
Giuseppe Mongiovi

**Uso de relevância semântica na melhoria da
qualidade dos resultados gerados pelos
métodos indutivos de aquisição de
conhecimento a partir de exemplos**

Campina Grande
1995

Giuseppe Mongiovi

**Uso de relevância semântica na melhoria da
qualidade dos resultados gerados pelos
métodos indutivos de aquisição de
conhecimento a partir de exemplos**

Tese apresentada ao Curso de Doutorado em
Engenharia Elétrica da Universidade Federal da
Paraíba, como requisito parcial à obtenção do título
de doutor em Engenharia Elétrica.

Área de Concentração: Inteligência Artificial

Orientador: Prof. José Antão Beltrão Moura
Universidade Federal da Paraíba

Campina Grande
Universidade Federal da Paraíba
1995



M743u

Mongiovi, Giuseppe.

Uso de relevância semântica na melhoria da qualidade dos resultados gerados pelos métodos indutivos de aquisição de conhecimento a partir de exemplos / Giuseppe Mongiovi. - Campina Grande, 1995.

232 f.

Tese (Doutorado em Engenharia Elétrica) - Universidade Federal da Paraíba, Centro de Ciências e Tecnologia, 1995.

"Orientação : Prof. Dr. José Antônio Beltrão Moura".

Referências.

1. Inteligência Artificial. 2. Aquisição de Conhecimento. 3. Métodos Indutivos. 4. Tese - Engenharia Elétrica. I. Moura, José Antônio Beltrão. II. Universidade Federal da Paraíba - Campina Grande (PB). III. Título

CDU 004.8(043)

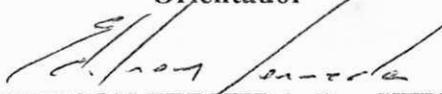
USO DA RELEVÂNCIA SEMÂNTICA NA MELHORIA DA QUALIDADE DOS
RESULTADOS GERADOS PELOS MÉTODOS INDUTIVOS DE AQUISIÇÃO E
CONHECIMENTO A PARTIR DE EXEMPLOS

GIUSEPPE MONGIOVI

Tese Aprovada em 09.06.1995



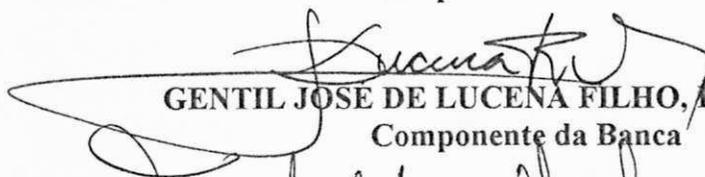
JOSÉ ANTÃO BELTRÃO MOURA, Ph.D.,
Orientador



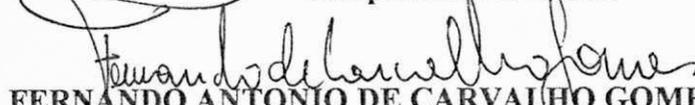
EDILSON FERNEDA, Dr., UFPB
Orientador



EMMANUEL PISECES LOPES PASSOS, Dr., IME,
Componente da Banca

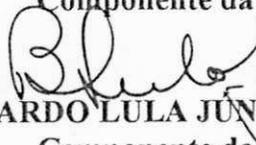


GENTIL JOSÉ DE LUCENA FILHO, Ph.D., CNPQ,
Componente da Banca



FERNANDO ANTONIO DE CARVALHO GOMES, Dr., UECE,
Componente da Banca

JOHN FRANKLIN ARCE, Ph.D., PUC/Santiago-Chile
Componente da Banca



BERNARDO LULA JÚNIOR, Dr., UFPB
Componente da Banca

CAMPINA GRANDE - PB

Junho - 1995

A meus pais Angelo (in memorian) e Serafina,
meu irmão Antonino,
minha esposa Gracinha, e
meus filhos Finuzza, Angelo e Melina.

Agradecimentos

Agradeço inicialmente ao grande Arquiteto do Universo, por essa obra magnífica que é a mãe natureza, a quem devo tudo o que sou e como sou.

À minha família, em particular aos meus pais Angelo e Serafina, meu irmão Antonino, minha esposa Gracinha e meus filhos Finuzza, Angelo e Melina, por todo amor, carinho, compreensão e incentivo, que me deram, me dão e, tenho certeza, me darão.

Aos meus alunos, em particular Fernando, Walfredo, Vasco, Tojal e Reginaldo, que foram a terra fértil de onde brotou este trabalho.

Ao meu orientador, prof. Antão, pelas sugestões, atenção, incentivo e confiança.

Ao meu co-orientador, prof. Edilson, pelo incentivo e sugestões.

Aos professores, membros da banca examinadora, Antão, Bernardo, Edilson, Emmanuel, Fernando, Gentil e John, por aceitarem participar da banca e, principalmente, pelo incentivo e pelas sugestões dadas, que, sem dúvida, foram de grande valia na redação final deste trabalho.

Aos atuais e ex-professores do Departamento de Sistemas e Computação, em particular Antão, Bernardo, Camilo, Chiquinho, Edilson, Fátima, Gentil, Hattori, Hélio, Izabel (Bibi), Joseluze, Pedrinho e Wanderley, pelo incentivo.

À Fátima Turnell e Angela, da COPELE, pela atenção.

Aos doutores Demóstenes Gonçalves, Maria Sidneuma e Renato Andrade, pela ajuda na preparação dos dados de alguns domínios utilizados nesta tese.

Sinceros agradecimentos às Instituições:

- Departamento de Sistemas e Computação, pelo apoio e facilidades concedidas;
- Grupo de Inteligência Artificial do DSC, pelo apoio dado em todas as fases de desenvolvimento deste trabalho;
- Coordenação de Pós Graduação em Engenharia Elétrica, pela atenção;
- CNPq, pelo apoio financeiro;
- Universidade Federal da Paraíba, por meio do Centro de Ciências e Tecnologia, pelo subsídios na participação de eventos científicos.

Finalmente, o meu muito obrigado a todos aqueles que, direta ou indiretamente, contribuíram de alguma forma com a realização deste trabalho.

Lista de figuras

FIGURA 1.1 - Uma classificação dos métodos de AC.....	04
FIGURA 1.2 Relação entre AA e AC.....	08
FIGURA 2.1 - Protocolo de um ambiente de aprendizagem.....	25
FIGURA 2.2 - O Ambiente de aprendizagem MOSCA.....	26
FIGURA 2.3 - Um Ambiente de aprendizagem para o paradigma conexionista.....	30
FIGURA 2.4 - Um Ambiente de aprendizagem para um CBL.....	32
FIGURA 2.5 - Um Ambiente de aprendizagem para um EBL.....	33
FIGURA 2.6 - Classificação dos sistemas de aprendizagem indutiva.....	35
FIGURA 2.7 - Um Ambiente de aprendizagem para o paradigma indutivo.....	37
FIGURA 2.8 - Possíveis situações de completude e consistência de uma hipótese induzida.....	45
FIGURA 3.1 - Exemplo de uma AD.....	46
FIGURA 3.2 - Conjunto de treinamento para o domínio "magro-gordo".....	51
FIGURA 3.3 - AD gerada pelo ID3.....	52

FIGURA 3.4 - AD gerada aleatoriamente.....	52
FIGURA 3.5 - AD correspondente as regras u_1 e u_2	56
FIGURA 3.6 - Algoritmo PRISM.....	61
FIGURA 4.1 - Exemplo de uma MR	66
FIGURA 4.2 - Exemplo de uma MRN.....	69
FIGURA 4.3 - Matriz de composição de especialistas.....	77
FIGURA 5.1 - Algoritmo ADEX	82
FIGURA 5.2 - AD expandida pelo ADEX a partir de uma AD ID3.....	83
FIGURA 5.3 - AD gerada pelo IDRT usando entropia normalizada	86
FIGURA 6.1 - Árvore de decisão gerada pelo EG2 para o domínio "brinquedo seguro".....	99
FIGURA 8.1 - Trechos do relatório de análise da variação dos parâmetros do algoritmo ISREG.....	128
FIGURA 8.2 - Relatório padrão de saída dos algoritmos.....	129
FIGURA 9.1 - Camadas da arquitetura do A4.....	148
FIGURA 9.2 - Arquitetura do A4.....	150
FIGURA 9.3 - As atribuições e conhecimento do gerente.....	152
FIGURA 10.1 - Tipos de valores de atributos.....	158
FIGURA 10.2 - Utilização de conhecimento hierárquico em ADs.....	161
FIGURA 10.3 - A classe MODELA DOMÍNIO no contexto do A4.....	162

FIGURA 10.4 - Composição de valores de atributos.....	163
FIGURA 10.5 - Fases da entrevista da classe modela domínio.....	168
FIGURA 11.1 - Fases do aprendizado no Modelo Neural Combinatório.....	171
FIGURA 11.2 - Grafo de conhecimento sobre Comunicação Interatrial (CIA), retirado de [Leão 88].....	175
FIGURA 11.3 - Distribuição da relevância semântica entre classes.....	185
FIGURA A.1 - Formulário de avaliação do grau de relevância de uma generalização (valores não generalizados).....	203
FIGURA A.2 - Formulário de avaliação do grau de relevância de uma generalização (valores generalizados).....	204

Lista de tabelas

TABELA 1.1 - Referências X objetivos da tese.....	17
TABELA 1.2 - Evolução cronológica das dissertações de mestrado envolvidas na solução dos subproblemas deste trabalho.....	19
TABELA 4.1 - <i>MRNs</i> de cinco especialistas.....	76
TABELA 5.1 - Comparação entre os diversos algoritmos para o domínio "magro-gordo".....	92
TABELA 6.1 - Conjunto de Treinamento para o domínio "brinquedo seguro".....	96
TABELA 6.2 - Informações de custo e generalização, para o domínio "brinquedo seguro", adaptado de [Nuñez 91].....	96
TABELA 7.1 - Comportamento da média versus o valor ajustado no cálculo da relevância.....	117
TABELA 7.2 - Matriz de relevância nebulosa para o domínio "brinquedo seguro".....	120
TABELA 7.3 - Resultados da avaliação das bases de conhecimento geradas, pelos algoritmos estudados, para o domínio "brinquedo seguro".....	120
TABELA 8.1 - Resultados dos algoritmos para o domínio "brinquedo seguro".....	129

TABELA 8.2 - Comparação objetiva entre os algoritmos semânticos e não semânticos para o domínio "Brinquedo seguro".....	131
TABELA 8.3 - Principais características dos domínios utilizados nos testes.....	133
TABELA 8.4 - Resultados dos algoritmos para o domínio "amenorréia".....	134
TABELA 8.5 - Comparação objetiva entre os algoritmos semânticos e não semânticos para o domínio Amenorréia.....	134
TABELA 8.6 - Resultados dos algoritmos para o domínio "Zoo".....	136
TABELA 8.7 - Comparação objetiva entre os algoritmos semânticos e não semânticos para o domínio "Zoo".....	137
TABELA 8.8 - Resultados dos algoritmos para o domínio "heart-disease-Cleveland".....	138
TABELA 8.9 - Comparação objetiva entre os algoritmos semânticos e não semânticos para o domínio "Heart-disease-Cleveland".....	139
TABELA 8.10 - Resultados dos algoritmos para o domínio "pima-indians-diabetes".....	141
TABELA 8.11 - Comparação objetiva entre os algoritmos semânticos e não semânticos para o domínio "Pima-indians-diabetes".....	142
TABELA 8.12 - Comparação objetiva entre os algoritmos semânticos e não semânticos para todos os domínios conjuntamente (média dos <i>índices de qualidade global</i>).....	145
TABELA 11.1. A tabela mostra o tamanho da TCOS de ordem 5, 7 e 9, para diversos números de evidências.....	174

TABELA 11.2. Mostra a razão entre o tamanho da TCROS e da TCOS para redes com ordens entre 1 e 7, referentes ao domínio do estudo de caso em ginecologia.....	182
TABELA A.1 - Codificação das classes utilizadas no experimento.....	205
TABELA A.2 - Codificação das generalizações utilizadas no experimento.....	206
TABELA A.3 - Resultado final do experimento.....	206
TABELA A.4 - Resultados por generalização.....	208

Lista de gráficos

GRÁFICO 7.1 - Convergência da relevância de uma regra.....	118
GRÁFICO 11.1 - Variação do parâmetro ordem da rede.....	186
GRÁFICO 11.2 - Variação da distribuição da relevância entre as classes, mantendo a intensidade relevância no valor típico de 25%.....	187
GRÁFICO 11.3-A - Variação da intensidade de relevância, mantendo a distribuição uniforme (proporcional a do cenário A da FIG. 11.3).....	188
GRÁFICO 11.3-B - Variação da intensidade de relevância, mantendo a distribuição dispersa(proporcional a do cenário B da FIG. 11.3).....	188
GRÁFICO A.1 - Resultado final do experimento.....	207
GRÁFICO A.2 - Resultados por generalização.....	208

Sumário

Resumo	xxiv
---------------------	------

1 Introdução

1.1 Inteligência artificial e sistemas baseados em conhecimento.....	01
1.2 Métodos de aquisição de conhecimento.....	03
1.2.1 Métodos cognitivos.....	04
1.2.2 Métodos automatizados.....	07
1.2.2.1 Métodos semi-automáticos.....	07
1.2.2.2 Métodos Automáticos.....	08
1.3. Objetivos da tese.....	12
1.4 Contribuições para a área de aquisição de conhecimento.....	15
1.5 Importância da tese.....	19
1.6 Estrutura da tese.....	20

2 Os Paradigmas da aprendizagem automática

2.1 Introdução.....	22
---------------------	----

2.1.1 O que é aprendizagem.....	22
2.1.2 Protocolos de aprendizagem.....	24
2.1.3 Tipos de inferências lógicas.....	27
2.2 O Paradigma conexionista.....	29
2.3 O Paradigma genético.....	30
2.4 O Paradigma analítico.....	31
2.4.1 Métodos baseados em casos.....	31
2.4.2. Métodos baseados em explicações.....	33
2.5 O Paradigma indutivo.....	34
2.5.1 Classificação da aprendizagem indutiva.....	35
2.5.2 Aprendizagem indutiva a partir de exemplos.....	37
2.5.2.1 Representação de conceitos, exemplos e conhecimento preliminar.....	38
2.5.1.2 Fundamentação lógica da aprendizagem indutiva.....	43
 3 Os Problemas sintático e semântico	
3.1 Indução em árvores de decisão.....	46
3.1.1 Árvores de decisão.....	46
3.1.2 A Família de algoritmos <i>TDIDT</i>	48
3.1.2.1 Procedimento geral.....	49

3.1.2.2 O Algoritmo ID3.....	50
3.1.2.3 Análise dos algoritmos <i>TDIDT</i>	53
3.2 O Problema semântico.....	54
3.3 O Problema sintático.....	55
3.4 Uma Solução para o problema sintático.....	58
3.4.1 O que mudar nos algoritmos <i>TDIDT</i>	58
3.4.2 O Algoritmo PRISM.....	59
3.4.2.1 O Porque do PRISM.....	59
3.4.2.2 Descrição do PRISM.....	60
3.4.3 Exemplo com o algoritmo PRISM.....	62
4 A Relevância semântica	
4.1 Conhecimento preliminar.....	63
4.2 Matriz de relevância.....	64
4.3 Matriz de relevância nebulosa.....	67
4.3.1 Definição da matriz de relevância nebulosa.....	68
4.3.2 Intensidade de relevância.....	69
4.3.3 Construção da matriz de relevância nebulosa.....	72
4.3.3.1 Matriz de composição de especialistas(<i>MCE</i>).....	73
4.3.3.2 Exemplo da construção de uma <i>MCE</i>	74

4.4 As Regras inúteis e a \mathcal{MRN}	77
---	----

5 Soluções do problema semântico via matriz de relevância

5.1 Evolução da solução do problema semântico.....	80
--	----

5.2 Soluções do problema semântico usando \mathcal{MR}	81
--	----

5.2.1 O Algoritmo ADEX.....	81
-----------------------------	----

5.2.2 O Algoritmo IDRT.....	84
-----------------------------	----

5.3 Uso da \mathcal{MRN} na solução do problema semântico.....	87
--	----

5.4 Uma solução abrangente para os problemas sintático e semântico.....	88
---	----

5.4.1 O Algoritmo RPRISM.....	88
-------------------------------	----

5.4.2 O Algoritmo FRPRISM.....	90
--------------------------------	----

5.5 Análise dos resultados parciais	91
---	----

6 Indução semântica de regras modulares, econômicas e generalizadas

6.1 Introdução.....	93
---------------------	----

6.2 Indução de árvores de decisão econômicas e generalizadas.....	94
---	----

6.2.1 Custo.....	94
------------------	----

6.2.2 Generalização.....	95
--------------------------	----

6.2.3 O Algoritmo EG2.....	97
6.3 O Algoritmo ISREG.....	100
6.3.1. Função de avaliação global.....	101
6.3.2 Procedimento geral do ISREG.....	103
6.3.3 O Algoritmo ISREG no Ambiente A4.....	105
6.4 Conclusão.....	107

7 Qualidade de uma base de conhecimento

7.1 Introdução.....	108
7.2 Formas de avaliação de uma base de conhecimento.....	109
7.2.1 Aspectos objetivos.....	109
7.2.2 Aspecto custo.....	110
7.2.3 Aspectos subjetivos.....	111
7.3 Utilização da relevância semântica na avaliação dos aspectos subjetivos de uma base de conhecimento.....	113
7.4 Definição do grau de relevância de uma base de conhecimento.....	114
7.5 Utilização das formas de avaliação de uma base de conhecimento.....	118
7.6 Avaliação de uma base de conhecimento para o domínio "brinquedo seguro".....	119

8 Análise dos resultados obtidos com os algoritmos semânticos

8.1 Introdução.....	122
8.2 Definição de um índice de qualidade global entre dois algoritmos.....	124
8.3 Procedimento de análise dos resultados.....	126
8.4 O domínio "brinquedo seguro".....	127
8.5 Domínios reais.....	132
8.5.1 Amenorréia.....	133
8.5.2 Zoo.....	135
8.5.3 Heart-disease-Cleveland.....	138
8.5.4 Pima-indians-diabetes.....	140
8.6 Análise final.....	143
8.6.1 Conclusões de carácter geral.....	144
8.6.2 Quadro resumo.....	145
8.6.3 O Algoritmo ISREG.....	146

9 A4: Ambiente de apoio à aquisição automática de conhecimento utilizando algoritmos indutivos

9.1 Objetivos e características do A4.....	148
--	-----

9.2 Arquitetura do A4.....	150
9.2.1 Gerente.....	151
9.2.2 Algoritmo indutivo.....	152
9.2.3 Entrada tratada.....	153
9.2.4 Modela domínio.....	153
9.2.5 Trata saída.....	154
9.2.6 Saída.....	156

10 Modelagem do domínio

10.1 Em que modelar o domínio.....	157
10.1.1 Os Exemplos.....	158
10.2.2 Conhecimento preliminar.....	159
10.2 A Classe modela domínio.....	161
10.2.1 Entrevistador.....	161
10.2.2 Estruturador.....	162
10.2.3 Trata ruído.....	164
10.3 Como realizar a modelagem.....	165
10.3.1 Alternativas de eliciação.....	166
10.3.2 O Processo de entrevista.....	167

11 Uso de semântica na melhoria do aprendizado em um modelo simbólico- conexionista

11.1 Introdução.....	170
11.2 Topologias previstas pelo MNC.....	172
11.2.1 Topologia Combinatorial de Ordem Superior (TCOS).....	172
11.2.2 Topologia eliciada.....	174
11.3 A Relevância semântica e a TCROS.....	176
11.3.1 A Relevância semântica.....	176
11.3.2 A Topologia Combinatorial Relevante de Ordem Superior (TCROS).....	178
11.4 Análise comparativa entre as topologias TCROS e TCOS.....	179
11.4.1 Um caso real (O Ginecol).....	181
11.4.2 Generalizando os resultados.....	182
11.4.2.1 Identificação e variação dos parâmetros.....	184
11.4.2.2 Os Resultados do estudo.....	185
11.5 Conclusão.....	189

12 Conclusões e trabalhos futuros

12.1 Conclusões.....	192
12.2 Trabalhos futuros.....	198

Apêndice A - Relevância semântica de uma generalização: definição e um estudo de caso.....	201
Apêndice B - Documentação dos domínios utilizados nos testes.....	209
Abstract.....	219
Referências Bibliográficas.....	220

Resumo

Os algoritmos indutivos de aquisição automática de conhecimento a partir de exemplos, em geral, apresentam um caráter puramente empírico, não usam conhecimento preliminar, observando apenas os aspectos quantitativos na geração de uma base de conhecimento. Procuram fornecer bases de conhecimento com um mínimo de tamanho e um máximo de acurácia, sem nenhuma preocupação com a qualidade semântica dos resultados gerados. Procurando preservar o caráter automático do processo de aquisição de conhecimento, com esta tese diminuimos o aspecto empírico do processo, propondo um tipo de conhecimento preliminar, na forma de relevância semântica, que é de fácil representação e eliciação. Esse conhecimento, de forma isolada ou combinado com outros tipos de conhecimento preliminar, como custo e generalização, é incorporado aos algoritmos indutivos visando a geração de bases de conhecimento de melhor qualidade, particularmente no aspecto semântico. Além disso, utilizamos a relevância semântica para definir um processo automático de avaliação da qualidade semântica de uma base de conhecimento, bem como para contornar o problema do grande esforço computacional apresentado pelo Modelo Neural Combinatório. Visando aumentar o grau de automação do processo como um todo, propomos ainda, nesta tese, o ambiente de apoio à aquisição automática de conhecimento A4. O ambiente auxilia na modelagem do domínio em exemplos e conhecimento preliminar e na avaliação qualitativa dos resultados fornecidos pelos algoritmos generalizadores.

1. Introdução

1.1 Inteligência artificial e sistemas baseados em conhecimento

O termo Inteligência Artificial (IA) foi criado por John MacCarthy, em 1956, na histórica conferência de Darmouth, que reuniu cientistas de diversas áreas para discutir como eles estavam usando o computador para simular os diversos aspectos da inteligência humana. Mas o que quer dizer IA? Varrendo a literatura, não vamos encontrar, entre os pesquisadores de peso na área, um consenso na definição. Cada um tem a sua visão do que seja IA. Contudo, tentaremos aqui, pelo menos, delinear quais os objetivos do estudo da IA. Para isto, apresentaremos duas definições que, no nosso entender, quando combinadas dão uma boa idéia do que seja a área.

A primeira, devida a Patrick H. Winston [Winston 87], é:

" IA é a área do conhecimento , relacionada a sistemas de computação, que procura imitar as complexas ações humanas (falar, andar, raciocinar, etc.)."

A segunda, de Elaine Rich [Rich 83], afirma :

" IA estuda como fazer os computadores realizarem bem certas tarefas que, embora complicadas para máquinas, são simples para seres humanos."

Implícita ou explicitamente, as duas definições levam a um ponto comum: Uso de conhecimento para simular ações humanas por meio de computador.

As pesquisas em IA têm se direcionado no sentido de procurar métodos específicos de solução de problemas, dependentes do domínio do conhecimento e do tipo do problema. Esta direção se deve ao insucesso da busca por métodos gerais, ocorrida durante a década de 60 e meados da de 70 [Jackson 86], quando se buscava desenvolver um resolvidor universal de problemas. Dessa experiência, percebeu-se que um grande poder de inferência não seria bastante para se desenvolver um sistema "inteligente" de utilidade prática. Mais importante que o poder de inferência era a quantidade e qualidade do conhecimento manipulado pelo sistema. Devido a isso, os programas passaram a ser desenvolvidos com ênfase no conhecimento, o que deu origem aos Sistemas Baseados em Conhecimento (*SBCs*). Os *SBCs* são programas construídos com o cuidado de se separar claramente o conhecimento do código que manipula esse conhecimento, i.e., realiza inferência sobre o mesmo. Essa separação é a principal característica de um *SBC*; ela evidencia a importância do conhecimento dentro do sistema e facilita a sua representação, aquisição, manipulação e manutenção.

