tais como o tratamento de valores desconhecidos e conflitantes.
3. Métodos para análise de informações baseados em entropia.

Em geral, *Grobian* oferece um subconjunto das facilidades oferecidas pela ferramenta *Rosetta*, a qual será melhor discutida na Seção 5.6.

5.3 Rough Enough

A ferramenta *Rough Enough* foi desenvolvida por Anders Torvill Bjorvand, fundador da Troll Data Inc³, Noruega. Essa ferramenta funciona embutida num módulo *run-*

²O termo em português é validação-cruzada, porém utilizou-se o termo em inglês por este ser o mais aceito na comunidade acadêmica.

³<http://www.trolldata.no>

time Paradox, embora seja capaz de importar vários outros formatos de tabelas de decisão para análise. Essa ferramenta possui uma *interface* muito simplificada, a qual permite ao usuário fazer as análises básicas utilizando a abordagem de Rough Sets.

5.4 Rough Sets Library

A biblioteca *Rough Sets Library* — RSL⁴ — foi desenvolvida em C++ por M. Gawryś e J. Sienkiewicz, membros do Instituto de Ciência da Computação da Universidade Warsaw de Tecnologia, Polônia. Essa biblioteca está atualmente na versão 2.0, a qual é formada por classes e métodos que auxiliam no desenvolvimento de novas ferramentas voltadas à aplicação de Rough Sets.

5.5 Column Importance

A ferramenta *Column Importance facility* — CI — é uma das ferramentas integrantes do MineSetTM (Rathjens, 1996) da Silicon Graphics⁵. Esta ferramenta é útil para determinar a importância de vários atributos para a classificações dos exemplos. Basicamente, CI utiliza uma medida chamada “pureza” que atribui números entre 0 e 100, os quais descrevem a importância das colunas (atributos) para a classificação. Existe a possibilidade de informar à CI a quantidade de atributos que ele deve considerar como mais importantes, ou mesmo deixar que ele decida livremente quantos atributos devem ser considerados.

5.6 Rosetta

A ferramenta *Rosetta* — *A Rough Sets Toolkit for Analysis of Data* — vem sendo continuamente aperfeiçoada desde a sua primeira versão em 1997, estando atualmente na versão 1.4.40. Essa ferramenta foi desenvolvida numa cooperação entre o Grupo de Sistemas de Conhecimento e o Grupo de Lógica. O primeiro grupo está localizado no Departamento de Computação e Ciência da Informação na Universidade Norueguesa de Ciência e Tecnologia, Noruega⁶; o segundo no Instituto de Matemática

⁴<ftp://ftp.ii.pw.edu.pl/pub/Rough/>

⁵<http://www.sgi.com>

⁶Knowledge Systems Group at NTNU, Norway — <http://www.ntnu.no>

da Universidade de Warsaw, Polônia⁷. O primeiro grupo projetou a arquitetura do sistema e a interface gráfica sob os cuidados de Aleksander Øhrn, mantenedor do site onde a ferramenta Rosetta está disponível para *download*⁸. O segundo grupo desenvolveu as bibliotecas baseadas em Rough Sets, denominada de Rough Sets Expert System — RSES — que atualmente está na versão 1.4.1.

A ferramenta Rosetta está baseada na relação de não-discernimento, podendo ser utilizada nas tarefas de descoberta de conhecimento, suportando desde as fases iniciais de pré-processamento através da computação de conjuntos mínimos de atributos e geração de regras, até a validação e análise das regras induzidas. A Figura 5.1 mostra a *interface* com todas as funcionalidades dessa ferramenta.

Figura 5.1: Opções que a ferramenta Rosetta oferece para auxiliar na Extração de Conhecimento

⁷Logic Group at Warsaw University, Poland — <http://www.warsaw.pl>

⁸<http://www.idi.ntnu.no/~aleks/rosetta>

Algumas das funcionalidades do Rosetta são brevemente descritas a seguir⁹:

- Importar/Exportar

1. Integração com bancos de dados via ODBC, o que permite a análise de dados em diferentes formatos.
2. Exporta regras, redutos, tabelas, gráficos e outros objetos para vários formatos, incluindo C⁺⁺, Matlab e Prolog, viabilizando assim a integração com outros tipos de ferramentas para análise de dados.
3. Geração de relatórios em HTML descrevendo os comandos que foram executados sobre a tabela de decisão.

- Pré-processamento

1. Métodos para tratar dados inexistentes ou desconhecidos nas tabelas de decisão.
2. Discretização de valores numéricos, tornando-os nominais, que é um dos requisitos para a aplicabilidade da abordagem Rough Sets.

- Computação

1. Suporta ambos aprendizados supervisionado e não-supervisionado, permitindo assim tanto a extração de regras como a descoberta de novos agrupamentos (*clusters*).
2. Suporta a definição de discernimento através de informações fornecidas pelo próprio usuário.
3. Computação eficiente na procura por redutos, empregando vários métodos para a aproximação de redutos, incluindo algoritmos genéticos.
4. Geração de regras de conhecimento simbólico como forma de tornar inteligível o conhecimento extraído ao ser humano.
5. Execução de *scripts*, o que torna a análise mais dinâmica pois o usuário pode definir os passos que ele deseja executar na análise dos dados, e a ferramenta realiza a execução desses passos.
6. Suporte à *cross-validation*, permitindo assim uma análise da precisão do conhecimento extraído na classificação de novos exemplos.

- Pós-processamento

⁹Estas funcionalidades, entre outras, serão melhor explanadas adiante.

1. Filtragem das regras e redutos baseada no suporte, tamanho e qualidade. A qualidade pode ser avaliada segundo algumas medidas nomeadas no Rosetta como Michalski, Coleman, Torgo, Brazdil, Pearson, Cohen, G2, J e Kononenko¹⁰.
- Validação e Análise
 1. Aplicação de regras simplificadas a novos exemplos, como forma de validar o conhecimento extraído frente a novas situações.
 2. Geração de matriz de confusão, curvas ROC e curvas de calibração para a visualização gráfica.
 3. Formas de medir a qualidade das regras extraídas.
 4. Utilitários estatísticos para testes de hipóteses.
 - Diversos
 1. *Clustering* através de relações de tolerância.
 2. Computação das partições através de Rough Sets de precisão variável, o que permite incrementar a precisão do conhecimento extraído.

Até o momento foram apresentadas de forma breve as principais características da ferramenta Rosetta. Como esta ferramenta foi escolhida para apoiar este trabalho, algumas de suas funcionalidades serão melhor descritas para fundamentar as escolhas feitas na direção do uso de um algoritmo ou outro nos diferentes passos da extração de conhecimento. Maiores informações sobre esta ferramenta e alguns termos aqui utilizados podem ser encontradas em (Øhrn, 1999a) e (Øhrn, 1999b).

Para facilitar a localização de cada opção na ferramenta Rosetta foram mantidos, também, os nomes em inglês.

5.6.1 Tratamento de Dados Incompletos

O tratamento de dados incompletos é feito segundo uma família de possíveis algoritmos chamada de *Completer*. Os algoritmos tomam como entrada uma tabela de decisão incompleta — com valores de atributos faltantes — e produz como saída uma tabela de decisão na qual não existem valores ausentes para quaisquer atributos.

¹⁰Referências a estas medidas de qualidade podem ser encontradas em (Øhrn, 1999a).

O tratamento dos dados incompletos deve ser feito porque a abordagem de Rough Sets não prevê valores ausentes. Nesta ferramenta, assim como na maioria dos sistemas de AM, os dados ausentes são representados por ? (ponto de interrogação). Se nenhum tratamento for dado aos valores ausentes a ferramenta os tratará como sendo um possível valor para o atributo, ou seja, não tratará os valores ausentes de forma apropriada.

Várias são as formas da ferramenta Rosetta tratar esses valores ausentes, as quais são brevemente descritas a seguir.

Remoção dos incompletos (Remove Incompletes) Remove todos os exemplos que possuem um ou mais valores de atributos ausentes. Esta opção deve ser utilizada com cautela, pois a remoção de exemplos com dados ausentes afeta o poder de descrição dos atributos, uma vez que devido a um único valor ausente em um exemplo todos os outros valores dos atributos daquele mesmo exemplo serão desconsiderados.

Substituição pela média (Mean/mode fill) Quando os valores ausentes são numéricos esta opção substitui os valores ausentes pelo valor da média dos valores observados nesse atributo. Se os valores ausentes correspondem a um atributo nominal, então o valor ausente é substituído pelo valor que ocorre com maior frequência para esse atributo. Note que ao substituir os valores numéricos ausentes pela média introduz-se um *bias* na tabela de decisão, pois os valores mais próximos à média passam a ter maior poder de representação. O mesmo ocorre quando os valores nominais ausentes são substituídos pelo valor mais frequente, pois este valor passará a ter maior poder de representação na tabela de decisão.

Substituição pela média condicionada à decisão (Conditioned mean/mode fill) Similar ao algoritmo anterior, mas neste caso os valores da média para atributos contínuos e os valores de maior frequência para atributos nominais são calculados levando-se em consideração as classes de decisão. Ou seja, se existem k classes então as médias e os valores de maior frequência são calculados como se existissem k conjuntos disjuntos, cada qual contendo apenas exemplos pertencentes à mesma classe. Deve ser observado que não há tratamento especial no caso do valor ausente ser o do atributo de decisão. Nesse caso, todo o exemplo é ignorado. Nesta opção para tratamento dos valores ausentes, os problemas citados na opção anterior se repetem com menor intensidade pelo fato da substituição estar condicionada à decisão. No entanto, por menor que

seja, algum *bias* é introduzido nas classes que possuem exemplos com valores ausentes.

Substituição combinatorial (Combinatorial completion) Esta opção para tratamento de valores ausentes expande cada valor ausente nos possíveis valores que o atributo pode assumir. Ou seja, cada exemplo é expandido em vários exemplos que são resultantes da combinação de todos os possíveis valores dos atributos incompletos. É claro que esta opção deve ser utilizada com cautela, pois o número de possíveis combinações pode ser muito grande dependendo do número de valores ausentes para cada exemplo e do número de possíveis valores que os atributos podem assumir. Nesta opção de substituição dos valores ausentes, a representação dos atributos é fortemente afetada, pois ao expandir o atributo de valor ausente em todos os possíveis valores que este possa assumir, implica-se na multiplicação da representação dos valores presentes nos demais atributos, inserindo portanto algum *bias*.

Substituição combinatorial condicionada (Conditioned combinatorial completion) Similar ao método anterior, mas neste caso a expansão dos valores é condicionada às classes de decisão, tal como no caso da *Substituição pela média condicionada à decisão*. Nesta opção para tratamento dos valores ausentes ocorre o mesmo problema da opção anterior, porém com menos impacto porque a substituição é condicionada à decisão embora algum *bias* seja inserido.

5.6.2 Discretização

A abordagem de RS não trabalha com valores contínuos e sim com valores discretos, ou seja, para os atributos numéricos é necessário aplicar um processo de discretização para torná-los nominais (categóricos).

Na ferramenta Rosetta há uma família de algoritmos chamada *Scaler*, para a qual dada uma tabela de decisão de entrada, uma tabela de decisão contendo somente valores discretos é retornada. O processo de discretização procura por pontos de “corte” que determinam intervalos apropriados. Todos os valores que pertencem a um certo intervalo são então mapeados para o mesmo valor. Com isso atributos que antes eram numéricos podem ser tratados como categóricos. Após o processo de discretização, os valores dos atributos passam a ser tratados como *strings*.

Os algoritmos de discretização da família *Scaler* pertencem a uma das seguintes três

categorias:

1. Cada atributo é considerado isoladamente e nenhum conhecimento sobre o atributo de decisão é utilizado no processo. Estes algoritmos são chamados de univariados e não-supervisionados.
2. Somente um atributo condicional é considerado por vez, mas a discretização é feita em conjunção com o atributo de decisão. Estes algoritmos são chamados de univariados e supervisionados.
3. Todos os atributos condicionais são considerados simultaneamente, e a discretização é feita em conjunção com o atributo de decisão. Estes algoritmos são chamados de multivariados e supervisionados.

Supostamente, os algoritmos de discretização deveriam transformar os atributos contínuos em atributos categóricos sem que houvesse nenhum tipo de perda de informação. Porém isso não ocorre, pois quando os valores numéricos passam a ser representados por intervalos determinados pelos pontos de “corte”, o poder de representação do conhecimento que cada valor possuía fica reduzido ao poder de representação do intervalo. Por exemplo, isso equivale a dizer que pessoas com idades de 4, 5, 7, 13, 15, 19 e 23 são jovens, ou seja, pessoas com idades no intervalo [4, 23] são tratadas como pessoas de mesma idade.

Embora isso pareça um problema, existem muitos algoritmos que conseguem calcular bons pontos de “corte” minimizando a perda de informação, mas essa perda sempre ocorre. Vale lembrar que alguns algoritmos para extração de conhecimento só trabalham na presença de valores nominais, como é o caso da abordagem de Rough Sets. Segue uma descrição dos diferentes métodos de discretização implementados na ferramenta Rosetta.

Raciocínio Booleano (Boolean reasoning algorithm) Este algoritmo é baseado na combinação dos “cortes” encontrados pelo algoritmo *Naive*, descrito posteriormente, com um procedimento de raciocínio booleano para descartar os pontos de “cor-te” excessivos. O conjunto restante é um conjunto mínimo de pontos de “cor-te” que preservam a relação de não-discernimento inerente ao sistema de decisão.

Primeiramente, o algoritmo cria uma função booleana f para o conjunto de candidatos a serem pontos de “corte”, e então, utiliza um método de minimiza-

ção baseado no conceito de primo implicante. As vezes, a abordagem de raciocínio booleano para a discretização resulta em nenhum ponto de “corte” para alguns atributos. Isso significa que estes atributos não são necessários para preservar a relação de não-discernimento, uma vez que este algoritmo calcula o reduto da tabela de decisão.

Este algoritmo tem uma ordem de complexidade de $O(|A||U|^3)$ no pior caso (onde $|A|$ é o número total de atributos e $|U|$ é o número total de exemplos). Assim, ele é computacionalmente custoso para tabelas de decisão muito grandes.

Discretização Manual (Manual discretization) Neste tipo de discretização o usuário é quem define os pontos de “corte” para discretizar um dado atributo, produzindo assim intervalos seguindo um critério próprio. Para utilizar este método de discretização é necessário ter conhecimento sobre o conteúdo da tabela de decisão ou ser auxiliado por um especialista do domínio.

Entropia (Entropy/MDL algorithm) Implementa um algoritmo que particiona recursivamente o conjunto de valores de cada atributo a fim de otimizar a medida de entropia local. Os valores ausentes dos atributos são ignorados na busca por pontos de “corte”.

Intervalos Iguais (Equal frequency binning) Esse tipo de discretização procura por $n - 1$ pontos de “corte”, dividindo o conjunto de exemplos em aproximadamente n partes iguais. Este tipo de discretização é não-supervisionada e univariada. Na maior parte dos casos não produz uma discretização suficientemente eficaz para uma boa extração de conhecimento.

Naive (Naive algorithm) Implementa uma forma de discretização que resulta em muitos “cortes”. Na maior parte dos casos produz mais pontos de “corte” do que necessário. No pior caso, cada valor observado é colocado num intervalo próprio. Este algoritmo ordena os valores de um determinado atributo em ordem crescente produzindo uma sequência tal como

$$v_a^1 < v_a^2 < v_a^3 < \dots v_a^{|V_a|}$$

onde v_a^i são os possíveis valores pertencentes a V que o atributo a pode assumir.

Em essência, os pontos de “corte” são colocados no valor médio entre v_a^i e v_a^{i+1} , exceto quando os exemplos que possuem estes valores também possuem o mesmo valor de decisão.

Semi-Naive (Semi-naive algorithm) Esta forma de discretização funciona similarmente à anterior, mas possui um mecanismo para manipular valores de exemplos que estão numa mesma vizinhança e pertencem a classes de decisão diferentes. Normalmente, este tipo de discretização resulta em menos pontos de “corte” que a discretização *Naive*, no entanto, ainda, o número de pontos de “corte” é maior que o necessário.

Raciocínio Booleano (RSES) (Boolean reasoning algorithm (RSES)) Esta é uma implementação mais eficiente da forma de discretização descrita em *Raciocínio Booleano*. O funcionamento é o mesmo, mas o resultado é obtido muito mais rápido. Nesta implementação os valores ausentes são tratados como valores positivos “infinitamente grandes”.

Embora sejam vários os algoritmos de discretização, alguns experimentos por nós elaborados mostraram que o método de discretização através do *Raciocínio Booleano (RSES)* é o que produz os melhores resultados, discretizando os valores contínuos com o menor conjunto de pontos de “corte” possível, além disso esta é uma observação feita no próprio manual da ferramenta (Øhrn, 1999b). No entanto, como este algoritmo faz uso dos redutos para o processo de discretização, alguns atributos podem não ser discretizados. Isso significa que estes atributos não são importantes, segundo a Teoria de Rough Sets, para a representação do conhecimento implícito na tabela de decisão. Porém, os redutos podem variar dependendo do algoritmo empregado. Além disso, quando comparado RS com outros indutores, deve-se observar se os atributos não discretizados serão importantes para esses outros indutores.

5.6.3 Redutores

Um dos pontos fundamentais da abordagem de Rough Sets é a procura por redutos, os quais são conjuntos mínimos de atributos relevantes que preservam o mesmo conhecimento (mesma relação de não-discernimento) representado pelo conjunto de todos os atributos. Ou seja, é uma forma de se reduzir a representação da tabela de decisão.

Na ferramenta Rosetta os algoritmos pertencentes a esta família são chamados de *Reducer*, os quais procuram pelos redutos utilizando diferentes heurísticas. Existem duas formas de se procurar os redutos:

1. *Full* : calcula os redutos relativos à tabela de decisão como um todo, preser-

vando assim a habilidade de distinguir os exemplos uns dos outros utilizando um conjunto mínimo de atributos.

2. *Object* : calcula os redutos relativos a um determinado exemplo, preservando a habilidade de se distinguir aquele exemplo dos demais na tabela de decisão. Normalmente, quando esta forma de calcular redutos é utilizada, seleciona-se um conjunto de exemplos para se distinguir dos demais.

No entanto, para o cálculo dos redutos nada se assume sobre a discretização dos dados. Assim, os redutos podem ser calculados sobre a tabela de decisão já discretizada ou sobre a tabela de decisão original que pode ter atributos contínuos. Entretanto, atributos contínuos têm grandes chances de figurarem como elementos dos redutos, uma vez que valores contínuos dificilmente são iguais entre os exemplos da tabela de decisão. Por outro lado, os redutos podem ser calculados sobre a tabela de decisão totalmente discretizada, a qual terá embutida em seus atributos o *bias* do algoritmo de discretização utilizado, interferindo de alguma forma no algoritmo que calcula os redutos.

Deve ser ressaltado que alguns dos algoritmos descritos nesta seção possuem limitações quanto ao tamanho da tabela de decisão (máximo de 500 exemplos). Maiores informações podem ser encontradas em (Øhrn, 1999a) e (Øhrn, 1999b).

Algoritmo Genético (Genetic Algorithm) Esta opção implementa um Algoritmo Genético para calcular redutos mínimos. Os redutos são recombinaados segundo uma função custo que avalia cada reduto e os mantém numa lista. Somente os redutos que satisfazem a função custo são mantidos nessa lista para serem recombinaados. A recombinação ocorre enquanto a lista de redutos sofre mudanças ou enquanto a qualidade da população aumenta segundo a função custo. No final deste processo cada reduto possui um valor de suporte associado como medida de qualidade daquele reduto na representação do conhecimento, preservando assim a relação de não-discernimento segundo um fator de credibilidade.

Algoritmo de Johnson (Johnson Algorithm) Nesta opção uma simples variação do algoritmo *greedy* é invocada para o cálculo de um único reduto. Este algoritmo tem um *bias* natural na busca de um simples primo implicante de tamanho mínimo. Neste algoritmo o reduto é inicialmente vazio, sendo acrescentado a este os atributos que maximizam uma função que atribui pesos a cada um dos

atributos. No entanto, por ser uma variação do algoritmo *greedy*, não há garantia que o único reduto calculado seja aquele que maximiza a função de pesos de forma global, talvez seja apenas um máximo local.

Holte 1R (Holte 1R) Retorna todos os conjuntos unitários de atributos, e ainda o conjunto de todas as regras geradas utilizando esses atributos.

Cálculo Manual (Manual reducer) Deixa que o usuário escolha os atributos que farão parte do reduto a ser utilizado em futuros cálculos.

Redutos Dinâmicos (Dynamic reducts (RSES)) Para o cálculo dos redutos um número de sub-tabelas é amostrado, de forma aleatória, da tabela de decisão original. Então, para cada sub-tabela os redutos são calculados utilizando um outro algoritmo qualquer e o reduto que aparecer com maior frequência como reduto das sub-tabelas é tomado como sendo o reduto principal. O cálculo de redutos utilizando esta opção pode ser computacionalmente custoso, mesmo para tabelas de decisão pequenas.

Cálculo Exaustivo (Exhaustive calculation (RSES)) O cálculo dos redutos é feito utilizando exaustivamente, *i.e.* sem qualquer tipo de heurística. Esta opção é apropriada somente para tabelas de decisão de tamanho moderado, pois o cálculo é exaustivo e o algoritmo é *NP-hard*.

5.6.4 Geração de Regras

A geração de regras é feita sobrepondo cada reduto — presente no conjunto dos redutos — sobre a tabela de decisão que originou aqueles redutos. Os antecedentes das regras são concatenados com conjunções de *atributo=valor* e o conseqüente é extraído segundo a decisão de cada um dos exemplos sendo sobrepostos. Deve ser ressaltado que no cálculo dos redutos ocorre apenas uma redução horizontal na tabela de decisão (número de atributos). Isso implica em um número de regras que as vezes pode ser maior que o número de exemplos, embora isso dependa do número de valores nominais dos atributos pertencentes aos redutos.

5.6.5 Classificação

Esta ferramenta, chamada de *BatchClassifier*, classifica todos os exemplos de uma dada tabela de decisão utilizando um dos classificadores que serão descritos a seguir.

Para cada classificação uma matriz de confusão é construída e um arquivo texto é gerado contendo informações detalhadas sobre o processo de classificação, tais como a predição e a real classificação de cada exemplo.

Em alguns casos, o classificador escolhido pode levar a várias alternativas para o valor de decisão de um exemplo, cada alternativa com um grau de certeza associado. Existem três opções para lidar com isso:

1. *Best*: classifica o exemplo como pertencente a classe de decisão que possui o maior grau de certeza associado.
2. *Prioritize*: classifica o exemplo como pertencente a classe de decisão cujo número de regras que a predizem for o maior.
3. *Refrain*: impede a classificação de exemplos para os quais o classificador indica múltiplas possibilidades de classificação.

Os classificadores que podem ser utilizados neste processo de classificação estão brevemente descritos a seguir.