Os *SBCs* mais conhecidos, de maior sucesso comercial e um dos maiores responsáveis pela evolução da IA nas últimas duas décadas são os Sistemas Especialistas (*SEs*). Um sistema especialista lida com problemas complexos de uma estreita área de aplicação do mundo real, requerendo assim, na sua construção, a intervenção de um ou mais especialistas do domínio. Deve ainda possuir a capacidade de justificar seus conselhos, análises ou conclusões e se for dotado de dados probabilísticos ou nebulosos deve lidar com incertezas [Rich 91].

Em um *SBC*, o conhecimento, devidamente representado, constitui a Base de Conhecimento (*BC*). Aquisição de Conhecimento (*AC*), por sua vez, é o processo que permite a transferência do conhecimento sobre a solução de um determinado problema para a *BC*. Além disso, precisamos de algoritmos que usem o conhecimento representado na *BC*. Estas tarefas estão a cargo do engenheiro do conhecimento, que é o profissional (geralmente especializado em computação, particularmente em IA) a conceber e construir os *SBCs*.

Portanto, a construção de um *SBC* envolve basicamente três tarefas relacionadas com o conhecimento: a representação, a aquisição e a manipulação. Das três tarefas, a aquisição de conhecimento tem sido um dos

mais importantes alvos de pesquisa em inteligência artificial. Isto deve-se ao fato da aquisição de conhecimento ser uma tarefa difícil, dispendiosa, mas de vital importância na qualidade final dos *SBCs*. De fato, essas características são tão marcantes que fazem delas o gargalo do processo de construção dos sistemas baseados em conhecimento [Feigenbaum 81]. Apesar de Feigenbaum ter identificado esse problema no início da década de 80, o gargalo da aquisição de conhecimento continua ainda hoje, sendo considerado um grande entrave na construção dos *SBCs*, não obstante o avanço verificado no desenvolvimento de ferramentas automatizadas de *AC*, vide, por exemplo, [Gaines 93]. A importância desse problema pode ser comprovada pela realização periódica de vários eventos científicos voltados exclusivamente à área de aquisição de conhecimento, como, por exemplo, os *Workshops* realizados na Europa (os "European Knowledge Acquisition Workshop") e no Canadá (os "Knowledge Acquisition for Knowledge-Based Systems Workshop").

Ao longo desta seção falamos várias vezes em conhecimento, mas afinal qual é o seu significado? Infelizmente, não temos uma resposta conclusiva para essa pergunta, pois conhecimento é um conceito de significado amplo e varia conforme o contexto em que é usado. Nesta tese, não objetivamos proceder a uma análise das várias nuances filosóficas ou científicas do conhecimento, mas sim considerar apenas o seu aspecto pragmático e passível de ser representado em uma máquina. Assim, no contexto deste trabalho, consideraremos conhecimento como sendo simplesmente um conjunto de informações organizadas.

1.2 Métodos de aquisição de conhecimento

Vários têm sido os métodos utilizados para contornar as grandes dificuldades do processo de aquisição de conhecimento. De um modo geral, conforme mostra a FIG. 1.1, esses métodos podem ser agrupados em duas grandes famílias: métodos cognitivos e métodos automatizados [Boy 87]. Os métodos cognitivos são caracterizados pela presença intensiva do engenheiro do conhecimento, que é encarregado de extrair e formalizar o conhecimento do(s) especialista(s). Os métodos automatizados são aqueles onde a máquina consegue obter, do ambiente que a envolve, o conhecimento necessário ao funcionamento do *SBC*.

1.2.1 Métodos cognitivos

Os métodos cognitivos têm como eixo básico o processo de "entrevistas" entre o engenheiro do conhecimento e o(s) especialista(s), diferindo entre si apenas pela forma de como essas "entrevistas" são conduzidas. Os métodos têm grande semelhança com as técnicas de análise de sistemas tradicionais e exigem do engenheiro do conhecimento o domínio completo de tais técnicas. Além disso, é necessário que o engenheiro do conhecimento tenha um razoável conhecimento do domínio e mantenha um intenso relacionamento com o(s) especialista(s). Devido a essas exigências e ao fato de que especialistas de alto nível são raros e conseqüentemente caros, o processo de \mathcal{AC} baseado em métodos cognitivos é bastante lento e dispendioso. De fato, a experiência mostra que, mesmo quando as exigências acima são satisfeitas, um engenheiro do conhecimento consegue acrescentar por dia apenas de uma a quatro regras na \mathcal{BC} [Quinlan 86] e [Shapiro 87]. Mesmo com o uso de técnicas específicas de condução de entrevistas, essa média pouco tem se alterado [Gaines 93].

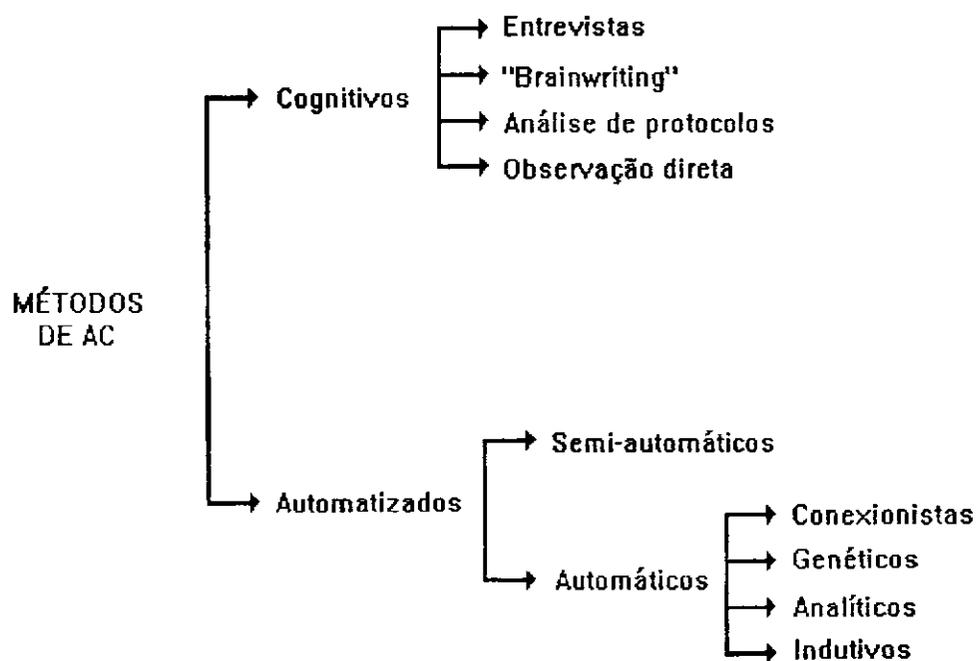


FIGURA 1.1 - Uma classificação dos métodos de \mathcal{AC}

Conforme mostra a FIG. 1.1, os métodos cognitivos podem ser divididos em quatro grandes classes: entrevistas, "*brainwriting*", análise de protocolos e observação direta [Boy 87].

As **entrevistas** são sessões nas quais o especialista é inquirido pelo engenheiro do conhecimento acerca do domínio em questão. As entrevistas, em geral, são pouco produtivas devido a razões como:

i) O vocabulário usado pelo especialista nas suas explicações geralmente é diferente do usado pelo engenheiro do conhecimento.

ii) Fatores humanos como humor, empatia e hostilidade interferem no processo de aquisição.

iii) Inexistência de uma metodologia "padrão" para conduzir as entrevistas dificulta o processo.

iv) Dificuldades encontradas pelo especialista para enunciar seu conhecimento. Estas dificuldades aparecem porque o conhecimento é compilado (o longo processo de aprendizagem faz com que o conhecimento se re-estruture de forma sucinta ao longo do tempo), subjetivo (cada especialista possui uma visão particular do domínio), volátil (uma afirmação que hoje é aproveitada como conhecimento, amanhã pode deixar de ser aproveitada total ou parcialmente) e formado por várias partes (diversos especialistas detêm, cada um, uma parte do conhecimento).

Vale salientar que, apesar das muitas limitações, as entrevistas continuam sendo o método mais utilizado, até pela falta de alternativas significativamente superiores.

"Brainwriting" é um método que permite obter conhecimento de um grupo de especialistas. Ele consiste basicamente na definição de regras para discussão entre os especialistas e propõe a substituição da comunicação oral pela escrita [Warfield 71]. As principais vantagens desse método são:

i) A produtividade do grupo aumenta, devido ao trabalho em paralelo.

ii) O domínio de personalidades fortes é eliminado.

iii) O silêncio e a ausência de crítica verbal propiciam uma reflexão franca e uma atmosfera criativa.

iv) As idéias minoritárias não são desprezadas e a responsabilidade pelo trabalho é dividida pelo grupo.

v) O ato de escrever permite evidenciar os mecanismos de análise e resolução de problemas.

Bons resultados têm sido obtidos com esta abordagem, porém ela ainda é demasiada sujeita a fatores humanos, além de ser bastante onerosa.

A **análise de protocolos** consiste na minuciosa observação do especialista em ação na solução de problemas reais. A partir daí tenta-se formalizar o processo que levou o especialista à solução. A análise de protocolos pode ser retrospectiva (quando o especialista é filmado resolvendo o problema e, posteriormente, é requisitado a explicar seu comportamento) ou concorrente (quando pede-se ao especialista para "pensar em voz alta") [Varejão 91]. O principal problema desta abordagem é que o conhecimento assim obtido tende a ser pouco geral.

Os métodos de **observação direta** são generalizações dos métodos anteriores. A aquisição é feita a partir da observação de documentos e manuais, entrevistas, estudo de ações e tarefas efetuadas pelo(s) especialista(s), etc. Devido a sua generalidade, bons resultados já foram conseguidos com este método [Boy 86], a custo, porém, de uma baixa produtividade.

Note que a \mathcal{AC} cognitiva não exclui totalmente o computador. Sistemas de edição e teste de consistência da \mathcal{BC} tais como o TEIRESIAS [Davis 82] e KREME [Abrett 89] têm sido propostos e utilizados proporcionando ganhos de produtividade para o engenheiro do conhecimento. Além do mais, para vários domínios, como por exemplo a área jurídica, os métodos cognitivos ainda são a única forma de \mathcal{AC} .

1. 2. 2 Métodos automatizados

Conforme mostra a FIG. 1.1, John Boose [Boose 90] divide os métodos automatizados em baseados em entrevistas (semi-automáticos) e baseados em aprendizagem automática (automáticos).

1. 2. 2. 1 Métodos semi-automáticos

Para contornar as dificuldades dos métodos cognitivos, principalmente a presença intensiva da figura eclética do engenheiro do conhecimento no processo, em meados da década de 80 surgiram os primeiros sistemas semi-automáticos de \mathcal{AC} . Nesses sistemas, as entrevistas com o(s) especialista(s) são realizadas diretamente pela máquina, "dispensando" assim a presença do engenheiro do conhecimento.

O primeiro sistema de peso considerado semi-automático foi o ETS ("Expertise Transfer System"). Foi desenvolvido por John Boose [Boose 84] e é baseado na teoria da psicologia das construções pessoais ("The Psychology Personal Construct Theory") [Kelly 55]. Este sistema, embora limitado, alcançou um grande sucesso na área de \mathcal{AC} , fazendo com que evoluísse para o ambiente de \mathcal{AC} AQUINAS [Boose 87]. O AQUINAS, além das facilidades básicas de um ambiente de \mathcal{AC} (eliciação, análise, teste e validação de uma \mathcal{BC}) ainda possui: um gerenciador de diálogos, que orienta o usuário ao longo de todo o ambiente; facilidades que permitem lidar com o conhecimento de uma forma estruturada e um módulo de combinação de conhecimento de vários especialistas.

Paralelamente ao AQUINAS, e dentro da mesma filosofia de ambientes, vários outros sistemas surgiram. Os mais representativos são: MOLE [Eshelman 88], KRITON [Diederich 88], KITTEN [Shaw 88] e o SALT [Marcus 87]. Embora em concepção sejam semelhantes, esses sistemas diferem entre si em vários aspectos. Os principais são: estratégias de condução das entrevistas, forma de quantificar e analisar os conceitos obtidos, tipos de problemas que resolvem e fontes de conhecimento auxiliar utilizadas.

Apesar do inegável avanço propiciado pelos sistemas semi-automáticos, aliviando o gargalo da aquisição de conhecimento, eles ainda são muito

limitados e, por que não dizer, rudimentares. As principais razões são: a) dependendo da complexidade do domínio, as entrevistas podem ser bastante longas e cansativas; b) em geral, esses sistemas só fornecem regras atômicas (regras com apenas uma condição), não capturando o fato de que certas conclusões são necessariamente derivadas de duas ou mais condições conjuntamente; c) são centrados fortemente no(s) especialista(s), fazendo pouco ou nenhum uso da experiência passada (informação catalogada); d) a maioria só dispõe de facilidades de \mathcal{AC} para resolver problemas de análise (diagnóstico, seleção, interpretação, etc.). Os poucos que permitem \mathcal{AC} para problemas de síntese (configuração, projeto, planejamento), o fazem de uma forma pouco convincente e para tipos específicos de problemas, como por exemplo o SALT, que aborda problemas de configuração.

1. 2. 2. 2 Métodos automáticos

Um passo adiante na automatização do processo de \mathcal{AC} leva à área do aprendizado por máquina ou aprendizagem automática (\mathcal{AA}) ("Machine Learning") [Michalski 87]. Esta área se preocupa com todas as formas pelas quais uma máquina pode captar conhecimentos do ambiente. Obviamente, quando este conhecimento obtido pela máquina puder ser apresentado como uma \mathcal{BC} , temos um método automático de aquisição de conhecimento (ou seja, uma técnica de \mathcal{AA} usada na \mathcal{AC}) [Cirne 91b]. A FIG. 1.2 mostra a relação entre aprendizagem automática e aquisição de conhecimento.

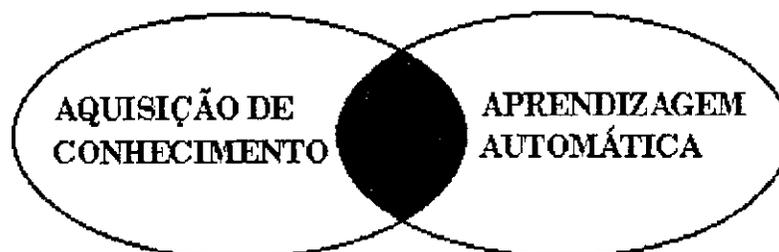


FIGURA 1.2 -- Relação entre \mathcal{AA} e \mathcal{AC}

Uma série de taxionomias definem \mathcal{AA} . A grande quantidade de classificações pode dar uma falsa impressão de crescimento da área. Mas, na realidade, o que existe é uma tumultuada e confusa coleção de definições.

Carbonell divide a aprendizagem automática em quatro paradigmas [Carbonell 89]: Conexionista (modelos baseados em redes neurais), Genético (sistemas classificadores), Analítico (aprendizado baseado em explicações e certas formas de analogias) e Indutivo (adquirir conceitos de conjuntos de exemplos e contra-exemplos) (veja FIG.1.1). Estes paradigmas surgem de diferentes raízes científicas, aplicando diferentes modelos computacionais e, muitas vezes, considerando de maneira diferente o que vem a ser um resultado bem sucedido. Mesmo assim, todos compartilham o mesmo objetivo de construir máquinas que aprendam para uma rica variedade de domínios. Uma descrição mais detalhada desses paradigmas é apresentada no capítulo 2.

Dos quatro paradigmas citados, o mais amplamente estudado e aplicado a sistemas baseados em conhecimento é o Indutivo. Neste paradigma, procura-se induzir uma descrição geral de um certo conceito (classe) a partir de um conjunto de exemplos e contra-exemplos. Este conjunto é conhecido como conjunto de treinamento ("training set"). A tarefa é construir uma descrição geral do conceito que explique (i.e., possa inferir por dedução e instanciação) todos os exemplos. Na verdade, este é um tipo especial de indução, conhecida como indução a partir de exemplos.

Restringiremos o escopo deste trabalho aos métodos automáticos indutivos a partir de exemplos, devido à sua maior aplicabilidade à aquisição de conhecimento para sistemas baseados em conhecimento. Portanto, a partir de agora, neste capítulo, toda referência ao aprendizado automático indutivo far-se-á aos métodos baseados em exemplos, mais especificamente aos orientados a dados (generalização orientada a dados - vide capítulo 2, FIG. 2.6).

Quando comparados com os métodos cognitivos e semi-automáticos, podemos relacionar como vantagens dos métodos automáticos indutivos:

v₁) Minimizar a presença do especialista, que é difícil de ser conseguida e quase sempre muito cara. Neste processo de aquisição de conhecimento, o trabalho é repartido: deixando-se uma pequena, mas significativa, parte, que é o aspecto qualitativo, com o especialista e o restante, a "força bruta", com a máquina.

v₂) Em alguns domínios é mais fácil adquirir conhecimento a partir de casos reais acontecidos (exemplos). Além do mais, o especialista pode achar mais

fácil relacionar casos de um domínio do que seguir o seu processo mental em uma resolução.

v₃) Ter o poder de captar conhecimento através de generalizações .

v₄) Produzir bases de conhecimento coerentes. Isto é, BCs livres de inconsistências, circularidades, etc.

Quanto aos pontos negativos dos métodos automáticos indutivos, podemos dizer que:

n₁) Há uma suposição básica de que as instâncias ou exemplos (normalmente representados por um conjunto de pares atributo/valor e uma classe), que servem como entrada para o algoritmo, são conhecidos de antemão. A prática tem mostrado que esta suposição não é fácil de ser satisfeita. A identificação dos atributos que formam os exemplos e a conseqüente construção do conjunto de treinamento também tem se mostrado um ponto de estrangulamento no processo de aquisição.

n₂) Há uma série de restrições e limitações, embora distintas de algoritmo para algoritmo, que têm que ser cumpridas na modelagem do mundo real para que possam ser aplicados aos algoritmos generalizadores. Estas exigências vão desde problemas de eficiência computacional a exigências conceituais pertinentes ao algoritmo.

n₃) A maioria dos algoritmos de indução assume, pura e simplesmente, que é conhecido um subconjunto de descrições e pretende-se, a partir delas, induzir a relação existente entre o conjunto das classes e o espaço de codificação. Essa atividade resulta de não existir qualquer modelo associado, nem ao processo de obtenção do conjunto de treinamento, nem às descrições desconhecidas.

n₄) A maioria desses métodos são puramente sintáticos: não usam informações semânticas sobre o domínio. Os algoritmos de indução usuais generalizam a partir de dados, sem referência ao conhecimento preliminar ("background knowledge") [Nuñez 91]. O conhecimento preliminar pode ser eliciado do especialista de várias formas diferentes, como relevância semântica [Mongiovi 90a], importância relativa dos exemplos [Pires 93], grafos de hierarquia IS-A e informações relativas ao custo dos atributos [Nuñez 91]. Ele contém informações relativas ao domínio, mas não

encontradas entre os dados. Sua utilidade está em facilitar o trabalho do algoritmo de indução e melhorar a qualidade do conhecimento induzido. A ausência de conhecimento preliminar fica especialmente evidente quando há casos raros no conjunto de treinamento ou quando se trabalha com conjuntos de treinamento pequenos, o que pode fazer com que coincidências sejam generalizadas como conhecimento [Cirne 91a].

n₅) Pelo fato de usarem pouco ou nenhum conhecimento preliminar relacionado com a semântica do domínio, os métodos indutivos não permitem automatizar o processo de avaliação qualitativa de seus resultados. Em geral, para esses métodos, apenas os aspectos quantitativos, como número de regras e comprimento médio de regras, e de acurácia têm sido observados.

n₆) Em geral os algoritmos indutivos não têm tratado, ou têm tratado apenas de uma maneira insatisfatória o problema do ruído no conjunto de treinamento, ruído esse que pode se apresentar na forma de "buracos", incerteza nas medidas dos atributos contínuos e exemplos ambíguos.

n₇) Um ponto negativo específico dos métodos indutivos, que geram árvores de decisão, é que a saída gerada pode ser pouco inteligível, pois árvores de decisão normalmente provocam a introdução de condições artificiais às conclusões [Cendrowska 88].

Poder-se-ia argumentar: se os métodos indutivos apresentam "poucas" vantagens e "muitas" desvantagens, por que utilizá-los? A resposta é simples: a **essência** dos métodos é muito boa e, se inteligentemente tratada, explorando-se mais o intelecto do especialista e menos a sua "força física", possibilitará a geração automática de bases de conhecimento coerentes e críveis. Portanto, qualquer esforço no sentido de melhorar a qualidade dos resultados dos métodos indutivos é compensador, fazendo com que o "pouco" (vantagens) fique cada vez maior e o "muito" (pontos negativos) cada vez menor.

Vários trabalhos têm sido apresentados para contornar os fatores negativos dos métodos indutivos. Entretanto, as abordagens têm se dado de uma forma focada em pontos específicos. Por exemplo, para diminuir a negatividade do ponto (n₁) Mizoguchi [Mizoguchi 91] propõe um método misto de aquisição de conhecimento, usa um método indutivo para gerar árvores de decisão (método automático) e um processo de entrevistas (método semi-automático) para eliciar, de uma forma gradual, os atributos do domínio. Pires [Pires 93],

contorna os pontos (n_3) e (n_6) propondo novos algoritmos de indução de árvores de decisão baseados na teoria das evidências de Dempster-Shafer, nas técnicas de segmentação e observando a cardinalidade do conjunto de treinamento completo e a frequência relativa dos exemplos. Para contornar o ponto negativo (n_4), Nuñez [Nuñez 91] e Mongiovi [Mongiovi 90a] utilizam conhecimento preliminar, o primeiro na forma de custo e generalização de valores de atributos, enquanto que o segundo na forma de relevância semântica. Já para o ponto (n_7) temos o excelente trabalho da Cendrowska [Cendrowska 88], que gera um conjunto de regras modulares em lugar de árvores de decisão, e os trabalhos de Quinlan [Quinlan 87] e Mingers [Mingers 89b] que propõem técnicas de simplificação de árvores de decisão.

Embora esses trabalhos tenham produzido bons resultados, eles são apenas soluções parciais para o problema como um todo. Na prática, entretanto, necessitamos de uma solução mais abrangente que enderece ao mesmo tempo, senão todos, pelo menos a maioria desses fatores negativos.

Esta tese de doutorado busca apresentar uma solução com tal abrangência.

1.3. Objetivos da tese

Devido à lentidão do processo de aquisição de conhecimento, vários métodos automáticos para esta tarefa foram propostos. Dentre eles, há aqueles que funcionam fazendo indução por computador, nos quais se destacam os da família *TDIDT* (Top Down Induction of Decision Trees) [Quinlan 86].

A meta principal dos algoritmos *TDIDT* é mapear um conjunto de exemplos em uma árvore de decisão de tamanho mínimo (altura e largura). Obviamente que, do ponto de vista puramente quantitativo, este objetivo é válido, pois implica em uma base de conhecimento reduzida. Entretanto, em geral, para serem utilizados em *SBCs*, os resultados obtidos pelos algoritmos *TDIDT* apresentam dois problemas críticos: um sintático e outro semântico.

Os algoritmos *TDIDT*, por definição, geram sua saída na forma de uma árvore de decisão. As árvores de decisão constituem um problema porque não equivalem a um conjunto de regras modulares, no sentido que sempre há um atributo presente em todas as regras. Isto permite, principalmente se existirem atributos com um grande número de valores, o aparecimento de

anomalias, pois nem todos os valores daquele atributo são relevantes para a conclusão dos elementos de classificação. Isto caracteriza o problema sintático.

A árvore de decisão gerada pelos algoritmos da família *TDIDT* é uma estrutura despida de informação semântica. Ela apenas mapeia, de forma reduzida, o conjunto de exemplos. Devido a este fato, pode ocorrer que condições irrelevantes apareçam na árvore. Isto ocorre porque, em aplicações reais, raramente trabalhamos com um conjunto completo de exemplos (i. e., um conjunto que cubra todas as combinações possíveis de condição x classe), o que permite o aparecimento de condições irrelevantes por simples coincidência. Uma circunstância em que este problema freqüentemente aparece é quando o conjunto de exemplos contém casos raros (casos que denotam classes que ocorrem apenas em pouquíssimos exemplos). Este é o problema semântico. Vale salientar que esse problema não é exclusivo dos algoritmos *TDIDT*, dele também sofrem os algoritmos indutivos que geram bases de conhecimento diretamente na forma de regras de produção (vide capítulo 5). Isso porque, em ambos os casos, os algoritmos indutivos não fazem uma distinção entre condições relevantes e irrelevantes.

O problema sintático foi apresentado por Cendrowska [Cendrowska 88] e o semântico foi identificado durante a pesquisa que viabilizou esta tese [Mongiovi 90a] e [Mongiovi 93a]. Na verdade, a questão da relevância, cuja não observância é a causa maior da manifestação do problema semântico, já havia sido abordada por Irving M. Copi [Copi 78]. Discorrendo sobre analogia, Copi propõe seis critérios importantes para a apreciação de um argumento analógico. No sexto critério, ele afirma textualmente:

*"A questão da relevância é de suprema importância. Um argumento baseado numa só análoga **relevante**, referente a um só exemplo, será muito mais convincente do que um outro argumento que assinale uma dúzia de pontos de semelhança **irrelevante** entre o exemplo da conclusão e uma porção de exemplos enumerados nas premissas."*

Além dos problemas sintático e semântico, a grande variedade de algoritmos generalizadores surgidos nos últimos anos nos leva às seguintes questões:

- a) Para um dado problema qual algoritmo utilizar?
- b) Como comparar ou mesmo fundir os resultados de dois ou mais algoritmos?

- c) O algoritmo embute algum tipo de simplificação ? Se não embute qual técnica de simplificação utilizar?
- d) Que tipo de restrições esses algoritmos impõem ao conjunto de treinamento?
- e) Usam algum tipo de conhecimento preliminar ? Se usam como obtê-lo?
- f) Como e em que modelar o domínio para um dado algoritmo?
- g) Como avaliar a qualidade semântica de uma base de conhecimento? Isto é, como medir o nível qualitativo de uma *BC* ?

Na tentativa de responder às questões acima e visando uma solução global para contornar a maioria dos pontos negativos dos métodos indutivos, com esta tese esperamos alcançar os seguintes objetivos:

Objetivos Principais

Estes objetivos referem-se ao aspecto semântico dos métodos indutivos de aquisição de conhecimento e constituem-se no foco principal da nossa tese. São eles:

- op₁) Propor um tipo de conhecimento preliminar, e uma forma de representá-lo, que relacione a relevância semântica entre os atributos e as classes de um dado domínio;
- op₂) Estudar o problema semântico, sugerindo alternativas que permitam contorná-lo;
- op₃) Propor uma solução para resolver simultaneamente ambos os problemas (sintático e semântico);
- op₄) Propor uma solução que contemple ao mesmo tempo os aspectos sintático, semântico de custo e generalização;
- op₅) Propor uma forma de avaliar a qualidade semântica de uma base de conhecimento;

Objetivos Secundários

Neste conjunto temos dois tipos de objetivos: um relacionado com os aspectos operacionais dos algoritmos indutivos e o outro referente ao uso da relevância semântica em outros métodos de aquisição automática de conhecimento. Embora os tenhamos chamados de secundários, pois não constituem o tema central do nosso trabalho, a realização desses objetivos é de grande importância para a nossa tese: os do primeiro tipo (aspecto operacional) porque permitem materializar as idéias para alcançar os objetivos principais, e os do segundo tipo (referentes à relevância semântica) porque reforçam a importância do uso de relevância semântica no processo de aquisição de conhecimento, independentemente do método a ser usado. Estes objetivos são:

- os₁) Desenvolver um ambiente de apoio à aquisição automática de conhecimento para os métodos indutivos;
- os₂) Apresentar uma forma automática de auxílio na modelagem do domínio para os métodos indutivos;
- os₃) Estudar a viabilidade de usar a relevância semântica para contornar o problema do grande esforço computacional apresentado por alguns métodos de aquisição de conhecimento, tais como:
 - Modelo Neural Combinatório [Machado 90], que gera regras a partir do treinamento de um tipo particular de rede neural ("backpropagation");
 - Algoritmo SDL (Stochastic Decision Lists) [Gomes 92] e [Gomes 93], que utiliza algoritmos estocásticos para gerar listas de decisão de complexidade limitada.

1.4 Contribuições para a área de aquisição de conhecimento

O processo de aquisição indutiva de conhecimento, a partir de exemplos, pode ser pensado como sendo composto de três fases: tratamento dos dados de entrada, algoritmos indutivos e tratamento das saídas.

Na primeira fase, lidamos com os mecanismos de preparação dos dados de entrada, obtendo a tabela de exemplos e as diversas formas de conhecimento preliminar. Para a segunda fase, devemos nos preocupar com a natureza intrínseca dos algoritmos indutivos, observando os seguintes aspectos: complexidade computacional, tratamento de ruídos, restrições impostas aos dados de entrada, uso de **conhecimento preliminar**, incrementabilidade e tipos de saídas geradas. Finalmente, na última fase devemos considerar, se for o caso, a possibilidade de utilizar técnicas de simplificação e mecanismos de composição de resultados advindos de vários algoritmos. Além disso, faz-se necessária uma análise das saídas geradas, onde deverão ser observados os aspectos tamanho, acurácia e nível de qualidade semântica.

Com esta tese, contribuimos para a área de aquisição de conhecimento nas três fases do processo acima mencionado, com ênfase maior para a segunda. Especificamente temos:

Primeira fase - contribuimos com um novo ambiente de apoio à aquisição automática de conhecimento, visando facilitar a operacionalização do processo como um todo, em particular auxiliar o usuário na modelagem do domínio (tabela de exemplos + conhecimento preliminar);

Segunda fase - contribuimos com a identificação do problema semântico e a proposição de um tipo de conhecimento preliminar, baseado nas relações de relevância semântica entre atributos e classes, tendo sido esse conhecimento utilizado para:

- a) propor soluções alternativas para o problema semântico;
- b) integrá-lo com outros tipos de conhecimento preliminar (custo e generalização), visando um novo algoritmo indutivo que considere ao mesmo tempo os aspectos sintático, semântico, de custo e generalização;
- c) podar o espaço de busca de métodos de aquisição de conhecimento que apresentam o problema da explosão combinatória;

Terceira fase - contribuimos com a definição de um processo de avaliação da qualidade semântica de uma base de conhecimento.

Como resultado deste trabalho tivemos a conclusão, sob nossa orientação, de quatro dissertações de mestrado [Cirne 92], [Vasco 93d], [Donato 94] e

[Alexandre 94b], uma tese de nossa autoria aprovada em concurso público para professor Titular da UFPB [Mongioli 93a] e 20 trabalhos, diretamente relacionados com a tese, publicados em eventos científicos da área, conforme mostra a TAB. 1.1. Além disso, orientamos a dissertação de mestrado [Gomes 89], que deu origem à esta linha de pesquisa.

TABELA 1.1 - Referências X objetivos da tese

Referência	Objetivos							
	principais					secundários		
	op1	op2	op3	op4	op5	os1	os2	os3
[Mongioli 90a]	F	F						R
[Cirne 90]	F	F						
[Oliveira 90]						P		
[Mongioli 91]	F	F						
[Cirne 91a]	F	F	F					
[Cirne 91b]	P	P	P					
[Vasco 92a]	P	P	P			F	R	
[Vasco 92b]	P	P	P			R	F	
[Cirne 92]*	F	F	F					
[Mongioli 93a]*	F	F	F			F	F	F
[Mongioli 93b]	R	F	F	F	F		R	
[Mongioli 93c]					F			
[Donato 93]								F
[Oliveira 93]						P		
[Vasco 93a]						R	F	
[Vasco 93b]						F	R	
[Vasco 93c]						F	R	
[Vasco 93d]*						F	F	
[Donato 94]*								F
[Oliveira 94]						P		
[Bezerra 94a]		P						
[Bezerra 94b]		P						
[Alexandre 94a]				F	F			
[Alexandre 94b]*				F	F			
[Alexandre 94c]				F				

* - tese/dissertação de mestrado

R - Referência razoavelmente relacionada com o objetivo

F - Referência fortemente relacionada com o objetivo

P - Referência um pouco relacionada com o objetivo

Forma operacional escolhida

No nosso entender, a meta principal de uma tese de doutorado consiste em identificar e resolver um problema não trivial e que traga alguma contribuição à área em que está inserido, sendo o seu desenvolvimento um veículo que traga para o doutorando, independência e capacidade gerencial (ou de orientação) em atividades de pesquisa avançada. Atividades de pesquisa essas, que levem preferencialmente à solução de problemas reais e que, assim, tragam benefícios à sociedade, que, direta ou indiretamente, é quem arca com o ônus financeiro dessas atividades. O doutorando pode realizar esse trabalho de uma forma isolada ou pode dividir o problema em subproblemas, de menor complexidade, resolvê-los com auxílio de alunos de mestrado e, posteriormente, aglutiná-los para resolver o problema como um todo.

Na forma isolada, o doutorando demonstra a sua independência ao conseguir apresentar soluções para o problema proposto, que em geral são soluções focadas e freqüentemente sem experimentação. A vantagem desta forma é que o mérito dos resultados é quase que exclusivamente do doutorando. Já na segunda forma, de caráter compartilhado, o doutorando demonstra a sua independência de pesquisa ao conseguir dividir o problema em subproblemas, gerenciar (orientar) as soluções desses subproblemas, compatibilizando as interfaces, de forma que, no final, as soluções das partes formem a solução geral do problema. A vantagem desta segunda forma é que, em geral, o doutorando pode investigar não só o foco central do problema, mas também observar as várias nuances periféricas, produzindo, assim, um trabalho mais amplo sem sacrificar a profundidade. Evidentemente, que, neste caso, o mérito dos resultados também fica compartilhado.

Por uma questão contextual, necessidade de orientar dissertações de mestrado do curso de Pós-Graduação em Informática da UFPB e, por que não dizer, ser esta uma forma mais agradável de trabalho, escolhemos a segunda forma, i. e. a compartilhada. A TAB. 1.2 mostra uma evolução cronológica do nosso trabalho, descrevendo os subproblemas e indicando as dissertações de mestrado, sob nossa orientação, diretamente com eles envolvidas.

TABELA 1.2 - Evolução cronológica das dissertações de mestrado envolvidas na solução dos subproblemas deste trabalho

ANO	SUBPROBLEMA	DISSERTAÇÃO
89	Um sistema de aquisição automática de conhecimento a partir de exemplos (APREND)	[Gomes 89]
90 e 91	. Identificação do problema semântico . Novo tipo de conhecimento preliminar (MR) . Soluções para o problema semântico via MR (algoritmos ID3X, ADEX e IDRT) . Solução dos problemas sintático/semântico (algoritmo RPRISM)	[Cirne 92]
92	Definição e construção da MRN (algoritmos IDRT e FRPRISM)	[Mongioli 93a]*
92 e 93	. Ambiente de apoio à aquisição automática de conhecimento (A4) . Modelagem do domínio (Tab. exemplos, custo e hierarquia)	[Vasco 93d]
93 e 94	Uso da MR no Modelo Neural Combinatório (topologias TCROS e TCROSES)	[Donato 94]
93 e 94	. Uso da MRN na avaliação semântica de uma base de conhecimento . Algoritmo indutivo, abrangendo os aspectos sintático, semântico, de custo e generalização	[Alexandre 94d]

* - Tese de Prof. Titular

1.5 Importância da tese

A área de inteligência artificial experimentou nas últimas duas décadas uma grande evolução, consolidando-se como uma das grandes áreas da informática. Os sistemas especialistas são uma significativa parte desse contexto e, sem sombra de dúvida, uns dos grandes responsáveis pela credibilidade que hoje se tem na IA. Isso deve-se ao fato dos *SEs* terem saído do laboratório para o campo, para resolver problemas do mundo real.

Além de lidar e solucionar problemas do mundo real, uma característica marcante dos sistemas especialistas é que eles conseguem justificar suas ações e resultados, propiciando, assim, uma grande confiança a seus

usuários. Evidentemente, que para um sistema especialista realizar a contento essa tarefa de solucionar justificando, é necessário que a sua base de conhecimento espelhe fielmente o conhecimento do(s) especialista(s) do domínio. Portanto, é imperativo o desenvolvimento de técnicas e ferramentas que, com um mínimo de esforço humano, possibilitem a construção de bases de conhecimento fidedignas, sem ambigüidades, sem contradições e de conteúdo claro e convincente. Contribuir para a materialização desse objetivo é o "Para que" da nossa tese.

1.6 Estrutura da tese

Embora a tese esteja estruturada fisicamente em doze capítulos, logicamente pode ser vista como composta de quatro partes. Na primeira parte, capítulos 1 e 2, localizamos o nosso trabalho no contexto da aquisição de conhecimento, mais especificamente na área da aprendizagem automática. Na segunda parte, capítulos de 3 a 8, que representa a essência do nosso trabalho, descrevemos como foram alcançados os objetivos principais propostos para esta tese. Na terceira parte, capítulos de 9 a 11, tratamos dos objetivos secundários, que, apesar da denominação, os consideramos de grande importância na validação das idéias descritas na segunda parte e, portanto, entendemos que esses capítulos devem também compor o corpo da tese. Na última parte, capítulo 12, apresentamos as conclusões e propostas para trabalhos futuros. Mais especificamente os capítulos tratam de:

No capítulo 2, apresentamos a idéia de protocolos de aprendizagem e descrevemos os paradigmas da aprendizagem automática, em particular o paradigma indutiva. Apresentamos uma classificação para esse paradigma e descrevemos um quadro formal de aprendizagem indutiva a partir de exemplos.

No capítulo 3, descrevemos, de forma resumida, os métodos indutivos de aquisição de conhecimento, em particular os algoritmos *TDIDT*. Apresentamos uma análise crítica desses algoritmos, caracterizando os problemas semântico e sintático a eles relacionados, e mostramos uma solução para o problema sintático, que é o algoritmo PRISM.

No capítulo 4, introduziremos o conceito de matriz de relevância, estendemos esse conceito para uma estrutura bem mais genérica, a matriz de relevância nebulosa, e propomos um processo de construção dessa nova estrutura.

No capítulo 5, para resolver o problema semântico, apresentamos os algoritmos ID3X, ADEX e IDRT, que utilizam os conceitos de matriz de relevância e matriz de relevância nebulosa. Descrevemos ainda os algoritmos RPRISM e FRPRISM, que, utilizando respectivamente os conceitos de matriz de relevância e matriz de relevância nebulosa, são soluções abrangentes para resolver simultaneamente ambos os problemas (o sintático e o semântico).

No capítulo 6, apresentamos um algoritmo que contemple ao mesmo tempo os aspectos sintático, semântico, de custo e generalização.

No capítulo 7, descrevemos uma forma de avaliar a qualidade semântica de uma base de conhecimento, com e sem generalização.

No capítulo 8, considerando alguns domínios reais, apresentamos uma análise dos resultados (bases de conhecimento) gerados pelos algoritmos abordados na tese. Para uma dada base, consideramos parâmetros tais como: tamanho, acurácia, nível de qualidade, custo, presença/ausência de regras inúteis, etc..

No capítulo 9, apresentamos a definição do A4: um ambiente de apoio à aquisição automática de conhecimento para os métodos indutivos.

No capítulo 10, apresentamos uma série de diretrizes e estratégias que auxiliam na modelagem do domínio para os métodos indutivos de *AC*. Particular ênfase é dada ao processo de estruturação de um exemplo.

No capítulo 11, descrevemos como, utilizando conhecimento preliminar, na forma de matriz de relevância, é possível melhorar o aprendizado em um modelo simbólico-conexionista, cuja base é o Modelo Neural Combinatório.

Finalmente, no capítulo 12, apresentamos as conclusões e algumas diretrizes para a continuação deste trabalho.

2 Os Paradigmas da aprendizagem automática

2.1 Introdução

Como vimos no capítulo introdutório deste trabalho, a automatização do processo de aquisição de conhecimento passa pela área da aprendizagem automática ("Machine Learning"). Vimos também que a classificação proposta por Carbonell, dividindo o aprendizagem automática em quatro paradigmas (Conexionista, Genético, Analítico e Indutivo), é atualmente a mais aceita pela comunidade científica da área. Para um melhor entendimento desses paradigmas, em particular o indutivo (objeto da nossa tese), alguns conceitos precisam ser previamente esclarecidos. Por exemplo, o que é aprendizagem? Que mecanismos de inferência esses paradigmas usam? Que agentes estão envolvidos no processo de aprendizagem? Como esses agentes se relacionam? Assim, neste capítulo, inicialmente discutiremos esses conceitos e posteriormente abordaremos cada um dos quatro paradigmas citados.

2.1.1 O que é aprendizagem

Aprendizagem é um termo abrangente que tem sido usado nos mais diversos contextos, mas nem sempre com a mesma semântica. Isto, segundo Herbert Simon [Simon 83], em parte, deve-se ao fato das pessoas falarem de diferentes tipos de aprendizagem, mas a suspeita maior é que isso também reflete a grande ambigüidade da palavra aprendizagem. Por exemplo, aprendemos a tabuada (um processo de memorização), aprendemos a andar

(um processo de descoberta), aprendemos linguagens (um processo de construção e entendimento, desenvolvendo a habilidade de sintetizar). Portanto, devido ao caráter multifacetado do termo aprendizagem, a sua definição é uma tarefa tão difícil quanto definir o próprio termo Inteligência Artificial e, como era de se esperar, não vamos encontrar na literatura uma definição que seja inteiramente satisfatória. Assim, optamos por apresentar aquelas que julgamos mais representativas e que freqüentemente são citadas nos trabalhos científicos da área.

Como em outras áreas da IA, Marvin Minsky [Minsky 75] oferece uma definição de aprendizagem humana que, embora resumida, é bastante significativa:

"Aprendizagem é fazer mudanças úteis na nossa mente"

Herbert Simon [Simon 83] apresenta uma definição mais focada, capturando a essência da aprendizagem humana que é de particular importância para a aprendizagem de máquina:

"Aprendizagem representa mudanças no sistema, que é adaptativo no sentido de realizar, na próxima vez, a mesma tarefa, ou tarefas similares, de uma forma mais eficiente e eficaz"

De forma similar à de Simon, Jaime Carbonell [Carbonell 89] afirma que:

"Aprendizagem é o resultado produzido por um processo em que se adquire a habilidade de executar novas tarefas que não podiam ser executadas antes, ou então executar antigas tarefas melhor (mais rapidamente, com maior precisão, etc.)"

Finalmente, enfatizando a importância que tem a representação em um processo cognitivo, Ryszard Michalski [Michalski 86] define:

"Aprendizagem é a construção ou modificação da representação do que está sendo experimentado"

Fica claro que as definições acima revelam tendências diferentes dos autores, Minsky se preocupa com o "humano", Simon e Carbonell procuram o objetivo eficiência, já Michalski se preocupa com a forma como a aprendizagem se reflete no sistema. Assim, devido às várias interpretações que podem ser dadas ao termo aprendizagem, consideraremos que ele não deve ser visto

como um termo de significado único e fossilizado, e sim, como um termo de semântica flutuante, cuja interpretação é função do contexto, refletindo mais o "o que" e o "para que" aprender.

2. 1. 2 Protocolos de aprendizagem

Implícita ou explicitamente as definições de aprendizagem, apresentadas na seção anterior, falam na figura de um agente¹ *aprendiz*. Um outro ponto muito importante a ser considerado num modelo de aprendizagem, que é imprescindível, é a figura do agente que ensina, podendo esse agente ser um *oráculo*, cuja veracidade de seu conhecimento é inquestionável, ou um *mestre*, que assume uma postura dialogante. Assim, o processo de aprendizagem tem que ser visto como consistindo de dois ou mais agentes e um mecanismo de relacionamento entre eles. À esse mecanismo de relacionamento dá-se o nome de *protocolo de aprendizagem*. Assim, a geração de conhecimento e sua evolução podem ser vistas como fruto de um diálogo entre os agentes envolvidos.

Um dos trabalhos pioneiros na área de protocolos de aprendizagem automática é o de Valiant [Valiant 84]. Ele defende que uma máquina com capacidade de aprender deve consistir de um "protocolo de aprendizagem" em conjunto com um "procedimento de dedução", protocolo esse que define a relação entre o "professor" e o "aprendiz".

Boucheron apresenta a idéia de um ambiente mínimo de aprendizagem, composto de um aprendiz e um oráculo [Boucheron 88]. O aprendiz consulta o oráculo a respeito da solução de um problema, e este lhe envia, de forma gradativa, um conjunto de problemas similares resolvidos.

Na mesma linha de Valiant e Boucheron, Shinohara e Miyano [Shinohara 91] apresentam, na área de aquisição de conceitos, o protocolo de aprendizagem denominado EXEMPLOS. Nesse protocolo, o professor fornece ao aprendiz um conjunto de exemplos positivos ou negativos do conceito a ser aprendido. Os autores do protocolo defendem um papel mais ativo do professor, permitindo-lhe, por exemplo, poder selecionar um pequeno, mas representativo, conjunto de "bons" exemplos, que se na prática pouco

¹ Neste contexto, o termo agente significa um sistema (humano ou artificial) capaz de produzir conhecimento e justificar suas ações. Devido a essas características, um tal agente é denominado racional [Sallantin 86], [Ferneda 92] e [Nguifo 93].

melhoram a qualidade da aprendizagem, pelo menos diminui o tempo de duração do processo.

Os três protocolos, acima descritos, se caracterizam pelo relacionamento entre dois agente, o *aprendiz* e o "*ensinante*" (professor ou oráculo). Um protocolo com tais características é mostrado no ambiente de aprendizagem da FIG. 2.1. Nesse ambiente, no caso da aprendizagem automática, os papéis do *aprendiz*, "*ensinante*" e *cliente* são desempenhados respectivamente pela máquina, especialista e usuário .

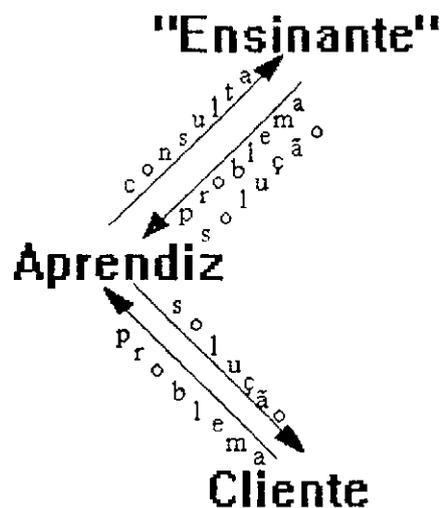


FIGURA 2.1 - Protocolo de um ambiente de aprendizagem.

O grande problema desses protocolos é que o *aprendiz* é um elemento passivo, sem poder de argumentação, limitando-se apenas a aprender o que lhe é fornecido (modelo da "escola clássica"), ficando, assim, difícil de se avaliar a qualidade da aprendizagem. A menos que se façam testes exaustivos, o que na maioria dos casos é infactível, esses protocolos de aprendizagem ficam restritos apenas à solução de problemas de análise, em particular problemas de diagnóstico e classificação.