Votação padrão (Standard voting) Implementa o processo de votação descrito em (Øhrn, 1999a) utilizando um conjunto específico de regras. Na etapa na qual as regras são “disparadas”, uma regra “dispara” se seus antecedentes não estiverem em conflito com o exemplo sendo classificado, e se a percentagem de termos no antecedente estiver abaixo de um certo limite t . Em alguns casos, e especialmente se as regras são geradas por redutos dinâmicos, pode acontecer que algumas regras sejam generalizações de outras. Se duas ou mais regras no conjunto de regras formam uma hierarquia, o algoritmo tem a opção de somente “disparar” a regra que for mais específica.

No processo de eleição cada regra precisa ter um certo número de votos em favor do valor de decisão que ela indica de acordo com a estratégia de voto selecionada. O coeficiente de certeza para cada possível valor de decisão é calculado pelo quociente do número total de votos para cada valor de decisão pelo fator de normalização.

Votação com rastreamento de exemplos (Voting with object tracking) Implementa a votação com rastreamento de exemplos tal como descrito em (Øhrn, 1999a), utilizando um conjunto de regras específico.

Naive Bayes (Naive Bayes) Implementa o classificador de Bayes, no qual para cada classe de decisão é calculada a probabilidade condicional de que a classe de decisão é a correta. Nesse método de classificação assume-se que os atributos dos exemplos são independentes. As probabilidades envolvidas são calculadas como frequências obtidas da tabela de decisão.

Votação padrão/afinada (RSES) (Standard/tuned voting (RSES)) Oferece uma forma de classificação similar à descrita no tópico sobre o método de classificação *Votação Padrão*, porém com dois adicionais:

1. *Majority*: similar ao algoritmo descrito no tópico sobre o método de classificação *Standard Voting*, mas sem tolerância a valores ausentes. Se alguma regra “disparar”, a classe de decisão que possuir o maior coeficiente de certeza será retornada.
2. *Tuned*: permite que o usuário incorpore valores de “distância” entre as classes de decisão.

Até o momento foram apresentados os principais métodos presentes na ferramenta Rosetta para extração de conhecimento. Nessa ferramenta, conforme os métodos são aplicados, é criada uma estrutura hierárquica contendo todos os resultados. Uma ilustração dessa estrutura pode ser vista na Figura 5.2.

Figura 5.2: Estrutura criada após a aplicação de alguns métodos no processo de Extração de Conhecimento

5.6.6 Automação

Como já mencionado, outra característica importante da ferramenta Rosetta é a possibilidade de executar *scripts* que automatizam a execução do processo de extração de conhecimento, ou mesmo possibilitam a execução de comandos de forma repetitiva.

Os *scripts* são arquivos texto que contém linhas de comandos e parâmetros, os quais são interpretados e executados dependendo do tipo de automação escolhida.

Existem três formas de automatizar as tarefas no Rosetta utilizando *scripts*, que são descritas a seguir.

Execução Serial (Pipeline script) Implementa a execução serial dos comandos em cascata, o que torna a saída da execução de um comando a entrada do comando que estiver imediatamente na sequência de execução. Se o *script* descreve uma sequência de algoritmos $A_1, \dots, A_n$ e se estes forem aplicados a uma estrutura inicial S , então tem-se que:

$$A_n(A_{n-1}(\dots(A_2(A_1(S)))\dots))$$

Execução Serial com Cross-Validation (Pipeline script with CV) Implementa *k-fold cross-validation* através da especificação via *script* das sequências de treinamento e teste. Os comandos são lidos e divididos em duas linhas de execução — uma para treinamento e outra para teste. O processo que ocorre é descrito a seguir:

1. Divide a tabela de decisão em k amostras disjuntas.
2. Coleta duas amostras (ou tabelas) disjuntas a partir da tabela de decisão inicial: uma tabela para treinamento contendo $k-1$ amostras e outra tabela para teste contendo uma única amostra, ambas obtidas no passo anterior.
3. Coloca a tabela de treinamento na sequência de execução do treinamento. Assume-se que a sequência de treinamento irá produzir um conjunto de regras ao final.
4. Coloca a tabela de teste na sequência de execução do teste. Assume-se que a sequência de teste irá produzir um ambiente propício à classificação. Essa sequência utilizará as regras produzidas pela sequência de treinamento.

5. Coleta estatísticas a partir da classificação produzida pela sequência de teste e as escreve num arquivo texto de $\log^{11}$.
6. Repete os passos anteriores k vezes variando sistematicamente a coleta das amostras do passo 2;
7. Calcula as estatísticas gerais e as escreve no arquivo texto de $\log$.

Utilizando *cross-validation* garante-se que cada exemplo presente na tabela de decisão de entrada estará na tabela de treinamento $k - 1$ vezes e uma vez na tabela de teste.

Execução Paralela (Batch script) Implementa a execução paralela dos comandos presentes no *script*. Se o *script* descreve uma sequência de algoritmos $A_1, \dots, A_n$ e se esses forem aplicados a uma estrutura inicial S , então tem-se que:

$$A_1(S), A_2(S), \dots, A_{n-1}(S), A_n(S)$$

A seguir é apresentado um exemplo de *script* que o Rosetta reconhece como válido:

```
%-----%
% Este script é utilizado para 10-fold-cross-validation          %
% os 4 primeiros passos referem-se aos 9-folds do treinamento e os %
% 2 outros passos são para o outro fold relativo ao teste.      %
%-----%

%Treinamento (4 passos)
%-----

RSESOthogonalScaler % Indica o uso do método de discretização booleana
                    % para discretizar o conjunto de treinamento
{MODE = Save; FILENAME = D:\cuts.txt} % Os pontos de "corte" serão
                                      % salvos no arquivo cuts.txt
                                      % para uso posterior

SAVGeneticReducer  % Indica a execução do Algoritmo Genético para
                   % calcular os redutos relativos a toda a tabela
                   % de decisão
{DISCERNIBILITY = Full}
RSESRuleGenerator  % Gera as regras utilizando os redutos encontrados
                   % no passo anterior
{ }

%Teste (2 passos)
%-----

RSESOthogonalFileScaler % Os pontos de corte são utilizados para
                       % discretizar os exemplos do conjunto de teste
```

¹¹Um arquivo de *log* é um arquivo que armazena todas as operações efetuadas no processo de *cross-validation*.

```
{MODE = Load; FILENAME = D:\cuts.txt}
BatchClassifier % O conjunto de teste será classificado com as regras
                  % geradas pelo treinamento, utilizando o método
                  % StandardVoter
{CLASSIFIER=StandardVoter}
%-----
```

5.7 A Biblioteca $\mathcal{MLC}^{++}$

Neste trabalho, além da ferramenta Rosetta, foram utilizados quatro indutores presentes na Biblioteca $\mathcal{MLC}^{++}$ (Kohavi et al., 1996). $\mathcal{MLC}^{++}$ é uma biblioteca de classes e ferramentas para Aprendizado de Máquina, implementada em C^{++} , a qual foi projetada para auxiliar a seleção de algoritmos apropriados para a realização de tarefas específicas.

O projeto da $\mathcal{MLC}^{++}$ iniciou-se em 1993 na universidade de Stanford, e em 1995 passou a estar sob a responsabilidade da Silicon Graphics.

A Biblioteca $\mathcal{MLC}^{++}$ é um software de domínio público, que inclui os códigos fontes e conta com mais de 30 algoritmos de AM. Especificamente, os seguintes quatro indutores foram utilizados neste trabalho:

1. ID3
2. $\mathcal{C4.5}$
3. $\mathcal{C4.5}$ -rules
4. $\mathcal{CN}^2$

Esses indutores são conhecidos na comunidade de AM e pertencem à categoria de aprendizado chamada *eager*, na qual o algoritmo, após extrair o conhecimento, descarta os exemplos e somente o conhecimento induzido é utilizado para a classificação de novos exemplos (Aha, 1997). Deve ser ressaltado que esses indutores reconhecem e tratam valores desconhecidos, os quais são indicados pelo símbolo de interrogação (?). Também, quando um exemplo não pode de ser classificado por esses indutores, a classe do exemplo é predita como sendo a classe majoritária (*default*). Uma descrição sucinta desses indutores é apresentada a seguir.

5.7.1 ID3

O ID3 (Quinlan, 1986) é um algoritmo muito simples para a extração de árvores de decisão, o qual não faz “podas” na árvore e nem regride, durante o processo de construção da árvore, para reconsiderar outras escolhas.

Um nó numa árvore de decisão representa um teste relativo ao valor de determinado atributo. O procedimento geral para construir a árvore de decisão consta dos seguintes passos (Quinlan, 1986):

1. Utilizando o conjunto de treinamento, um atributo é escolhido de acordo com algum critério, como o nó raiz e os possíveis valores que esse atributo assume determinam os ramos a partir deste nó.
2. Para cada subconjunto de exemplos nesses ramos o processo é repetido enquanto cada subconjunto contiver exemplos de diferentes classes.
3. Uma vez que um desses subconjuntos contenha apenas exemplos de uma mesma classe, ao invés de se criar um novo nó, uma folha é criada e nomeada com a classe a qual todos os exemplos daquele subconjunto pertençam.

Quando uma nova instância necessita ser classificada, começa-se pelo nó raiz da árvore de decisão e os testes vão seguindo nó a nó comparando os valores dos atributos do novo exemplo com os valores condicionais dos ramos, até que uma folha seja alcançada. Então, a classe do novo exemplo é predita como sendo a classe da folha.

A versão original do ID3 utiliza como critério para decidir quais os nós que irão compor a árvore de decisão o critério de ganho de informação, o qual é calculado utilizando a entropia. O critério utilizado na implementação do ID3 presente na biblioteca $\mathcal{MLC}^{++}$ é chamado de *Informação de Correlação Normalizada* — ICN — o qual é muito similar ao critério do ganho de informação, e também é baseado na entropia. Ele é determinado por:

$$ICN = \frac{Entropia}{\log_2(NúmeroNósFilhos)}$$

A versão do ID3 presente na biblioteca $\mathcal{MLC}^{++}$ é capaz de reconhecer e tratar valores desconhecidos, embora a versão original (Quinlan, 1986) deste algoritmo não tenha esse suporte.

5.7.2 C4.5

C4.5 (Quinlan, 1993) é um dos sucessores do ID3. Muitas extensões foram feitas ao algoritmo original, tais como aumento na eficiência computacional, tratamento de valores contínuos e de valores desconhecidos, uso de janelas — *i.e.* desenvolvimento de várias árvores de decisão — e o uso do critério de ganho de proporção, ao invés do ganho de informação utilizado na versão original do ID3. O uso desse novo critério corrigiu uma séria deficiência da versão original, a qual possuía um forte *bias* em favor dos atributos que possuem muitos valores.

5.7.3 C4.5-rules

C4.5-rules (Quinlan, 1993) parte da árvore de decisão originalmente produzida pelo C4.5 e deriva dessa árvore um conjunto de regras não ordenadas na forma *if <complexo> then <classe>* (Baranauskas & Monard, 2000).

Em geral, regras assumem a forma:

$$if < complexo > then < classe = C_i >$$

onde C_i é uma das k possíveis classes $\{C_1, C_2, \dots, C_k\}$; *< complexo >* representa a condição da regra e *< classe = C_i >* a conclusão. O *< complexo >* é uma disjunção de conjunções de testes de valores de atributos, da forma:

$$A_i \text{ op } Valor$$

onde A_i é um atributo; *op* é um operador pertencente ao conjunto $\{=, \neq, <, \leq, >, \geq\}$ e *Valor* é um valor válido do atributo A_i . Um exemplo é coberto por uma regra se satisfaz tanto a condição quanto a conclusão da regra.

É importante notar que o C4.5-rules generaliza o conhecimento representado na árvore de decisão removendo condições descartáveis — *i.e.* condições irrelevantes que não afetam a conclusão — sem afetar a precisão e retém apenas regras consideradas mais interessantes.

5.7.4 $\mathcal{CN}^2$

O $\mathcal{CN}^2$ (Clark & Niblett, 1987; Clark & Niblett, 1989; Clark & Boswell, 1991) é um algoritmo de Aprendizado de Máquina que induz um conjunto de regras não ordenadas no mesmo formato do $\mathcal{C4.5}$ -rules (Baranauskas & Monard, 2000).

Para classificar um novo exemplo utilizando as regras induzidas, todas as regras são testadas e o conjunto das regras satisfeitas é obtido. Se mais de uma classe for predita pelas regras satisfeitas, então é considerado o número de exemplos cobertos por cada uma dessas regras. Depois somam-se esses valores para encontrar a classe mais provável. Por exemplo, considerando as três regras seguintes, relacionadas à classificação de um robô como inimigo ou amigo:

se cabeça=quadrada	e segura=arma	então classe=inimigo	cobre	[15,1]
se tamanho=alto	e voa=não	então classe=amigo	cobre	[1,10]
se aparência=zangado		então classe=inimigo	cobre	[20,0]

sendo que [15,1] denota que a primeira regra cobre 15 exemplos de treinamento da classe inimigo e 1 da classe amigo. Dado um novo exemplo de um robô que possui cabeça quadrada, carrega uma arma, alto, não voa e está zangado, todas as três regras são satisfeitas. Neste caso, o $\mathcal{CN}^2$ resolve o conflito somando os valores, no exemplo considerado obtém-se [36,11], das regras satisfeitas e então prediz a classe mais comum — inimigo.

5.8 Considerações Finais

Neste capítulo foram apresentadas algumas das ferramentas existentes que utilizam a abordagem Rough Sets, bem como outras ferramentas e indutores utilizados no desenvolvimento deste trabalho. A ferramenta Rosetta possui todos os recursos para a análise da aplicabilidade e eficiência de RS aplicado a AM. Essa ferramenta possui ainda vários mecanismos de apoio para uma efetiva aplicação e análise dessa abordagem, tais como métodos de discretização, *cross-validation*, gráficos, entre outros. As demais ferramentas, possuem somente as funcionalidades básicas para analisar a aplicabilidade de Rough Sets na extração de conhecimento. Assim, devido às características da ferramenta Rosetta, esta foi escolhida como ferramenta principal no apoio de nossos estudos. Ainda foram apresentadas outras ferramentas e indutores

bem conhecidos da área de Aprendizado de Máquina, os quais foram utilizados para realizar diversos estudos e comparações, descritos no próximo capítulo, com a abordagem de Rough Sets.

Capítulo 6

Redutos na Abordagem Filtro para Seleção de Atributos

6.1 Considerações Iniciais

A evolução tecnológica permitiu a coleta e o armazenamento de uma quantidade cada vez maior de dados. Com a aplicação cada vez mais freqüente de técnicas de AM a estas bases de dados, o problema de focalizar-se na informação mais relevante tornou-se muito importante. Assim, um dos principais problemas em AM é a seleção de atributos relevantes.

Existem diversas razões para a realização da seleção de atributos. Uma dessas razões é que a maioria dos algoritmos de AM computacionalmente viáveis não trabalham bem na presença de um grande número de atributos, *i.e.* SA pode melhorar a precisão dos classificadores gerados por estes algoritmos. Outra razão é que a seleção de atributos melhora a habilidade de compreensão dos dados pelos humanos e também, por exemplo, das regras de indução geradas por algoritmos simbólicos de AM. Uma terceira razão para a realização da SA é o alto custo para a aquisição das informações, já que em muitos domínios a coleta de dados pode ser muito cara. Finalmente, a seleção de atributos pode reduzir os custos de processamento de grandes quantidades de dados.

A principal funcionalidade da Teoria de Rough Sets é o cálculo dos redutos. Como já visto anteriormente, os redutos são subconjuntos de atributos que preservam a relação de não-discernimento de todo o conjunto de dados. Assim, os atributos pertencentes aos redutos são atributos selecionados segundo o conceito de relevância

de RS.

Existem três formas principais de se fazer seleção de atributos (Baranauskas & Monard, 1998b). Filtro é uma das abordagens para SA, e neste capítulo serão mostrados resultados experimentais e comparativos entre os redutos e outros métodos para seleção de atributos pertencentes a essa abordagem. Ainda, serão apresentados os conjuntos de dados naturais e a organização esquemática desses experimentos.

6.2 Abordagens para a Seleção de um Subconjunto de Atributos

As diferentes abordagens para realizar a SSA podem ser agrupadas em:

1. *Embedded*

Nesta abordagem a tarefa da seleção de atributos é realizada internamente pelo próprio algoritmo. Em outras palavras, dado um conjunto de exemplos representado no formato atributo-valor, o próprio algoritmo de AM é capaz de decidir quais são os atributos relevantes para representar o conhecimento extraído. Algoritmos pertencentes a categoria *eager*, tais como *C4.5*, *C4.5-rules* e *CN2* utilizam esta abordagem, como visto na Seção 5.7.

2. *Filtro*

Esta abordagem para seleção de atributos introduz um processo separado, o qual ocorre antes da aplicação do algoritmo de AM propriamente dito. A idéia é filtrar, segundo algum critério, os atributos irrelevantes (John et al., 1994). Esse passo de pré-processamento considera características gerais do conjunto de dados para selecionar alguns atributos e excluir outros. Sendo assim, métodos de filtros são independentes do algoritmo de indução que, simplesmente, receberá como entrada a saída fornecida pelo filtro.

Segundo (Blum & Langley, 1997), um dos esquemas mais simples de filtragem é a avaliação de cada atributo, baseada na sua correlação com o conceito meta, escolhendo os k atributos que fornecem o melhor valor. Este método é comumente empregado em tarefas de categorização de textos, os quais têm mostrado bons resultados empíricos utilizando esse tipo de filtro.

Embora os indutores utilizados neste trabalho possuam uma abordagem *embedded* para a seleção de atributos, é possível empregar a priori a abordagem *fil-*

tro em conjunto com esses indutores. Basta que os atributos do conjunto de dados sejam filtrados por algum *filtro* e em seguida seja feita uma nova representação do conjunto de dados contendo apenas os atributos selecionados pelo *filtro*. Então, os indutores são aplicados sobre esse conjunto de dados de dimensão menor, pois em geral contém um menor número de atributos que o conjunto de dados inicial.

3. *Wrapper*

Esta abordagem também ocorre externamente ao algoritmo de AM, porém utilizando o mesmo algoritmo como uma caixa preta para analisar o conjunto de atributos. Nesta abordagem o critério para avaliar a relevância de atributos é a precisão do classificador.

Em outras palavras, métodos *wrapper* geram um subconjunto candidato de atributos, executam o algoritmo de indução considerando apenas este subconjunto de atributos selecionado do conjunto de treinamento, e utilizam a precisão resultante do classificador induzido para avaliar o subconjunto de atributos em questão. Este processo é repetido para cada subconjunto de atributos até que o melhor subconjunto de atributos seja encontrado ou outro critério de parada seja satisfeito.

Um argumento utilizado com muita frequência para apoiar a utilização da abordagem *wrapper* é que o mesmo algoritmo de indução que vai usar o subconjunto de atributos selecionado deve prover uma estimativa melhor de precisão que um outro algoritmo, o qual pode possuir um *bias* de indução totalmente diferente (Baranauskas et al., 1999).

Porém, a maior desvantagem dos métodos *wrapper* é o custo computacional, o qual resulta da repetida execução do algoritmo de indução para avaliar cada subconjunto de atributos a ser considerado (Baranauskas & Monard, 1998a).

Neste trabalho foi investigado o uso de redutos para a seleção de atributos relevantes, utilizando a abordagem filtro. Na próxima seção serão descritos os conjuntos de dados utilizados nos experimentos.

6.3 Conjuntos de Dados

Os experimentos foram realizados utilizando vários conjuntos de dados naturais de diferentes domínios. A maior parte dos conjuntos de dados foram obtidos do Reposi-

tório da UCI Irvine (Blake et al., 1998), exceto os conjuntos de dados nomeados Smoke e TA, os quais foram obtidos, respectivamente, a partir dos endereços a seguir:

- <http://lib.stat.cmu.edu/datasets/csb/>
- <http://www.stat.wisc.edu/p/stat/ftp/pub/loh/treeprogs/datasets/>

Com o objetivo de ajudar nas comparações, os conjuntos de dados escolhidos também possuem diferentes tipos de atributos. Os atributos podem ser contínuos ou nominais, o que não implica que estes tipos apareçam isoladamente em cada conjunto de dados. Foram ainda selecionados conjuntos de dados que contêm valores ausentes para validar o tratamento deste tipo de problema pelos diferentes algoritmos utilizados nos experimentos. Segue uma descrição geral dos conjuntos de dados.

6.3.1 Descrição Geral

Nesta seção são descritos resumidamente os conjuntos de dados utilizados nos experimentos. Descrições mais detalhadas sobre os conjuntos de dados utilizados podem ser encontradas em (Lee et al., 1999).

TA Este conjunto de dados consiste em medidas da qualidade do ensino num período de três semestres regulares e dois semestres de verão. As medidas são relativas a 151 professores assistentes do Departamento de Estatística da Universidade de Wisconsin – Madison.

Bupa Este conjunto de dados consiste em predições de quando um paciente tem ou não desordens no fígado com base em vários testes sanguíneos e no consumo de álcool.

Pima Neste conjunto de dados todos os pacientes são mulheres com idade mínima de 21 anos e pertencentes à linhagem de Índios Pima que vivem próximos a Phoenix, Arizona, USA. O problema é predizer quando uma paciente terá resultado positivo para o teste de diabetes.

Breast-cancer2 Este conjunto de dados é um dos conjuntos nomeados Breast Cancer que estão na UCI, no qual o problema é predizer sobre a recorrência de cancer de mama.

CMC Os exemplos presentes neste conjunto de dados são relativos a mulheres casadas que não estavam grávidas ou não sabiam se estavam grávidas no momento da entrevista. O problema consiste em prever o método contraceptivo escolhido por cada mulher (nenhum, método a curto prazo, método a longo prazo) com base nas características demográficas e sócio-econômicas de cada uma delas.

Breast-cancer Neste conjunto de dados o problema é prever quando uma amostra de tecido da mama extraído de uma paciente possui tumor benigno ou maligno.

Smoke Este conjunto de dados está relacionado ao problema de prever atitudes resultantes da restrição ao fumo em locais de trabalho (proibição, restrição, sem restrição) com base em leis, ambiente ou variáveis sócio-econômicas.

Hungarian Neste conjunto de dados os exemplos são relativos a diagnósticos de doenças cardíacas.

Hepatitis O conteúdo deste conjunto de dados está relacionado a previsão da expectativa de vida de pacientes com hepatite.

Na próxima seção são apresentadas as principais características de cada um dos conjuntos de dados, tais como o número de exemplos, o número de atributos e o percentual relativo à classe majoritária, entre outras.