Para contornar o problema acima, Philippe Reitz [Reitz 92] dotou o *aprendiz* de um mecanismo de argumentação, isto é, a pedido do especialista, o *aprendiz* apresenta uma justificativa para seus resultados. Assim, analisando as explicações recebidas, o especialista terá meios de julgar o conteúdo do que foi aprendido, podendo, se achar necessário, solicitar ao *aprendiz* que faça uma revisão no que até então aprendeu. Devido às contradições entre suas crenças e as do "*ensinante*", o *aprendiz* também pode

fazer espontaneamente essa revisão. Vários autores reconheceram a importância desse mecanismo de argumentação, entre eles podemos citar os trabalhos de [Fisher 91], [Boucheron 92], [Ferneda 92] e [Nguifo 93].

Em seu protocolo de aprendizagem, além do mecanismo de argumentação, Reitz faz uma distinção entre o *mestre* e o *oráculo*, e ainda introduz um agente "provocador", denominado *sonda*. Assim, esse protocolo é composto de cinco agentes com papéis distintos. São eles: *aprendiz*, *oráculo*, *sonda*, *mestre* e *cliente*. Como mostra a FIG. 2.2, o ambiente de aprendizagem MOSCA² [Reitz 92] define um protocolo de aprendizagem que faz interagir esses cinco agentes.

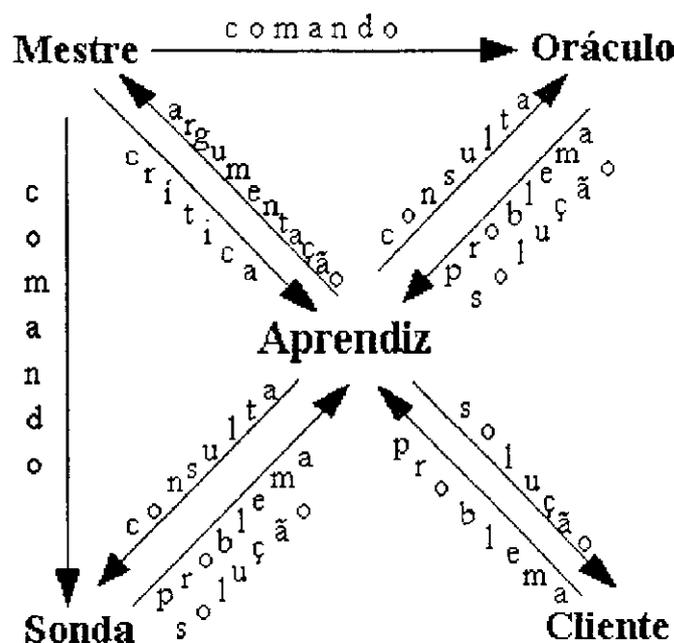


FIGURA 2.2 - O Ambiente de aprendizagem MOSCA

A seguir descreveremos, de uma maneira resumida e informal, o papel de cada um dos cinco agentes da FIG. 2.2 .

Aprendiz - constrói, de forma evolutiva, uma hipótese para explicar adequadamente um conjunto de pares <problema, solução> de um dado domínio.

Oráculo - fornece ao *aprendiz*, a pedido deste ou do *mestre*, problemas com as respectivas soluções, permitindo ao *aprendiz* construir a sua base de

² Mosca = Mestre + Oráculo + Sonda + Cliente + Aprendiz

aprendizagem. As soluções apresentadas são consideradas corretas e, conseqüentemente, irrefutáveis.

Sonda - tem um papel semelhante ao do *oráculo*, exceto que suas soluções não são necessariamente corretas, podendo, conseqüentemente, ser refutadas. Seu objetivo maior é testar o conhecimento do *aprendiz*, forçando-o a argumentar.

Mestre - dialoga com o *aprendiz* recebendo argumentação e enviando-lhe críticas. Solicita ao *oráculo* para enviar ao *aprendiz* novos pares <problema, solução>, permitindo, assim, ao *aprendiz* rever o seu conhecimento. O *mestre* também aciona o *sonda* para avaliar o conhecimento do *aprendiz*.

Cliente - simplesmente submete problemas ao *aprendiz* e aguarda as respectivas soluções.

Teoricamente, o papel de cada um dos agente da FIG. 2.2 pode ser desempenhado por um ser humano ou máquina. Na prática, entretanto, duas situações são de particular interesse:

1. Sistemas de aprendizagem automática

A máquina desempenha o papel do *aprendiz* e o ser humano os demais papéis.

2. Sistemas tutorias inteligentes

O ser humano desempenha o papel do *aprendiz* e a máquina os demais papéis.

2.1.3 Tipos de inferências lógicas

Por definição, os agentes racionais (artificiais), mencionados na seção anterior, são sistemas computacionais dotados de um mecanismo de raciocínio. Sendo a lógica simbólica um formalismo bastante poderoso de raciocínio, nada mais natural do que incorporar à esses agentes regras de inferência lógica, possibilitando, assim, automatizar o processo de raciocínio.

As três principais regras de inferência lógica, que servem como mecanismo de base para os agentes racionais, são: dedução, indução e abdução. Se para a

dedução existe uma definição bem precisa e formal, o mesmo não acontece com a indução e a abdução. A seguir, através de idéias extraídas de [Firebaugh 88] e [Nguifo 93], apresentaremos uma descrição esclarecedora desses conceitos.

Dedução

É um processo de raciocínio cuja conclusão é baseada em premissas previamente postas. Um exemplo clássico de dedução é:

Socrates é um homem, os homens são mortais, então Socrates é mortal.

De uma maneira formal em lógica das proposições temos:

se X é verdade e X implica \mathcal{Y} é verdade, então \mathcal{Y} é verdade (modus ponens), ou

$$\frac{X, X \rightarrow \mathcal{Y}}{\mathcal{Y}}$$

A dedução preserva a verdade e é uma inferência de raciocínio do geral para o particular.

Indução

É um princípio de raciocínio cuja conclusão a respeito de todos os membros de uma classe é baseada na observação de apenas alguns poucos elementos dessa classe. Por exemplo, se sabemos que:

Chico Buarque é um grande artista e tem consciência social
 Caetano Veloso é um grande artista e tem consciência social
 Jô Soares é um grande artista e tem consciência social

poderemos inferir logicamente que

"todos os grandes artistas têm consciência social"

Isto poderá ou não ser verdadeiro, no caso não é. Em lógica das proposições temos:

se sempre que X é verdade Y é verdade, então X implica Y , ou

$$\frac{X, Y}{X \rightarrow Y}$$

A indução é uma inferência de raciocínio do particular para o geral, apesar de não preservar a verdade, é um mecanismo bastante útil de generalização.

Abdução

É uma forma de lógica dedutiva que provê apenas uma "inferência plausível". Isto é, é uma heurística no sentido de que proporciona uma conclusão aceitável, consistente com as informações disponíveis, mas que na realidade pode ou não ser verdadeira. Por exemplo, vamos considerar o seguinte sistema lógico, consistindo de uma regra geral e uma proposição específica:

Todo adolescente tem problemas existenciais
João tem problemas existenciais

então, poderemos inferir que "João é um adolescente". Essa afirmação, como na indução, pode ou não ser verdadeira. Em lógica das proposições temos: se X implica Y é verdade e Y é verdade, então X é verdade, ou

$$\frac{X \rightarrow Y, Y}{X}$$

Embora não preserve a verdade, através da teoria das probabilidades e técnicas de estatística, a abdução pode fornecer a inferência mais provável entre as muitas possíveis.

2.2 O Paradigma conexionista

Consiste de modelos baseados em redes neurais artificiais. As redes neurais artificiais funcionam tentando imitar diretamente a maneira pela qual o cérebro humano trabalha. Então, nada mais natural que as redes neurais aprendam de maneira análoga ao cérebro, isto é, o aprendizado é feito através de alterações na rede [Soucek 89]. A apresentação de padrões entrada/saída, sem nenhuma preocupação em definir o processamento que transforma a entrada na saída, é suficiente para ensinar uma rede.

A FIG. 2.3 mostra um ambiente de aprendizagem para os métodos conexionistas. Nesse ambiente, o *aprendiz* aprende a resolver um problema, proposto pelo *cliente*, construindo hipóteses a partir de pares <problema, solução> (exemplos) fornecidos pelo *oráculo*. Durante a aprendizagem o usuário não tem meios de influenciar no comportamento do sistema, exceto pela escolha dos exemplos e dos parâmetros da rede, que, geralmente, são previamente fixados.

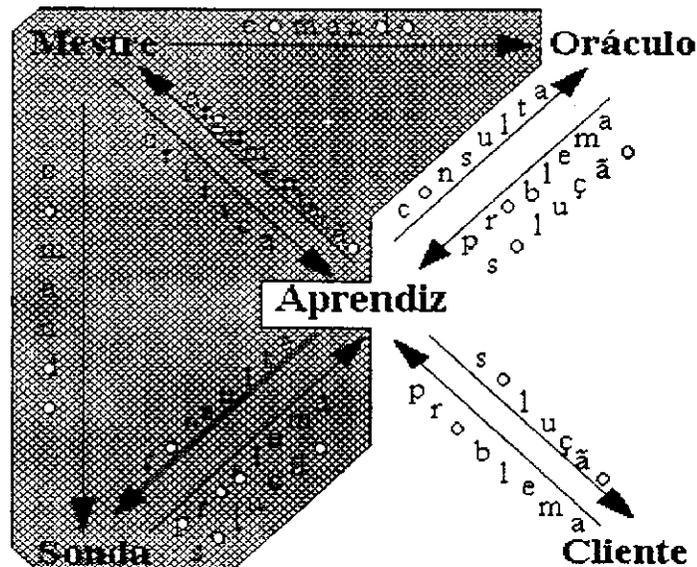


FIGURA 2.3 - Um Ambiente de aprendizagem para o paradigma conexionista

Um dos maiores inconvenientes da abordagem conexionista é que esses sistemas não conseguem explicitar a linha de raciocínio utilizada na solução de um problema. Em alguns tipos de redes neurais, onde os elementos de processamento (neurônios) representam conceitos, é possível obter uma base de conhecimento explícita [Machado 89]. Este procedimento caracteriza uma forma de aquisição automática de conhecimento.

2.3 O Paradigma genético

Este paradigma tem como eixo os algoritmos genéticos, que se baseiam na teoria da evolução e seleção natural de Darwin [Goldberg 89]. Esses algoritmos, introduzidos pelo grupo de John Holland [Holland 75], na Universidade de Michigan, partem de uma população de indivíduos para produzir uma população melhor adaptada a um dado meio. As variações das descrições de conceitos correspondem a indivíduos de uma mesma

espécie, e combinações destes indivíduos são testadas contra uma função objetivo (critério de seleção). Os algoritmos genéticos codificam uma busca paralela através do espaço de conceitos, onde cada processo tenta maximizar a função objetivo.

Neste paradigma, o ambiente é geralmente representado, através de suas entradas e saídas, por cadeias de símbolos de comprimento fixo, em um alfabeto {0, 1, #}, onde 0 indica ausência em uma posição, 1 a presença e # o desconhecido ou (0 ou 1). Essa cadeia, comumente chamada de cromossomo, serve de material genético que sofre mutações (alterações aleatórias nos valores das posições) resultantes da aplicação de operadores genéticos [Austin 90].

Neste paradigma, a qualidade dos resultados depende fortemente de um grande número inicial da população e seu uso é conveniente apenas quando o espaço de hipóteses é muito grande, não estruturado e que o custo do teste das hipóteses é relativamente baixo.

Um ambiente de aprendizagem para o paradigma genético é similar ao dos métodos conexionistas, conforme mostrado na FIG. 2.3.

2.4 O Paradigma analítico

O paradigma analítico, chamado por alguns de paradigma dedutivo, é baseado em aprendizagem a partir de poucos exemplos e um substancial conhecimento do domínio em questão. Utiliza experiência de problemas passados para guiar quais cadeias dedutivas percorrer quando da resolução de novos problemas, ou para fornecer regras de controle que tornem mais eficientes as pesquisas do domínio.

Segundo [Nguifo 93], os métodos que seguem o paradigma analítico podem ser classificados em baseados em casos (CBL: Case-Based Learning) e baseados em explicações (EBL: Explanation-Based Learning).

2.4.1 Métodos baseados em casos

Os métodos baseados em casos, ou também conhecidos como baseados em analogia, se baseiam na transferência de conceitos de um objeto (base) para outro (meta), a partir da informação que eles são semelhantes. Assim, a

solução de um problema (meta) consiste na busca das informações adequadas utilizadas na solução de outros problemas similares (bases). Obviamente, é necessário muita heurística para saber, entre outras coisas, o seguinte:

- a) qual a base mais adequada?
- b) que conceitos devem ser utilizados dentre os disponíveis?
- c) como obter as propriedades analógicas?
- d) como transferir essas propriedades entre a base e a meta?
- e) como interpretar o resultado do problema proposto à luz das propriedades definidas?

O processo de aprendizagem por analogia, em geral, é construído a partir dos três mecanismos básicos de inferência: indução, abdução e dedução. Num primeiro passo, a indução e a abdução podem ser usadas para se obter a base mais adequada (item (a)), os conceitos pertinentes (item (b)) e as propriedades analógicas (item (c)). Numa segunda fase, a dedução será usada para mapear a solução do problema análogo escolhido, a base, para a solução do problema proposto, a meta (item (d)).

De uma forma simplificada, conforme mostra a FIG. 2.4, um ambiente de aprendizagem tipo CBL se limita à uma troca de mensagens entre o *aprendiz* e o *cliente*. O *cliente* transmite um problema ao *aprendiz* e este lhe retorna uma solução, que nada mais é do que a solução de um problema análogo ao proposto.

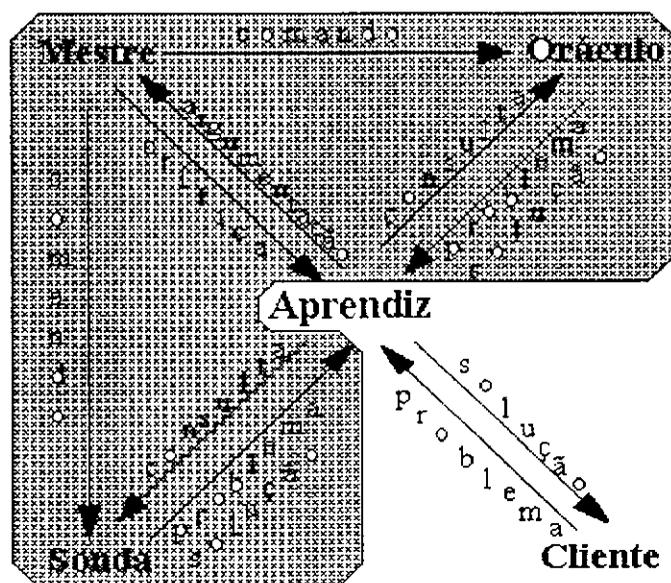


FIGURA 2.4 - Um Ambiente de aprendizagem para um CBL.

Um exemplo do uso dessa técnica, e que procura responder ao questionamento acima, pode ser encontrado em um sistema de agentes racionais proposto por [Py 92]. Py decompõe o processo de analogia em três fases: a busca da dependência entre a meta e a base (itens (a), (b) e (c)), a busca da similaridade (item (d)) e a avaliação da inferência (item (e)).

2.4.2. Métodos baseados em explicações

Os métodos baseados em explicações utilizam um tipo de aprendizagem onde a generalização deve ser justificada por explicações. Eles explicam porque um dado exemplo é um elemento de um conceito, sendo o resultado da aprendizagem a descrição de um ou mais conceitos. É um tipo de aprendizagem supervisionada, semelhante a de um professor que fornece exemplos de um novo conceito a seus alunos. A inferência é fundamentada na explicação das observações, sendo este um tipo de raciocínio dedutivo e baseado no conhecimento do domínio que está sendo tratado.

A aprendizagem EBL consiste de uma fase de explicação (utilizando o mecanismo de dedução), seguida eventualmente de uma fase de generalização (utilizando o mecanismo de indução) que permite atualizar o conhecimento até então aprendido.

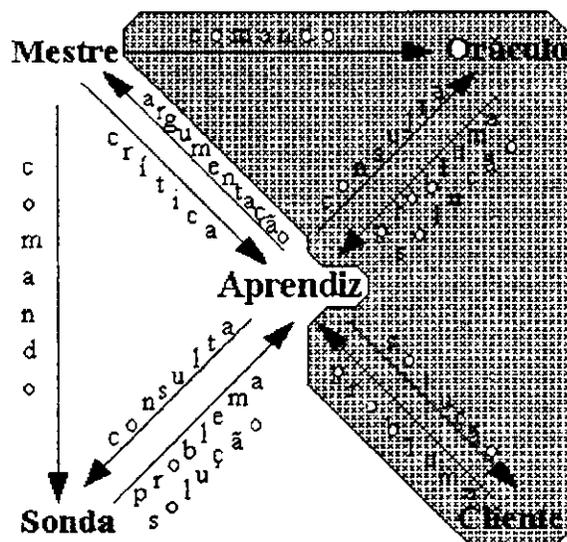


FIGURA 2.5 - Um Ambiente de aprendizagem para um EBL

A FIG. 2.5 mostra um ambiente de aprendizagem para os métodos tipo EBL. Nesse ambiente, sob o comando do *mestre*, o *sonda* envia um par <problema,

solução> ao *aprendiz*, que vai procurar uma *explicação* para a solução do problema e a transmite ao mestre. À luz das críticas recebidas do *mestre*, o *aprendiz* faz uma revisão de seu conhecimento.

Dois sistemas que usam a abordagem EBL são LEX [Mitchell 86] e PRODIGY [Minton 90]. Um dos maiores inconvenientes desses sistemas é que exigem uma definição prévia dos conceitos a serem aprendidos.

2.5 O Paradigma indutivo

A busca de padrões para uma coleção de observações aparentemente caóticas, ou a procura de uma descrição geral para um conjunto de alguns poucos fatos dispersos, tem sido um tópico de pesquisa fascinante e de grande interesse. A partir de meados da década de oitenta, o domínio dessa técnica ganhou também uma grande importância prática, visto ser ela a chave para impulsionar o processo de aquisição de conhecimento, em particular o automático. Aquisição essa que, apesar dos avanços experimentados pela área, ainda se constitui no gargalo do processo de desenvolvimento de modernos sistemas ditos inteligentes. O domínio dessa técnica é alcançado por um processo denominado de *aprendizagem indutiva*, que é um dos tópicos centrais da aprendizagem automática.

Além de possibilitar a automação do processo de aquisição de conhecimento, os métodos indutivos podem ainda ser utilizados para refinar bases de conhecimento previamente desenvolvidas por especialistas humanos, utilizando técnicas manuais ou semi-automáticas. Nessas bases, os programas indutivos podem detectar e retificar inconsistências, remover redundâncias, cobrir lacunas, ou ainda, simplificar regras (processo de generalização).

Basicamente, existem dois modos de utilizar os programas indutivos: como uma ferramenta interativa de aquisição de conhecimento a partir de exemplos, ou como parte de um sistema de aprendizagem automática. No primeiro, o usuário fornece os exemplos de treinamento e exerce um rígido controle no uso do programa. No segundo modo, um programa indutivo é um componente de um sistema integrado de aprendizagem, onde os outros componentes geram os exemplos de aprendizagem. Um exemplo desse segundo modo é o sistema de aprendizagem LEX para integração simbólica [Mitchell 86].

2.5.1 Classificação da aprendizagem indutiva

Do ponto de vista prático, tal como auxiliar na formação de uma base de conhecimento, a aprendizagem indutiva pode ser classificada em dois tipos: *aprendizagem a partir de exemplos* e *aprendizagem a partir de observações* (vide FIG. 2.6).

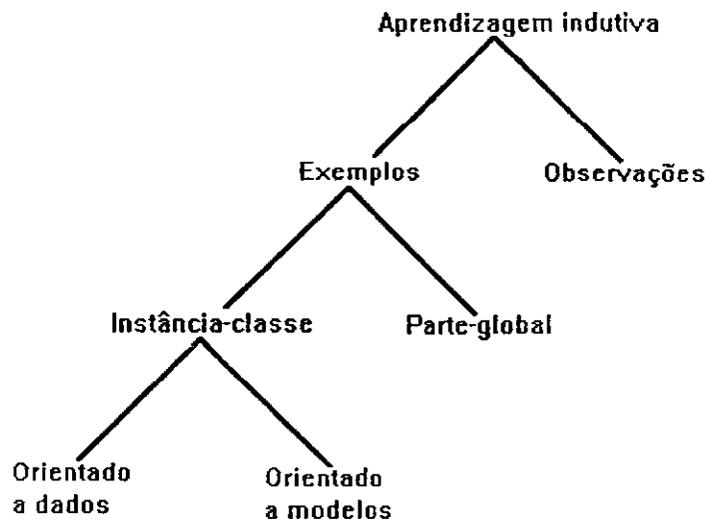


FIGURA 2.6 - Classificação dos sistemas de aprendizagem indutiva

Aprendizagem a partir de exemplos

Na *aprendizagem a partir de exemplos*, também conhecida como aquisição de conceitos, as observações (exemplos) são caracterizações de objetos (situações, processos, etc.) pré-classificados por um "professor" em uma ou mais classes (conceitos). Cada hipótese induzida pode ser vista como um conjunto de regras de reconhecimento de um conceito, tal que se um objeto satisfaz uma dessas regras, então ele representa esse conceito. Por exemplo, uma regra de reconhecimento do conceito "filósofo" poderia ser:

"um homem que medita bastante, vive aparentemente tranquilo e é indiferente aos preconceitos e convenções sociais é um filósofo"

Conforme mostra a FIG. 2.6, na aprendizagem indutiva a partir de exemplos, há dois tipos de generalização: parte-global e instância-classe.

Na generalização parte-global, o objetivo é conceituar um objeto a partir de exemplos que representam partes do objeto. Por exemplo, o sistema SPARC [Dietterich 86] se utiliza de indução parte-global para gerar regras.

Instância-classe caracteriza um sistema de aprendizagem que induz uma descrição geral de uma classe quando informamos uma coleção de instâncias (ou exemplos) daquela classe. As instâncias podem ser representações de objetos físicos, sons, imagens, ações ou sintomas, por exemplo.

A generalização instância-classe permite duas formas de se chegar às descrições gerais de um objeto: orientada a dados ("data-driven") e orientada a modelos ("model-driven") [El-Khomi 88]. Nos métodos orientados a dados, o conjunto de treinamento não possui, a priori, nenhuma descrição geral das classes envolvidas. Já nos métodos orientados a modelos, o conjunto de treinamento passou previamente por um processo de generalização. Esse processo, por exemplo, pode ser o fornecimento de regras obtidas de uma forma cognitiva. O programa INDUCE [Dietterich 81] se utiliza de um método orientado a modelos para generalizar regras.

Aprendizagem a partir de observações

Na *aprendizagem a partir de observações*, conhecida como generalização descritiva ("*descriptive generalization*"), a meta é determinar uma descrição geral (uma lei) caracterizando uma coleção de observações, ou seja, produzir descrições especificando as propriedades dos objetos pertencendo à uma certa classe. Por exemplo, observando-se que os grandes pintores renascentistas Leonardo da Vinci, Boticelli e Michelangelo eram italianos e que o pintor renascentista René d'Anjou era francês, poderemos concluir que:

"A maioria dos grandes pintores renascentistas eram italianos"

Assim, em contraste com a aquisição de conceitos, que induz hipóteses para classificar objetos em classes, a generalização descritiva produz descrições que especificam propriedades de objetos pertencendo a uma certa classe.

Restringiremos o escopo deste trabalho aos métodos automáticos indutivos a partir de exemplos, devido à sua maior aplicabilidade à aquisição de conhecimento para sistemas baseados em conhecimento. Portanto, a partir de agora, toda referência à aprendizagem automática indutiva far-se-á

especificamente aos métodos baseados em exemplos (generalização orientada a dados).

A FIG. 2.7 mostra um ambiente de aprendizagem para os métodos indutivos. Nesse ambiente, o *oráculo* (especialista e/ou engenheiro do conhecimento) envia ao *aprendiz* um conjunto de pares <problema, solução> e este tenta induzir uma hipótese capaz de explicar esse conjunto de pares.