6.3.2 Características dos Conjuntos de Dados

Na Tabela 6.1 é apresentado um resumo das principais características de cada um dos conjuntos de dados utilizados neste trabalho. É mostrado, o número de exemplos (#Exemplos), número e percentual de exemplos duplicados (aparecem mais que uma vez) ou conflitantes (possuem o mesmo conjunto atributo-valor mas diferente classe de decisão), número de atributos (#Atributos) contínuos e nominais, o erro majoritário e se o conjunto de dados tem ao menos um valor ausente¹.

Os conjuntos de dados são apresentados em ordem crescente do número de atributos, tal como serão apresentados em futuras tabelas e gráficos. A Figura 6.1 mostra a dimensão dos conjuntos de dados, *i.e.* o número de atributos e o número de exemplos de cada um deles. Deve ser observado que devido a grande variação na dimensão, o número de exemplos na Figura 6.1 é apresentado na escala logarítmica, *i.e.* $\log_{10}(\text{\#Exemplos})$.

¹Essas informações foram obtidas utilizando o utilitário *MCC++ info*.

Conjuntos de dados	#Exemplos	Duplicados ou conflitantes (%)	#Atributos (cont.,nom.)	Classe	%Classe	Erro Majoritário	Valores Ausentes
ta	151	45 (39.13%)	5 (1,4)	1 2 3	32.45% 33.11% 34.44%	65.56% na classe 3	N
bupa	345	4 (1.16%)	6 (6,0)	1 2	42.03% 57.97%	42.03% na classe 2	N
pima	769	1 (0.13%)	8 (8,0)	0 1	65.02% 34.98%	34.98% na classe 0	N
breast-cancer2	285	2 (0.7%)	9 (4,5)	recurrence no-recurrence	29.47% 70.53%	29.47% na classe no-recurrence	S
cmc	1473	115 (7.81%)	9 (2,7)	1 2 3	42.70% 22.61% 34.69%	57.30% na classe 1	N
breast-cancer	699	8 (1.15%)	9 (9,0)	2 4	65.52% 34.48%	34.48% na classe 2	S
smoke	2855	29 (1.02%)	13 (2,11)	0 1 2	5.29% 25.18% 69.53%	30.47% na classe 2	N
hungarian	294	1 (0.34%)	13 (13,0)	presence absence	36.05% 63.95%	36.05% na classe absence	S
hepatitis	155	0 (0%)	19 (6,13)	die live	20.65% 79.35%	20.65% na classe live	S

Tabela 6.1: Características dos Conjuntos de Dados

Figura 6.1: Dimensão dos Conjuntos de Dados

A próxima seção mostra como os experimentos foram organizados e conduzidos.

6.4 Organização dos Experimentos

Uma série de experimentos para a seleção de atributos através da abordagem filtro foi realizada utilizando os algoritmos e conjuntos de dados descritos nas Seções 5.7 pág. 66 e 6.3 pág. 73, respectivamente.

É importante notar que os dados originais não foram pré-processados de forma alguma na tentativa de remover ou substituir valores ausentes, ou mesmo transformar valores contínuos em categóricos. Além disso, os indutores foram executados utilizando os valores padrão para todos os parâmetros, ou seja, nenhuma tentativa foi feita para melhorar o desempenho de qualquer um dos indutores utilizados.

Os experimentos realizados podem ser divididos em dois passos principais — Figura 6.2²:

1. No primeiro passo os indutores *C4.5*, *ID3*, *CI* e *Rosetta* são utilizados como filtros na seleção de atributos relevantes.
2. No segundo passo os atributos selecionados no passo 1 são utilizados para calcular o erro dos classificadores induzidos por *C4.5-rules* e *CN2*, bem como o número de regras induzidas pelos dois.

Figura 6.2: Passos do Experimento

É importante notar que quando utiliza-se o *Rosetta* como filtro o resultado é um conjunto de redutos. Na configuração padrão, a ferramenta *Rosetta* calcula os redutos de forma que os redutos resultantes possua a mesma habilidade de discernir os exemplos uns dos outros. Portanto, cada reduto é um subconjunto de atributos selecionados, sendo que o número de atributos pode diferir entre os redutos.

²Embora a ferramenta *Rosetta* seja capaz de gerar regras, esta não foi utilizada para este fim, pois alguns experimentos iniciais feitos por nós mostraram que esta ferramenta tende a gerar um número de regras muito maior que os demais indutores.

Neste trabalho decidimos selecionar como filtro o reduto com o menor número de atributos. Nossa escolha está calcada no princípio de **Occam's Razor** (Mitchell, 1997) que diz:

“Prefira a hipótese que represente os dados de forma mais simples.”

Por exemplo, se a ferramenta Rosetta encontrar cinco redutos diferentes que possua a mesma relação de não-discernimento do conjunto de todos os atributos do conjunto de dados, selecionamos o reduto com menor número de atributos. É claro que essa escolha introduz algum *bias* em nossos experimentos.

6.5 Resultados Experimentais

Nesta seção são apresentados os resultados obtidos através dos experimentos³ descritos na seção anterior. Os resultados são apresentados em diversas tabelas relacionadas a cada conjunto de dados.

6.5.1 Descrição das Tabelas

Para cada conjunto de dados são apresentadas duas tabelas:

1. A primeira tabela descreve cada um dos atributos do conjunto de dados original: número do atributo (a numeração dos atributos começa em zero), tipo (contínuo ou nominal) e nome do atributo. Para os atributos nominais, o número máximo de valores possíveis (tal como descrito no arquivo *names* da Biblioteca *MCC++*) e o número atual de valores (o número encontrado pelo utilitário *MCC++ info*) são mostrados. Deve-se notar que para os atributos nominais, um número de valores atuais maior que o número de possíveis valores indica que existem valores ausentes para aquele atributo específico, embora o inverso não seja verdadeiro.
2. A segunda tabela mostra todos os atributos, os atributos selecionados pelos filtros, o erro na classificação, bem como o número de regras induzidas por

³Todos os experimentos foram executados em uma estação de trabalho Silicon Graphics *Indigo 2*, exceto os experimentos que envolviam a execução da ferramenta Rosetta que foram executados em um Pentium III 500Mhz padrão. Os tempos de execução da abordagem filtro foram muito próximos e por isso não surtem efeito na análise dos resultados.

$\mathcal{CN}2$ e $\mathcal{C}4.5$ -rules. Para especificar o experimento realizado é utilizada a notação $SSA(método, indutor)$, na qual:

- $método \in \{f\}$ indica o tipo de método utilizado para a seleção de atributos. Embora esta notação pareça incomum pelo fato de existir apenas um método, ela foi usada a fim de manter a uniformidade na notação com outros trabalhos experimentais realizados, tais como (Lee et al., 1999; Pila & Monard, 2001a) que envolvem os métodos *wrapper-forward*, *wrapper-backward* e *filtro*;
- $indutor \in \{\mathcal{C}4.5, ID3, CI, RS\}$ indica o algoritmo ou ferramenta utilizada como filtro.

Esta tabela apresenta para cada $SSA(método, indutor)$, o subconjunto de atributos selecionados, o número de atributos no subconjunto selecionado ($\#A$), a proporção de atributos selecionados ($\%A$), a média e o desvio padrão do erro dos indutores $\mathcal{CN}2$ e $\mathcal{C}4.5$ -rules calculados através de 10-fold-cross-validation⁴ (10-cv) utilizando todos os atributos, bem como os atributos selecionados em cada $SSA(método, indutor)$ considerado, e a quantidade de regras induzidas para cada um dos subconjuntos de atributos selecionados pelos diversos filtros utilizando os indutores $\mathcal{C}4.5$ -rules e $\mathcal{CN}2$. Nessa tabela, os erros podem estar assinalados com dois símbolos especiais:

- (a) • indica que esse erro é maior que o erro da classe majoritária, considerando somente a média do erro;
- (b) $\triangle$ indica que o erro é significativo com 95% de nível de confiança.

6.5.2 TA

Resultados experimentais obtidos para o conjunto de dados TA.

Número do Atributo	Nome do Atributo	#Valores Distintos		
		possíveis	atuais	tipo
#0	Eng-speaker	-	2	Nominal

continua na próxima página

⁴10-fold-cross-validation — cv — (validação cruzada) é executada dividindo-se o conjunto de dados em 10 conjuntos disjuntos (folds) de exemplos de tamanho aproximadamente iguais. O indutor é treinado e testado 10 vezes. A cada vez é testado em um *fold* e treinado com o restando do conjunto de dados menos os exemplos constantes naquele *fold*. O erro estimado por esse método é a média dos erros estimados pelos 10 diferentes *folds*.

continuação da página anterior

Número do Atributo	Nome do Atributo	#Valores Distintos		
		possíveis	atuais	tipo
#1	Course-inst	-	25	Nominal
#2	Course	-	26	Nominal
#3	Sem	-	2	Nominal
#4	Class-size	-	46	Contínuo

Tabela 6.5.1: TA – Descrição dos Atributos

Filtro	Atributos			Erro no 10-cv		Total de Regras	
	Selecionados	#A	%A	$\mathcal{CN}^2$	$\mathcal{C4.5}$ -rules	$\mathcal{CN}^2$	$\mathcal{C4.5}$ -rules
nenhum	todos	5	100.00%	51.67±3.42	53.58±6.00	61	17
SSA(f,CI)	0 1 2 3	4	80.00%	50.28±3.92	50.25±5.25	65	14
SSA(f,C4.5)	0 1 2 3 4	5	100.00%	51.67±3.42	53.58±6.00	70	17
SSA(f,ID3)	0 1 2 3 4	5	100.00%	51.67±3.42	53.58±6.00	63	17
SSA(f,RS)	1 2 4	3	60.00%	51.06±4.03	48.33±5.86	64	19

Tabela 6.5.2: TA – Atributos Selecionados, Erros na Classificação e Número de Regras

6.5.3 Bupa

Resultados experimentais obtidos para o conjunto de dados **Bupa**.

Número do Atributo	Nome do Atributo	#Valores Distintos		
		possíveis	atuais	tipo
#0	mcv	-	26	contínuo
#1	alkphos	-	78	contínuo
#2	sgpt	-	67	contínuo
#3	sgot	-	47	contínuo
#4	gammagt	-	94	contínuo
#5	drinks	-	16	contínuo

Tabela 6.5.3: Bupa – Descrição dos Atributos

Filtro	Atributos			Erro no 10-cv		Total de Regras	
	Selecionados	#A	%A	$\mathcal{CN}^2$	$\mathcal{C4.5}$ -rules	$\mathcal{CN}^2$	$\mathcal{C4.5}$ -rules
nenhum	todos	6	100.00%	35.35±2.01	34.13±2.85	34	11
SSA(f,CI)	4	1	16.67%	45.21±1.98●△	41.42±2.85	40	2
SSA(f,C4.5)	0 1 2 3 4 5	6	100.00%	35.35±2.01	34.13±2.85	34	11
SSA(f,ID3)	0 1 2 3 4 5	6	100.00%	35.35±2.01	34.13±2.85	37	11
SSA(f,RS)	0 1 2	3	50.00%	38.53±2.94	42.62±2.49●△	46	3

Tabela 6.5.4: Bupa – Atributos Selecionados, Erros na Classificação e Número de Regras

6.5.4 Pima

Resultados experimentais obtidos para o conjunto de dados **Pima**.

Número do Atributo	Nome do Atributo	#Valores Distintos		
		possíveis	atuais	tipo
#0	Number	-	17	contínuo
#1	Plasma	-	136	contínuo
#2	Diastolic	-	47	contínuo
#3	Triceps	-	51	contínuo
#4	Two	-	186	contínuo
#5	Body	-	248	contínuo
#6	Diabetes	-	517	contínuo
#7	Age	-	52	contínuo

Tabela 6.5.5: Pima – Descrição dos Atributos

Filtro	Atributos			Erro no 10-cv		Total de Regras	
	Selecionados	#A	%A	C $\mathcal{N}$ 2	C4.5-rules	C $\mathcal{N}$ 2	C4.5-rules
nenhum	todos	8	100.00%	25.12 $\pm$ 1.97	25.87 $\pm$ 1.07	56	6
SSA(f,CI)	0 1 4 5 6 7	6	75.00%	25.13 $\pm$ 1.49	26.53 $\pm$ 0.78	58	7
SSA(f,C4.5)	0 1 2 4 5 6 7	7	87.50%	23.69 $\pm$ 1.22	26.39 $\pm$ 1.13	53	8
SSA(f,ID3)	0 1 2 3 4 5 6 7	8	100.00%	25.12 $\pm$ 1.97	25.87 $\pm$ 1.07	56	6
SSA(f,RS)	1 2 6	3	37.50%	29.15 $\pm$ 1.31 Δ	27.71 $\pm$ 1.49	88	4

Tabela 6.5.6: Pima – Atributos Selecionados, Erros na Classificação e Número de Regras

6.5.5 Breast Cancer2

Resultados experimentais obtidos para o conjunto de dados **Breast Cancer2**.

Número do Atributo	Nome do Atributo	#Valores Distintos		
		possíveis	atuais	tipo
#0	Age	-	44	contínuo
#1	Age-at-meno	-	3	nominal
#2	Tumor-size	-	23	contínuo
#3	Involved-nodes	-	18	contínuo
#4	Node-capsule	3	3	nominal
#5	Degree-of-malig	-	3	contínuo
#6	Breast	-	2	nominal
#7	Breast-Quadrant	6	6	nominal
#8	Irradiation	-	2	nominal

Tabela 6.5.7: Breast Cancer2 – Descrição dos Atributos

Filtro	Atributos			Erro no 10-cv		Total de Regras	
	Selecionados	#A	%A	C $\mathcal{N}$ 2	C4.5-rules	C $\mathcal{N}$ 2	C4.5-rules
nenhum	todos	9	100.00%	27.03 $\pm$ 2.29	27.71 $\pm$ 1.73	40	12
SSA(f,CI)	1 2 3 4 5 6 7 8	8	88.89%	27.71 $\pm$ 1.68	29.46 $\pm$ 2.48	47	17
SSA(f,C4.5)	0 1 3 4 5 6 7 8	8	88.89%	29.16 $\pm$ 2.75	24.19 $\pm$ 2.37	48	6
SSA(f,ID3)	0 1 2 3 4 5 6 7 8	9	100.00%	27.03 $\pm$ 2.29	27.71 $\pm$ 1.73	40	12

continua na próxima página

continuação da página anterior						
Filtro	Atributos			Erro no 10-cv		Total de Regras
	Selecionados	#A	%A	CN2	C4.5-rules	CN2 C4.5-rules
SSA(f,RS)	0 2 3 5 7	5	55.56%	27.75±2.79	25.70±2.37	44 9

Tabela 6.5.8: Breast Cancer2 – Atributos Selecionados, Erros na Classificação e Número de Regras

6.5.6 Cmc

Resultados experimentais obtidos para o conjunto de dados **Cmc**.

Número do Atributo	Nome do Atributo	#Valores Distintos		
		possíveis	atuais	tipo
#0	Wage	-	34	contínuo
#1	Wedu	-	4	nominal
#2	Hedu	-	4	nominal
#3	Nchi	-	15	contínuo
#4	Wrel	-	2	nominal
#5	Work	-	2	nominal
#6	Hocu	-	4	nominal
#7	Stdliv	-	4	nominal
#8	Medexp	-	2	nominal

Tabela 6.5.9: Cmc – Descrição dos Atributos

Filtro	Atributos			Erro no 10-cv		Total de Regras	
	Selecionados	#A	%A	CN2	C4.5-rules	CN2	C4.5-rules
nenhum	todos	9	100.00%	49.64±1.01	45.90±1.38	174	36
SSA(f,CI)	0 1 2 3 4 5 6 7 8	9	100.00%	49.64±1.01	45.90±1.38	180	36
SSA(f,C4.5)	0 1 2 3 4 5 6 7 8	9	100.00%	49.64±1.01	45.90±1.38	176	36
SSA(f,ID3)	0 1 2 3 4 5 6 7 8	9	100.00%	49.64±1.01	45.90±1.38	174	37
SSA(f,RS)	0 1 2 3 4 5 6 7 8	9	100.00%	49.64±1.01	45.90±1.38	173	35

Tabela 6.5.10: Cmc – Atributos Selecionados, Erros na Classificação e Número de Regras

6.5.7 Breast Cancer

Resultados experimentais obtidos para o conjunto de dados **Breast Cancer**.

Número do Atributo	Nome do Atributo	#Valores Distintos		
		possíveis	atuais	tipo
#0	Clump Thickness	-	10	contínuo
#1	Uniformity of Cell Size	-	10	contínuo
#2	Uniformity of Cell Shape	-	10	contínuo
#3	Marginal Adhesion	-	10	contínuo
#4	Single Epithelial Cell Size	-	10	contínuo
#5	Bare Nuclei	-	10	contínuo
#6	Bland Chromatin	-	10	contínuo
#7	Normal Nucleoli	-	10	contínuo

continua na próxima página

continuação da página anterior				
Número do Atributo	Nome do Atributo	#Valores Distintos		
		possíveis	atuais	tipo
#8	Mitoses	-	9	contínuo

Tabela 6.5.11: Breast Cancer – Descrição dos Atributos

Filtro	Atributos			Erro no 10-cv		Total de Regras	
	Selecionados	#A	%A	$\mathcal{CN}^2$	$\mathcal{C4.5}$ -rules	$\mathcal{CN}^2$	$\mathcal{C4.5}$ -rules
nenhum	todos	9	100.00%	4.87 ± 0.77	4.29 ± 0.60	18	8
SSA(f,CI)	0 1 2 3 4 5 6 7 8	9	100.00%	4.87 ± 0.77	4.29 ± 0.60	19	8
SSA(f,C4.5)	0 1 2 3 4 5 6 8	8	88.89%	4.44 ± 0.61	4.29 ± 0.60	14	7
SSA(f,ID3)	0 1 2 3 4 5 6 7	8	88.89%	5.16 ± 0.86	4.86 ± 0.80	18	8
SSA(f,RS)	0 3 5 6	4	44.44%	$6.72 \pm 0.79 \triangle$	4.29 ± 0.67	31	7

Tabela 6.5.12: Breast Cancer – Atributos Selecionados, Erros na Classificação e Número de Regras

6.5.8 Smoke

Resultados experimentais obtidos para o conjunto de dados **Smoke**.

Número do Atributo	Nome do Atributo	#Valores Distintos		
		possíveis	atuais	tipo
#0	Weight	-	128	contínuo
#1	Time	-	2	nominal
#2	Work1	-	2	nominal
#3	Work2	-	2	nominal
#4	Residence	-	2	nominal
#5	Smoking1	-	2	nominal
#6	Smoking2	-	2	nominal
#7	Smoking3	-	2	nominal
#8	Smoking4	-	2	nominal
#9	Knowledge	-	13	nominal
#10	Sex	-	2	nominal
#11	Age	-	73	contínuo
#12	Education	-	5	nominal

Tabela 6.5.13: Smoke – Descrição dos Atributos

Filtro	Atributos			Erro no 10-cv		Total de Regras	
	Selecionados	#A	%A	$\mathcal{CN}^2$	$\mathcal{C4.5}$ -rules	$\mathcal{CN}^2$	$\mathcal{C4.5}$ -rules
nenhum	todos	13	100.00%	$32.18 \pm 0.64 \bullet \triangle$	$32.54 \pm 0.68 \bullet \triangle$	426	22
SSA(f,CI)	1 2 3 4 5 6 7 8 9 10 12	11	84.62%	$35.02 \pm 0.71 \bullet \triangle$	$33.21 \pm 0.82 \bullet$	410	26
SSA(f,C4.5)	0 1 2 3 4 5 6 7 8 9 10 11 12	13	100.00%	$32.18 \pm 0.64 \bullet \triangle$	$32.54 \pm 0.68 \bullet \triangle$	423	22
SSA(f,ID3)	0 1 2 3 4 5 6 7 8 9 10 11 12	13	100.00%	$32.18 \pm 0.64 \bullet \triangle$	$32.54 \pm 0.68 \bullet \triangle$	426	22
SSA(f,RS)	0 2 3 4 5 6 7 8 9 11 12	11	84.62%	$32.01 \pm 0.82 \bullet \triangle$	$33.10 \pm 1.01 \bullet \triangle$	474	37

Tabela 6.5.14: Smoke – Atributos Selecionados, Erros na Classificação e Número de Regras

6.5.9 Hungarian

Resultados experimentais obtidos para o conjunto de dados **Hungarian**.

Número do Atributo	Nome do Atributo	#Valores Distintos		
		possíveis	atuais	tipo
#0	age	-	38	contínuo
#1	sex	-	2	contínuo
#2	cp	-	4	contínuo
#3	trestbps	-	31	contínuo
#4	chol	-	153	contínuo
#5	fbs	-	2	contínuo
#6	restecg	-	3	contínuo
#7	thalach	-	71	contínuo
#8	exang	-	2	contínuo
#9	oldpeak	-	10	contínuo
#10	slope	-	3	contínuo
#11	ca	-	2	contínuo
#12	thal	-	3	contínuo

Tabela 6.5.15: Hungarian – Descrição dos Atributos

Filtro	Atributos			Erro no 10-cv		Total de Regras	
	Selecionados	#A	%A	CN2	C4.5-rules	CN2	C4.5-rules
nenhum	todos	13	100.00%	21.44±2.19	20.05±2.90	25	11
SSA(f,CI)	1 2 4 5 6 7 8 9 11 12	10	76.92%	21.79±2.22	20.41±2.18	30	8
SSA(f,C4.5)	0 1 2 3 4 5 6 7 8 9 10	11	84.62%	20.02±2.62	19.40±2.66	25	12
SSA(f,ID3)	0 1 2 3 4 5 7 8 9 10 12	11	84.62%	21.09±2.23	18.03±2.21	25	11
SSA(f,RS-b)	4 7 9	3	23.07%	26.17±3.11	20.75±3.61	43	2

Tabela 6.5.16: Hungarian – Atributos Selecionados, Erros na Classificação e Número de Regras

6.5.10 Hepatitis

Resultados experimentais obtidos para o conjunto de dados **Hepatitis**.

Número do Atributo	Nome do Atributo	#Valores Distintos		
		possíveis	atuais	tipo
#0	age	-	49	continuous
#1	female	2	2	nominal
#2	steroid	2	3	nominal
#3	antivirals	2	2	nominal
#4	fatigue	2	3	nominal
#5	malaise	2	3	nominal
#6	anorexia	2	3	nominal
#7	liver-big	2	3	nominal
#8	liver-firm	2	3	nominal
#9	spleen-palpable	2	3	nominal
#10	spiders	2	3	nominal
#11	ascites	2	3	nominal
#12	varices	2	3	nominal
#13	bilirubin	-	34	continuous
#14	alk-phosphate	-	83	continuous
#15	sgot	-	84	continuous

continua na próxima página

continuação da página anterior				
Número do Atributo	Nome do Atributo	#Valores Distintos		
		possíveis	atuais	tipo
#16	albumin	-	29	continuous
#17	protine	-	44	continuous
#18	histology	2	2	nominal

Tabela 6.5.17: Hepatitis – Descrição dos Atributos

Filtro	Atributos			Erro no 10-cv		Total de Regras	
	Selecionados	#A	%A	CN2	C4.5-rules	CN2	C4.5-rules
nenhum	todos	19	100.00%	16.18±1.80	20.54±3.02	19	10
SSA(f,CI)	2 3 5 8 10 11 13 16 17 18	10	52.63%	20.09±3.42	18.71±3.36	25	7
SSA(f,C4.5)	0 1 3 4 5 7 8 10 11 15 16 17	12	63.16%	14.86±2.53	18.75±2.03	20	10
SSA(f,ID3)	0 3 7 10 11 13 14 16 17	9	47.37%	18.17±2.21	19.46±2.44	22	6
SSA(f,RS)	0 10 16	3	15.79%	20.66±3.01●△	18.71±3.86	28	2

Tabela 6.5.18: Hepatitis – Atributos Selecionados, Erros na Classificação e Número de Regras

Na próxima seção são apresentadas comparações dos resultados mostrados anteriormente.