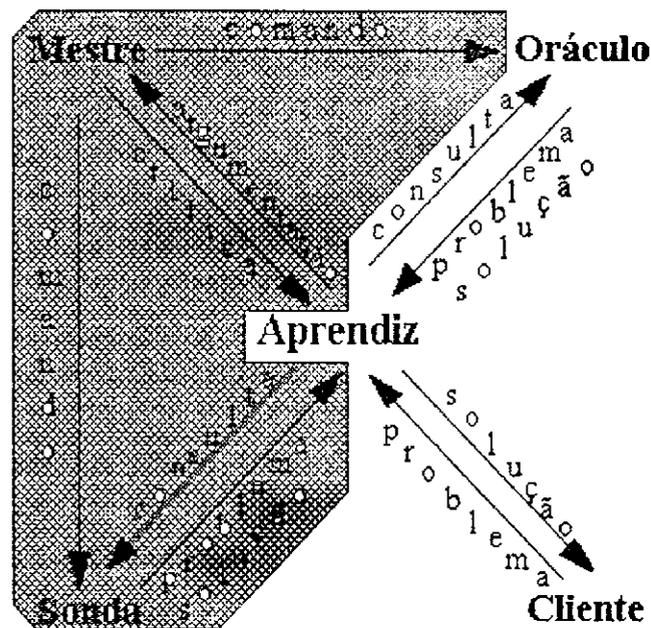


FIGURA 2.7 - Um Ambiente de aprendizagem para o paradigma indutivo

2.5.2 Aprendizagem indutiva a partir de exemplos

O processo da aprendizagem indutiva pode ser visto como a pesquisa de uma descrição geral plausível (assertiva indutiva) que explica o conjunto de dados de entrada (os exemplos) e é útil para prever a classificação de novos exemplos.

Para um dado conjunto de exemplos existe, potencialmente, um número infinitamente grande de hipóteses suscetíveis de explicar esses exemplos. Assim sendo, na aprendizagem indutiva, além dos exemplos, é necessário fornecer um conjunto de restrições objetivando reduzir esse número infinitamente grande de escolhas possíveis a uma única hipótese, ou a um pequeno conjunto das hipóteses mais favoráveis. Essas restrições poderão ser de caráter geral ou específicas do domínio.

As restrições de caráter geral, independentes do domínio, são incorporadas aos próprios algoritmos de aprendizagem. Por exemplo, entre as hipóteses possíveis, o algoritmo poderia escolher a mais "curta", que no caso das árvores de decisão corresponderia escolher a árvore de menor número de nós.

As restrições específicas, são informações adicionais a respeito do domínio, que são fornecidas aos algoritmos indutivos juntamente com os exemplos. De posse dessas informações, o algoritmo poderá escolher, por exemplo, a hipótese mais "econômica", ou se for o caso a de "melhor" significado. Esse conjunto de informações dependentes do domínio, previamente fornecido aos algoritmos, é denominado de **conhecimento preliminar** ("*background knowledge*").

Ao processo de aprendizagem indutiva, a partir de exemplos, baseado em algoritmos que não utilizam conhecimento preliminar dá-se o nome de **aprendizagem por indução empírica**.

2.5.2.1 Representação de conceitos, exemplos e conhecimento preliminar

Num sistema de aprendizagem indutiva, os exemplos e o conhecimento preliminar (quando for o caso) são a entrada do sistema e os conceitos a saída. Essas entradas e saídas são objetos que devem ser descritos em uma certa linguagem, podendo essa descrição ser classificada em superficial ou estruturada [Michalski 83].

Na descrição superficial, também conhecida como *descrição de atributos*, apenas as propriedades globais dos objetos são especificadas. Por exemplo, na descrição de um automóvel poderíamos listar seu custo, modelo, fabricante, ano, etc. Nenhuma estrutura interna dos objetos é apresentada, podendo a descrição dos atributos ser facilmente expressa em uma linguagem baseada na lógica proposicional.

Na *descrição estrutural*, por outro lado, os objetos são vistos como estruturas consistindo de vários componentes, onde descrevem-se não apenas esses componentes mas também o relacionamento entre eles. Por exemplo, na *descrição estrutural*, um automóvel pode ser representado em termos de motor, carroceria, cabine e rodas, bem como uma descrição das relações existentes entre esses componentes. Uma descrição estrutural pode ser expressa em lógica dos predicados, onde cada componente é descrito

globalmente através de variáveis e predicados unários, e as relações entre componentes podem ser expressas como funções e predicados k-ários.

Um método de aprendizagem que usa uma linguagem baseada na *descrição de atributos* tende a ser relativamente eficiente e fácil de implementar, mas fica limitado no seu poder de expressividade, dificultando, assim, o seu uso em várias aplicações do mundo real. Por outro lado, um método que usa uma linguagem baseada na *descrição estrutural*, embora permite representar objetos de estruturas complexas e relações entre esses objetos, ou seus componentes, tende a ser bastante ineficiente e a apresentar grandes problemas de implementação.

Em função das facilidades apresentadas pelas linguagens baseadas na *descrição de atributos*, a grande maioria dos sistemas indutivos têm utilizada essa linguagem na representação de suas entradas e saídas. Entretanto, vale ressaltar que ultimamente vem existindo uma tendência no sentido de também se construir sistemas indutivos que utilizam linguagens de *descrição estrutural*. Esse é o caso, por exemplo, dos sistemas tipo ILP (Inductive Logic Programming) [Muggleton 92], onde a linguagem de descrição é a da *programação em lógica*.

Representação dos exemplos

Na maioria dos trabalhos de aprendizagem indutiva a partir de exemplos, as classes (conceitos) são descritas por um conjunto de características (atributos) e o processo de aprendizagem se baseia nessa informação sobre as classes. Por sua vez, cada característica pode assumir um conjunto de valores que, conforme o tipo desses valores, pode ser classificada em *nominal*, *linear* ou *estruturada* [Michalski 83].

- **nominal** - em que o conjunto de valores é definido por um conjunto de símbolos, ou nomes independentes. Por exemplo, o atributo COR pode assumir os valores {verde, branco, vermelho}.
- **linear** - quando o conjunto de valores é totalmente ordenado, podendo esses valores ser contínuos, por exemplo no caso do atributo TEMPERATURA, ou discretos, como seria no caso do atributo IDADE.

• **estruturada** - o conjunto de valores é dividido em subconjuntos disjuntos e a cada subconjunto está associado uma hierarquia de generalização, representada através de uma árvore. Um nó, nessa árvore, representa um conceito mais geral do que os conceitos representados pelos nós filhos. Por exemplo, o atributo FORMA contendo os valores {triângulo, quadrado, pentágono, círculo, elipse}, representado pela árvore

(FORMA, (triângulo, quadrado, pentágono, círculo, elipse))

poderia ser decomposto em

(FORMA, (POLÍGONO, CÔNICA))

onde: (POLÍGONO, (triângulo, quadrado, pentágono)) e
(CÔNICA, (círculo, elipse))

Assim, considerando-se que , para um dado domínio, existe:

- um espaço cartesiano de atributos $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \dots \times \mathcal{A}_n$, em que \mathcal{A}_i é o atributo de ordem i ;
- um conjunto de classes a ser aprendidas $\mathcal{E} = \{E_1, E_2, \dots, E_{n_E}\}$;
- uma relação, \mathcal{R} , entre o espaço dos atributos e o conjunto das classes

$$\mathcal{R} \subseteq \mathcal{A} \times \mathcal{E}$$

Definimos um conjunto de treinamento, (CT), como sendo um subconjunto da relação \mathcal{R} . Portanto, um exemplo, que é um elemento do CT , nada mais é do que um conjunto de pares atributo-valor, ao qual está associado uma classe, ou

$$\text{exemplo} = (\{(\mathcal{A}_i = v_{ik}) \mid \mathcal{A}_i \in \mathcal{A} \text{ e } v_{ik} \in C_j(\mathcal{A}_i)\}, E_j)$$

onde $C_j(\mathcal{A}_i)$ representa o conjunto de valores de \mathcal{A}_i e $E_j \in \mathcal{E}$.

Na prática, para avaliar o conhecimento induzido a partir do CT , freqüentemente utiliza-se um conjunto de teste (CT_s), que é definido de forma

análoga ao *CT*. Esses dois conjuntos são partes de um conjunto maior, denominado *tabela de exemplos* que, obviamente, também é um subconjunto de \mathcal{R} .

À essa forma de descrever os exemplos, baseada na lógica proposicional, dá-se o nome de *linguagem de atributos valorados*, "attribute-values language" em inglês.

Representação do conhecimento induzido

A representação do conhecimento induzido a partir de exemplos, depende do tipo de algoritmo de aprendizagem utilizado. Objetivando proporcionar ao ser humano uma melhor interpretação do conhecimento gerado, a maioria dos algoritmos indutivos utiliza uma representação simbólica para esse conhecimento. As formas mais utilizadas de representação simbólica são:

- **Regras** do tipo "se <premissa> então <conclusão>", em que a <premissa> é formada por uma conjunção de condições (pares atributo-valor) e a <conclusão> representa um elemento de classificação (classe). Esta é uma das formas mais legíveis para o ser humano.
- **Árvores de decisão**, são árvores nas quais o teste associado a cada nó corresponde a um atributo e a cada ramo da árvore está associado um valor específico do atributo. As folhas representam as classes.
- **Listas de decisão** - são listas de pares

$$(p_1, \mathcal{E}_1), (p_2, \mathcal{E}_2), \dots, (p_n, \mathcal{E}_n)$$

onde cada p_i é uma premissa, cada \mathcal{E}_i é uma classe e a premissa p_n é sempre verdadeira. Cada elemento de uma lista é interpretado como uma regra do tipo "se ... então... senão", assim uma lista de decisão é vista como um conjunto ordenado de regras, ou seja "se p_1 então \mathcal{E}_1 senão se p_2 ... senão \mathcal{E}_n ".

Embora seja possível converter uma forma de representação para qualquer uma das outras, apenas a conversão de árvore de decisão para regras se constitui numa tarefa trivial. As demais conversões, além de não serem simples, podem produzir resultados de difícil entendimento ao ser humano.

Representação do conhecimento preliminar

De um modo geral o conhecimento preliminar pode ser expresso de três formas: *características* de atributos e/ou exemplos, *funções* de valores de atributos e *relações*, podendo essas *relações* ser entre valores de um mesmo atributo, ou atributos distintos, ou entre classes e pares atributo-valor.

Na primeira forma, uma ou mais *características* a respeito dos atributos ou exemplos são descritas. Por exemplo, poderíamos ter a descrição do custo de cada atributo e/ou a importância de cada exemplo no conjunto de treinamento.

Na segunda forma, o conhecimento preliminar consiste na criação de novos atributos e, para cada exemplo do conjunto de treinamento, os valores desses novos atributos serão *função* dos valores que os antigos atributos assumem para esse exemplo [Lavrac 94].

Para a terceira forma, podemos distinguir três tipos de *relações*:

- relação entre valores de um mesmo atributo - o conhecimento preliminar corresponde à descrição estrutural dos atributos, conforme vimos anteriormente ("representação dos exemplos");
- relação entre valores de atributos distintos - igualmente à segunda forma, o conhecimento preliminar consiste na criação de novos atributos, sendo que agora os únicos valores que os novos atributos podem assumir são *verdade* ou *falso* (se, no exemplo, os valores dos atributos correspondentes satisfazem ou não a relação), i.e., as relações são funções booleanas.
- relação entre classes e pares atributo-valor - neste tipo de conhecimento preliminar descreve-se a importância (relevância semântica) que cada par atributo-valor tem na conclusão das classes, sendo a estrutura de matriz uma forma de representar essa relação [Mongiovi 90a].

2.5.1.2 Fundamentação lógica da aprendizagem indutiva

Não havendo um consenso na definição formal de um quadro de aprendizagem indutiva a partir de exemplos, a descrição que apresentaremos nesta seção está baseada em três trabalhos significativos, [Muggleton 92], [Kietz 93] e [Lavrac 94], que, embora diferem nas particularidades e simbologia, no geral convergem para uma mesma idéia.

Para um dado domínio \mathcal{D} , um sistema de aprendizagem indutiva, a partir de exemplos, recebe como entrada um conjunto de exemplos ε , representados em uma linguagem $\mathcal{L}_{\varepsilon}$, e possivelmente um conhecimento preliminar \mathcal{B} a respeito do domínio \mathcal{D} , descrito em uma linguagem $\mathcal{L}_{\mathcal{B}}$. O conjunto ε é constituído de dois subconjuntos. O conjunto dos exemplos *positivos* ε^+ , que descrevem o conceito a ser aprendido (*conceito-meta*) e o conjunto dos exemplos *negativos* ε^- , que são os contra-exemplos desse *conceito-meta*.

A tarefa do sistema de aprendizagem consiste em induzir uma hipótese \mathcal{H} , em uma linguagem $\mathcal{L}_{\mathcal{H}}$, que juntamente com \mathcal{B} *explique* os exemplos de ε . Como usualmente existe mais de uma hipótese nessas condições, é necessário definir, em $\mathcal{L}_{\mathcal{H}}$, uma relação de ordem de preferência que possibilite escolher a "melhor" hipótese.

Assim, um sistema de aprendizagem indutiva a partir de exemplos pode ser definido como segue:

Dados:

- o nome do conceito C a ser aprendido;
- um conjunto não vazio de exemplos $\varepsilon = \varepsilon^+ \cup \varepsilon^-$ de C , em $\mathcal{L}_{\mathcal{E}}$, onde ε^+ é o conjunto não vazio de exemplos *positivos* e ε^- é o conjunto possivelmente vazio de exemplos *negativos*;
- um (possivelmente vazio) conhecimento preliminar \mathcal{B} em $\mathcal{L}_{\mathcal{B}}$;
- uma linguagem $\mathcal{L}_{\mathcal{H}}$ de descrição de hipóteses;
- uma relação de ordem de preferência \leq em $\mathcal{L}_{\mathcal{H}}$

Encontrar: uma hipótese (descrição de C) $\mathcal{H} \subseteq \mathcal{L}_{\mathcal{H}}$, tal que:

- i) $\forall e \in \varepsilon^+, \mathcal{B} \cup \mathcal{H} \models e$, i. e., \mathcal{H} é **completa**, cobrindo todos os exemplos *positivos*;
- ii) $\forall e \in \varepsilon^-, \mathcal{B} \cup \mathcal{H} \not\models e$, i. e., \mathcal{H} é **consistente**, não cobrindo nenhum exemplo *negativo*;
- iii) $\forall \mathcal{H}^*$ satisfazendo (i) e (ii), $\mathcal{H} \leq \mathcal{H}^*$, i. e., de todas as soluções satisfazendo as condições acima \mathcal{H} é a melhor com relação a \leq .

As condições de **completude** (i) e de **consistência** (ii) constituem a *fundamentação lógica* dos algoritmos de aquisição de conceitos a partir de exemplos.

Em um sistema de aprendizagem de múltiplos conceitos (c_1, c_2, \dots, c_n), para cada conceito c_i o conjunto ε^+ (ε_i^+) é formado por todos os exemplos de c_i , enquanto que o conjunto ε^- (ε_i^-) é formado por todos os contra-exemplos de c_i , que são os exemplos de todos os conceitos, exceto os de c_i , ou seja

$$\varepsilon_i^- = \varepsilon^- = \bigcup_{j \neq i} \varepsilon_j^+$$

É importante observar que os sistemas que lidam com dados imperfeitos (ruído), de uma certa forma relaxam as condições de completude e consistência. Por exemplo, se um paciente apresenta duas doenças (confirma duas classes), seu quadro sintomático deve satisfazer a descrição das duas doenças e, portanto, neste caso, a condição de consistência não se verifica. A FIG. 2.8 mostra as quatro possíveis situações de completude e consistência de uma hipótese induzida [Lavrac 93].

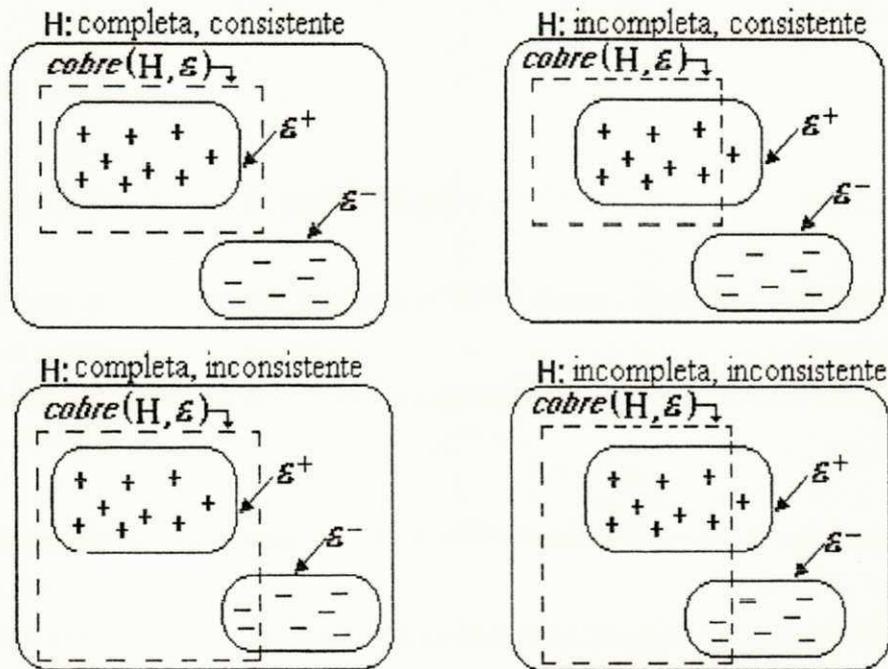


FIGURA 2.8 - Possíveis situações de completude e consistência de uma hipótese induzida.

3 Os Problemas sintático e semântico

3.1 Indução em árvores de decisão

Grande parte das soluções propostas para fazer indução por computador geram como resultado uma *Árvore de Decisão (AD)*, um formalismo bastante conhecido e utilizado para representar conhecimento classificatório.

3.1.1 Árvores de decisão

Uma *AD* é uma árvore onde as folhas são os elementos de classificação (i. e., as classes), os nós não-terminais representam os atributos dos fatos concretos observados no mundo real e os ramos denotam os valores destes atributos. Ela é construída de forma que, ao percorrer o caminho da raiz para uma das folhas, identificamos condições (pares ATRIBUTO = valor) suficientes para classificar a folha.

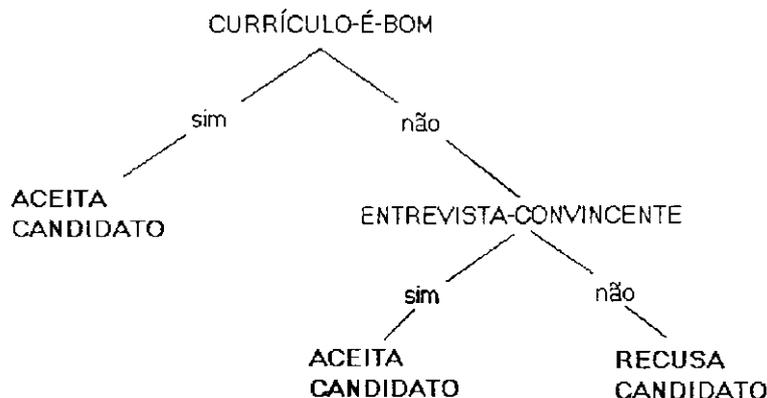


FIGURA 3.1 - Exemplo de uma *AD*

Por exemplo, na \mathcal{AD} da FIG. 3.1, as condições CURRICULO-É-BOM = não e ENTREVISTA-CONVINCENTE = sim são suficientes para afirmar que o candidato será aceito.

Há várias técnicas de aprendizagem automática que permitem a construção de \mathcal{AD} s a partir de conjuntos de treinamento (CT s). Na indução de árvores de decisão, um elemento do CT geralmente representa um exemplo ou evento do mundo real e é descrito como uma coleção de condições, a qual está associada uma classe. A maior parte dos algoritmos de indução que são baseados em \mathcal{AD} s fazem parte da família $TDIDT$ [Quinlan 86]. Estes algoritmos são caracterizados por construir a \mathcal{AD} da raiz para as folhas. Eles fazem a divisão sucessiva do conjunto de treinamento usando o melhor (por exemplo, o mais informativo) atributo como divisor. O processo é repetido recursivamente até que um teste de parada seja satisfeito (tipicamente, até que todos os exemplos da folha sejam de uma única classe) [Van de Vilde 89].

Mapeamento em Regras

É comum mapear árvores de decisão em regras, para possibilitar o uso de ferramentas baseadas em regras para construção de SBC s. Além do mais, o uso de regras facilita a explanação, pois não é mais necessário mostrar toda a \mathcal{AD} quando uma explicação for solicitada. Quando utilizamos regras, apenas aquelas utilizadas são exibidas.

O mapeamento de uma \mathcal{AD} em regras pode ser feito de forma trivial. São construídas tantas regras quantas folhas houver na árvore. Cada folha é o conseqüente de uma regra que tem como antecedente a conjunção dos pares "NO = ramo" encontrados no caminho que vai da raiz à folha em questão. Note que nem sempre há um mapeamento que transforme um conjunto de regras em uma \mathcal{AD} (maiores detalhes na seção 3.3).

Neste trabalho, frequentemente usaremos regras mapeadas de \mathcal{AD} s porque, devido a possibilidade de nomeá-las, é mais fácil desenvolver nossa argumentação com as regras do que com a árvore propriamente dita. Entretanto, a argumentação não perde generalidade, pois as regras obtidas trivialmente de uma dada \mathcal{AD} nada mais são do que uma outra forma de representar a árvore.

Algoritmos que geram árvores de decisão

Há duas abordagens usadas pelos métodos de indução orientados a dados que usam \mathcal{AD} s: a incremental e a \mathcal{TDIDT} .

Os algoritmos incrementais funcionam transformando a \mathcal{AD} , inicialmente vazia, a cada exemplo obtido. Como exemplos de métodos incrementais temos o ID4 [Schlimmer 86], o ID5 [Utgoff 88], o ID5R [Utgoff 89] e o IDL [Van de Vilde 89]. Uma análise das condições em que o uso desses algoritmos é favorável é apresentada em [Bezerra 94a] e [Bezerra 94b].

Para os algoritmos \mathcal{TDIDT} , é necessário que todos os exemplos estejam disponíveis a priori. Eles geram a \mathcal{AD} recursivamente, da raiz para as folhas, escolhendo, a cada passo, o "melhor" atributo. Os algoritmos \mathcal{TDIDT} mais famosos são o ID3 [Quinlan 83], o C4 [Quinlan 87] e o CART [Breiman 84].

O problema sintático (veja seção 3.3) provém do uso de \mathcal{AD} s como forma de representar o conhecimento adquirido. Portanto, tanto os algoritmos incrementais quanto os \mathcal{TDIDT} dele sofrem, embora a forma pela qual eles constroem a árvore seja diferente. Já o problema semântico (veja seção 3.2), pode ser tão bem resolvido tanto pela abordagem \mathcal{TDIDT} quanto pela incremental. Devido a isto, por simplicidade, trataremos apenas dos algoritmos \mathcal{TDIDT} no restante deste trabalho.

3.1.2 A Família de algoritmos \mathcal{TDIDT}

Os diversos algoritmos da família \mathcal{TDIDT} constroem \mathcal{AD} s para tarefas de classificação. Estas árvores são construídas a partir da raiz, descendo até as folhas. Daí o nome \mathcal{TDIDT} .

O conhecimento representado nas \mathcal{AD} s permite classificar objetos ou situações em classes. De posse da tabela de exemplos (conjunto de exemplos que descrevem situações do mundo real), o engenheiro de conhecimento obtém o conjunto de treinamento usando parte dos exemplos dessa tabela. O \mathcal{CT} é usado para gerar a \mathcal{AD} . Os exemplos restantes constituem o conjunto de teste, que é usado para avaliar a \mathcal{AD} obtida.

A tabela de exemplos pode ter diferentes origens. Ela pode vir de uma base de dados existente (por exemplo, um fichário médico com o quadro histórico de vários pacientes). Alternativamente, os exemplos podem ser um conjunto tutorial cuidadosamente preparado por um especialista (por exemplo, os casos clássicos de um determinado domínio).

Cada exemplo está definido por um conjunto de condições (pares ATRIBUTO = valor) e o valor da classe correspondente. Cada atributo a_i pode assumir um número finito de valores v_{i1}, \dots, v_{in_i} . Obviamente, o número de atributos também é finito.

Cada nó não-terminal de uma \mathcal{AD} , representa o teste de um atributo e tem tantos filhos quanto forem os valores que o atributo representado puder assumir. Os ramos que ligam o atributo representado aos seus filhos denotam esses valores. Cada folha é rotulada com a classe que descreve os exemplos, que contém os pares ATRIBUTO = valor do caminho que liga a folha à raiz.

3.1.2.1 Procedimento geral

Os algoritmos *TDIDT* começam com uma \mathcal{AD} vazia, que é expandida gradualmente até classificar todos os exemplos do \mathcal{CT} . O procedimento geral é o seguinte [Castiñeira 90]:

DADOS um conjunto de treinamento \mathcal{D} ;
 uma condição de parada $t(\mathcal{D})$; e
 uma função de avaliação $aval(\mathcal{D}, a)$

SE todas as instâncias em \mathcal{D} satisfazem a condição de término $t(\mathcal{D})$
ENTÃO RETORNE o valor da classe

SENÃO PARA CADA atributo a , CALCULE o valor da função $aval(\mathcal{D}, a)$
 SEJA a_m o atributo que possui o melhor valor de $aval(\mathcal{D}, a)$
 DIVIDA o conjunto \mathcal{D} em subconjuntos com valores v_{m1}, \dots, v_{mn_m} do
 atributo a_m

APLIQUE recursivamente o algoritmo a cada subconjunto de
 treinamento \mathcal{D}_k ($1 \leq k \leq n_m$)

O critério de parada t pode ser definido tanto para construir \mathcal{AD} s que classificam todos os elementos do conjunto de treinamento em domínios determinísticos, quanto para decidir pela não expansão da árvore quando os exemplos fornecidos forem insuficientes. A função de avaliação *aval* denota quão bom é um atributo, no sentido do atributo levar à menor \mathcal{AD} possível. Várias funções de avaliação foram propostas, visando obter \mathcal{AD} s tão pequenas quanto possível. Temos o cálculo de entropia, a qui-quadrado, a estatística G, o índice de diversidade GINI, a medida proporcional de ganho [Mingers 89a]. Dentre elas, a mais utilizada é o cálculo de entropia [Klir 88].

3. 1. 2. 2 O Algoritmo ID3

O algoritmo mais conhecido da família *TDIDT* é o ID3 (Iterative Dichotomizer 3) [Quinlan 83]. Além disso, ele é um dos precursores da família e é um algoritmo simples. Devido a essas características e tendo em vista que o raciocínio aqui desenvolvido não é dependente das particularidades de nenhum algoritmo *TDIDT*, o ID3 será o único algoritmo *TDIDT* descrito neste capítulo.

O ID3 encerra a construção da \mathcal{AD} quando todos os elementos do CT pertencerem à mesma classe. A função de avaliação usada baseia-se no cálculo de entropia. A entropia mede o grau de incerteza de um conjunto de eventos. Assim, quanto menor a entropia de um dado conjunto de eventos, menor será seu grau de incerteza. O objetivo do ID3 é, portanto, obter a entropia nula, i. e., a eliminação da incerteza.

A função de avaliação \mathcal{H} de um atributo a é a média ponderada das entropias das partições do conjunto de treinamento segundo a (ou seja, a cada valor de a corresponde uma partição). Em formulas, temos:

$$\mathcal{H}(a) = \sum_{i=1}^{n_v} \frac{n_{vi}}{n} \mathcal{H}(a = v_i)$$

onde: n_v é o número de valores do atributo a

n_{vi} é o número de exemplos em que o atributo a possui o valor v_i

n é o número de exemplos no conjunto de treinamento

Por sua vez, a entropia da partição do CT formada por todos os casos em que a possui o valor v_i é dada por:

$$\mathcal{H}(a = v_i) = \sum_{j=1}^m -p(\mathcal{E}_j | v_i) \cdot \log_2(p(\mathcal{E}_j | v_i))$$

onde:

m é o número de classes no conjunto de treinamento;

$p(\mathcal{E}_j | v_i)$ é a probabilidade de se ter, no conjunto de treinamento, um exemplo com classe \mathcal{E}_j , dado que a assume valor v_i ;

Por exemplo, para o CT da FIG. 3.2, o cálculo da entropia do atributo DIABÉTICO seria feito como segue:

$$\mathcal{H}(\text{DIABÉTICO} = \text{sim}) = -1/4 \cdot \log_2(1/4) - 3/4 \cdot \log_2(3/4) = 0,8113$$

$$\mathcal{H}(\text{DIABÉTICO} = \text{não}) = -3/4 \cdot \log_2(3/4) - 1/4 \cdot \log_2(1/4) = 0,8113$$

$$\mathcal{H}(\text{DIABÉTICO}) = 4/8 \cdot 0,8113 + 4/8 \cdot 0,8113 = 0,8113$$

	COME	IDADE	VEGETARIANO	DIABÉTICO	CLASSE
ex ₁	pouco	velho	sim	sim	MAGRO
ex ₂	médio	velho	sim	não	MAGRO
ex ₃	muito	velho	não	sim	GORDO
ex ₄	pouco	velho	não	não	MAGRO
ex ₅	médio	jovem	não	sim	GORDO
ex ₆	pouco	jovem	sim	não	MAGRO
ex ₇	muito	velho	não	não	GORDO
ex ₈	médio	jovem	não	sim	GORDO

FIGURA 3.2 - Conjunto de treinamento para o domínio "magro-gordo"

A \mathcal{AD} gerada pelo ID3, correspondente ao conjunto de treinamento da FIG. 3.2, está na FIG. 3.3. Abaixo de cada classe estão os números dos exemplos que foram mapeados pelo caminho que começa na raiz e vai até a classe.



FIGURA 3.3 - \mathcal{AD} gerada pelo ID3

As regras correspondentes à árvore, utilizando o mapeamento trivial, são:

$R_1(\text{ID3})$: se COME = pouco então MAGRO

$R_2(\text{ID3})$: se COME = médio e IDADE = jovem então GORDO

$R_3(\text{ID3})$: se COME = médio e IDADE = velho então MAGRO

$R_4(\text{ID3})$: se COME = muito então GORDO

A vantagem de usar uma boa função de avaliação está no tamanho da \mathcal{AD} obtida. Como comparação, observe a árvore a FIG. 3.4, que foi gerada, usando como função de avaliação uma função aleatória, a partir dos mesmos dados de entrada usados pelo ID3 para gerar a \mathcal{AD} acima (i. e., o CT da FIG. 3.2).



FIGURA 3.4 - \mathcal{AD} gerada aleatoriamente

3. 1. 2. 3 Análise dos algoritmos *TDIDT*

De posse das informações apresentadas neste capítulo sobre os algoritmos *TDIDT*, já podemos levantar as principais vantagens e desvantagens desses algoritmos.

Entre as vantagens destacamos:

- i) A participação do especialista e do engenheiro do conhecimento na construção da *BC* fica minimizada.
- ii) A *BC* é gerada a partir de exemplos ocorridos no dia-a-dia do(s) especialistas(s). Os exemplos são fáceis de serem obtidos (em relação à aquisição cognitiva) e permitem tratamento probabilístico.
- iii) A *BC* assim obtida é livre de inconsistência [Gascuel 88], eliminando a necessidade de verificação de sua coerência [Kgyuen 87].
- iv) O processo não se limita à áreas específicas, isto é, pode, a princípio, ser utilizado em qualquer domínio.

Apesar do sucesso alcançado, estes métodos apresentam algumas limitações de cunho prático. As mais conhecidas são:

- i) São aplicados somente em domínios onde tanto os elementos de classificação como os valores de suas características (atributos) são mutuamente exclusivos.
- ii) Não permitem manipular atributos com valores contínuos.
- iii) Falham na sua tentativa de construir a *AD* quando, entre os exemplos, existem exemplos ambíguos, caracterizando um domínio não determinístico.
- iv) Não geram regras com incerteza.

3.2 O Problema semântico

As *ADs* geradas pelos algoritmos da família *TDIDT* são estruturas despidas de semântica. Elas apenas mapeiam, de forma reduzida, o *CT* usado como entrada. Devido a este fato, estes algoritmos podem gerar, em certas circunstâncias, regras sem significado, que na prática são inúteis. Este fenómeno é observado principalmente na resolução de problemas reais (i. e, problemas em domínios que têm grande número de classes e condições).

Isso ocorre porque, em problemas reais, o *CT* geralmente só explicita uma minúscula parte de todos os exemplos possíveis dentro do domínio. Assim, podemos ter, por coincidência, que uma dada classe seja determinada por uma condição irrelevante (para a classe). Uma circunstância em que isto frequentemente acontece é quando o conjunto de exemplos contém casos raros (casos caracterizados por condições que ocorrem apenas em pouquíssimos exemplos). Vale ressaltar que o conjunto de treinamento é pouco abrangente, não por vontade do engenheiro do conhecimento ou de algum elemento humano envolvido no processo de aquisição, mas devido ao tamanho do problema em si.

O problema semântico foi descoberto durante a validação do APREND [Gomes 88] [Gomes 89], sistema de *AC* baseado no ID3. O objetivo era construir um SE em ginecologia, o que representava um domínio com 18 classes e 44 atributos, cada um com, em média, 3,47 valores [Mongiovi 90b]. O *CT* foi obtido de um fichário médico. Para nossa surpresa, entre outras aberrações, o APREND (leia-se, ID3) concluiu que a condição "IDADE = criança" era suficiente para diagnosticar VAGINITE (sic!). Isto aconteceu porque no *CT*, por coincidência, havia poucas crianças e todas elas estavam com vaginite (já que os dados foram obtidos de fichário médico). Portanto, o atributo IDADE era o de menor entropia. Além disso, o sistema APREND encerrou neste ponto a construção do ramo correspondente ao caso, já que todos os exemplos ficam classificados apenas com "IDADE = criança". O caso raro ficou, portanto, mapeado numa regra inútil, uma vez que a premissa da regra acima ("a paciente é criança"), na prática é irrelevante para concluir se a paciente está ou não com vaginite. Vale salientar que denominamos as regras com este tipo de problema de "inúteis" porque, embora semanticamente erradas, elas de fato mapeiam o *CT*.

Este tipo de problema pode aparecer até mesmo em casos "de brinquedo". Observe, por exemplo, a \mathcal{AD} da FIG. 3.3, que foi gerada pelo ID3. As regras $R_2(\text{ID3})$ e $R_3(\text{ID3})$, embora não mapeando casos raros, têm, como se pode ver, pouca utilidade prática e são ditas inúteis. Isto porque DIABÉTICO e VEGETARIANO são fatores muito mais indicativos do que IDADE, para concluirmos se alguém é magro ou gordo. Além do mais, o atributo COME não é importante para a conclusão com o valor "médio".

Na verdade, há também uma outra justificativa para esse comportamento apresentado pelos algoritmos \mathcal{TDIDT} . Em casos reais, apenas uma parcela das condições é relevante para a determinação de uma dada classe.

Por exemplo, a condição "ESTADO-CIVIL = casado" não é, de forma alguma, tão importante quanto "FEBRE = alta" para determinar a classe VIROSE. Porém, os algoritmos \mathcal{TDIDT} consideram todas as condições potencialmente iguais para a conclusão de uma determinada classe. É devido a este comportamento (não considerar as condições de forma diferenciada, tendo em vista o que elas significam para a classe em determinação) que o problema apresentado nesta seção foi chamado de problema semântico.

3.3 O Problema sintático

Uma das características mais desejáveis nos \mathcal{SBCs} é que eles sejam facilmente atualizados e modificados. Esta característica provém da modularidade da \mathcal{BC} , i. e., da possibilidade de alterar o conhecimento relativo a uma parte do domínio mantendo o resto da \mathcal{BC} intocado.

As \mathcal{ADs} não são uma forma de representação de conhecimento que proporciona modularidade. Em outras palavras, o conhecimento de áreas específicas do domínio está pulverizado por toda a árvore. Há, entretanto, um problema maior: existe conhecimento que pode ser representado muito mais facilmente em regras que em \mathcal{ADs} . Considere, por exemplo, o seguinte par de regras:

- u₁) se VEGETARIANO = sim então MAGRO
- u₂) se COME = pouco então MAGRO

Note que as regras são suficientes para classificar MAGRO, em relação ao *CT* da FIG. 3.2. Entretanto, não há uma *AD* correspondente a estas regras, porque elas não compartilham nenhum atributo (i. e., não há um atributo comum para ser colocado na raiz).

A *AD* mais simples que classifica as instâncias cobertas por regras que não têm atributos em comum, necessariamente adiciona pelo menos uma condição extra em uma das regras, o que a torna mais específica e frequentemente exige a adição de novos ramos para cobrir eventuais instâncias excluídas pela especialização. No nosso exemplo, tivemos a sorte de não precisar adicionar novos ramos, pois a *AD* da FIG. 3.5 cobre todas as instâncias classificadas por u_1 e u_2 . Todavia, o que mais comumente ocorre nestes casos é que a *AD* é bastante mais complexa que as regras [Cendrowska 88].

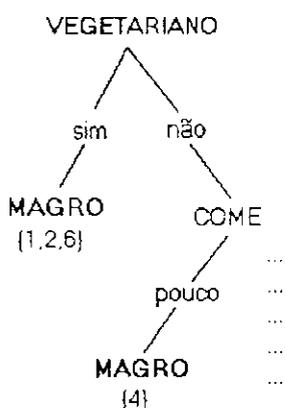


FIGURA 3.5 - *AD* correspondente as regras u_1 e u_2

Representar numa *AD* o conhecimento contido em um conjunto de regras não mapeável diretamente em uma árvore traz sérias desvantagens. Além de aumentar o número de condições necessárias para obter algumas classificações, as explicações obtidas da *AD* (ou de regras diretamente mapeadas da *AD*) são mais difíceis de entender. É que as condições adicionadas artificialmente na *AD* muitas vezes são irrelevantes para a classe que determinam. Em outras palavras, o especialista não acredita na explicação porque ela contém, na premissa, condições irrelevantes para a conclusão da classe em questão. Por exemplo, no sistema especialista ginecológico gerado pelo APREND (veja seção 3.2) todas as regras, com exceção de "se IDADE = criança então VAGINITE", têm a condição "IDADE =

adulto", o que em nada contribui para a clareza das explicações obtidas do sistema.

Outro ponto negativo da conversão forçada de regras em AD s é que o conhecimento fica menos "disponível". Considere as regras u_1 e u_2 e a árvore da FIG. 3.5. Se soubermos apenas a evidência $COME = pouco$, conseguimos chegar a conclusão através das regras. Já através da AD , não conseguimos.

Voltando ao acréscimo desnecessário de condições à BC , vale ressaltar que, em muitos domínios, isto inviabiliza o SBC . De fato, obter o valor de determinados atributos custa tempo e/ou dinheiro. Para se ter idéia do problema, imagine o gasto de tempo e dinheiro que traria um sistema especialista que tivesse como BC uma AD com o atributo TOMOGRAFIA-COMPUTADORIZADA na raiz. Pior ainda, a maioria das classes determinadas pela sub-árvore relativa ao valor negativo de TOMOGRAFIA-COMPUTADORIZADA poderiam ser concluídas sem a ajuda desse exame.

O problema é mais grave quando há atributos com grande número de valores, pois aumentam as chances de aparecer valores que não são úteis à conclusão. Colocando de outra forma, ao calcular a função de avaliação dos atributos, os algoritmos que usam AD como saída geralmente fazem a média entre os resultados da função calculada com o atributo assumindo todos os seus valores. O problema aparece quando, apesar do resultado da função não ser "bom" para alguns valores, a média ainda é a melhor.

Observando o conjunto de regras obtidas pelo ID3 para o CT da FIG. 3.2, vemos que a regra $R_3(ID3)$ poderia ser substituída com vantagem por u_1 , pois esta última é mais representativa, tem um número menor de condições e o novo conjunto de regras ainda continua mapeando o CT . No entanto, nenhum algoritmo da família $TDIDT$ seria capaz de fazer isto, pois após tal substituição, não há mais uma AD trivialmente equivalente a esse novo conjunto de regras.

Além do mais, regras não-modulares são pouco "robustas", pois cada exemplo é classificado apenas por uma regra. Com regras modulares, isto pode mudar. De fato, veja que os exemplos ex_1 e ex_6 ficam cobertos, após a substituição de $R_3(ID3)$, por duas regras ($R_1(ID3)$ e u_1).

Pelas questões discutidas nesta seção concluímos que, além do problema semântico, as \mathcal{AD} s geradas pelos algoritmos \mathcal{TDIDT} também apresentam um conjunto de pontos negativos, que são de natureza estrutural. Justamente devido a essa natureza estrutural é que esse conjunto de pontos negativos foi denominado de problema sintático.

3. 4 Uma Solução para o problema sintático

Vários métodos já foram propostos para contornar o problema sintático. A simplificação de \mathcal{AD} s é possível pela identificação de partes comuns de ramos distintos, mas a explosão combinatória do número de comparações que precisam ser feitas torna este método aplicável apenas para pequenas árvores. Além disso, a simplificação pode ser feita de diversas formas pois pode haver mais de uma parte comum para um dado conjunto de ramos. A questão que aparece então é qual a melhor simplificação que pode ser feita. A solução para isto envolve consulta a especialistas e/ou o uso de um outro programa de indução que gera novas regras a partir das antigas [Cendrowska 88]. Além do mais, para um mesmo domínio, existem diferenças significativas entre as diversas técnicas de simplificação de \mathcal{AD} s, visto que a qualidade dos resultados dessas técnicas depende muito de certas características do domínio, como, por exemplo, número de classes e nível de ruído nos dados.

Uma alternativa extremamente interessante consiste em evitar o uso de árvores, pela indução direta de regras modulares. Contornamos, assim, os problemas que aparecem quando tentamos solucionar os problemas intrínsecos às \mathcal{AD} s (isto é, os problemas sintáticos). Os algoritmos da família AQ [Michalski 83] e o algoritmo PRISM, proposto por Cendrowska, fazem exatamente isso.

3. 4. 1 O que mudar nos algoritmos \mathcal{TDIDT}

A causa principal do problema sintático é a maneira pela qual uma \mathcal{AD} representa o conhecimento. Quando colocamos um atributo em um nó qualquer de uma \mathcal{AD} , estamos nos obrigando a usar todos os seus valores nos ramos (isto é, colocamos na \mathcal{BC} todas as condições possíveis que contêm o atributo). Ora, frequentemente ocorre que nem todas as condições são úteis

para a conclusão das classes. Entretanto, temos que colocar as condições inúteis na \mathcal{AD} para viabilizar o uso das condições interessantes às conclusões.

Pensando nos algoritmos \mathcal{TDIDT} , o problema sintático pode ser creditado ao uso de funções de avaliação. Funções de avaliação de atributos sempre fornecem uma indicativa global da qualidade do atributo. Elas não contêm informações individualizadas sobre a qualidade das condições que podem ser compostas com o atributo.

Portanto, a mudança que deve ser feita nos algoritmos \mathcal{TDIDT} para resolver o problema sintático consiste no uso de indicadores individualizados no lugar dos globais. Assim, enquanto os \mathcal{TDIDT} escolhem "bons" atributos, podemos escolher boas condições ou boas combinações de condições (complexos), evitando o uso forçado de condições irrelevantes.

3.4.2 O Algoritmo PRISM

3.4.2.1 O Porque do PRISM

Entre os algoritmos que geram diretamente regras modulares podemos destacar os da família AQ e o PRISM. Embora em tese sejam semelhantes, esses algoritmos diferem em um aspecto, que consideramos fundamental. Enquanto os algoritmos AQS buscam melhores conjunção de condições (melhores complexos), ou seja melhores indicadores globais, o algoritmo PRISM busca melhores condições de uma forma individualizada. Esse fato resulta em dois pontos negativos para os AQS. Primeiro, é que ao escolher o melhor complexo, com um certo número finito k de componentes, os AQS consideram todas as possíveis combinações das condições (de tamanho máximo k), apresentando assim uma complexidade exponencial, que será tanto mais acentuada quanto maior for o valor de k . Segundo, é que devido ao caráter globalizante da escolha do melhor complexo, nada garante que todos os componentes desse complexo sejam úteis para a conclusão da regra.

Apesar da complexidade do PRISM ser quadrática, ela é bem melhor que a exponencial dos AQS. Além disso, usa técnicas de indução semelhantes às dos \mathcal{TDIDT} , diferindo apenas na estratégia de indução. Por esses motivos, pelo fato de escolher condições individualizadas e devido aos pontos negativos,

acima mencionados, apresentados pelos AQS, o algoritmo PRISM será o utilizado neste trabalho como base para a solução do problema sintático. Gaines e Shaw [Gaines 93] também salientaram os aspectos positivos do PRISM e utilizam idéias semelhantes em seu algoritmo INDUCT, que gera diretamente regras modulares a partir de dados (exemplos) sujeitos a ruídos.

3.4.2.2 Descrição do PRISM

O algoritmo PRISM [Cendrowska 88] gera as regras para cada um dos elementos de classificação separadamente. Desta forma, a inclusão artificial de condições desnecessárias nas regras é evitada.

A quantidade de informação (*information content*) de um determinado elemento de classificação \mathcal{E}_j é calculada pela fórmula:

$$I(\mathcal{E}_j) = \log_2(1/p(\mathcal{E}_j))$$

onde: $p(\mathcal{E}_j)$ é a probabilidade de um exemplo ser da classe \mathcal{E}_j

Por exemplo, o cálculo da quantidade de informação da classe MAGRO no CT da FIG. 3.2 é feito da seguinte forma:

$$I(\text{MAGRO}) = \log_2(1/p(\text{MAGRO})) = \log_2(1/0,5) = 1$$

As regras são construídas de forma que o ganho de informação ("*information gain*") pelas condições da premissa seja igual ou superior a $I(\mathcal{E}_j)$. O processo termina quando todos os exemplos relativos a \mathcal{E}_j tiverem sido utilizados para gerar regras. O ganho de informação para a classificação de \mathcal{E}_j devido à presença da condição C na premissa é calculada pela seguinte fórmula:

$$I(\mathcal{E}_j | C) = \log_2(p(\mathcal{E}_j | C)/p(\mathcal{E}_j))$$

onde: $p(\mathcal{E}_j | C)$ é a probabilidade de um exemplo, dado que contém a condição C , pertencer à classe \mathcal{E}_j .