6.6 Análise dos Resultados

Nas próximas seções são mostradas tabelas que resumem os resultados obtidos nos experimentos a fim de facilitar a análise dos resultados.

6.6.1 Número de Atributos Selecionados

A Tabela 6.2 mostra, para cada conjunto de dados, o número total de atributos, o número de atributos selecionados pelos diversos filtros. A última linha dessa tabela (Total) apresenta o percentual de todos os atributos selecionados por cada método de SSA considerando todos os conjuntos de dados.

Considerando o trabalho relatado em (Lee et al., 1999), no qual foram feitos estudos envolvendo as abordagens filtro e wrapper para a seleção de atributos relevantes, o número de atributos selecionados pelo filtro CI é sempre menor ou igual ao número de atributos selecionados pelos filtros C4.5 e ID3, ou seja

$$\#SSA(f,CI) \leq \#SSA(f,C4.5) \text{ e } \#SSA(f,CI) \leq \#SSA(f,ID3)$$

No entanto, utilizando os redutos para a seleção de atributos relevantes resultou que o número de atributos selecionados pelo RS é sempre menor ou igual ao número de atributos selecionados pelos filtros $\mathcal{C}4.5$ e ID3, ou seja

$$\#SSA(f,RS) \leq \#SSA(f,\mathcal{C}4.5) \text{ and } \#SSA(f,RS) \leq \#SSA(f,ID3)$$

Além disso, o número de atributos selecionados pelo RS é menor ou igual ao número de atributos selecionados pelo CI, exceto para o conjunto de dados bupa. Pode-se observar que o percentual global dos atributos selecionados pelo RS é menor que 50%, enquanto que o percentual global dos atributos selecionados pelo CI é maior que 70%.

Tal como esperado, devido aos indutores $\mathcal{C}4.5$ e ID3 induzirem árvores de decisão, o número de atributos selecionados por ambos algoritmos é mais ou menos o mesmo, com exceção do conjunto de dados hepatitis. Além disso, o percentual global dos atributos selecionados é aproximadamente 85%.

A partir destes resultados e considerando somente o número de atributos selecionados em cada um dos quatro filtros — CI, $\mathcal{C}4.5$, ID3 e RS — é possível concluir que RS seleciona o menor número de atributos.

Conjunto de Dados	#A	SSA			
		(f,CI)	(f, $\mathcal{C}4.5$)	(f,ID3)	(f,RS)
ta	5	4	5	5	3
bupa	6	1	6	6	3
prima	8	6	7	8	3
breast cancer2	9	8	8	9	5
cmc	9	9	9	9	9
breast cancer	9	9	8	8	4
smoke	13	11	13	13	11
hungarian	13	10	11	11	3
hepatitis	19	10	12	9	3
Total	100%	74.73%	86.81%	85.71%	48.35%

Tabela 6.2: Número de Atributos Selecionados

6.6.2 Comparação sem Seleção de Atributos e com Seleção de Atributos utilizando Filtro

Para determinar quando a diferença entre dois algoritmos — por exemplo, A_1 e A_2 — é significativa ou não, vários gráficos são apresentados nesta seção, cada um com quatro barras para cada conjunto de dados. Cada barra corresponde à diferença da média do erro dividida pela diferença do desvio padrão. Quando o tamanho das bar-

ras é maior que dois (em módulo), os resultados são significativos com 95% de nível de confiança.

As comparações são feitas de forma que A_2 representa o indutor utilizando os atributos selecionados e A_1 é o indutor utilizando todos os atributos. Quando a barra está abaixo de zero significa que A_2 superou A_1 , ou seja, o uso dos atributos selecionados pelo filtro melhorou a precisão em relação ao algoritmo padrão.

Para cada conjunto de dados, a combinação da média $m(A_2 - A_1)$, o desvio padrão $dp(A_2 - A_1)$ e a diferença no desvio padrão são calculados conforme as equações a seguir:

$$m(A_2 - A_1) = m(A_2) - m(A_1) \quad (6.1)$$

$$dp(A_2 - A_1) = \sqrt{\frac{dp(A_2)^2 + dp(A_1)^2}{2}} \quad (6.2)$$

$$md(A_2 - A_1) = \frac{m(A_2 - A_1)}{dp(A_2 - A_1)} \quad (6.3)$$

A Tabela 6.3 mostra os resultados obtidos pela aplicação da Equação 6.3 para cada indutor (*indutor*) utilizando os atributos selecionados pela aplicação das abordagens filtro (SSA(f,*indutor*)).

Conjunto de Dados	SSA(f,CI) -CN2	SSA(f,C4.5) -CN2	SSA(f,ID3) -CN2	SSA(f,RS) -CN2
ta	1.97	0.00	0.00	-0.16
bupa	5.39	0.00	0.00	1.26
pima	0.44	-0.11	0.00	2.41
breast cancer2	-0.44	-1.18	0.00	0.29
cmc	0.00	0.00	0.00	-0.40
breast cancer	0.00	-1.01	-0.38	2.37
smoke	6.33	0.00	0.00	-0.23
hungarian	-0.45	0.19	-0.70	1.76
hepatitis	1.59	-1.11	-0.33	1.81

Conjunto de Dados	SSA(f,CI) -C4.5-rules	SSA(f,C4.5) -C4.5-rules	SSA(f,ID3) -C4.5-rules	SSA(f,RS) -C4.5-rules
ta	-0.70	0.00	0.00	-1.04
bupa	3.35	0.00	0.00	3.17
pima	1.62	0.00	0.00	1.42
breast cancer2	0.65	0.83	0.00	-0.97
cmc	0.00	0.00	0.00	0.00
breast cancer	0.00	0.00	0.00	0.64
smoke	-0.47	0.00	0.00	0.65
hungarian	-0.17	-0.17	0.11	0.21
hepatitis	-0.01	-1.51	0.07	-0.53

continua na próxima página

continuação da página anterior				
Conjunto de Dados	SSA(f,CI) -C4.5	SSA(f,C4.5) -C4.5	SSA(f,ID3) -C4.5	SSA(f,RS) -C4.5

Tabela 6.3: Diferença no Desvio Padrão dos Erros

As Figuras 6.3 e 6.4 mostram graficamente os resultados da Tabela 6.3.

Figura 6.3: $CN2$ – Diferença no Desvio Padrão dos Erros

Nesses gráficos, para cada conjunto de dados, as barras correspondem à comparação entre o indutor utilizando os atributos selecionados pelo algoritmo filtro e o indutor utilizando todos os atributos (nenhuma seleção de atributos).

Considerando somente os casos nos quais a abordagem filtro superou o indutor padrão com 95% de nível de confiança, ou ainda, nos casos que o indutor padrão superou a abordagem filtro com 95% de nível de confiança, tem-se para o indutor $CN2$ — Figura 6.3:

- Para o conjuntos de dados bupa e smoke, existem dois casos para os quais o indutor padrão superou a abordagem filtro.
- Para o conjuntos de dados pima e breast cancer, existem dois casos para os quais o indutor padrão superou o filtro RS.

E, para o indutor $\mathcal{C}4.5$ -rules — Figura 6.4:

- Para o conjunto de dados bupa, existem dois casos para os quais o indutor padrão superou as abordagens filtro CI e RS.

Na Tabela 6.4 são mostradas as melhorias da precisão para os indutores $\mathcal{C}4.5$ -rules e $\mathcal{CN}2$ quando utilizam os atributos selecionados pelos filtros ou todos os atributos (nenhuma seleção de atributos).

Conjuntos de Dados	SSA				#	#
	(f,CI)	(f,CI)	(f,RS)	(f,RS)	$\triangle$	∇
	CN2	$\mathcal{C}4.5$ -rules	CN2	$\mathcal{C}4.5$ -rules		
ta					0	0
bupa					0	3
pima	∇	∇		∇	0	1
breast cancer2			∇		0	0
cmc					0	0
breast cancer			∇		0	1
smoke	∇				0	1
hungarian					0	0
hepatitis					0	0
# $\triangle$	0	0	0	0	0	
# ∇	2	1	2	1		6

Tabela 6.4: Mudança na Precisão com Nível Significativo

Observe que na Tabela 6.4 são mostrados apenas os filtros CI e RS comparados com o indutor padrão. Isso porque não houve nenhuma mudança significativa com 95% de nível de confiança para os filtros $\mathcal{C}4.5$ e ID3.

Na Tabela 6.4, os casos que o filtro superou o indutor padrão com nível de confiança de 95% estão destacados com $\triangle$, enquanto que os casos que o indutor padrão superou o filtro com o referido nível de confiança estão destacados com ∇ .

Levando em consideração o nível de confiança de 95%, pode-se notar através da Tabela 6.4 que a abordagem filtro não superou o indutor padrão em nenhum caso, enquanto que o indutor padrão superou as abordagens filtro em 6 casos. Especificamente, quando utilizou-se RS como filtro, não houve caso em que este filtro superasse o indutor padrão, embora em 3 casos o indutor padrão o superou com 95%

Figura 6.4: C4.5-rules – Diferença no Desvio Padrão dos Erros

de nível de confiança.

Embora em nenhum caso o filtro superou o indutor padrão com nível de confiança de 95% como mostrado na Tabela 6.4, nós decidimos investigar estes resultados mais a frente. Uma das razões é que a abordagem filtro é um método muito rápido, em contraste com outras abordagens para seleção de atributos (Pila & Monard, 2001a). Além disso, em alguns casos, como, por exemplo, o alto custo na aquisição dos atributos, pode ser válido considerar a possibilidade de permitir um pequeno aumento no erro da classificação se alguns atributos custosos⁵ fossem descartados.

⁵Entende-se por atributos custosos aqueles de difícil aquisição em termos temporal e/ou financeiro.

6.6.3 Outros Resultados para SSA Utilizando Filtros

Nesta seção serão apresentados resultados experimentais que envolvem a seleção de atributos relevantes considerando aceitável um leve acréscimo no erro da classificação quando menos atributos são utilizados.

Algumas figuras são apresentadas, mostrando para cada conjunto de dados e indutor utilizado como filtro, a diferença do desvio padrão do erro, bem como o coeficiente que representa a proporção de atributos descartados depois da SSA utilizando cada filtro. Este coeficiente é calculado como mostrado a seguir:

$$Dec(f, D) = 1 - \frac{|Atributos_f|}{|Atributos_D|}$$

onde $|Atributos_D|$ é o total de atributos presentes no conjunto de dados D e $|Atributos_f|$ é o número de atributos selecionados utilizando o filtro f . Assim, $Dec(f, D)$ representa o percentual de atributos descartados após a SSA.

Nas figuras apresentadas a seguir, $Dec(f, D)$ é representado no eixo vertical esquerdo e o filtro correspondente no eixo direito. Assim, o filtro que aparece no canto superior direito é aquele que descartou mais atributos. Observar que é possível haver empate.

A diferença no desvio padrão do erro se refere ao indutor utilizando apenas os atributos selecionados pelo filtro correspondente comparado com o indutor utilizando todos os atributos. Com isso, barras para esquerda indicam vantagem do filtro enquanto que barras para direita indicam desvantagem.

Levando-se em consideração somente o percentual de atributos descartados pela SSA, pode-se observar que Rough Sets é similar ou melhor que os outros filtros, exceto para o conjunto de dados bupa, para o qual o filtro CI descartou mais atributos — Figure 6.6.

Contudo, o erro de classificação deve ser levado em conta para a escolha de um par conveniente (*Filtro, Indutor*), tal que o aumento no erro de classificação seja aceitável considerando o decréscimo no número de atributos. Assim, essa escolha torna-se subjetiva, pois irá depender de qual medida será priorizada, se a classificação ou o decréscimo no número de atributos. A seguir são feitas algumas considerações sobre essa escolha para cada um dos conjuntos de dados analisados.

Para o conjunto de dados *TA* — Figura 6.5 — SSA(f,RS) é apropriado para os dois indutores.

Figura 6.5: Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Conjunto de Dados Ta

Para o conjunto de dados *Bupa* — Figura 6.6 — SSA(f,RS) é a melhor opção, mas somente para o indutor $\mathcal{CN}2$. Na verdade, este conjunto de dados mostrou os piores resultados quando aplicados os métodos de filtro.

Figura 6.6: Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Conjunto de Dados Bupa

Para o conjunto de dados *Pima* — Figura 6.7 — SSA(f,RS) é apropriado, mas somente para o indutor $\mathcal{C4.5}$ -rules, e SSA(f,CI) para o indutor $\mathcal{CN}2$. Contudo, se o erro de classificação for o maior interesse, então SSA(f,C4.5) deve ser utilizado para os dois indutores.

Figura 6.7: Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Conjunto de Dados Pima

Para o conjunto de dados *Breast Cancer2* — Figura 6.8 — SSA(f,RS) é mais apropriado para o indutor C4.5-rules, enquanto que SSA(f,CI) deve ser utilizado com CN2.

Figura 6.8: Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Conjunto de Dados Breast Cancer2

Para o conjunto de dados *Cmc* — Figura 6.9 — todos os atributos parecem ser relevantes, uma vez que nenhum dos filtros foi capaz de descartar qualquer atributo.

Para o conjunto de dados *Breast Cancer* — Figura 6.10 — SSA(f,RS) é apropriado para o indutor C4.5-rules, porém não é apropriado para o indutor CN2, uma vez que o indutor padrão superou o filtro com 95% de nível de confiança. Para o indutor CN2,

Figura 6.9: Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Conjunto de Dados Cmc

$SSA(f,C4.5)$ é mais apropriado.

Figura 6.10: Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Conjunto de Dados Breast Cancer

Para o conjunto de dados *Smoke* — Figura 6.11 — $SSA(f,RS)$ é apropriado para o indutor $CN2$, e $SSA(f,CI)$ para o indutor $C4.5$ -rules.

Para o conjunto de dados *Hungarian* — Figura 6.12 — $SSA(f,RS)$ é apropriado para o indutor $C4.5$ -rules, e $SSA(f,CI)$ é apropriado para o indutor $CN2$. Novamente, se o erro de classificação for o maior interesse, então $SSA(f,CI)$ deve ser utilizado para os dois indutores.

Figura 6.11: Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Conjunto de Dados Smoke

Figura 6.12: Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Conjunto de Dados Hungarian

Para o conjunto de dados *Hepatitis* — Figura 6.13 — $SSA(f,RS)$ é apropriado para o indutor $\mathcal{C}4.5$ -rules, enquanto $SSA(f,ID3)$ para o indutor $\mathcal{CN}2$. Contudo, se o erro de classificação for o maior interesse, então $SSA(f,\mathcal{C}4.5)$ é uma boa opção para os dois indutores.

Figura 6.13: Diferença no Desvio Padrão dos Erros e Decréscimo no #A para o Conjunto de Dados Hepatitis

Até o momento os experimentos estavam centralizados na análise do número de atributos selecionados em cada um dos conjuntos de dados utilizando cada um dos filtros. Nas próximas duas seções serão apresentadas tabelas e comparativos sobre o número de regras geradas quando os indutores utilizam os atributos selecionados pelos filtros.

6.6.4 Número de Regras Induzidas

As Tabelas 6.5 e 6.6 mostram para os indutores $\mathcal{C}4.5$ -rules e $\mathcal{CN}2$, respectivamente, o número de regras geradas em cada conjunto de dados utilizando os atributos selecionados pelos filtros CI, $\mathcal{C}4.5$, ID3 e RS. Nessas tabelas também são apresentados o número total, a média e o desvio padrão em relação à quantidade de regras induzidas. Maiores detalhes podem ser encontrados em (Pila & Monard, 2001b).

Um resultado importante é que o número médio de regras induzidas pelo indutor $\mathcal{C}4.5$ -rules utilizando os atributos selecionados pelo filtro RS é menor que o número médio de regras induzidas por este mesmo indutor utilizando qualquer um dos outros subconjuntos de atributos. Para ilustrar estes resultados, seja $\#TotalRegras(Indu-$

tor, Filtro) o número total de regras induzidas pelo *Indutor* utilizando os atributos selecionados pelo *Filtro*, *i.e.*

$$\begin{aligned}
 \#TotalRegras(\mathcal{C}4.5\text{-rules}, SSA(f, RS)) &\leq \\
 \#TotalRegras(\mathcal{C}4.5\text{-rules}, SSA(f, CI)) &\leq \\
 \#TotalRegras(\mathcal{C}4.5\text{-rules}, SSA(f, ID3)) &\leq \\
 \#TotalRegras(\mathcal{C}4.5\text{-rules}, SSA(f, \mathcal{C}4.5)) &\leq \\
 \#TotalRegras(\mathcal{C}4.5\text{-rules}, todos) &
 \end{aligned}$$

Outro resultado importante e contrário ao anterior, é que para o indutor $\mathcal{CN}2$ o número médio de regras geradas utilizando todos os atributos é menor que o número médio de regras geradas utilizando qualquer um dos outros subconjuntos de atributos, *i.e.*

$$\begin{aligned}
 \#TotalRegras(\mathcal{CN}2, todos) &\leq \\
 \#TotalRegras(\mathcal{CN}2, SSA(f, ID3)) &\leq \\
 \#TotalRegras(\mathcal{CN}2, SSA(f, CI)) &\leq \\
 \#TotalRegras(\mathcal{CN}2, SSA(f, \mathcal{C}4.5)) &\leq \\
 \#TotalRegras(\mathcal{CN}2, SSA(f, RS)) &
 \end{aligned}$$

Este último resultado confirma que o $\mathcal{CN}2$ trabalha melhor quando é deixado que ele faça sua própria seleção de atributos. Na verdade, o número de regras induzidas pelo $\mathcal{CN}2$ aumenta conforme o número de atributos selecionados decresce. Por exemplo, $SSA(f, RS)$ selecionou, na média, o menor número de atributos, e o $\mathcal{CN}2$ induziu o maior número de regras (991), considerando todos os conjuntos de dados. Por outro lado, $\mathcal{C}4.5\text{-rules}$ induziu o menor número de regras (118) nesse caso. Também, nas Tabelas 6.5 e 6.6 pode ser observado que o $\mathcal{CN}2$ possui uma tendência a induzir maior quantidade de regras que o $\mathcal{C}4.5\text{-rules}$. Na verdade, os resultados para todos os conjuntos de dados e filtros mostram que o número de regras induzidas pelo $\mathcal{CN}2$ é maior que o número de regras induzidas pelo $\mathcal{C}4.5\text{-rules}$, *i.e.*

$$\#TotalRegras(\mathcal{CN}2, \text{Todos ou } SSA) > \#TotalRegras(\mathcal{C}4.5\text{-rules}, \text{Todos ou } SSA)$$

Conjuntos de Dados	Número de Regras (%A)					utilizando Filtro		
	Todos	(f, CI)	(f, $\mathcal{C}4.5$)	(f, ID3)	(f, RS)	Total	Média	Desv-pad
ta	17	14 (80.00%)	17 (100.00%)	17 (100.00%)	19 (60.00%)	67	16.75	2.06
bupa	11	2 (16.67%)	11 (100.00%)	11 (100.00%)	3 (50.00%)	27	6.75	4.92
pima	6	7 (75.00%)	8 (87.50%)	6 (100.00%)	4 (37.50%)	25	6.25	1.71
breast cancer2	12	17 (88.89%)	6 (88.89%)	12 (100.00%)	9 (55.56%)	44	11.00	4.69
cmc	36	36 (100.00%)	36 (100.00%)	36 (100.00%)	36 (100.00%)	144	36.00	0.00
breast cancer	8	8 (100.00%)	7 (88.89%)	8 (88.89%)	7 (44.44%)	30	7.50	0.58
smoke	22	26 (84.62%)	22 (100.00%)	22 (100.00%)	37 (84.62%)	107	26.75	7.09
hungaria	11	8 (76.92%)	12 (84.62%)	11 (84.62%)	2 (23.07%)	33	8.25	4.50
hepatitis	10	7 (52.63%)	10 (63.16%)	6 (47.37%)	2 (15.79%)	25	6.25	3.30

continua na próxima página

continuação da página anterior

Conjuntos de Dados	Número de Regras (%A)					utilizando Filtro
	Todos	(f,CI)	(f,C4.5)	(f,ID3)	(f,RS)	Total Média Desv-pad
Total	133	125	129	129	119	
Média	14.78	13.89	14.33	14.44	13.11	
Desv-pad	9.28	10.90	9.58	9.91	14.01	

Tabela 6.5: Número de Regras Induzidas pelo C4.5-rules

Conjuntos de Dados	Número de Regras (%A)					utilizando Filtro		
	Todos	(f,CI)	(f,C4.5)	(f,ID3)	(f,RS)	Total	Média	Desv-pad
ta	61	65 (80.00%)	63 (100.00%)	63 (100.00%)	64 (60.00%)	255	63.75	0.96
bupa	34	40 (16.67%)	34 (100.00%)	37 (100.00%)	46 (50.00%)	157	39.25	5.12
pima	56	58 (75.00%)	53 (87.50%)	56 (100.00%)	88 (37.50%)	255	63.75	16.30
breast cancer2	40	47 (88.89%)	48 (88.89%)	40 (100.00%)	44 (55.56%)	179	44.75	3.59
cmc	174	180 (100.00%)	176 (100.00%)	174 (100.00%)	173 (100.00%)	703	175.75	3.10
breast cancer	18	19 (100.00%)	14 (88.89%)	18 (88.89%)	31 (44.44%)	82	20.50	7.33
smoke	426	410 (84.62%)	423 (100.00%)	426 (100.00%)	474 (84.62%)	1743	435.75	25.62
hungaria	25	30 (76.92%)	25 (84.62%)	25 (84.62%)	43 (23.07%)	123	30.75	8.50
hepatitis	19	25 (52.63%)	20 (63.16%)	22 (47.37%)	28 (15.39%)	95	23.75	3.50
Total	853	874	884	861	991			
Média	94.78	95.11	98.22	95.67	110.11			
Desv-pad	133.15	132.26	130.03	132.71	143.60			

Tabela 6.6: Número de Regras Induzidas pelo CN2

6.7 Considerações Finais

Neste capítulo foram apresentados resultados experimentais relativos à seleção de atributos relevantes através da abordagem filtro, utilizando tanto RS como outros três filtros. Comparações em relação ao número de atributos selecionados, precisão e número de regras induzidas — com os atributos selecionados por cada filtro — foram também apresentadas. Ficou explícito que a abordagem de Rough Sets é uma possível escolha para a seleção de atributos e, em geral, escolhe o menor número de atributos relevantes. Outro resultado interessante é que o número de regras induzidas pelo C4.5-rules diminui, enquanto que para o CN2 aumenta, conforme o número de atributos selecionados decresce. Isso mostra que os resultados do CN2 tendem a ser melhores se a seleção dos atributos relevantes for deixada por conta dele. Já o C4.5-rules tende a gerar menor quantidade de regras quando utilizado em conjunto com o filtro RS, uma vez que este seleciona a menor quantidade de atributos.