Por exemplo, tomando como base o CT da FIG. 3.2, o ganho de informação da condição "COME = pouco" para a classe MAGRO é dado por

$$I(\text{MAGRO} \mid \text{COME}=\text{pouco}) = \log_2(p(\text{MAGRO} \mid \text{COME}=\text{pouco})/p(\text{MAGRO})) \\ = \log_2(1/0,5) = 1$$

Já o ganho de informação de IDADE = jovem para GORDO é

$$I(\text{GORDO} \mid \text{IDADE}=\text{jovem}) = \log_2(p(\text{GORDO} \mid \text{IDADE}=\text{jovem})/p(\text{GORDO})) \\ = \log_2(0,6667/0,5) = 0,4151$$

Como o objetivo é ter as menores regras possíveis, o algoritmo se resume a escolher qual condição usar num dado instante. Para atingir esta meta basta escolher o maior valor de $I(E_j \mid C)$. Porém, como $p(E_j)$ é o mesmo para todo C , isso se reduz a obter o maior $p(E_j \mid C)$. As provas que o algoritmo PRISM produz regras corretas e completas em relação ao conjunto de exemplos usado como entrada, podem ser encontradas no trabalho original de Cendrowska [Cendrowska 88]. A FIG. 3.6 apresenta o algoritmo PRISM em pseudo-código.

```

prism(CT)
PARA TODA classe  $E_j$  em CONJUNTO_CLASSES(CT)
  ENQUANTO há exemplo em CT com  $E_j$ 
    lista_condições  $\leftarrow \emptyset$ 
    CT_AUX  $\leftarrow$  CT
    REPITA
      ACHE, em CT_AUX, C tal que  $p(E_j \mid C)$  seja máximo
      lista_condições  $\leftarrow$  lista_condições + C
      CT_AUX  $\leftarrow$  {X | (X é exemplo de CT_AUX) e (X contém C) }
      RETIRE o atributo de C de CT_AUX
    ATÉ QUE CONJUNTO_CLASSES(CT_AUX) = {  $E_j$  }
    NOVA_REGRA  $\leftarrow$  SE lista_condições ENTÃO  $E_j$ 
    RETIRE de CT todos os exemplos classificados por NOVA_REGRA
  RESTAURE CT original

```

FIGURA 3.6 - Algoritmo PRISM

3. 4. 3 Exemplo com o algoritmo PRISM

Utilizando como entrada o *CT* da FIG. 3.2, o algoritmo PRISM produz:

$R_1(\text{PRISM})$: se VEGETARIANO = sim então MAGRO

$R_2(\text{PRISM})$: se COME = pouco então MAGRO

$R_3(\text{PRISM})$: se VEGETARIANO = não e IDADE = jovem então GORDO

$R_4(\text{PRISM})$: se COME = muito então GORDO

Note que, embora tenha solucionado o problema levantado na seção 3.3 (a regra $u_1 = R_1(\text{PRISM})$ agora pertence a \mathcal{BC}), o PRISM não resolveu a questão levantada na seção 3.2, pois, ainda há, na regra $R_3(\text{PRISM})$, uma condição com IDADE, para decidir entre MAGRO e GORDO. Ou seja, o PRISM resolve o problema sintático mas não o semântico.

4 A Relevância semântica

4.1 Conhecimento preliminar

Os algoritmos indutivos orientados a dados ("data driven") são caracterizados por produzirem generalizações a partir de uma grande quantidade de exemplos, mas usando pouca ou freqüentemente nenhuma teoria a respeito do domínio em questão. Esses algoritmos devem construir generalizações basicamente a partir de exemplos empíricos. Devido a isso, alguns autores chamam esse tipo de indução de indução empírica [Pires 93].

Contudo, para a maioria dos casos reais, essa faceta eminentemente empírica do processo pode resultar em algoritmos que produzem resultados de forma ineficiente e de qualidade questionável, pelo menos do ponto de vista semântico.

Para contornar os aspectos negativos acima mencionados, uma solução encontrada, por alguns pesquisadores da área, foi a de diminuir o caráter puramente empírico do processo. Para isso, combinaram as informações extraídas dos dados (exemplos) com o conhecimento preliminar ("background knowledge") [Mongiovi 90a], [Nuñez 91] e [Pires 93]. O Conhecimento preliminar, que consiste no conhecimento disponível sobre o domínio antes do uso do algoritmo generalizador, tem um caráter de complementaridade ao conjunto de exemplos e pode contribuir de duas formas na aprendizagem: melhorando a qualidade e utilidade do conhecimento gerado e facilitando o trabalho dos algoritmos de aprendizagem, através da diminuição do seu espaço de busca.

O conhecimento preliminar pode se apresentar em diversas formas diferentes, como: relevância dos pares atributo-valor para cada classe, grafos de hierarquia IS-A, informação quanto ao custo dos atributos, importância de cada exemplo no conjunto de treinamento, etc. Esses diversos tipos de conhecimento preliminar tratam aspectos específicos do domínio e podem ser usados de forma isolada ou simultaneamente. Por exemplo, [Cirne 92] e [Mongiovi 93a] usam relevância semântica para resolver o problema semântico, já [Nuñez 91] usa grafos de hierarquia IS-A e informações sobre custo para gerar árvores de decisão econômicas e com um alto nível de generalização.

Poderíamos argumentar que a eliciação do conhecimento preliminar, por si só, já seria um outro problema de aquisição de conhecimento e que como tal, provavelmente, teríamos uma volta aos métodos cognitivos (manuais) de aquisição de conhecimento, com todos os seus inconvenientes. Mas esse argumento pode ser facilmente refutado. Os maiores inconvenientes das técnicas manuais estão nas dificuldades de se estabelecer técnicas genéricas e eficazes de eliciação do conhecimento pleno do domínio. O que estamos propondo aqui é se obter apenas alguns aspectos bastantes específicos do conhecimento do domínio, podendo ser eliciado de uma forma objetiva, através, por exemplo, de entrevistas focadas [Vasco 92b].

A seguir descreveremos um desses tipo de conhecimento preliminar, que denominamos de relevância semântica, cujo objetivo maior é fornecer subsídios aos algoritmos indutivos para produzir bases de conhecimento de melhor qualidade e utilidade.

4.2 Matriz de relevância

Os algoritmos de indução de ADs da família $TDIDT$ apresentam um problema, pois consideram os dados apenas do ponto de vista sintático. Uma solução para este problema consiste no uso de informações semânticas no processo de obtenção da AD . Assim, produziríamos Árvores de Decisão Semânticas ($ADSs$), i. e., árvores de decisão cuja construção observou a relevância com que cada condição determina uma dada classe.

É importante lembrar que esse tipo de problema não é exclusivo dos algoritmos $TDIDT$, ele também existe em algoritmos generalizadores de

outras famílias, como por exemplo os da família AQ. Isto porque, o problema não está nem no tipo de representação de conhecimento, usada pelos algoritmos, nem na forma como os algoritmos geram o conhecimento. O problema reside, isto sim, no fato de que, em geral, os algoritmos indutivos são guiados apenas pelo conjunto de treinamento, ou seja apenas por informações estatísticas.

Portanto, para contornar o problema acima, necessitamos obter de alguma forma o conhecimento a respeito do relacionamento semântico entre as condições e as classes de um dado domínio. A fonte mais natural para a obtenção desse tipo de informação é o especialista. Para isso, definimos uma estrutura chamada Matriz de Relevância (\mathcal{MR}) [Mongiovi 90a], que permite facilmente a representação e a eliciação desse tipo de conhecimento preliminar.

Definição da matriz de relevância

Uma \mathcal{MR} para um dado domínio é o relacionamento de relevância entre os elementos de classificação e os atributos deste domínio. Em uma \mathcal{MR} a linha i representa o atributo a_i e a coluna j o elemento de classificação \mathcal{E}_j . Um elemento r_{ij} de uma \mathcal{MR} indica o grau de relevância do atributo a_i para a classificação de \mathcal{E}_j . Este grau é dado pelo conjunto de valores do atributo a_i que são relevantes para a conclusão de \mathcal{E}_j . O elemento r_{ij} é, na verdade, um subconjunto do conjunto de valores do atributo a_i . Se r_{ij} for vazio, significa que o atributo a_i é irrelevante na classificação de \mathcal{E}_j . Ao passo que, r_{ij} igual ao conjunto de valores de a_i expressa que a_i é totalmente relevante na classificação de \mathcal{E}_j . Uma \mathcal{MR} é dita completa se todos os atributos forem totalmente relevantes para todas as classes. Uma \mathcal{MR} é dita vazia se todos os seus elementos forem iguais ao conjunto vazio.

Uma \mathcal{MR} é semelhante à uma rede de repertório ("rating grid") dos sistemas tipo ETS. A grande diferença é que enquanto numa rede de repertório o relacionamento é global (cada atributo (característica), como um todo, se relaciona com os elementos de classificação), numa \mathcal{MR} o relacionamento é específico (os pares atributo/valor é que se relacionam com os elementos de classificação). Além do mais, numa rede de repertório o relacionamento é total (todos os elementos de classificação se relacionam com todas as

características), ao passo que numa \mathcal{MR} apenas os relacionamentos tidos como relevantes são considerados. Isto minimiza a participação do especialista no processo de aquisição deste tipo de conhecimento.

Por exemplo, uma \mathcal{MR} para o domínio do CT da FIG. 3.2 (este domínio é caracterizado pelos atributos COME (que assume os valores pouco, médio ou muito), DIABÉTICO (valores sim ou não), IDADE (valores jovem ou velho) e VEGETARIANO (valores sim ou não), e pelas classes MAGRO e GORDO) aparece na figura abaixo.

	MAGRO	GORDO
COME	{pouco}	{muito}
DIABÉTICO	{não}	{sim}
IDADE	ϕ	ϕ
VEGETARIANO	{sim}	{não}

FIGURA 4.1 - Exemplo de uma \mathcal{MR}

A FIG. 4.1 mostra que:

- i) O atributo COME é relevante para classificar MAGRO no valor pouco, e, no valor muito, para classificar GORDO.
- ii) O atributo DIABÉTICO é relevante para classificar GORDO no valor sim e MAGRO no valor não.
- iii) O atributo IDADE é irrelevante para classificar MAGRO e GORDO (casos deste tipo alertam ao engenheiro do conhecimento que o atributo IDADE possivelmente deve ser eliminado do CT).
- iv) O atributo VEGETARIANO é relevante para classificar GORDO no valor não e MAGRO no valor sim.

Note que a \mathcal{MR} não é uma técnica de aquisição de conhecimento na acepção pura da palavra, pois não há nela informação suficiente para construir uma BC . Entretanto, ela é muito fácil de ser construída pelo especialista. Por exemplo, um especialista gastou apenas 2 horas na

construção de uma \mathcal{MR} para o domínio ginecológico onde foi detectado o problema semântico, que tem 18 classes e 44 atributos (com, em média, 3,47 valores por atributo).

A idéia, com o uso da \mathcal{MR} , é repartir o trabalho: deixamos uma pequena parte com o especialista e o restante com a máquina, de forma a obter um bom resultado no menor tempo possível. O ideal seria eliminar a participação do especialista. Todavia, as soluções até hoje propostas não foram boas ou gerais o suficiente para produzir resultados que dispensem a aprovação final do especialista. Esta dificuldade de construir métodos automáticos de aquisição de conhecimento realmente gerais e eficazes se deve à grande complexidade do processo de aquisição.

4.3 Matriz de relevância nebulosa

Dentro da nossa concepção de abordagem dos problemas dos métodos indutivos, o conceito de matriz de relevância desempenha um papel fundamental. Tanto pode ser usado na solução do problema semântico (vide capítulo 5), como também pode servir de apoio na modelagem do domínio (capítulo 9). Por exemplo, se na \mathcal{MR} consta que um dado atributo é irrelevante para todas as classes do domínio, isto é um indicativo para eliminar esse atributo do conjunto de treinamento. Entretanto, temos dois problemas com a \mathcal{MR} definida na seção 4.2. O primeiro de natureza estrutural, enquanto que o segundo está relacionado com a obtenção da própria \mathcal{MR} .

Durante a construção de \mathcal{MR}_S , por parte dos especialistas, por exemplo o Ginecol (capítulo 3), constatamos que vários termos linguísticos, tais como *pouco* relevante, *mais ou menos* relevante e *muito* relevante, não puderam ser representados. Isto deve-se ao fato de que, numa \mathcal{MR} só é possível indicar se um par atributo/valor é ou não relevante para uma dada classe, isto é, os elementos da matriz são apenas conjuntos cantorianos.

Além do mais, para um dado domínio não existe a \mathcal{MR} , existe, isto sim, uma \mathcal{MR} definida por um especialista naquele domínio. Em outras situações semelhantes de aquisição de conhecimento, como no caso de redes de repertório [Boose 84] e grafos de conhecimento [Machado 90], a prática mostrou que os resultados apresentados (leia-se BCs) variam, as vezes de

forma antagônica, de um especialista para outro. Esta variação deve-se a fatores tais como: contexto, escola do especialista e grau de vivência na área. Obviamente, estes fatores também influenciam a construção de MR_s .

Nos casos acima (redes de repertório e grafos de conhecimento), a solução encontrada para o problema da não unicidade de uma BC foi compor as diversas bases de conhecimento fornecidas pelos especialistas. Na composição, podem ser observadas as importâncias relativas dos especialistas, premiando os consensos sem desprezar as opiniões dissidentes. Esta solução da composição se aplica perfeitamente ao caso das MR_s .

Portanto, necessitamos de uma estrutura que permita ao mesmo tempo, modelar os termos linguísticos, acima mencionados, e representar o resultado da composição das MR_s . Para atender a essas necessidades, a solução que nos pareceu mais adequada foi estender as MR_s de forma a permitir que seus elementos fossem representados por conjuntos nebulosos (uma extensão dos conjuntos cantorianos - maiores detalhes veja [Klir 88]). Com o uso de conjuntos nebulosos, podemos representar mais fielmente como um atributo é relevante para a conclusão de uma determinada classe. A este novo tipo de MR chamaremos de Matriz de Relevância Nebulosa (MRN) [Mongiovi 93a] e [Mongiovi 93b].

4.3.1 Definição da matriz de relevância nebulosa

Uma matriz de relevância nebulosa para um dado domínio é o relacionamento de relevância entre os elementos de classificação e os atributos deste domínio. Em uma MRN a linha i representa o atributo a_i e a coluna j o elemento de classificação E_j . Um elemento r_{ij} de uma MRN representa a relevância do atributo a_i para a conclusão da classe E_j , sendo definido como um conjunto nebuloso que tem como universo o conjunto de valores de a_i . Se r_{ij} for vazio, significa que o atributo a_i é irrelevante na classificação de E_j . Ao passo que, r_{ij} igual ao conjunto de valores de a_i (função de pertinência igual a 1 (um) para todos os valores de a_i) expressa que a_i é totalmente relevante na classificação de E_j . No caso particular de todos os valores da função de pertinência de cada r_{ij} serem iguais a 0 (zero) ou 1 (um), temos uma simples MR .

Por exemplo, uma \mathcal{MRN} para o mesmo domínio da \mathcal{MR} da FIG. 4.1 (domínio do \mathcal{CT} da FIG. 3.2) é a da FIG. 4.2.

	MAGRO	GORDO
COME	1.0/pouco + 0.2/médio	0.3/médio + 1.0/muito
DIABÉTICO	1.0/não	1.0/sim
IDADE	ϕ	0.5/velho
VEGETARIANO	1.0/sim	1.0/não

FIGURA 4.2 - Exemplo de uma \mathcal{MRN}

Comparando a \mathcal{MR} da FIG. 4.1 com a \mathcal{MRN} da FIG. 4.2, observamos que esta última possibilita expressar certos fatos que a primeira não permite. Por exemplo, na \mathcal{MRN} da FIG. 4.2 foi possível indicar que a condição "IDADE = velho" é *mais ou menos* relevante (valor da função de pertinência igual a 0,5) para concluir GORDO. Isto jamais seria possível na \mathcal{MR} .

4.3.2 Intensidade de relevância

Para medir o quanto um atributo é relevante para a conclusão de uma ou mais classes, definiremos uma grandeza que denominamos de Intensidade de Relevância (IR). Para uma dada \mathcal{MRN} , a IR é um número no intervalo $[0, 1]$ e pode ser de uma condição (par atributo/valor) ou de um atributo. Por sua vez, cada uma dessas IRs pode ser definida para uma classe específica ou para todo o conjunto de classes do domínio. Além dessas grandezas, definiremos também a intensidade de relevância do conjunto de atributos para todo o conjunto de classes, que denominaremos de intensidade de relevância da matriz.

Seja uma $\mathcal{MRN}(n_a * n_c)$ com as seguintes características:

- $\mathcal{A} = \{a_1, a_2, \dots, a_{n_a}\}$ - conjunto de n_a atributos;

- $\mathcal{E} = \{ \mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_{n_c} \}$ - conjunto de n_c classes;
- r_{ij} - a relevância do atributo a_i para a classe \mathcal{E}_j ;
- v_{ik} - k -ésimo valor do atributo a_i ;
- $\mathcal{F}_{ij}(v_{ik})$ - função de pertinência do conjunto nebuloso r_{ij} num ponto (valor) v_{ik} .

IR de uma condição para uma classe ($IR_{v_{ik}}\mathcal{E}_j$)

Mede a relevância de uma condição ($a_i = v_{ik}$) para uma classe \mathcal{E}_j , é dada por:

$$IR_{v_{ik}}\mathcal{E}_j = \mathcal{F}_{ij}(v_{ik})$$

IR de uma condição para um conjunto de classes ($IR_{v_{ik}}\mathcal{E}$)

Mede a relevância de uma condição ($a_i = v_{ik}$) para todo o conjunto de classes do domínio, é definida como:

$$IR_{v_{ik}}\mathcal{E} = \left(\sum_{j=1}^{n_c} IR_{v_{ik}}\mathcal{E}_j \right) / n_c$$

Se esta intensidade for nula significa que o par (a_i, v_{ik}) é totalmente irrelevante na modelagem do domínio, sendo assim um forte indicativo de que o valor v_{ik} deveria ser suprimido do atributo a_i , ou então deveria se processar uma redefinição dos valores desse atributo.

IR de um atributo para uma classe ($IR_{a_i}\mathcal{E}_j$)

Mede a relevância de um atributo a_i para uma classe \mathcal{E}_j , é definida como

$$IR_{a_i}\mathcal{E}_j = \left(\sum_{k=1}^{n_{vi}} IR_{v_{ik}}\mathcal{E}_j \right) / n_{vi}$$

onde n_{vi} é o número de valores do atributo a_i .

IR de um atributo para um conjunto de classes ($IR_{a_i \mathcal{E}}$)

Mede a relevância do atributo a_i para o conjunto de classes \mathcal{E} do domínio, é definida como

$$IR_{a_i \mathcal{E}} = \left(\sum_{j=1}^{n_c} IR_{a_i \mathcal{E}_j} \right) / n_c = \left(\sum_{k=1}^{n_{vi}} IR_{v_{ik} \mathcal{E}} \right) / n_{vi}$$

Se esta medida for igual a zero, significa que o atributo a_i é irrelevante para o domínio, devendo assim ser eliminado do conjunto de treinamento.

Intensidade de relevância da matriz nebulosa ($IR_{\mathcal{MN}}$)

Mede a relevância do conjunto de atributos \mathcal{A} para o conjunto de classes \mathcal{E} , é definida como

$$IR_{\mathcal{MN}}(\mathcal{A}, \mathcal{E}) = \left(\sum_{i=1}^{n_a} IR_{a_i \mathcal{E}} \right) / n_a$$

Esta medida fornece uma indicação da "força semântica" da matriz, quanto maior for seu valor maior será a informação semântica do domínio. Entretanto, a medida que o valor dessa grandeza se aproxima de seus valores extremos, 0 ou 1, o poder de distinção dos atributos para com as classes vai diminuindo, sendo isso um indicativo de que o domínio não foi adequadamente modelado, pois em ambos os casos os atributos são igualmente inúteis na distinção das classes. No primeiro caso, valores próximos de 0, porque não dispomos de nenhum conhecimento semântico do domínio e, portanto, os atributos não tem nenhuma importância na distinção dos elementos de classificação. No segundo caso, valores próximos de 1, porque, em sendo os atributos totalmente relevantes para todo o conjunto de classes, eles continuam sem condições de distinguir essas classes.

Portanto, a intensidade de relevância de uma matriz pode servir como parâmetro de avaliação na modelagem do domínio. Se o valor da IR for muito baixo, devemos procurar incluir novos atributos do domínio que sejam relevantes para as classes. Da mesma maneira, se esse valor for muito alto, indica que existem vários atributos dependentes entre si e, devido a isso, devemos eliminar alguns desses atributos e incluir outros, que sejam relevantes para algumas classes e irrelevantes para outras. Para domínios reais, os valores da intensidade de relevância da matriz têm se situado na faixa entre 0,15 e 0,60, com uma freqüência maior em torno de 0,3.

4. 3. 3 Construção da matriz de relevância nebulosa

Dentro da nossa concepção de aquisição de conhecimento, combinando técnicas indutivas e conhecimento preliminar, na forma de relevância semântica, a fidelidade da MRN para um dado domínio é um fator preponderante no sucesso dos resultados obtidos. Portanto, é imperativo a obtenção de uma MRN com um alto grau de representatividade do conhecimento semântico do domínio.

Uma forma simples de se obter uma MRN seria pedir diretamente a um especialista do domínio que a construísse. Entretanto, esta aparente simplicidade leva à duas questões. Primeiro, dependendo do domínio, dificilmente encontraríamos um especialista com conhecimento de conjuntos nebulosos e que, ao mesmo tempo, fosse capaz de quantificar os termos linguísticos inerentes a esses conjuntos. Segundo, mesmo que vários especialistas com essas características fossem disponíveis, qual deles seria o ideal? Será que especialistas sem as características acima mencionadas não seriam dotados de um conhecimento mais profundo e, portanto, não seriam mais úteis?

A nossa proposta, para o problema acima, consiste em combinar, em uma só matriz, o conhecimento de vários especialistas de um mesmo domínio. Esse conhecimento pode ser expresso tanto na forma de MRs (de fácil construção) ou mesmo diretamente na forma de $MRNs$. A essa matriz resultante denominamos de Matriz de Composição de Especialistas (MCE). A idéia da combinação de conhecimento de múltiplos especialistas tem sido usada tanto em técnicas de aquisição de conhecimento cognitivas como semi-automáticas.

Embora essa idéia da combinação não seja nova, o seu uso na combinação de conhecimento via \mathcal{MR}_s e \mathcal{MRN}_s o é.

4.3.3.1 Matriz de composição de especialistas(\mathcal{MCE})

Uma \mathcal{MCE} é uma \mathcal{MRN} formada a partir de duas ou mais \mathcal{MR}_s e/ou \mathcal{MRN}_s fornecidas por diversos especialistas em um mesmo domínio. A composição observará não só as funções de pertinência dos elementos das matrizes componentes, mas também o nível de conhecimento de cada especialista envolvido. Esse nível, embora sendo um conceito subjetivo, será representado por um fator de ponderação.

Seja um domínio \mathcal{D} modelado pelos seguintes elementos:

- $\mathcal{A} = \{ a_1, a_2, \dots, a_{n_a} \}$ - conjunto de n_a atributos;
- $\mathcal{E} = \{ \mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_{n_c} \}$ - conjunto de n_c classes;
- $\mathcal{M} = \{ \mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_{n_m} \}$ - conjunto de n_m \mathcal{MR}_s e/ou \mathcal{MRN}_s fornecidas por especialistas distintos no domínio \mathcal{D} ;
- r_{sij} - relevância do atributo a_i para a classe \mathcal{E}_j na matriz \mathcal{M}_s ;
- $\mathcal{F}r_{sij}(v_{ik})$ - função de pertinência do conjunto nebuloso r_{sij} no valor v_{ik} da matriz \mathcal{M}_s ;
- $\mathcal{P}s_j$ - fator de ponderação (importância) do especialista s para a classe \mathcal{E}_j ;
- $\mathcal{F}Pr_{ij}(v_{ik})$ - função de pertinência do conjunto nebuloso r_{ij} no valor v_{ik} da \mathcal{MCE} ;

com $1 \leq i \leq n_a$, $1 \leq j \leq n_c$, $1 \leq s \leq n_m$ e $1 \leq k \leq n_{v_i}$.