No próximo capítulo são apresentados experimentos e resultados envolvendo um estudo de casos de um conjunto de dados do mundo real.

Capítulo 7

Talassemia α — Um Estudo de Caso

7.1 Considerações Iniciais

Neste capítulo será descrito o estudo de caso realizado a respeito de um conjunto de dados sobre talassemia α . O objetivo deste estudo de caso é verificar a aplicação dos métodos de AM abordados nos capítulos anteriores em um conjunto de dados do mundo real. O estudo de um conjunto de dados real envolve uma série de problemas, tal como desbalanceamento de exemplos (Batista, 2000; Batista et al., 1999; Batista et al., 2000), limpeza de dados e outros, os quais não são enfrentados quando os experimentos são realizados utilizando-se conjuntos de dados obtidos de repositórios, como o Repositório de Dados da UCI-Irvine (Blake et al., 1998), conhecidos como conjuntos de dados naturais. O principal motivo é que conjuntos de dados naturais são previamente tratados de forma a não conter imperfeições grosseiras. Porém, essas imperfeições nos dados torna o estudo de caso interessante, por propiciar um contato direto com informações reais e com o especialista do domínio. A seguir são mostradas as principais fases de um estudo de caso.

7.2 Fases do Estudo de Caso

O estudo de caso é uma tarefa que pode ser dividida em quatro fases — Figura 7.1:

1. Definição e compreensão do domínio: nesta fase ocorre o primeiro contato com o especialista do domínio. São esclarecidas as principais características do conjunto de dados e o especialista informa que tipo de conhecimento ele

desejaria que fosse extraído do conjunto de dados, embora existam casos para os quais o especialista não sabe que tipo de conhecimento possa ser extraído, e espera algum conhecimento novo.

2. Limpeza e pré-processamento dos dados: nesta fase são eliminados, segundo indicações do especialista, atributos que não fazem parte do domínio do problema em estudo. Além disso, os atributos e seus valores são transformados para o formato que os algoritmos de AM reconhecem.
3. Aplicação dos métodos de AM: nesta fase, os métodos de AM são aplicados para, por exemplo, selecionar atributos relevantes, extrair conhecimento na forma de regras e árvores de decisão¹ e testar o conhecimento extraído.
4. Avaliação do conhecimento: nesta fase os resultados são analisados e o conhecimento extraído é avaliado pelo especialista do domínio.

Figura 7.1: Fases da Extração de Conhecimento

A seguir são descritas as atividades realizadas em cada uma dessas fases.

7.3 Definição e Compreensão do Domínio

Nas próximas duas seções são apresentadas a descrição do domínio ao qual pertence o conjunto de dados e o tipo de conhecimento que o especialista deseja que seja

¹Por estarmos tratando de Aprendizado de Máquina Simbólico, os algoritmos aplicados são aqueles que podem expressar o conhecimento na forma de regras ou árvores de decisão.

extraído desses dados.

7.3.1 Dados sobre Hematologia — Talassemia α

O Laboratório de Hematologia do Departamento de Patologia Clínica da Faculdade de Ciências Médicas da Universidade Estadual de Campinas — UNICAMP — realiza a rotina diária dos hemogramas provenientes de todos os ambulatórios do complexo hospitalar da UNICAMP. Não raramente, são observados dados hematimétricos alterados, com o volume corpuscular médio (VCM) e a hemoglobina corpuscular média (HCM) reduzidos, em indivíduos nos quais a taxa global de hemoglobina se encontra dentro da faixa de normalidade (sem anemia) e a HbA₂ em níveis normais ou diminuídos (Borges, 2000).

A microcitose e a hipocromia, sem o concomitante aumento da HbA₂, podem ser resultantes da presença de talassemia α , de anemia por deficiência de ferro ou, ocasionalmente, da anemia por doenças crônicas. Não raramente, indivíduos com microcitose e hipocromia, sem anemia e sem a elevação da HbA₂, são detectados em exames hematológicos de rotina. Com o objetivo de investigar a contribuição da talassemia α nestes casos, foram analisados 339 indivíduos adultos, atendidos nos ambulatórios do Hospital das Clínicas da UNICAMP, apresentando níveis de hemoglobina (Hb) maiores ou iguais a 12g/dL para mulheres e 14g/dL para homens, volume corpuscular médio das hemácias (VCM) e hemoglobina corpuscular média (HCM) menor ou igual a 80fl e 27pg, respectivamente, e percentuais de HbA₂ normais ou diminuídos ($\leq 3,4\%$). Os dados obtidos são de relevância clínica, uma vez que a microcitose e a hipocromia são comumente interpretadas como sinais de deficiência de ferro.

As síndromes talassêmicas α representam uma alteração genética freqüente, de distribuição mundial, ocasionada pela deficiência de síntese das cadeias α da Hb. A redução de cadeias promove uma menor formação de Hb nos eritrócitos, causando microcitose e hipocromia, e levando ao acúmulo das outras cadeias cuja síntese está normal.

O diagnóstico da talassemia α apresenta dificuldades. Os heterozigotos das talassemias α^0 e α^+ e os homozigotos da talassemia α^+ só podem ser diagnosticados na vida adulta por análise de DNA, pois a Hb Bart's desaparece após o período neonatal; além disso, a pouca quantidade produzida nos heterozigotos da talassemia α^+ (1-3%) pode muitas vezes não ser detectada pelos métodos rotineiramente empregados. A microcitose e a hipocromia são alterações hematológicas que não podem ser consideradas como exclusivas da talassemia α , uma vez que são características das demais

talassemias, como as talassemias β , da anemia ferropriva e, eventualmente, ocorrem nas anemias das doenças crônicas (Borges, 2000).

O conteúdo deste conjunto de dados foi fornecido pela Dra. Maria de Fátima Sonati (especialista) com o objetivo de avaliar a presença de talassemia α em função das demais informações presentes no hemograma, excluindo-se os pacientes atendidos nos ambulatórios de Hematologia Clínica, em função da elevada frequência de anemias carenciais e para se evitar o direcionamento das amostras.

7.3.2 Objetivos

Segundo a especialista, a presença da talassemia α nos exemplos deste conjunto de dados foram determinadas através do exame de DNA, o qual demanda tempo além de ser financeiramente custoso. A especialista desejava que fosse possível determinar a existência de talassemia α utilizando somente os dados do hemograma, ou seja, uma determinação da talassemia α sem a necessidade do exame de DNA. Deve ser observado que a determinação da existência da talassemia α através do exame de DNA é feita sem a necessidade do conhecimento de qualquer valor dos demais atributos. Nesse exame, a técnica utilizada é a Reação em Cadeia da Polimerase, sendo auto-suficiente para a determinação da talassemia α , embora custosa. Sendo assim, qualquer relação explícita que os algoritmos de AM venham a descobrir entre os valores dos atributos presentes nos hemogramas² (conjunto de dados) e a decisão, pode representar conhecimento novo, ou mesmo confirmar algum tipo de relação que possa ocorrer entre esses atributos e a decisão.

Após a definição dos objetivos da extração do conhecimento, o conjunto de dados foi submetido ao pré-processamento e limpeza dos dados. Essa fase será descrita a seguir.

7.4 Limpeza e Pré-Processamento dos Dados

O conjunto de dados original, contendo 339 exemplos e utilizado neste estudo, está organizado em um documento no formato *planilha do Excel*. Cada linha da planilha constitui um exemplo obtido nos laboratórios do Hospital das Clínicas da UNICAMP. Inicialmente a planilha foi fornecida contendo os seguintes atributos para cada paciente:

²Hemogramas demandam pouco tempo e são de baixo custo.

1. NT: identificador do exemplo;
2. Idade: idade do paciente relacionado ao exemplo;
3. Cor: cor da pele;
4. Sexo: sexo do paciente (F ou M)
5. GV: nível de glóbulos vermelhos;
6. HB: nível de hemoglobina;
7. HT: nível hematológico;
8. VCM: volume corpuscular médio das hemácias;
9. HCM: hemoglobina corpuscular média;
10. RDW: amplitude da distribuição dos eritrócitos;
11. Padrão Eletroforético: o valor deste atributo está relacionado ao exame de DNA;
12. HBA₂: nível de hemoglobina A2;
13. HBF: nível de hemoglobina F;
14. Ferritina Sérica: nível das reservas de ferro no organismo;
15. PCR1: resultados da técnica da Reação em Cadeia da Polimerase para deleção $\alpha^{3.7}$;
16. PCR2: resultados da técnica da Reação em Cadeia da Polimerase para deleção $\alpha^{4.2}$;
17. PCR3: resultados da técnica da Reação em Cadeia da Polimerase para deleção $-\text{MED}$;
18. PCR4: resultados da técnica da Reação em Cadeia da Polimerase para deleção $(-\alpha)^{20.5}$;
19. PCR5: resultados da técnica da Reação em Cadeia da Polimerase para a forma não delecional $\alpha^{\text{HphI}}\alpha$;
20. PCR6: resultados da técnica da Reação em Cadeia da Polimerase para as formas não delecionais α^{NcoI} e $\alpha\alpha^{\text{NcoI}}$;
21. PCR7: resultados da técnica da Reação em Cadeia da Polimerase para a forma não delecional $\alpha^{\text{TsaDI}}\alpha$;

A aquisição dos valores do atributo de decisão é custosa, como já exposto anteriormente, sendo que os valores de alguns dos atributos apresentados são determinados pelo exame de DNA. Levando-se em consideração essa informação e as necessidades da especialista, foi possível a exclusão de alguns atributos que não deveriam fazer parte do processo de extração de conhecimento, uma vez que os valores desses atributos foram determinados posteriormente através do exame de DNA, ou mesmo, segundo a especialista, não são diretamente relevantes para o problema em questão. Os atributos excluídos do conjunto de dados original foram:

- NT: por ser apenas o identificador do exemplo;

- GEL, Padrão Eletroforético, HBA₂, HBF e Ferritina Sérica: por serem atributos cujos valores foram determinados posteriormente utilizando os resultados do DNA;
- PCR2, PCR3, PCR4, PCR5, PCR6 e PCR7: por serem outros resultados da técnica da Reação em Cadeia da Polimerase não ligados diretamente à determinação da presença de talassemia α .

Após a exclusão desses atributos, o conjunto de dados possuía apenas atributos relacionados a informações extraídas do hemograma. Os valores do atributo PCR1, que são encontrados através do exame de DNA, contém os valores relativos à presença de talassemia α com 100% de confiança nessa determinação. Em outras palavras, o atributo PCR1 é a classe (decisão) do conjunto de exemplos.

Inicialmente, os possíveis valores para o atributo PCR1 eram *hetero*, *homo* e *normal*. Ambos valores *hetero* e *homo* indicam a presença de talassemia α , porém com o diferencial de saber se a doença era proveniente do pai ou da mãe, enquanto que o valor *normal* indica que o paciente não possui talassemia α .

Como o interesse da especialista está relacionado somente à determinação da talassemia α sem a necessidade de saber sua origem, e ainda devido ao fato da classe *homo* possuir poucos exemplos (apenas 5.6%) do total, foi decidido substituir o atributo PCR1 por um novo atributo de decisão, o qual denominamos PCR, com apenas dois valores: normal e anormal. O valor *anormal* representa os exemplos de ambas classes *hetero* e *homo* do atributo PCR1.

As características gerais do conjunto de dados, após essa limpeza, constam na Tabela 7.1.

#Exemplos	Duplicados ou conflitantes (%)	#Atributos (cont.,nom.)	Classe	%Classe	Erro Majoritário	Valores Ausentes
339	1 (0.29%)	9 (7,2)	normal anormal	51.9% 48.1%	48.1% na classe normal	N

Tabela 7.1: Característica do Conjuntos de Dados Talassemia α

Na Tabela 7.2 são descritas as características de cada um dos atributos desse conjunto de dados³.

Número do Atributo	Nome do Atributo	Descrição do Atributo	Valores		
			#possíveis	#atuais	tipo
#1	Idade	Idade do paciente	—	64	contínuo

continua na próxima página

³Informações obtidas através do utilitário *Info* da *MCC++*.

continuação da página anterior

Número do Atributo	Nome do Atributo	Descrição do Atributo	#possíveis	Valores #atuais	tipo
#2	Cor	Cor da Pele	2	2	discreto
#3	Sexo	Sexo	2	2	discreto
#4	GV	Nível de glóbulos vermelhos	—	150	contínuo
#5	HB	Nível de hemácias	—	44	contínuo
#6	HT	Nível Hematológico	—	123	contínuo
#7	VCM	Volume corpuscular médio	—	90	contínuo
#8	HCM	Hemoglobina corpuscular média	—	56	contínuo
#9	RDW	Amplitude da distribuição dos eritrócitos	—	97	contínuo
#10	PCR	Determina a presença/ausência de talassemia α	2	2	discreto

Tabela 7.2: Descrição dos Atributos do Conjuntos de Dados Talassemia α

Algumas estatísticas a respeito dos valores dos atributos desse conjunto de dados são apresentadas na Tabela 7.3 e podem servir como parâmetros de comparação nos resultados a seguir apresentados.

Número do Atributo	Média	Mediana	Desvio Padrão	Mínimo	Máximo	1°. Quartil	3°. Quartil
#1	38.47	38.00	15.69	14.00	79.00	26.00	8.00
#2	—	—	—	—	—	—	—
#3	—	—	—	—	—	—	—
#4	5.22	5.13	0.52	4.13	7.67	4.84	5.47
#5	13.38	13.10	1.21	12.00	19.20	12.40	14.10
#6	40.16	39.40	4.31	3.91	58.10	37.50	42.60
#7	77.10	78.00	3.04	64.10	80.40	75.50	79.50
#8	25.57	25.80	1.27	20.50	27.40	24.90	26.50
#9	16.78	16.10	2.98	12.30	38.30	15.00	17.40
#10	—	—	—	—	—	—	—

Tabela 7.3: Estatísticas dos Atributos do Conjunto de Dados Talassemia α

Na próxima seção são apresentados os experimentos realizados, a fim de extrair conhecimento utilizando esse conjunto de dados, e os resultados alcançados.

7.5 Extração de Conhecimento: Experimentos e Resultados

A seguir são descritos os principais passos que compuseram os experimentos realizados com o conjunto de dados sobre talassemia α .

7.5.1 Descrição dos Experimentos

Como já mencionado, o objetivo deste estudo é a construção de um classificador capaz de classificar novos exemplos como pertencentes à classe dos portadores de talassemia α (anormal), ou como pessoas não portadores dessa doença (normal) usando somente as informações obtidas de hemogramas de pacientes. Para alcançar esse objetivo, foram realizados vários experimentos envolvendo a seleção de atributos relevantes e extração de conhecimento no formato de regras. Os experimentos foram realizados seguindo quatro passos, os quais estão esquematizados na Figura 7.2:

1. **Passo 1:** utilizando a ferramenta Rosetta, foi feita a seleção de atributos relevantes utilizando redutos como filtro. Vale lembrar que na procura por redutos a ferramenta Rosetta pode achar vários redutos, ou seja, vários subconjunto de atributos relevantes segundo a abordagem de Rough Sets. Novamente, levando-se em consideração a idéia que é melhor representar as hipóteses com o menor número de atributos possível (Mitchell, 1997), foram escolhidos alguns redutos com o menor número de atributos.

No entanto, como já exposto anteriormente, para calcular os redutos a ferramenta Rosetta não leva em consideração se o conjunto de dado está totalmente discretizado ou ainda contém atributos contínuos. Na verdade, assume-se que todos os atributos são discretos, o que pode favorecer a seleção de atributos contínuos que forem tratados como discretos. Para avaliar isso os redutos foram calculados sobre o conjunto de dados sem qualquer tipo de discretização, bem como sobre o conjunto de dados discretizado utilizando dois métodos diferentes de discretização (Entropia e RB).

2. **Passo 2:** de posse dos vários subconjuntos de atributos selecionados pela ferramenta Rosetta, esses foram avaliados sobre sua real capacidade de representação do conjunto de dados (desempenho).
3. **Passo 3:** os redutos selecionados segundo o desempenho foram utilizados para representar o conjunto de dados inicial em vários outros conjuntos de dados reduzidos, cada qual contendo apenas os atributos pertencentes a cada reduto.
4. **Passo 4:** os conjuntos de dados reduzidos foram submetidos aos indutores $\mathcal{C}4.5$ -rules, $\mathcal{CN}2$ e Rosetta para a avaliação tanto do erro na classificação quanto do número de regras induzidas.

Figura 7.2: Experimentos Realizados sobre o Conjunto de Dados Talassemia α

Os resultados obtidos pela aplicação dos passos definidos anteriormente no conjunto de dados em estudo são apresentados na próxima subseção.

7.5.2 Resultados Obtidos

O primeiro passo foi calcular os redutos do conjunto de dados, para então avaliar quais redutos poderiam ser utilizados para representar o conjunto de dados de forma reduzida. Os resultados relacionados aos atributos seleccionados são descritos na próxima seção.

7.5.2.1 Atributos Seleccionados

A seleção de atributos relevantes segundo a abordagem de Rough Sets (reduto) foi feita levando em consideração três formas diferentes de representação do conjunto de dados em estudo:

1. Conjunto de dados original — Tabela 7.2.
2. Conjunto de dados discretizado pelo método de Raciocínio Booleano.
3. Conjunto de dados discretizado pelo método da Entropia⁴.

Deve ser observado que existem duas possíveis formas de calcular os redutos na ferramenta Rosetta. Calcular os redutos de forma que os atributos pertencentes ao reduto representem da forma mais apropriada o conjunto de dados na forma reduzida,

⁴Maiores informações sobre esse método estão descritas na Seção 5.6.

ou ainda calcular os redutos relativos a alguns exemplos. Neste último caso os redutos resultantes são suficientes apenas para diferenciar os demais exemplos daqueles utilizados para o cálculo dos redutos. Como o objetivo deste estudo de caso é a construção de um classificador no formato de regras capazes de prever a classe de novos exemplos, os redutos foram calculados utilizando a primeira forma (opção *Full* na ferramenta Rosetta). Os resultados obtidos encontram-se na Tabela 7.4.

Original			
Atributos Seleccionados	#A	%A	Desempenho (%)
4 9	2	22.22%	99.71%
1 5 9	3	33.33%	99.71%
1 8 9	3	33.33%	99.71%
5 8 9	3	33.33%	99.71%
1 5 8	3	33.33%	99.71%
7 8 9	3	33.33%	99.71%
5 7 9	3	33.33%	99.71%
1 7 8	3	33.33%	99.71%
1 5 7	3	33.33%	99.71%
6 8 9	3	33.33%	99.71%
6 7 9	3	33.33%	99.71%
6 7 8	3	33.33%	99.71%
5 6 9	3	33.33%	99.71%
5 6 7	3	33.33%	99.71%
1 6 9	3	33.33%	99.71%
1 6 8	3	33.33%	99.71%
1 6 7	3	33.33%	99.71%
1 5 6	3	33.33%	99.71%
1 4 8	3	33.33%	99.71%
4 7 8	3	33.33%	99.71%
1 4 7	3	33.33%	99.71%
4 6 8	3	33.33%	99.71%
4 6 7	3	33.33%	99.71%
1 4 6	3	33.33%	99.71%
4 5 6	3	33.33%	99.71%
1 4 5	3	33.33%	99.71%
2 5 7 8	4	44.44%	99.71%
2 5 6 8	4	44.44%	99.71%
Discretizado — Raciocínio Booleano			
Atributos Seleccionados	#A	%A	Desempenho (%)
1 4 5 6 7 8 9	7	77.78%	99.71%
Discretizado — Entropia			
Atributos Seleccionados	#A	%A	Desempenho (%)
1 2 3 4 7 8	6	66.67%	98.82%
1 2 3 5 7 8	6	66.67%	98.82%

Tabela 7.4: Conjunto de Dados – Atributos Seleccionados

Os redutos presentes na Tabela 7.4 são calculados levando em consideração a relação de não-discernimento da Teoria de Rough Sets. Utilizando a ferramenta Rosetta foi possível obter os índices de correlação entre os atributos. Esses índices de correlação estão na Tabela 7.5.

Atributos	1	2	3	4	5	6	7	8	9
2	-0.073								
3	0.141	0.141							
4	-0.006	0.118	0.581						
5	0.013	0.111	0.679	0.824					
6	0.019	0.114	0.574	0.795	0.843				
7	0.110	-0.011	0.034	-0.337	0.128	0.047			

continua na próxima página

<i>continuação da página anterior</i>									
Atributos	1	2	3	4	5	6	7	8	9
8	0.032	-0.041	0.054	-0.366	0.160	-0.032	0.866		
9	0.071	-0.140	-0.082	0.045	-0.090	-0.030	-0.216	-0.246	
10	0.177	-0.271	-0.063	-0.095	-0.086	-0.061	0.065	0.057	0.285

Tabela 7.5: Índice de Correlação entre os Atributos

Utilizando a correlação entre os atributos, foram extraídos outros dois subconjuntos de atributos que possuem os maiores índices de correlação (em módulo) com a decisão (atributo 10), quando comparados com os demais. Assim, os atributos $\{2, 9\}$ e $\{1, 2, 9\}$ formaram dois novos subconjuntos de atributos relevantes a serem investigados posteriormente.

Analisando a Tabela 7.4 pode-se observar que os redutos calculados através do conjunto de dados sem qualquer discretização e discretizado pelo método de Raciocínio Booleano — RB — são os que possuem melhor desempenho (99.71%). Essa medida de desempenho está relacionada ao poder de discernimento de cada um dos redutos e foi obtida através da ferramenta Rosetta. Entretanto, os redutos calculados com o conjunto de dados original, contém um número bem menor de atributos (máximo de 4 atributos) que o reduto calculado com o conjunto de dados discretizado pelo método RB (7 atributos). Assim, foi decidido trabalhar com os primeiros redutos.

No entanto, todos os redutos calculados através do conjunto de dados original possuem o mesmo desempenho. Sendo assim, foram escolhidos quatro desses redutos. Dessa forma, um total de seis subconjuntos de atributos foram selecionados no Passo 1 — Figura 7.2 — para comparar seu poder de representação ao conjunto de todos os atributos do conjunto de dados original.

1. $\{2, 9\}$: obtido através dos índices de correlação;
2. $\{4, 9\}$: primeiro reduto calculado que contém dois atributos;
3. $\{1, 2, 9\}$: obtido através dos índices de correlação;
4. $\{1, 5, 9\}$: primeiro reduto calculado que contém três atributos;
5. $\{1, 6, 8\}$: reduto formado pelos três atributos mais frequentes nos demais redutos;
6. $\{2, 5, 7, 8\}$: primeiro reduto calculado que contém quatro atributos.

Logo após, esses subconjuntos foram utilizados para representar o conjunto de dados na forma reduzida correspondente, ou seja, seis conjuntos de dados reduzidos.

7.5.2.2 Avaliação da Precisão

Uma vez selecionados os subconjuntos de atributos relevantes, eles foram submetidos à avaliação quanto a precisão na representação do conjunto de dados em estudo. Para isso, os indutores $\mathcal{C}4.5$ -rules, $\mathcal{CN}2$ e Rosetta foram aplicados aos conjuntos de dados reduzidos.

Na Tabela 7.6 são mostrados os erros obtidos na classificação utilizando *10-fold-cross-validation*. Vale lembrar que o erro e a precisão são medidas complementares. Nessa tabela a primeira coluna representa os subconjuntos de atributos utilizados para representar o conjunto de dados, enquanto que a primeira linha indica os indutores utilizados para a avaliação. Nas linhas da tabela estão os erros na classificação seguidos do desvio padrão do erro. Por exemplo, o valor 32.47 ± 3.06 indica que houve um erro de 32.47% com desvio padrão de 3.06% quando o conjunto de dados com todos os atributos (conjunto de dados original) foi submetido ao indutor $\mathcal{C}4.5$ -rules.

Talassemia 10-cv	$\mathcal{C}4.5$ -rules	$\mathcal{CN}2$	Rosetta
Todos	32.47 ± 3.06	36.33 ± 3.25	34.25 ± 8.30
{2, 9}	28.06 ± 2.69	33.38 ± 2.35	32.17 ± 6.25
{4, 9}	34.53 ± 1.68	41.94 ± 2.39	35.12 ± 9.60
{1, 2, 9}	28.94 ± 2.95	37.77 ± 3.17	37.71 ± 10.50
{1, 5, 9}	32.76 ± 2.68	43.37 ± 2.33	41.01 ± 9.26
{1, 6, 8}	41.89 ± 3.53	50.14 ± 2.60	42.47 ± 9.27
{2, 5, 7, 8}	36.90 ± 1.83	46.59 ± 2.25	43.36 ± 7.45

Tabela 7.6: Erros na Classificação do Conjunto de Dados

Na Tabela 7.6 é possível observar que para todos os indutores e conjuntos de dados, os menores erros foram obtidos quando o SSA {2, 9} foi utilizado. Embora os erros (28.06 ± 2.69 para $\mathcal{C}4.5$ -rules, 33.38 ± 2.35 para $\mathcal{CN}2$ e 32.17 ± 6.25 para Rosetta) pareçam altos, vale lembrar que o erro da classe majoritária é de 48.1%. Em outras palavras, se o erro do classificador for menor que o erro da classe majoritária, significa que houve algum aprendizado.

Na tentativa de diminuir o erro obtido, foi utilizada a informação fornecida pelo especialista que o atributo 9 (RDW) é o que possui uma relação mais direta com o atributo de decisão (PCR), informação confirmada pelo índice de correlação entre ambos, 0.285 na Tabela 7.5.

A Figura 7.3 ilustra a distribuição dos exemplos segundo os valores do atributo RDW. Quartiles são valores que particionam um conjunto de dados em quatro grupos, cada um deles contendo 25% das medidas. O 1º. quartil indica que para aquele valor de RDW existem 25% dos exemplos que estão abaixo dele. O 3º. quartil indica que para

aquele valor de RDW existem 75% dos exemplos que estão abaixo dele. O intervalo de confiança de 95% — $IC_{95\%}(A)$ — para um atributo A qualquer é obtido da seguinte forma:

$$IC_{95\%}(A) = [média(A) - 2 \times desvio_padrão(A), média(A) + 2 \times desvio_padrão(A)]$$

Esse intervalo informa que, no caso do atributo RDW, 95% dos exemplos possuem valores de RDW entre 10.8198 e 22.7442. Os valores que ficam fora dessa faixa são chamados de *aberrantes*⁵, os quais estão representados por * na Figura 7.3. Normalmente, recomenda-se que os valores aberrantes sejam excluídos de qualquer tipo de modelagem, porque esses valores podem produzir desvios nos modelos.

Figura 7.3: *Boxplot* para o Atributo RDW

Para avaliar a influência dos valores aberrantes do atributo RDW no erro da classificação, os exemplos que contém esses valores — 18 exemplos — foram excluídos do conjunto de dados em três etapas, de forma a eliminá-los gradativamente e, a cada

⁵A determinação dos valores aberrantes foi feita utilizando o software *MINITAB 13.3 for Windows* — <http://www.minitab.com>

etapa, o erro da classificação utilizando *10-fold-cross-validation* foi calculado. A influência dos valores aberrantes foi medida sobre o conjunto de dados contendo todos os atributos.

Na Tabela 7.7 são mostrados os erros da classificação utilizando todo o conjunto de dados e o conjunto de dados excluindo-se os 9, 14 e 18 exemplos aberrantes com maior valor de RDW, respectivamente. Pode ser observado que a exclusão dos exemplos aberrantes implicou num aumento gradual no erro da classificação para todos os indutores. Isso pode ser explicado pelo fato de que, apesar dos exemplos conterem valores aberrantes para RDW, eles são importantes ao induzir o classificador. A exclusão desses exemplos implica na geração de um classificador que não consegue classificar exemplos com características próximas àqueles excluídos, incorrendo assim no acréscimo do erro.

Conjunto de Dados	C4.5-rules	CN2	Rosetta
Total	32.47 $\pm$ 3.06	36.33 $\pm$ 3.25	34.25 $\pm$ 8.30
Total–9 aberrantes	32.82 $\pm$ 1.84	39.83 $\pm$ 3.01	35.27 $\pm$ 8.15
Total–14 aberrantes	33.24 $\pm$ 2.46	40.31 $\pm$ 3.01	36.84 $\pm$ 7.90
Total–18 aberrantes	35.86 $\pm$ 3.08	41.14 $\pm$ 2.05	38.10 $\pm$ 7.84

Tabela 7.7: Erros na Classificação Excluindo os Valores Aberrantes em Três Etapas

Uma vez avaliada a importância de manter todos os exemplos do conjunto de dados, mesmo que alguns deles contenham valores de atributos que sejam aberrantes do ponto de vista estatístico, resolveu-se medir a quantidade de regras geradas pelos indutores utilizando os subconjuntos de atributos selecionados.

7.5.2.3 Número de Regras Induzidas

Na Tabela 7.8 são mostradas as quantidades de regras induzidas utilizando-se todos os exemplos, considerando os diferentes subconjuntos de atributos. É importante considerar essa tabela conjuntamente com a Tabela 7.6 que mostra o erro na classificação. Por exemplo, considerando o subconjunto de atributos {2, 9}, o C4.5-rules gerou 4 regras com um erro e desvio padrão esperado de 28.06 ± 2.69 na classificação de exemplos nunca vistos. Analogamente, CN2 gerou 44 regras com erro 33.38 ± 2.35 e Roseta 10 regras com erro 32.17 ± 6.25 .

Atributos	C4.5-rules	CN2	Rosetta
Todos	7	31	318
{2, 9}	4	44	10
continua na próxima página			

continuação da página anterior			
Atributos	$\mathcal{C}4.5$ -rules	$\mathcal{CN}2$	Rosetta
{4, 9}	2	109	15
{1, 2, 9}	4	74	35
{1, 5, 9}	4	51	88
{1, 6, 8}	2	64	55
{2, 5, 7, 8}	2	71	45

Tabela 7.8: Número de Regras Induzidas

Pode ser observado que o indutor $\mathcal{C}4.5$ -rules foi o que induziu o menor número de regras em todos os casos. Além disso, não considerando o experimento com todos os atributos e o subconjunto de atributos {1, 5, 9} o número de regras induzidas por $\mathcal{CN}2$ foi maior que as induzidas por Rosetta. Considerando somente $\mathcal{C}4.5$ -rules e $\mathcal{CN}2$ é importante notar que os resultados expostos em (Pila & Monard, 2001b) se confirmaram, *i.e.* o $\mathcal{C}4.5$ -rules tem uma tendência a gerar menor quantidade de regras quando o conjunto de dados possui um subconjunto dos atributos, enquanto que o $\mathcal{CN}2$ tende a gerar mais regras. O Rosetta parece preferir um subconjunto de atributos, gerando nesses casos um número menor de regras. Na próxima seção é apresentado o conhecimento extraído no formato de regras, bem como algumas considerações sobre esse conhecimento.

7.6 Avaliação do Conhecimento

Analisando a Tabela 7.6 pode-se notar que o subconjunto de atributos que propiciou o menor erro na classificação foi o {2, 9} quando utilizado com qualquer um dos três indutores. No entanto, como o maior interesse desse estudo de caso está no conhecimento simbólico induzido, é desejável que o número de regras seja o menor possível, da mesma forma que o erro na classificação. Assim, seria possível ter um conhecimento sintético e com um bom poder preditivo (erro baixo na classificação).

O conhecimento induzido por $\mathcal{C}4.5$ -rules utilizando os atributos {2, 9} é mais simbólico (somente 4 regras) e com o melhor poder preditivo (28.06 ± 2.69 de erro). As regras induzidas são:

```

Rule 1:
    RDW <= 15.2
    -> class ANORMAL [73.2%] (cover 95)

Rule 5:
    Cor = NEG
    RDW <= 19.4

```

```

-> class ANORMAL [70.1%] (cover 58)

Rule 6:
  Cor = NEG
  RDW > 19.4
-> class NORMAL [79.4%] (cover 6)

Rule 2:
  Cor = BCA
  RDW > 15.2
-> class NORMAL [69.6%] (cover 180)

Default class: NORMAL

```

Nas regras induzidas pelo $\mathcal{C}4.5$ -rules são mostradas duas informações adicionais. O percentual indica a taxa de acerto daquela regra sobre os exemplos que a regra cobre e o número de exemplos que a regra cobre. Para esse conjunto de regras induzidas pelo $\mathcal{C}4.5$ -rules, a matriz de confusão, sobre todos os exemplos, está representada na Tabela 7.9. A matriz de confusão informa a quantidade de exemplos classificados corretamente como pertencentes a uma classe e o número de exemplos classificados incorretamente para aquela mesma classe. Por exemplo, na primeira linha da Tabela 7.9, tem-se 113 exemplos da classe ANORMAL classificados corretamente como sendo da classe ANORMAL e 50 exemplos dessa classe classificados incorretamente como sendo da classe NORMAL.

(a)	(b)	← classificado como
113	50	(a): classe ANORMAL
40	136	(b): classe NORMAL

Tabela 7.9: Matriz de Confusão para as Regras Induzidas pelo $\mathcal{C}4.5$ -rules

As regras induzidas pelo Rosetta utilizando os atributos {2, 9} foram:

Regras	Cobertura	Precisão
Cor(BCA) AND RDW([*, 15.0]) $\Rightarrow$ PCR(ANORMAL) OR PCR(NORMAL)	49 [37, 12]	[75.51%, 24.49%]
Cor(NEG) AND RDW([15.9, 17.0]) $\Rightarrow$ PCR(ANORMAL) OR PCR(NORMAL)	24 [14, 10]	[58.33%, 41.67%]
Cor(BCA) AND RDW([18.1, *]) $\Rightarrow$ PCR(NORMAL) OR PCR(ANORMAL)	57 [44, 13]	[77.19%, 22.80%]
Cor(BCA) AND RDW([15.9, 17.0]) $\Rightarrow$ PCR(NORMAL) OR PCR(ANORMAL)	65 [50, 15]	[76.92%, 23.07%]
Cor(NEG) AND RDW([15.0, 15.9]) $\Rightarrow$ PCR(ANORMAL) OR PCR(NORMAL)	25 [19, 6]	[76.00%, 24.00%]
Cor(BCA) AND RDW([17.0, 18.1]) $\Rightarrow$ PCR(NORMAL) OR PCR(ANORMAL)	30 [20, 10]	[66.67%, 33.33%]
Cor(NEG) AND RDW([*, 15.0]) $\Rightarrow$ PCR(ANORMAL) OR PCR(NORMAL)	27 [23, 4]	[85.19%, 14.81%]
Cor(NEG) AND RDW([17.0, 18.1]) $\Rightarrow$ PCR(ANORMAL) OR PCR(NORMAL)	12 [9, 3]	[75.00%, 25.00%]
Cor(BCA) AND RDW([15.0, 15.9]) $\Rightarrow$ PCR(NORMAL) OR PCR(ANORMAL)	40 [20, 20]	[50.00%, 50.00%]
Cor(NEG) AND RDW([18.1, *]) $\Rightarrow$ PCR(NORMAL) OR PCR(ANORMAL)	10 [7, 3]	[70.00%, 30.00%]

Pode ser observado que todas as regras levam a duas decisões, ou seja, o subconjunto de atributos {2, 9} faz com que o conjunto de dados sobre talassemia α seja considerado como pertencente a região de borda, e por isso a indução de regras não determinísticas. Para cada uma das regras é informada a cobertura da regra e a cobertura

individual de cada decisão. É informada ainda a precisão de cada decisão associada à regra. Por exemplo, a primeira regra cobre 49 exemplos, sendo que 37 estão relacionados à classe ANORMAL e 12 à classe NORMAL. Desses exemplos, para a decisão ANORMAL há uma precisão de 75.51%, enquanto que para a decisão NORMAL a precisão é de 24.49%.

A matriz de confusão gerada pelo Rosetta — Tabela 7.10 — da mesma forma que a gerada pelo $\mathcal{C}4.5$ -rules, mostra o número de exemplos de uma determinada classe que são preditos correta e incorretamente.

Classe Atual	Predita	
	ANORMAL	NORMAL
ANORMAL	122	41
NORMAL	55	121

Tabela 7.10: Matriz de Confusão para as Regras Induzidas pelo Rosetta

Embora as regras induzidas não apresentaram conhecimento inesperado para a especialista, elas foram úteis para confirmar algumas suspeitas que a especialista possuía em relação à determinação da talassemia α em função dos demais atributos. A primeira suspeita confirmada foi que a *Cor da pele* (atributo 2) possui grande poder de decisão para separar os pacientes que possuem ou não a doença, ou seja, que a Cor é um atributo relevante para a decisão. A suspeita era que a talassemia α é mais freqüente nos indivíduos negros. Analisando ambos conjuntos de regras pode-se observar que a maioria das regras que possuem *Cor=NEG* no antecedente cobrem mais exemplos pertencentes à classe ANORMAL.

Outro conhecimento confirmado é em relação ao atributo RDW (atributo 9), o qual possui grande influência na determinação da talassemia α . Nas regras induzidas pelo indutor $\mathcal{C}4.5$ -rules o ponto de “corte” da discretização ficou muito próximo ao ponto de “corte” encontrado em (Borges, 2000) que relata estudos da área médica para a determinação da talassemia α utilizando, entre outros, métodos estatísticos.

7.7 Considerações Finais

Neste capítulo foi apresentado um estudo de caso com um conjunto de dados reais utilizando a abordagem de RS, *i.e.* os redutos como forma de seleção de atributos. Os dados presentes nesse conjunto são relativos a informações colhidas nos ambulatórios do complexo hospitalar da UNICAMP. Essas informações estão relacionados ao diagnóstico da presença ou ausência da talassemia α em pacientes que aparente-

mente possuem apenas anemia. O diagnóstico da talassemia α com 100% de previsão só é possível através do exame de DNA, o qual é financeiramente custoso. O objetivo deste estudo consistiu em verificar a possibilidade de determinar a presença de talassemia α em função de dados extraídos de hemogramas, o qual é um exame financeiramente econômico. Após a seleção de atributos foi constatado que os atributos *Cor* e *RDW* são os mais relevantes na determinação da doença, tal como o especialista do domínio já havia suposto. Utilizando somente estes dois atributos para o passo de extração de conhecimento, foi obtido um conjunto de regras com alguns pontos “corte” que confirmaram o conhecimento prévio da especialista no domínio.

Entretanto, ainda que os dois melhores classificadores induzidos possuam um erro de predição bem mais baixo que o erro majoritário, esse erro não é suficientemente pequeno para considerar que o conjunto de dados fornecido de hemogramas de pacientes é suficiente para diagnosticar a doença com uma boa margem de segurança.

Também, considerando o poder de predição de cada uma das regras que fazem parte desses dois classificadores (quatro regras utilizando *C4.5*-rules e dez regras utilizando Rosetta) não foi possível isolar nenhuma regra que permitisse diagnosticar a doença para algum subconjunto de exemplos (pacientes) com boa margem de segurança do ponto de vista médico.

Pelo interesse do problema, a especialista pretende fornecer mais exemplos no futuro a fim de repetirmos os experimentos e verificar se o erro de predição diminui consideravelmente e/ou alguma regra com maior poder de predição é encontrada.

No próximo capítulo são apresentadas as conclusões deste trabalho, bem como trabalhos futuros relacionados.

Capítulo 8

Conclusões

PREVISÕES apontam que a quantidade de dados que serão armazenados nos computadores nos próximos cinco anos será maior que a quantidade armazenada nos últimos trinta anos. Essas previsões baseiam-se no crescimento exponencial do uso da maior rede de computadores do mundo, a Internet. Com isso, a maior parte dos dados das empresas estarão armazenados nos computadores e uma manipulação adequada desses dados faz-se necessária.

A principal preocupação está em como gerenciar essa crescente quantidade de dados. Essa preocupação fundamenta-se na premissa que os dados de uma empresa são um de seus maiores patrimônios. Na verdade, os dados armazenados durante anos de trabalho possuem implicitamente boa parte da memória corporativa da empresa. Por exemplo, se um analista financeiro de um banco trabalhou durante anos na concessão de empréstimos, os dados registrados por ele devem conter implicitamente as direções que o levaram a tomada da decisão (conceder ou não o empréstimo). Portanto, uma transformação desses dados em conhecimento pode proporcionar um auxílio inteligente à tomada de decisões dentro da empresa.

Uma área da Inteligência Artificial que estuda métodos para a indução automática de conhecimento é o Aprendizado de Máquina, o qual pode ser Supervisionado ou Não-Supervisionado. Em AM Supervisionado o conhecimento a respeito do domínio é normalmente representado na forma de experiências passadas resolvidas com sucesso pelo especialista do domínio, as quais contém, portanto, conhecimento implícito utilizado pelo especialista.

Essas experiências são normalmente chamadas de exemplos ou simplesmente dados. Frequentemente essas experiências estão representadas em um vetor atributo-

valor, tais como os registros dos bancos de dados. Quando a indução de conhecimento está relacionada a exemplos rotulados, diz-se ser um aprendizado supervisionado. Os rótulos, chamados de classe, podem ter valores contínuos ou categóricos, sendo que o aprendizado que envolve esses tipos de classe são chamados de regressão e classificação, respectivamente.

Portanto, a tarefa do sistema de AM é extrair uma representação generalizada — classificador — capaz de representar o conhecimento implícito nos dados de forma a mapear os valores dos atributos nos rótulos. Logicamente, essa representação interna ao classificador é feita utilizando os atributos e as classes. No entanto, para que o classificador seja capaz de prever a classe dos exemplos é necessário que os atributos caracterizem os exemplos de forma coerente.

Em alguns conjuntos de dados a caracterização dos exemplos é feita com muitos atributos e pode ser necessário que o sistema de AM focalize-se apenas nos atributos mais relevantes. Assim, um dos principais problemas de AM é a seleção de atributos relevantes.

Embora a seleção de atributos venha sendo estudada faz algum tempo, com o crescimento do montante de dados a serem analisados, é uma área que sempre requer novas metodologias. Algumas justificativas para se fazer seleção de atributos incluem, entre outras: a maioria dos sistemas de AM, computacionalmente viáveis, não trabalham bem na presença de muitos atributos, ou seja, a precisão dos classificadores pode ser melhorada com a seleção de atributos; a representação interna dos classificadores pode ser feita utilizando menos atributos, melhorando assim a compreensão pelos seres humanos frente a essa representação; e, em alguns domínios o custo da coleta das informações pode ser reduzido, pois serão coletadas somente informações relacionadas aos atributos relevantes.

A principal abordagem para seleção de atributos é chamada de Seleção de um Subconjunto de Atributos, cujo objetivo é encontrar um conjunto mínimo de atributos que seja capaz de manter o mesmo poder de representação de todos os atributos. A SSA está relacionada à relevância de cada um dos atributos. Porém, existem diversas formas de relevância para selecionar atributos, como, por exemplo, relevância relativa à descrição do conhecimento, relevância relativa à precisão do classificador e relevância relativa ao poder de distinção entre os exemplos.

Neste trabalho é tratada a SSA utilizando algoritmos de AM simbólicos, especificamente algoritmos que expressam o conhecimento induzido na forma de regras, bem como o uso de Rough Sets para SSA. Nessa teoria matemática a representação do

conhecimento é feita utilizando os redutos, que são subconjunto mínimos de atributos capazes de manter a relação de distinção entre os exemplos e por isso foram por nós investigados como forma de realizar SSA.

Para avaliar a aplicabilidade dos redutos como forma de seleção de atributos relevantes, foram feitos vários experimentos e comparações envolvendo Rough Sets e alguns algoritmos de indução freqüentemente utilizados pela comunidade de AM. Nesses experimentos foram avaliados o número de atributos selecionados, o erro do classificador, bem como o número de regras induzidas utilizando os atributos selecionados. Os experimentos foram conduzidos sobre nove conjuntos de dados naturais obtidos do Repositório da UCI. Neste trabalho também foram analisados exemplos provenientes de um conjunto de dados do mundo real da área médica, cujo conteúdo está relacionado à determinação da talassemia α .

Nos experimentos realizados sobre os conjuntos de dados naturais pode ser observado que a utilização dos redutos como forma de SSA não trouxe resultados significativos com 95% de confiança, embora a abordagem de RS tenha sido a que, na média, selecionou menor quantidade de atributos dentre as demais analisadas. Quanto ao número de regras induzidas utilizando os atributos selecionados pelos redutos, a análise dos resultados mostra que há uma forte dependência entre ambos. Porém, essa relação de dependência pode variar dependendo do indutor e do conjunto de dados utilizado. Isso serve para reforçar a grande importância das avaliações experimentais dos sistemas de AM.

Embora a abordagem de RS tenha apresentado alguns bons resultados quanto a utilização de redutos para a seleção de atributos, existem alguns problemas a serem considerados nessa abordagem. Um dos problemas de Rough Sets está relacionado à necessidade de discretizar os dados. Sabe-se que no processo de discretização sempre há perda de informação, pois os valores dos atributos passam a estar relacionados a intervalos determinados pelos pontos de “corte” (Félix et al., 2000). Assim, valores originalmente distintos, passam a ser tratados como iguais após o processo de discretização. Logicamente, nesse caso, o conhecimento implícito é perdido. Outro problema está relacionado à indução de regras. Na abordagem RS a indução de regras é feita sobrepondo o reduto no conjunto de dados. Dessa forma, as regras são meras traduções do formato atributo-valor do reduto para o formato de conjunção de disjunções. Nesse tipo de indução de regras a generalização, ponto primordial do aprendizado, é negligenciada. Na verdade, o classificador possui bom desempenho somente se o conjunto de dados contiver alguns exemplos capazes de representar todos os demais, caso contrário as regras geradas não são capazes de representar o

conhecimento de forma generalizada, como é o caso das regras induzidas pelos indutores $\mathcal{CN}2$ e $\mathcal{C4.5}$ -rules.

Diferentemente da análise de conjuntos de dados naturais, a análise de conjuntos de dados do mundo real envolve alguns problemas adicionais, tais como o acesso inicial aos dados, a caracterização dos objetivos da extração do conhecimento pelo especialista, a limpeza e pré-processamento dos dados e a avaliação, pelo especialista do domínio, do conhecimento extraído. O conjunto de dados reais analisado neste trabalho possui uma característica que o distingue de outros conjuntos de dados do mundo real. As classes (rótulos) dos exemplos foram obtidas empregando uma técnica da área médica que não utiliza os valores dos atributos presentes nesse conjunto de dados, pois o objetivo da especialista é verificar a viabilidade de determinar a classe de novos exemplos em função de atributos cuja determinação é mais econômica. Assim, qualquer relação encontrada que pudesse prever a classe utilizando esses atributos seria vista como novidade. Os redutos foram por nós utilizados como forma de selecionar os atributos relevantes e os resultados vieram a confirmar o conhecimento prévio da especialista em relação ao poder de predição de dois atributos. Quanto à precisão, os resultados não foram suficientes para se obter um classificador confiável, pois o erro ainda é relativamente alto para a área médica. Ainda, utilizando esses atributos selecionados pelo reduto, pode-se avaliar o conhecimento extraído na forma de regras. Na avaliação do conhecimento extraído, mais uma vez o conhecimento prévio da especialista foi confirmado. Isso mostra que a efetiva aplicação de sistemas de AM depende muito da interação do engenheiro do conhecimento com o especialista do domínio. Em outras palavras, a nossa conclusão é que a aplicação efetiva de métodos de Aprendizado de Máquina é viável desde que a interação entre diferentes áreas do conhecimento e o experimentalismo sejam levados em consideração.

Como trabalhos futuros pretende-se continuar a análise desse conjunto de dados do mundo real, após a especialista coletar novos exemplos, com o objetivo de construir um classificador com maior precisão, o que incrementaria a confiabilidade das decisões tomadas na determinação da talassemia α frente a novos exemplos.

Finalmente, pretende-se dar continuidade do estudo de outras metodologias para seleção de atributos relevantes, bem como de outros conjuntos de dados do mundo real.

Referências

- Aha, D. W. (1997). Lazy learning. *Artificial Intelligence Review*, 11:7–10.
- Baranauskas, J. A. & Monard, M. C. (1998a). Metodologias para a seleção de atributos relevantes. XIII Simpósio Brasileiro de Inteligência Artificial.
- Baranauskas, J. A. & Monard, M. C. (1998b). Metodologias para seleção de atributos. *Workshop de Teses e Dissertações do Simpósio Brasileiro de Inteligência Artificial (SBIA)*. <http://www.fmrp.usp.br/~augusto/ps/SBIA98.web.ps.zip>.
- Baranauskas, J. A. & Monard, M. C. (2000). An unified overview of six supervised symbolic machine learning inducers. Technical Report 103, ICMC-USP. ftp://ftp.icmc.sc.usp.br/pub/BIBLIOTECA/rel_tec/Rt_103.ps.zip.
- Baranauskas, J. A., Monard, M. C., & Horst, P. S. (1999). Evaluation of feature selection by wrapping around the CN2 inducer. *Encontro Nacional de Inteligência Artificial (ENIA/SBC)*, pages 315–326. <http://www.fmrp.usp.br/~augusto/ps/ENIA99.web.ps.zip>.
- Batista, G. E. A. P. A. (2000). Pré-processamento de dados em aprendizado de máquina supervisionado. Minidissertação para Qualificação de Doutorado, ICMC-USP.
- Batista, G. E. A. P. A., Carvalho, A. C. P. L., & Monard, M. C. (1999). Aplicando seleção unilateral em conjuntos de exemplos desbalanceados: Resultados iniciais. In *Anais II Encontro Nacional de Inteligência Artificial - ENIA 99*, pages 327–340.
- Batista, G. E. A. P. A., Carvalho, A. C. P. L., & Monard, M. C. (2000). Applying one-sided selection to unbalanced datasets. In *Proceedings of the Mexican Congress on Artificial Intelligence (MICAI-2000), Lecture Notes in Artificial Intelligence*, pages 315–325.
- Beaubouef, T. & Lang, R. (1998). Rough set techniques for uncertainty management in automated story generation. *Communications of the ACM*, 4:326–331.

- Beaubouef, T., Petry, F. E., & Arora, G. (1998). Information-theoretic measures of uncertainty for rough sets and rough relational databases. *Journal of Information Sciences*, pages 185–195.
- Blake, C., Keogh, E., & Merz, C. (1998). Uci irvine repository of machine learning databases. <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
- Bloedorn, E. & Michalski, R. S. (1998). Data-Driven Construtive Induction. *IEEE Intelligent Systems*, 13(2):30–37. March/April 1998.
- Blum, A. L. & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, pages 245–271.
- Bonikowski, Z. (1998). Extensions and intentions in the rough set theory. *Journal of Information Sciences*, pages 149–167.
- Borges, E. (2000). Contribuição da talassemia α como causa de microcitose e hipocromia em uma população brasileira. Dissertação de Mestrado, UNICAMP.
- Caruana, R. A. & Freitag, D. (1994). How useful is relevance ? *Working Notes of the AAAI Fall Symposium on Relevance*, pages 25–29.
- Clark, P. & Boswell, R. (1991). Rule induction with CN2: Some recent improvements. In Kodratoff, Y., editor, *Proceedings of the 5th European Conference EWSL 91*, pages 151–163. Springer-Verlag.
- Clark, P. & Niblett, T. (1987). Induction in noise domains. In Bratko, I. & Lavrač, N., editors, *Proceedings of the 2nd European Working Session on Learning*, pages 11–30, Wilmslow, UK. Sigma.
- Clark, P. & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, 3(4):261–283.
- Øhrn, A. (1999a). *Discernibility and Rough Sets in Medicine: Tools and Applications*. PhD thesis, Norwegian University on Science and Technology.
- Øhrn, A. (1999b). Rosetta: Technical reference manual. Technical report, Knowledge System Group, Norwegian University on Science and Technology, NO.
- Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34.
- Félix, L. C. M., Rezende, S. O., Monard, M. C., & Caulkins, C. W. (2000). Transforming a regression problem into a classification problem using hybrid discretization. *Computación y Sistemas. Special issue in Artificial Intelligence*, pages 44–52.

- Guan, J. W. & Bell, D. A. (1998). Rough computational methods for information systems. *Artificial Intelligence*, pages 77–103.
- Hu, X. (1995). *Knowledge Discovery in Databases: An Attribute-Oriented Rough Set Approach*. PhD thesis, University of Regina.
- Hu, X. & Cercone, N. (1994). Discovery of decision rules in relational databases: A rough set approach. *CIKM'94*, page 9.
- John, G., Kohavi, R., & Pflieger, K. (1994). Irrelevant features and the subset selection problem. In Kaufmann, M., editor, *Proceedings of the Eleventh International Conference on Machine Learning*, pages 167–173, San Francisco, CA.
- Kasabov, N. K. (1996). *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*. The MIT Press.
- KDD 95 (1995). *Proceedings of the First International Conference on Knowledge Discovery and Data Mining KDD-95*, Menlo Park, CA. American Association for Artificial Intelligence.
- KDD 96 (1996). *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining KDD-96*, Menlo Park, CA. American Association for Artificial Intelligence.
- Kira, K. & Rendell, L. (1992). A practical approach to feature selection. In Kaufmann, M., editor, *Proceedings of the Ninth International Conference on Machine Learning*, pages 249–256, Aberdeen, Scotland.
- Koczkodaj, W. W., Orlowski, M., & Marek, V. W. (1998). Myths about rough set theory. *Communications of the ACM*, pages 102–103.
- Kohavi, R., Sommerfield, D., & Dougherty, J. (1996). Data mining using $\mathcal{MLC}^{++}$: A machine learning library in C++. *Tools with IA*, pages 234–245.
- Komorowski, J. & Øhrn, A. (1999). Modelling prognostic power of cardiac tests using rough sets. *Artificial Intelligence in Medicine*, pages 167–191.
- Komorowski, J., Pawlak, Z., Polkowski, L., & Skowron, A. (1999). Rough sets: A tutorial. Technical report, Warsaw University.
- Kryszkiewicz, M. (1998). Rough set approach to incomplete information systems. *Information Sciences*, pages 39–49.
- Lee, H. D. (2000). Seleção e construção de features relevantes para o aprendizado de máquina. Dissertação de Mestrado, ICMC-USP.

- Lee, H. D., Monard, M. C., & Baranauskas, J. A. (1999). Empirical comparison of wrapper and filter approaches for feature subset selection. Technical Report 94, ICMC-USP. ftp://ftp.icmc.sc.usp.br/pub/BIBLIOTECA/rel_tec/Rt_94.ps.zip.
- Lin, T. Y. & Cercone, N. (1997). *Rough Sets and Data Mining: Analysis of Imprecise Data*. Kluwer Academic Publishers.
- Lingras, P. (1998). Comparasion of neofuzzy and rough neural networks. *Information Sciences*, pages 207–215.
- Mitchell, T. M. (1997). *Machine Learning*. WCB/McGraw-Hill.
- Newell, A. & Simon, H. A. (1972). *Human Problem Solving*, chapter 1. Prentice hall.
- Pawlak, Z. (1982). Rough sets. *International Jornal of Computer and Information Sciences*, pages 341–356.
- Pawlak, Z. (1996). Rough sets, rough relations and rough functions. *Fundamenta Informaticae*, 27, pages 103–108.
- Pawlak, Z. (1998). An inquiry into anatomy of conflicts. *Journal of Information Sciences*, pages 65–78.
- Pawlak, Z., Grzymala-Busse, J., Slowinski, R., & Ziarko, W. (1995). Rough sets. *Communications of the ACM*, pages 89–95.
- Pila, A. D. & Monard, M. C. (2001a). Rough sets reducts as a filter approach for feature subset selection: An empirical comparison with wrapper and other filters. Technical Report 134, ICMC-USP. ftp://ftp.icmc.sc.usp.br/pub/BIBLIOTECA/rel_tec/Rt_134.ps.zip.
- Pila, A. D. & Monard, M. C. (2001b). Rules induction using rough sets reducts as feature subset selection: An empirical comparison with other filter approaches. Technical Report 139, ICMC-USP. ftp://ftp.icmc.sc.usp.br/pub/BIBLIOTECA/rel_tec/Rt_139.ps.zip.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.
- Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann. San Francisco, CA.
- Rathjens, D. (1996). MinesetTM user's guide. Silicon Graphics, Inc.
- Rezende, S. O. & Pugliesi, J. B. (1998). Aquisição de conhecimento explícito ou manual. Technical Report 37, ICMC-USP. http://labic.icmc.sc.usp.br/didatico/PostScript/rt_ac.ps.zip.

- Russel, S. J. & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall.
- Slowinski, R. (1995). Rough set approach to decision analysis. *AI Expert*, March:19–25.
- Smolenski, P. (1990). *Tensor product variable binding and the representation of symbolic structures in connectionist systems*, chapter 1. Prentice hall.
- Solheim, H. G. & Øyvind Tuset Aasheim (1996). Rough sets as a framework for data mining. Technical report, The Norwegian University of Science and Technology, NO.
- Spillman, R. (1990). Managing with belief uncertainty functions. *AI Expert*, May:44–49.
- Stein, R. (1993). The dempster-shafer theory of evidential reasoning. *AI Expert*, August:26–31.
- Szladow, A. & Ziarko, W. (1993). Rough sets: Working with imperfect data. *AI Expert*, July:36–41.
- Tsomoto, S. (1998). Automated extraction of medical expert systems rules from clinical databases based on rough set theory. *Information Sciences*, pages 67–84.
- Weiss, S. M. & Kulikowski, C. A. (1990). *Computer Systems that Learn*. Morgan Kaufmann Publishers, Inc.
- Yao, Y. Y. (1998). A comparative study of fuzzy sets and rough sets. *Journal of Information Sciences*, pages 227–242.

Apêndice A

Teoria da Evidência de Dempster-Shafer

A.1 Considerações Iniciais

A vida real exige muito mais do que a solução de simples questões tais como “Quem fez isso?”. Frequentemente é necessário tomar decisões e formular conclusões utilizando algumas evidências, as quais podem ser incompletas e conflitantes. Essas situações frequentemente envolvem a escolha de uma entre várias interpretações possíveis. A Teoria de Dempster-Shafer — D-S — pode ser vista como uma generalização da Teoria das Probabilidades, e portanto fornece uma abordagem mais efetiva para lidar com informações incertas (Spillman, 1990). Em alguns casos, a teoria de D-S pode ser especialmente atrativa por ser uma abordagem altamente intuitiva para o tratamento de incerteza.

No caso de existir várias evidências independentes (dados observados) e ser possível realizar algumas inferências gerais relacionadas a cada uma dessas evidências, então a teoria de D-S permite combinar essas evidências de uma forma probabilística consistente a fim de se estabelecer uma estimativa mais completa sugerida pelo conjunto de evidências.

Utilizando a teoria de D-S, vários conjuntos alternativos de hipóteses podem ser derivados de uma simples coleção de evidências. Cada um desses conjuntos têm uma faixa de crença associada, chamada de intervalo de crença.

Neste apêndice, baseado principalmente no artigo de (Stein, 1993), serão apresen-

tados os conceitos básicos da Teoria de Dempster-Shafer com o objetivo de realizar possíveis comparações com a Teoria de Rough Sets.

A.2 Raciocínio Evidencial

Quando existe a tentativa de determinar a validade de uma dada hipótese é interessante incorporar a maior quantidade de informações, ou evidências, possíveis. A teoria de D-S permite fazer justamente isso.

Para aplicar a teoria de D-S, primeiramente é necessário definir o domínio do problema, *i.e.* o conjunto de valores θ , o qual é completo e portanto contém todas as possíveis hipóteses. Por exemplo, no caso da predição da bolsa de valores, θ pode ser {NMG, -5%, -1%, 0%, +1%, +5%, PMG}, onde cada elemento é uma hipótese indicando as mudanças dos valores na bolsa nas próximas 24 horas, e os termos NMG e PMG indicam, respectivamente, mudanças negativas muito grandes e mudanças positivas muito grandes desses valores.

O próximo passo é coletar evidências e construir conjuntos refinados de hipóteses, baseados nas regras e/ou dados históricos da bolsa de valores. Esses conjuntos indicam que cada parte da evidência implica relativamente em θ . Por exemplo, um analista pode encontrar com 60% de crença que se as tendências do mercado se mantiverem, a ocorrência de uma determinada notícia poderá aumentar o volume de negócios da bolsa de valores. Portanto, se essa notícia ocorrer, ela suportaria o conjunto de hipóteses $H_1 = \{+1\%, +5\%, \text{PMG}\}$ que é o conjunto de todos os elementos de θ que representa um crescimento na movimentação da bolsa.

Identificada essa primeira possível hipótese H_1 , deve-se então atribuir uma probabilidade básica (*basic probability assignment*) — bpa — utilizando o conhecimento do especialista. Uma bpa é um valor baseado na função de probabilidade de massa, a qual representa o total de massa de crença na evidência que aponta exatamente para um conjunto de hipóteses particular. Sendo uma probabilidade, o valor da bpa está no intervalo $[0,1]$. A bpa não inclui a crença na evidência de subconjuntos da hipóteses. Por exemplo, uma evidência apontando para uma mudança de exatamente 5% (e não para 1% ou PMG) não deveria ser incluída na bpa para H_1 . Pode-se então atribuir os valores da bpa para os conjuntos H_1 e θ . Uma vez que o analista (especialista) da bolsa de valores indicou que existia um suporte de 60% para um conjunto particular de hipóteses, calcula-se as bpa como segue:

$$\begin{aligned} \{1\%, 5\%, \text{PMG}\} &= H_1 & \text{bpa} &= 0.60 \\ \theta & & \text{bpa} &= 1 - 0.60 = 0.40 \end{aligned}$$

Note que θ contém o conjunto H_1 bem como seu complemento, H_1^c — composto de quatro elementos $\{\text{NMG}, -5\%, -1\%, 0\%\}$. Utilizando a teoria de D-S, seria errado atribuir o valor 0.40 somente para H_1^c , pois não existe nenhuma evidência dizendo que os 40% restantes contradizem H_1 . Sabe-se apenas que a evidência parcial atual suporta um coeficiente de 0.60. O que se pretende dizer é que existe uma crença de 60% de que os novos itens observados indicam um crescimento no movimento, H_1 . Inversamente, há 40% de crença de que o novo item observado não oferece informação adicional. Portanto, os 40% de probabilidade restantes são atribuídos ao conjunto todo de discernimento, o qual contém H_1 e H_1^c . Em algum ponto posterior, e na presença de novas evidências, essa probabilidade de 40% pode ser reduzida ainda mais.

Esse último ponto é crucial para o entendimento da teoria de D-S. O primeiro passo é quantificar o fenômeno da maneira mais intuitiva possível. Então, as probabilidades vão mudando conforme novas evidências são adicionadas ao problema. É algo familiar ao jogo *Twenty Questions*¹. Por exemplo, após a primeira questão, “A pessoa toca bateria?” (“sim”), os participantes provavelmente desenvolveriam hipóteses a respeito de Ringo Starr, Desi Arnaz, Tito Puente, entre outros músicos. Os participantes ainda veriam pouca razão em achar que a evidência suporta a hipótese de que a pessoa seja o Richard Feynman, um físico ganhador do Prêmio Nobel. Esse último ponto poderia ser expressado como:

$$\begin{aligned} \{\text{Feynman}\} &= H_1 & \text{bpa} &= 0.001 \\ \theta & & \text{bpa} &= 1 - 0.001 = 0.999 \end{aligned}$$

Contudo, após a questão “A pessoa esteve também envolvida no desenvolvimento da bomba nuclear?” (“sim”), a massa de 0.999 para a bpa em θ é convertida para suportar $\{\text{Feynman}\}$ ². $\{\text{Feynman}\}$ havia sido eliminado após a primeira pergunta (atribuindo 0.999 à bpa como *não* $\{\text{Feynman}\}$).

Tal como *Twenty Questions*, a teoria de D-S agrupa várias pequenas evidências para conseguir um efeito “convicente”, ainda que nenhuma evidência consiga isso isoladamente. Isso leva a uma questão: “Como a teoria de D-S combina as diferentes evidências?”

¹Jogo onde os participantes tentam adivinhar uma pessoa ou objeto através do questionamento de uma série de perguntas.

²Antes de ser um renomado físico, Richard Feynman foi um percussionista semi-profissional.

Para adicionar uma nova evidência em conjuntos de hipóteses pré-existentes, é necessário calcular a intersecção de todos os conjuntos de hipóteses correntes e o novo conjunto. Deve-se notar que θ será sempre um dos conjuntos de hipóteses. A bpa associada para cada novo subconjunto é simplesmente o produto das bpas das duas hipóteses que formam o subconjunto. Novos subconjuntos formados por essas intersecções tornam-se novos conjuntos de hipóteses. É importante observar que devido a θ estar sempre presente e porque a união de θ com qualquer outro conjunto de hipóteses H é sempre H , surgem algumas propriedades:

1. Nenhuma hipótese é “destruída” ou “perdida”.
2. A ordem de apresentação das evidências é irrelevante.
3. Evidências que suportam pouco uma hipótese, por exemplo 10%, não diminuem o suporte de outras evidências que suportam fortemente a hipótese, tal como em teoria de probabilidades.

A.3 Intervalo de Crença

Enquanto a combinação inicial das evidências e o cálculo das bpas representam pontos importantes, a teoria de D-S fornece medidas adicionais para manipular evidências e assim fazer projeções de qual hipótese é a mais aceitável. Essa informação está resumida no intervalo de crença, o qual descreve um conjunto de hipóteses não somente em termos de sua crença ou do peso das evidências que a suportam, mas também em termos de possibilidades e dos pesos das evidências que não negam a hipótese. Com isso, tem-se um intervalo de possibilidade.

Os intervalos de crença são calculados para cada conjunto de hipóteses final após todas as evidências terem sido apresentadas e combinadas. Um intervalo de crença tem a forma:

$$H \text{ } [Bel(H) \text{ } Pl(H)],$$

onde H é o conjunto de hipóteses em questão. O intervalo de crença é, por convenção, delimitado por colchetes. A crença, representada por $Bel(H)$ ³, é a probabilidade de que a hipótese H seja suportada baseada somente nas evidências examinadas.

³*Belief*: crença

Ela é calculada pela combinação das bpas associadas à H com as bpas de todas as hipóteses que são subconjuntos de H . A possibilidade, representada por $Pl(H)$ ⁴, é o grau máximo de crença para H , considerando as outras evidências que suportam a negação de H (evidências conflitantes). Ela é calculada pela subtração da crença de todos os subconjuntos de H_1^c de 1.0.

$$Pl(H) = 1 - Bel(H^c)$$

sendo $Bel(H^c)$ calculada somando todas as bpas de todos os conjuntos que são disjuntos de H ; $Bel(H^c)$ nunca pode exceder $Pl(H)$.

Após essa breve introdução sobre a teoria de D-S, será apresentado um exemplo clássico de aplicação da teoria, onde os conceitos vistos e a demonstração de alguns cálculos ficarão mais claros.

A.4 Aplicação da Teoria de Dempster-Shafer

O exemplo abaixo está baseado numa investigação policial, onde o principal personagem é um detetive que tenta desvendar um crime, combinando as evidências coletadas ao longo da investigação. A investigação começa numa sala onde houve um assassinato. O detetive tem quatro suspeitos:

1. o amigo da vítima (F);
2. um rival da vítima (R);
3. o filho irresponsável da vítima (S);
4. o mordomo da vítima (B).

A investigação coleta quatro evidências, as quais são analisadas. Com base em sua experiência de investigação, o detetive atribui algumas probabilidades a cada uma das hipóteses por ele considerada, tal como na Tabela A.1.

O detetive, utilizando a teoria de D-S, combina essas evidências. O resultado da combinação das duas primeiras evidências é mostrado na Tabela A.2. Na tabela, os conjuntos no topo e mais à esquerda são as hipóteses suportadas pelas evidências originais. Os valores internos são os novos conjuntos de hipóteses e bpas formados

⁴ *Plausibility*: possibilidade

Evidência	Suspeitos / Raciocínio Envolvido	BPA
Toco de cigarro	$\{\text{não F}\}=\theta-\{F\}=\{S,B,R\}$ Amigo não fuma.	60%
Discussão recente com o morto	$\{S,B\}$ Muitas pessoas presenciaram discussões com o morto.	80%
Marca de sapato perto da vítima	$\{E,B\}$ Tamanho do sapato é o mesmo dos suspeitos.	60%
Fio de cabelo louro	$\{F\}$ O amigo é o único suspeito com cabelo louro.	30%

Tabela A.1: Hipóteses e as probabilidades de cada hipótese estar correta

calculando a intersecção dos conjuntos da borda da tabela. Esses novos conjuntos serão mantidos para futuras combinações.

Toco de Cigarro					
Argumento		$\{S,B,R\}$	60%	θ	40%
$\{S,B\}$	80%	$\{S,B\}$	0.480	$\{S,B\}$	0.320
θ	20%	$\{S,B,R\}$	0.120	θ	0.080

Tabela A.2: Combinando as duas primeiras evidências

Deve-se notar que existem duas ocorrências do conjunto de hipóteses $\{S,B\}$. As bpas associadas a esses dois conjuntos serão combinadas para formar uma única bpa e um único conjunto de hipótese:

$$\begin{array}{llll}
 \{S, B\} & \text{bpa} & = & (0.48 + 0.32) = 0.80 \\
 \{S, B, R\} & \text{bpa} & & = 0.12 \\
 \theta & \text{bpa} & & = 0.08
 \end{array}$$

É importante ressaltar que bpa é uma medida da massa de evidência que suporta exatamente um dado conjunto de hipóteses. Tal como mostrado no exemplo anterior, uma evidência que suporta exatamente um subconjunto de hipóteses de H , não suporta necessariamente o conjunto H exatamente. Por exemplo, o conjunto $\{S, B, R\}$ contém o subconjunto $\{S, B\}$, mas tem uma bpa menor que $\{S, B\}$.

O resultado de considerar as outras duas evidências — marca deixada pelo sapato e fio de cabelo louro — estão condensadas nas Tabelas A.3 e A.4. Os conjuntos que estão mais à esquerda na tabela são aqueles que foram calculados em passos anteriores e foram mantidos. Os conjuntos que estão no topo da tabela representam as novas evidências.

No caso onde foi adicionada a última evidência (Tabela A.4), o conjunto vazio — $\emptyset$ — aparece várias vezes. Isso ocorre quando são combinadas hipóteses disjuntas, por

Marca de Sapato perto da Vítima					
Anterior		{F,B}	60%	θ	40%
{S,B}	80%	{B}	0.480	{S,B}	0.320
{S,B,R}	12%	{B}	0.072	{S,B,R}	0.048
θ	8%	{F,B}	0.048	θ	0.032

Tabela A.3: Resultado da inserção da terceira evidência

Fio de Cabelo Louro					
Anterior		{F}	30%	θ	70%
{B}	55%	$\emptyset$	0.166	{B}	0.386
{F,B}	5%	{F}	0.014	{F,B}	0.034
{S,B}	32%	$\emptyset$	0.096	{S,B}	0.224
{S,B,R}	5%	$\emptyset$	0.014	{S,B,R}	0.034
θ	3%	{F}	0.010	θ	0.022

Tabela A.4: Resultado da inserção da quarta evidência

exemplo, {F} e {B}. Isso indica que existem duas hipóteses suportadas e que são diretamente conflitantes entre si. Para eliminar o efeito desse tipo de ocorrência, é necessário fazer alguns ajustes. Seja k a bpa relativa a todas as ocorrências de $\emptyset$, então pode-se normalizar os resultados dividindo as probabilidades por $1 - k$, o que efetivamente descarta a porção de evidência que está em conflito. A Tabela A.5 mostra os resultados dessa normalização.

Hipóteses	BPA	BPA Normalizada
{B}	0.39	0.53
{S, B}	0.22	0.31
{S, B, R}	0.03	0.05
{F, B}	0.03	0.05
{F}	0.02	0.03
θ	0.02	0.03
$\emptyset$	0.28	—
$k = \text{bpa}(\emptyset) =$	0.28	
$1 - k =$	0.72	

Tabela A.5: Resultados da normalização

O próximo passo consiste em calcular os intervalos de crença associados à cada hipótese. Desde que o conjunto {B} é um conjunto composto de uma única hipótese, este não possui subconjuntos. Como {F} é a única hipótese não-trivial em conflito com {B}, o cálculo do intervalo de crença para {B} é simples:

$$\begin{aligned}
\text{Bel}(\{B\}) &= \text{bpa de } \{B\} = 0.53, \\
\text{Pl}(\{B\}) &= 1 - \text{Bel}(\text{not } \{B\}) \\
&= 1 - \text{Bel}(\{S, R, F\}) \\
&= 1 - \text{Bel}(\{F\}) \quad (\{F\} \text{ é o único subconjunto de } \{S, R, F\}) \\
&= 1 - 0.03 \\
&= 0.97
\end{aligned}$$

Portanto, o intervalo de crença é: $\{B\}$ [0.53 0.97].

O cálculo do intervalo de crença para $\{S, B\}$ é um pouco mais complexo:

$$\begin{aligned}
\text{Bel}(\{S, B\}) &= \text{bpa de } \{S, B\} + \text{bpa de } \{S\} + \text{bpa de } \{B\} \\
&= (0.31 + 0 + 0.53) \\
&= 0.84, \\
\text{Pl}(\{S, B\}) &= 1 - \text{Bel}(\text{not } \{S, B\}) \\
&= 1 - \text{Bel}(\{R, F\}) \\
&= 1 - (\text{bpa de } \{R, F\} + \text{bpa de } \{R\} + \text{bpa de } \{F\}) \\
&= 1 - (0 + 0 + 0.03) \\
&= 0.97
\end{aligned}$$

Portanto, o intervalo de crença é: $\{S, B\}$ [0.84 0.97].

Os intervalos de crença para cada uma das hipóteses, omitidos os passos intermediários, são:

$\{B\}$	[0.53 0.97]
$\{S, B\}$	[0.84 0.97]
$\{S, B, R\}$	[0.89 0.97]
$\{F, B\}$	[0.61 1.00]
$\{F\}$	[0.03 0.11]

A.5 Interpretação dos Resultados

Não há uma heurística simples para determinar qual é a hipótese ótima ou “correta” (Stein, 1993). Sabe-se que, quanto maior for $\text{Bel}(H)$, é mais provável que o conjunto de hipóteses H contenha a conclusão correta. Contudo, em geral, quanto mais elementos são adicionados ao conjunto de hipóteses H , maior será $\text{Bel}(H)$. Isso parece um tanto quanto paradoxal, pois a utilidade do conjunto de hipóteses é, em geral,

inversamente proporcional ao número de hipóteses possíveis contidas no conjunto. Além disso, quanto maior for o intervalo de crença, mais incerteza será adicionada ao resultado. Uma forma simples para se obter as conclusões, é a interpretação direta dos intervalos de crença para cada conjunto unitário de hipóteses. Essa idéia aplicada ao exemplo do detetive, deriva os seguintes intervalos de crença:

$\{B\}$	[0.53 0.97]
$\{F\}$	[0.03 0.11]
$\{S\}$	[0.00 0.39]
$\{R\}$	[0.00 0.08]

Observando os intervalos de crença, nota-se que existe uma hipótese que contém grande massa de crença: $\{B\}$. Isso leva a crer que o mordomo cometeu o assassinato, o que levaria o detetive a interrogá-lo para coletar novas evidências, e assim adicioná-las e aplicar a teoria de D-S com o intuito de diminuir a incerteza das conclusões.

Entretanto, não deve ser descartado o fato da existência de várias evidências conflitantes. Precisamente, mais de 1/4 das evidências foram descartadas quando ocorreu a normalização. Isso deve ser levado em consideração, pois afeta a crença no resultado final. O ideal seria que não houvessem evidências conflitantes a ponto de necessitar normalizar as bpas. Por exemplo, se durante o interrogatório dos suspeitos o amigo da vítima $\{F\}$ confessar o crime, essa nova evidência deveria ser adicionada com 100% (bpa = 1.0), o que certamente tem um impacto dramático nos resultados conseguidos com a aplicação da teoria de Dempster-Shafer.

A.6 Considerações Finais

Neste capítulo foram apresentadas as noções da teoria de Dempster-Shafer, com o objetivo de mostrar o tratamento de incerteza representada pelo intervalo de crença, bem como ressaltar que essa teoria é altamente subjetiva, pois as bpas são atribuídas por um especialista — no exemplo, o detetive.

Apêndice B

Teoria de Fuzzy Sets

B.1 Considerações Iniciais

A Teoria de Fuzzy Sets tem como objetivo fornecer métodos para especificar o quão bem um objeto satisfaz uma descrição vaga (Russel & Norvig, 1995). Por exemplo, se for considerada a proposição “João é alto”. Será que isso é verdade dado que a altura de João é 1.70 m? A maioria das pessoas hesitaria em responder “sim” ou “não”. Na verdade a maioria responderia “talvez” ou “mais ou menos”. Deve-se notar que essa não é uma questão de incerteza com relação a altura de João, pois tem-se certeza sobre sua altura (1.70 m). A incerteza reside no significado do termo lingüístico “alto”.

Com base nessas considerações serão apresentadas neste apêndice as noções básicas sobre fuzzy sets, com o objetivo de fundamentar qualquer comparação que possa ser feita em relação à Teoria de Rough Sets.

B.2 Fuzzy Sets

A noção de fuzzy sets¹ foi introduzida por Lotfi A. Zadeh em 1965, o qual desenvolveu muitos dos métodos de lógica fuzzy utilizando essa simples noção. Zadeh levou alguns anos até que sua teoria fosse entendida e aplicada por outros cientistas.

A maneira convencional de representar elementos u de um conjunto crisp X é através

¹Dentre os significados possíveis da palavra fuzzy, *vago* ou *difuso* parecem ser os mais apropriados. Neste trabalho usaremos o termo *fuzzy* por ser amplamente aceito pela comunidade.

da *função característica*:

$$\mu_X(u) = \begin{cases} 1, & \text{se } u \in X \\ 0, & \text{se } u \notin X \end{cases}$$

Dado um elemento u é possível decidir se esse elemento pertence (1) ou não (0) ao conjunto X .

Em fuzzy sets um dado elemento pode pertencer parcialmente a um conjunto. O grau de pertinência é definido através de uma generalização da função característica, chamada de *função de pertinência*:

$$\mu_X(u) : U \rightarrow [0, 1]$$

onde U é chamado de *universo*, e X é um *subconjunto fuzzy* de U .

Os valores da função de pertinência são números reais no intervalo $[0,1]$, onde 0 significa que o objeto não é um membro do conjunto e 1 significa que ele pertence com certeza ao conjunto. Cada valor da função é chamado de *grau de pertinência*. A Figura B.1 mostra três funções de pertinência representando três conjuntos fuzzy denominados “pequeno”, “médio” e “alto”, onde todos eles são valores fuzzy da variável “altura”². Nota-se que o valor 170 cm pertence ao conjunto fuzzy “médio” com um grau de pertinência 0.2, e ao mesmo tempo pertence ao conjunto fuzzy “alto” com grau de pertinência 0.7.

Se o universo é *discreto*, uma função de pertinência pode ser definida por um conjunto finito, tal como segue:

$$X = \mu(u_1)/u_1 + \mu(u_2)/u_2 + \dots + \mu(u_n)/u_n$$

ou simplesmente

$$X = \sum \mu_i/u_i$$

onde o símbolo “/” separa os graus de pertinência $\mu(u_i)$ dos elementos do universo $\mu_i \in U$, e + representa a união dos elementos. De uma forma simplificada, um conjunto fuzzy é representado como uma sequência de pares de “grau de pertinência/valor”: $\{0/150, 0.3/160, 0.68/170, 0.9/180, 1/190, 1/250\}$. A principal diferença entre um conjunto crisp e um conjunto fuzzy é ilustrada na Figura B.2.

²Na terminologia de fuzzy sets, altura é chamada de *variável fuzzy* ou *variável lingüística*. Os valores que essa variável pode assumir (“pequeno”, “médio” e “alto”) são chamados de *qualificadores*.

Figura B.1: Função de pertinência representando a variável “altura”

Conjuntos crisp utilizam bordas bem definidas, enquanto que em conjuntos fuzzy não existe uma borda bem definida para se afirmar se um determinado elemento pertence ou não ao conjunto. Analisando a Figura B.2 nota-se que para os valores de temperatura 14.999 e 15.001 os graus de pertinência estão muito próximos em relação ao conjunto fuzzy “médio”. No entanto, esses valores pertencem a conjuntos distintos se levada em consideração a borda rígida dos conjuntos crisp “boa” e “média”. Essa característica faz com que os conjuntos fuzzy também sejam conhecidos como *conjuntos nebulosos*. A Figura B.3 mostra a diferença entre um conjunto crisp e um conjunto fuzzy do ponto de vista das bordas. No conjunto crisp, pode-se afirmar com total certeza que u_1 não pertence ao conjunto e que u_2 pertence ao conjunto. Para o conjunto fuzzy, u_3 e u_4 pertencem ao conjunto com algum grau de verdade. Logicamente, u_4 deverá possuir um maior grau de verdade em relação a u_3 .

Algumas noções básicas de conjuntos fuzzy são definidas a seguir:

Definição B.2.1 (Suporte) O suporte (support) de um conjunto fuzzy X é o subconjunto do universo U , para o qual todos os elementos possuem um grau de pertinência diferente de zero (Figura B.4).

$$\text{supp}(A) = \{u \mid u \in U, \mu_X(u) > 0\}$$

Por exemplo, na Figura B.3 o suporte para o conjunto fuzzy “temperatura média” é o intervalo $[10, 30]$. Um conjunto fuzzy X pode ser inteiramente formulado por seu

Figura B.2: Representação de conjuntos crisp e fuzzy como subconjuntos do domínio (universo) U

suporte:

$$X = \{\mu_X(u)/u \mid u \in \text{supp}(X)\}$$

Definição B.2.2 (Cardinalidade) A cardinalidade (cardinality) de um conjunto crisp é definida como o número de elementos pertencentes ao conjunto, enquanto que a cardinalidade em conjuntos fuzzy $M(X)$ é definida como:

$$M(X) = \sum \mu_X(u), u \in U$$

Definição B.2.3 (Conjunto Potência) O conjunto potência (power set) de A é o conjunto formado por todos os subconjuntos fuzzy de A .

Definição B.2.4 (Conjunto Fuzzy Normal) Um conjunto fuzzy A é chamado de conjunto fuzzy normal se sua função de pertinência produz ao menos um valor 1 quando aplicada aos elementos do universo U .

Definição B.2.5 (α -cut) Todo conjunto fuzzy X pode ser representado pelo seu α -cut, o qual pode ser definido como fraco ou forte (Figura B.5). Um α -cut de um conjunto

Figura B.3: Representação de conjuntos crisp e fuzzy do ponto de vista da existência de bordas bem definidas

fuzzy X é um subconjunto X_α do universo U que consiste dos elementos que pertencem ao conjunto fuzzy X com grau de pertinência maior (fraco) ou maior-igual (forte) ao valor $\alpha \in [0, 1]$.

A teoria de fuzzy sets pode ser considerada como uma extensão da teoria clássica dos conjuntos, a qual teve seus operadores estendidos à teoria de fuzzy sets. Esses operadores serão apresentados na próxima seção.

B.3 Operações e Propriedades

Como já mencionado, conjuntos crisp são um caso especial de conjuntos fuzzy, onde somente dois graus de pertinência existem, 0 e 1, e a borda que delimita os conjuntos é bem definida. Todas as definições, provas, e teoremas que se aplicam a conjuntos fuzzy, devem ser válidos quando a incerteza torna-se nula, ou seja, quando os conjuntos fuzzy tornam-se conjuntos crisp.

Deve-se definir alguns operadores sobre conjuntos fuzzy. Uma função análoga à função de pertinência é utilizada na Figura B.6 para representar algumas operações sobre os conjuntos fuzzy. Os seguintes operadores são definidos sobre dois conjuntos fuzzy X e Y , os quais pertencem ao mesmo universo U .

1. *União*, $X \cup Y$:

$$\mu_{X \cup Y}(u) = \mu_X(u) \vee \mu_Y(u), \text{ para todo } u \in U, \text{ onde } \vee \text{ denota MAX;}$$

2. *Intersecção*, $X \cap Y$:

Figura B.4: Suporte do conjunto fuzzy X

$\mu_{X \cap Y}(u) = \mu_X(u) \wedge \mu_Y(u)$, para todo $u \in U$, onde $\wedge$ denota MIN;

3. *Igualdade*, $X = Y$:

$\mu_X(u) = \mu_Y(u)$, para todo $u \in U$;

4. *Complemento*, *not* X , $\neg X$:

$\mu_{\neg X}(u) = 1 - \mu_X(u)$, para todo $u \in U$;

5. *Concentração*, $CON(X)$:

$\mu_{CON(X)}(u) = (\mu_X(u))^2$, para todo $u \in U$; esta operação é utilizada como o modificador lingüístico “muito”;

6. *Dilatação*, $DIL(X)$:

$\mu_{DIL(X)}(u) = (\mu_X(u))^{0.5}$, para todo $u \in U$; esta operação é utilizada como o modificador lingüístico “mais ou menos”;

7. *Subconjunto*, $X \subseteq Y$:

$\mu_X(u) \leq \mu_Y(u)$, para todo $u \in U$;

8. *Produto Algébrico*, $X \cdot Y$:

Figura B.5: α -cut de um conjunto fuzzy

Figura B.6: Cinco operações com dois conjuntos fuzzy A e B

$$\mu_{X \cdot Y}(u) = \mu_X(u) \cdot \mu_Y(u), \text{ para todo } u \in U;$$

9. *Soma Limitada*, $X| + |Y$:

$$\mu_{X|+|Y}(u) = \max\{1, \mu_X(u) + \mu_Y(u)\}, \text{ para todo } u \in U;$$

10. *Diferença Limitada*, $X| - |Y$:

$$\mu_{X|-|Y}(u) = \min\{0, \mu_X(u) - \mu_Y(u)\}, \text{ para todo } u \in U;$$

11. *Produto Limitado*, $X| \cdot |Y$:

$$\mu_{X| \cdot |Y}(u) = \max\{0, \mu_X(u) + \mu_Y(u) - 1\}, \text{ para todo } u \in U;$$

12. *Normalização*, $NORM(X)$:

$$\mu_{NORM(X)}(u) = \mu_X(u) / MAX\{\mu_X(u)\}, \text{ para todo } u \in U;$$

13. *Soma Algébrica*, $X + Y$:

$$\mu_{X+Y}(u) = \{\mu_X(u) + \mu_Y(u)\}, \text{ para todo } u \in U.$$

A Lei de De Morgan é válida para a união, intersecção, soma e diferença algébrica.

As operações sobre os conjuntos fuzzy possuem as propriedades: *associativa*, *comutativa* e *distributiva*, isto é, se X , Y e Z são conjuntos fuzzy:

1. *Associativa*: $(X * Y) * Z = X * (Y * Z)$;
2. *Comutativa*: $X * Y = Y * X$ (não é válida para diferença limitada);
3. *Distributiva*: $X * (Y \circ Z) = (X \circ Y) * (X \circ Z)$.

onde $*$ e $\circ$ denotam qualquer um dos operadores citados anteriormente.

Um ponto muito importante e que distingue os conjuntos fuzzy dos conjuntos crisp, é que os primeiros rompem com a *Lei do Meio Excluído* e a *Lei da Contradição*. Portanto, os resultados abaixo podem ser verdadeiros:

$$X \cup \neg X \neq U$$

$$X \cap \neg X \neq \emptyset$$

Ou seja, a união de um conjunto fuzzy X com o seu complementar $\neg X$ não é necessariamente igual ao conjunto universo U . E a intersecção entre ambos não é necessariamente igual ao conjunto vazio ($\emptyset$).

B.4 Medidas de Ambigüidade

Medir a *ambigüidade* de um conjunto fuzzy é uma característica interessante. A ambigüidade de um conjunto fuzzy pode ser medida através de sua *entropia* (*entropy*):

$$E(X) = \frac{M(X \cap \neg X)}{M(X \cup \neg X)}$$

onde M denota a cardinalidade do conjunto. Quanto maior a entropia, maior é a ambigüidade do conjunto fuzzy. Obviamente, conjuntos crisp têm entropia igual a 0. Outra forma de medir a entropia de um conjunto fuzzy X é a aplicação da fórmula

abaixo:

$$E(X) = -k \sum \{\mu_X(u_i) \cdot \log \mu_X(u_i) + \mu_{\neg X}(u_i) \cdot \log \mu_{\neg X}(u_i)\}, \forall u \in U.$$

onde $k > 0$ é uma constante.

Outras medidas interessantes são apresentadas para medir a *distância* e a *similaridade* entre conjuntos fuzzy. Uma representação gráfica é mostrada na Figura B.7.

Figura B.7: A região em preto representa quantitativamente as medidas de similaridade (a) e distância (b) entre conjuntos fuzzy.

A similaridade S entre dois conjuntos fuzzy X e Y pode ser interpretada como uma forma de quantificar o quanto o conjunto X é igual ao conjunto Y . A similaridade pode ser medida calculando a *possibilidade* P e a *necessidade* N conforme a equação a seguir, que está ilustrada na Figura B.8.

$$S = \begin{cases} P(X/Y), & \text{se } N(X/Y) > 0.5 \\ (N(X/Y) + 0.5) * P(X/Y), & \text{caso contrário} \end{cases}$$

onde $P(X/Y) = \max\{\min\{\mu_X(u), \mu_Y(u)\}\}$, para todo $u \in U$; $N(X/Y) = 1 - P(\neg X/Y)$. Para o exemplo, $N(X/Y) = 0.2$, e $S = (0.2 + 0.5) * 0.8 = 0.56$.

Figura B.8: Representação gráfica do cálculo da similaridade S entre dois conjuntos fuzzy X e Y com base na possibilidade P e na necessidade N

B.5 Considerações Finais

Existem vários tipos de incerteza no mundo real. Um desses tipos está relacionado à incerteza dos termos lingüísticos (por exemplo, “João é alto se sua altura é 1.70 m?”). Para contornar essa incerteza e relativizar a pertinência de um dado elemento a um dado conjunto foi desenvolvida a teoria de fuzzy sets, a qual foi apresentada neste capítulo abordando apenas os aspectos básicos de sua formulação.