Definimos a função de pertinência de cada elemento da \mathcal{MCE} como sendo a média ponderada das funções de pertinência dos respectivos elementos das matrizes componentes. É dada por

$$\mathcal{F} Pr_{ij}(v_{ik}) = \left(\sum_{s=1}^{n_m} \mathcal{F} r_{sij}(v_{ik}) * \mathcal{P}_{s_j} \right) / \sum_{s=1}^{n_m} \mathcal{P}_{s_j}$$

De uma forma subjetiva, fixaremos os valores de \mathcal{P}_{s_j} num intervalo [$\mathcal{P}_{mín}$, $\mathcal{P}_{máx}$], por exemplo [0, 10]. Onde, $\mathcal{P}_{mín}$ representa o fato de que o conhecimento do especialista na classe \mathcal{E}_j do domínio é nulo ao passo que, $\mathcal{P}_{máx}$ significa que o especialista tem um conhecimento profundo daquela classe do domínio. Teoricamente, a fixação dos valores de \mathcal{P}_{s_j} é um outro problema complexo o que poderia nos levar a definir uma outra matriz de composição de importância de especialistas.

Por que consideramos a média ponderada na composição das funções de pertinência? As alternativas mais viáveis que se apresentam são: média aritmética, média ponderada e interseção ou união de conjuntos nebulosos. A média aritmética em casos de valores diametralmente opostos é extremamente falha. A interseção de conjuntos nebulosos nos daria o conhecimento do consenso que, em muitos casos, poderia resultar em uma grande pobreza semântica. Além do mais a intensidade de relevância da matriz resultante seria mínima. A união dos conjuntos, embora aparentemente produza uma grande riqueza semântica (intensidade de relevância máxima), tem a desvantagem de acolher as opiniões de todos os especialistas de uma forma indiscriminada. Paradoxalmente, em muitos casos este fato pode resultar em uma matriz semanticamente pobre. Portanto, a média ponderada nos permite aproveitar a contribuição de cada especialista, permitindo assim que seja considerado seu nível de especialização em cada classe.

4.3.3.2 Exemplo da construção de uma MCE

Para exemplificar a construção de uma MCE, o domínio escolhido foi um subconjunto das linguagens de programação de alto nível. O motivo da escolha é óbvio: disponibilidade de vários especialistas (consultamos cinco professores do Departamento de Sistemas e Computação da UFPB).

O domínio foi modelado pelos seguintes conjuntos:

CLASSES = { FORTRAN, COBOL, C, PROLOG, LISP }

ATRIBUTOS/VALORES:

A1 = TIPO = {(V11=numérico), (V12 = simbólico), (V13 = misto)}

A2 = CAMPO_DE_APLICAÇÃO = { (V21 = manipulação_simbólica),
(V22 = software_numérico),
(V23 = software_básico),
(V24 = software_comercial) }

A3 = FACILIDADE_DE_OBTENÇÃO = { (V31 = grande), (V32 = média),
(V33 = pouca) }

Queremos esclarecer que os atributos acima foram escolhidos apenas para exemplificar o processo de construção de uma *MCE*. Eles estão longe de representar as características essenciais que seriam necessárias na classificação de uma linguagem. Num caso real, outros atributos (por exemplo, legibilidade, portabilidade, facilidade de aprendizagem, etc.) deveriam ser considerados.

Utilizando uma escala de 0 a 10, pedimos aos especialistas que indicassem o seguinte:

- a) Seu próprio nível de conhecimento em cada uma das linguagens do domínio (fator de ponderação P_{sj});
- b) O grau de relevância que cada par atributo/valor tem na classificação de cada linguagem (valor da função de pertinência $F_{rs_{ij}}(v_{ik})$).

Os dados fornecidos pelos especialistas estão resumidos na TAB. 4.1. Nessa tabela podemos observar que:

- 1) Os pares (A1, V11) e (A2, V22) têm um grau de casamento de 87%
- 2) Os pares (A1, V12) e (A2, V21) têm um grau de casamento de 85%

OBS: O grau de casamento foi obtido utilizando o conceito de distância relativa entre dois pontos (distância_real/distância_máxima), tendo sido calculado diretamente a partir dos valores fornecidos pelos especialistas.

As observações (1) e (2) são um forte indicativo de que os atributos A1 e A2 não são independentes entre si e, portanto, a modelagem do domínio deveria ser revista. De fato, durante a coleta dos dados, notamos uma certa

dificuldade dos especialistas em distinguir claramente o atributo A1(TIPO) de A2 (CAMPO_DE_APLICAÇÃO).

TABELA 4.1 - \mathcal{MRN} s de cinco especialistas

	ESPECIALISTA		A1 TIPO			A2 CAMPO DE APLICAÇÃO				A3 FACILIDADE DE OBTENÇÃO		
	Código	Nível	V11	V12	V13	V21	V22	V23	V24	V31	V32	V33
A P R I S I O	E1	9	10	6	7	0	10	3	4	9	1	0
	E2	10	10	4	7	0	10	2	2	10	0	0
	E3	7	9	4	4	4	9	4	7	10	0	0
	E4	8	10	3	3	3	10	2	1	10	0	0
	E5	10	10	4	7	4	10	1	5	10	0	0
	média			9.9	4.2	5.8	2.1	9.9	2.8	3.7	9.9	0.2
T O B O	E1	4	3	3	3	3	3	1	5	9	1	0
	E2	5	5	5	5	5	2	0	10	10	0	0
	E3	8	5	0	0	0	0	0	10	7	3	0
	E4	3	1	3	3	3	1	2	10	8	2	0
	E5	8	3	0	1	0	0	0	8	9	1	0
	média			3.7	1.6	1.9	1.6	0.9	0.3	8.5	8.5	1.5
D	E1	6	8	8	8	7	10	10	8	9	0	0
	E2	5	5	4	5	4	5	9	2	10	0	0
	E3	9	7	5	5	5	7	10	8	10	0	0
	E4	5	2	6	6	8	3	10	5	10	0	0
	E5	8	7	7	7	7	7	10	8	10	0	0
	média			6.1	6.0	5.9	6.2	6.1	9.8	6.6	9.9	0.0
P R O C E S S O	E1	9	7	10	8	10	6	1	7	5	5	0
	E2	0	-	-	-	-	-	-	-	-	-	-
	E3	5	0	10	5	10	0	0	7	10	0	0
	E4	0	-	-	-	-	-	-	-	-	-	-
	E5	5	3	7	5	7	3	1	1	8	2	0
	média			4.1	9.2	6.4	9.2	3.6	0.7	5.4	7.1	2.9
L I S P	E1	6	7	10	8	10	6	1	7	3	7	0
	E2	5	5	10	8	10	5	5	0	0	10	0
	E3	0	0	10	5	10	0	0	5	7	3	0
	E4	0	-	-	-	-	-	-	-	-	-	-
	E5	1	3	7	5	7	3	1	0	7	2	1
	média			5.8	9.7	7.7	9.8	5.3	2.7	3.6	2.1	7.8

A \mathcal{MCE} correspondente à TAB. 4.1 é dada pela \mathcal{MRN} da FIG. 4.3.

		FORTRAN	COBOL	C	PROLOG	LISP
A1	v11	0.99	0.37	0.61	0.41	0.58
	v12	0.42	0.16	0.60	0.92	0.97
	v13	0.58	0.19	0.59	0.64	0.77
A2	v21	0.21	0.16	0.62	0.92	0.98
	v22	0.99	0.09	0.61	0.36	0.53
	v23	0.28	0.03	0.98	0.07	0.27
	v24	0.37	0.85	0.66	0.54	0.36
A3	v31	0.99	0.85	0.99	0.71	0.21
	v32	0.02	0.15	0.00	0.29	0.78
	v33	0.00	0.00	0.00	0.00	0.01

FIGURA 4.3 - Matriz de composição de especialistas

Da análise dos dados da MRN da FIG. 4.3, podemos concluir que o atributo A3 (FACILIDADE_DE_OBTENÇÃO) é inútil na classificação das linguagens e, portanto, deveria ser excluído do conjunto de atributos do domínio. De fato, observe que as intensidades de relevância dos pares (condições) (A3, v32) e (A3, v33) para o conjunto de classes, da referida matriz, são iguais a 0,248 e 0,002 respectivamente. O primeiro valor (0,248) é bem menor do que 1 (um), enquanto que o segundo é praticamente 0 (zero). Isto significa que, em termos reais, essas duas condições são irrelevantes para classificar uma linguagem e, portanto, poderiam ser eliminadas. Como o atributo A3 tem três valores, com a eliminação dos dois primeiros, ele ficaria reduzido apenas a um valor significativo. Ora, um atributo com apenas um valor não tem nenhum poder de decisão (classificação) e, por isso, não deve ser usado na modelagem de um domínio.

4.4 As Regras inúteis e a MRN

Vimos, no capítulo 3, que o problema semântico é decorrente da geração de regras inúteis. Definimos, também, que uma regra é inútil se não tiver pelo menos uma condição relevante em sua premissa. Isto é, uma regra sem um "mínimo de relevância semântica" não deveria ser utilizada numa base de conhecimento, pois, aos olhos do especialista, ela não tem nenhuma importância na classificação de um certo elemento.

No caso do uso de uma \mathcal{MRN} , nem sempre temos o fato de que uma condição é ou não relevante. Temos, isto sim, que uma condição é relevante para um determinado elemento de classificação com uma certa intensidade, que varia no intervalo $[0, 1]$. A questão que se coloca é, qual deve ser o "mínimo de relevância semântica" de uma regra para que ela possa se considerada útil? Observe, que agora, por exemplo, mesmo uma regra com várias condições relevantes poderia, na prática, ser considerada inútil, caso as intensidades de relevância dessas condições fossem todas muito pequenas (próximas de 0,1), ou seja, caso essas condições fossem muito pouco relevantes. Portanto, para responder à pergunta acima, necessitamos definir uma grandeza que reflita, de uma forma conjunta, a relevância de todas as condições da regra, não apenas de condições individuais. À essa grandeza denominamos de grau de relevância de uma regra.

Grau de relevância de uma regra

Considerando-se que existam:

- uma regra $\mathcal{R}_r = (\mathcal{P}_r, \mathcal{E}_j)$ em que \mathcal{P}_r é a premissa desta regra e \mathcal{E}_j a sua conclusão;
- um conjunto de n_c condições representando a premissa \mathcal{P}_r de \mathcal{R}_r , $\mathcal{P}_r = \{C_1, C_2, \dots, C_{n_c}\}$;
- uma condição $C_m = (a_i = v_{i\kappa})$ em que a_i é um atributo e $v_{i\kappa}$ o seu κ -ésimo valor associado;

Definimos o grau de relevância $Gr(\mathcal{R}_r)$ de uma regra \mathcal{R}_r como sendo

$$Gr(\mathcal{R}_r) = \left(\sum_{C_m \in \mathcal{P}_r} IR(C_m, \mathcal{E}_j) / n_c \right) / (1 + \sigma^2)$$

onde,

- $IR(C_m, \mathcal{E}_j)$ é a intensidade de relevância da condição C_m para o elemento \mathcal{E}_j ;
- σ^2 é a variância da relevância das condições que compõem \mathcal{R}_r .

Regra inútil

Definimos como sendo inútil, toda regra cujo grau de relevância for menor que um certo valor pré-determinado κ , denominado limiar de utilidade.

No nosso entender, o limiar de utilidade, representando o valor mínimo a partir do qual um especialista considera uma regra como semanticamente válida, depende da intensidade de relevância da \mathcal{MRN} fornecida para o domínio, e seu valor poderia representar uma fração dessa intensidade. Na ausência de um estudo mais aprofundado para se obter essa fração, um valor razoável para o limiar de utilidade seria $\kappa = 0,25$, que representa o termo linguístico *pouco relevante*.

5 Soluções do problema semântico via matriz de relevância

Os algoritmos de indução de ADs da família $TDIDT$ apresentam um problema, pois consideram os dados apenas do ponto de vista sintático. Uma solução para este problema consiste no uso de informações semânticas no processo de obtenção da AD . Assim produziríamos Árvores de Decisão Semânticas ($ADSs$), i. e., árvores de decisão cuja construção observou a relevância com que cada condição determina uma dada classe.

É necessário, portanto, obter de alguma forma o conhecimento a respeito do relacionamento semântico entre as condições e as classes. A fonte mais natural para a obtenção deste tipo de informação é o especialista. Para isto, no capítulo 4, definimos uma estrutura chamada Matriz de Relevância (MR), posteriormente estendida para Matriz de Relevância Nebulosa (MRN), que permite obter e representar esse conhecimento do especialista.

5.1 Evolução da solução do problema semântico

Utilizando o conceito de matriz de relevância, as soluções para o problema semântico foram surgindo de uma forma gradual e evolutiva.

O primeiro algoritmo proposto para solucionar o problema semântico foi o ID3X [Mongiovi 90a]. Ele expande a árvore gerada pelo ID3, com a ajuda da MR , de forma a colocar condições relevantes naqueles caminhos em que todas as condições são irrelevantes. O ID3X foi posteriormente generalizado para

expandir \mathcal{AD} s geradas por qualquer algoritmo \mathcal{TDIDT} . Esta generalização foi batizada de ADEX [Cirne 90].

Apesar da expansão semântica de uma árvore \mathcal{TDIDT} melhorar a qualidade desta árvore, colocando condições relevantes nas premissas das regras, ainda restam alguns problemas. Como a expansão só faz acrescentar nós à árvore, esta pode assumir proporções inadequadas. Além disso, só é garantida uma condição relevante na premissa, mesmo quando esta tem um grande número de condições.

Para contornar os problemas apresentados pela expansão das \mathcal{AD} s, propomos a construção direta da \mathcal{AD} levando em consideração não só o aspecto sintático como também o aspecto semântico. Esta construção será guiada por uma Função de Avaliação Pragmática (\mathcal{FAP}) que engloba ambos os aspectos. À este novo algoritmo (que utiliza a \mathcal{FAP} como função de avaliação) chamamos de IDRT ("Induction of Decision Relevant Trees") [Mongioli 91].

Os dois algoritmos propostos acima (ADEX e IDRT) fazem uso da matriz de relevância. Conforme visto no capítulo 4, esse conceito foi estendido para um conceito mais geral, que é o da matriz de relevância nebulosa. Em função disso, analisaremos também, neste capítulo, o que mudar nesses algoritmos para que eles possam usar a maior potencialidade da \mathcal{MRN} .

5.2 Soluções do problema semântico usando \mathcal{MR}

5.2.1 O Algoritmo ADEX

Nesta seção apenas o ADEX será descrito, pois o ID3X nada mais é que o ADEX aplicado ao ID3. Usando o ADEX, o resultado é a expansão da \mathcal{AD} gerada por um algoritmo \mathcal{TDIDT} . A finalidade dessa expansão é colocar a semântica obtida do especialista (contida na \mathcal{MR}) na árvore. Pretendemos, desta forma, evitar que se tenha uma regra onde toda a premissa é irrelevante para a conclusão, isto é, evitar regras inúteis.

A expansão é feita de modo a colocar pelo menos uma condição relevante em todo caminho que vai da raiz a uma folha qualquer. Ou seja, toda regra terá pelo menos um componente relevante em sua premissa. Abaixo, segue o

algoritmo ADEX, que expande uma \mathcal{AD} gerada por um algoritmo \mathcal{TDIDT} em uma \mathcal{ADS} .

ADEX(\mathcal{AD})

PARA TODA folha f (composta pelo elemento de classificação \mathcal{E} e pelo respectivo conjunto de exemplos \mathcal{D}) da \mathcal{AD} TENTAR_EXPANDIR(f)

TENTAR_EXPANDIR(f)

SE nenhuma das condições pertencentes ao caminho ($raiz, f$) é relevante ENTÃO

SEJA a_i um atributo não pertencente ao caminho ($raiz, f$)

SEJA $\mathcal{V}_i = \{\mathcal{V}_{i_1}, \mathcal{V}_{i_2}, \dots, \mathcal{V}_{i_n}\}$ o conjunto de valores que são relevantes para a_i concluir \mathcal{E}

SEJA $\mathcal{A} = \{a_1, a_2, \dots, a_k\}$ o conjunto de todos os a_i 's tais que existe um exemplo pertencente a \mathcal{D} com o valor de a_i pertencente a \mathcal{V}_i

SE \mathcal{A} não é vazio ENTÃO

SEJA a_m o atributo de *melhor* função de avaliação pertencente a \mathcal{A}

SUBSTITUA f pelo nó não-terminal a_m ligado pelos ramos

$\mathcal{V}_{m_1}, \mathcal{V}_{m_2}, \dots, \mathcal{V}_{m_n}$ e * às folhas f_1, f_2, \dots, f_n e f_* , respectivamente

ASSOCIE a cada uma destas folhas o elemento \mathcal{E}

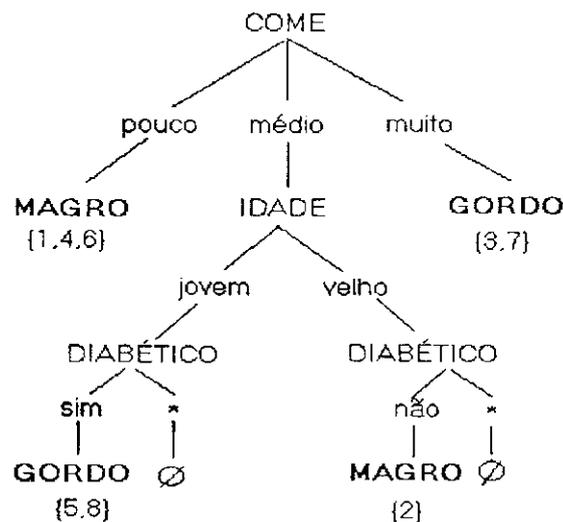
REDISTRIBUA o conjunto de exemplos \mathcal{D} entre elas

TENTAR_EXPANDIR(f_*)

SENÃO marca f como inútil

FIGURA 5.1 - Algoritmo ADEX

No algoritmo ADEX, o símbolo *, filho do atributo a_m , representa todos os valores não relevantes de a_m . A aplicação do ADEX à árvore da FIG. 3.3 (capítulo 3), usando a \mathcal{MR} da FIG. 4.1 (capítulo 4), está na figura abaixo.

FIGURA 5.2 - \mathcal{AD} expandida pelo ADEX a partir de \mathcal{AD} ID3

As regras correspondentes à árvore acima são:

$R_1(\text{ADEX})$: se $\text{COME} = \text{pouco}$ então MAGRO

$R_2(\text{ADEX})$: se $\text{COME} = \text{médio}$ e $\text{IDADE} = \text{jovem}$ e $\text{DIABÉTICO} = \text{sim}$
então GORDO

$R_3(\text{ADEX})$: se $\text{COME} = \text{médio}$ e $\text{IDADE} = \text{velho}$ e $\text{DIABÉTICO} = \text{não}$
então MAGRO

$R_4(\text{ADEX})$: se $\text{COME} = \text{muito}$ então GORDO

Os ramos que têm como valor o * não aparecem nas regras, já que eles não acrescentam nenhuma relevância às regras originalmente obtidas. Ou seja, o símbolo * é transparente para o gerador de regras.

Da comparação das regras originais com as regras expandidas pelo ADEX, notamos que a regra $R_2(\text{ID3})$ inicialmente tida como inútil foi substituída, com o uso do algoritmo ADEX, pela regra útil $R_2(\text{ADEX})$. Além disso, a outra regra inútil $R_3(\text{ID3})$ foi transformada na regra útil $R_3(\text{ADEX})$. A conclusão de que as regras $R_2(\text{ADEX})$ e $R_3(\text{ADEX})$ são úteis se deve ao fato de que, na prática, o atributo DIABÉTICO, pertencente a essas regras, é bastante significativo para distinguir entre MAGRO e GORDO.

Embora no exemplo apresentado todas as regras inúteis foram substituídas por regras úteis, na prática isto nem sempre acontece [Cirne 90] [Mongiovi 90a]. Entretanto, isso não é devido à uma limitação do ADEX e sim à pouca

informação que eventualmente possa estar contida na \mathcal{MR} . Mesmo quando o ADEX não retira regras inúteis seu uso é recompensador, pois as regras inúteis ficam marcadas de forma que quando o especialista for revisar o banco de regras obtido, pode concentrar sua atenção sobre elas.

Até nos casos extremos, a \mathcal{MR} completa ou vazia, o ADEX ainda produz resultados coerentes. Em ambos os casos ele não muda a árvore original. Se a \mathcal{MR} for completa, concluímos que todas as regras são úteis (já que tudo é relevante). Ao passo que, se a \mathcal{MR} for vazia, obtemos que todas as regras são inúteis (como nada é relevante, tudo é inútil).

5.2.2 O Algoritmo IDRT

Como foi visto na seção 5.1, este algoritmo constrói a \mathcal{AD} utilizando uma função de avaliação pragmática (\mathcal{FAP}) para englobar tanto o aspecto sintático como o semântico. Isto é, o algoritmo constrói uma \mathcal{ADS} .

A \mathcal{FAP} aqui proposta é definida de forma a ponderar a informação retirada do \mathcal{CT} (aspecto sintático) com a informação obtida da \mathcal{MR} (aspecto semântico). Esta ponderação será variável, permitindo ao usuário enfatizar, de acordo com a necessidade, o aspecto desejado (sintático ou semântico). O principal fator que influencia a escolha da ponderação é o grau de confiança que se tem na \mathcal{MR} . A \mathcal{FAP} aqui definida implica que a construção da \mathcal{AD} será feita utilizando primeiro os atributos de \mathcal{FAP} mais alta, i. e., o melhor atributo a expandir será aquele de maior \mathcal{FAP} . Portanto, para um dado atributo a_i , o valor da \mathcal{FAP} m foi definido como:

$$m(a_i) = p.\mathcal{N}(a_i) + (1-p).\mathcal{IR}(a_i)$$

onde: p é o fator de ponderação, sendo $0 \leq p \leq 1$,

$\mathcal{N}(a_i)$ é a função de avaliação sintática normalizada e

$\mathcal{IR}(a_i)$ é a intensidade de relevância do atributo a_i para o conjunto de classes \mathcal{E} do domínio (equivalente à $\mathcal{IR}_{a_i \in \mathcal{E}}$ do capítulo 4).

Na ausência de uma melhor indicação para compor uma \mathcal{FAP} equilibrada, um valor razoável para p é $1/2$.

$\mathcal{N}(a_i)$ é uma função de avaliação convencional (i. e., uma função de avaliação sintática) normalizada para o intervalo $[0, 1]$. A normalização deverá ser feita de forma que os "melhores" atributos obtenham os valores mais altos. Por exemplo, se a função de avaliação convencional considerada for a entropia (*ent*), uma boa normalização seria:

$$\mathcal{N}(a_i) = \exp(-ent(a_i))$$

onde: *exp* denota a exponenciação neperiana e *ent* denota a função entropia.

Antes de definir $IR(a_i)$, algumas outras definições se fazem necessárias. Consideremos, então:

- n_{vi} como sendo o número de valores do atributo a_i
- n_E como sendo o número de classes do domínio
- $IRv(a_i = v_k)$ como sendo a intensidade de relevância do valor v_k do atributo a_i
- $\mathcal{MR}[i, j]$ como sendo o elemento (conjunto) da Matriz de Relevância que está na linha i e coluna j

Podemos agora definir a intensidade de relevância do atributo a_i como segue:

$$IR(a_i) = (1/n_{vi}) \cdot \sum_{k=1}^{n_{vi}} IRv(a_i = v_k)$$

onde:

$$IRv(a_i = v_k) = (1/n_E) \cdot \sum_{j=1}^{n_E} f(a_i, v_k, E_j)$$

sendo:

$$f(a_i, v_k, E_j) = \begin{cases} 1, & \text{se } v_k \text{ pertence a } \mathcal{MR}[i, j] \\ 0, & \text{caso contrário} \end{cases}$$

A partir dos mesmos dados de entrada utilizados no exemplo com o ADEX (CT da FIG. 3.2 e \mathcal{MR} da FIG. 4.1 dos capítulos 3 e 4 respectivamente), o IDRT, usando $\exp(-ent(a_i))$ como função de avaliação sintática normalizada, gerou a \mathcal{AD} mostrada na figura abaixo.

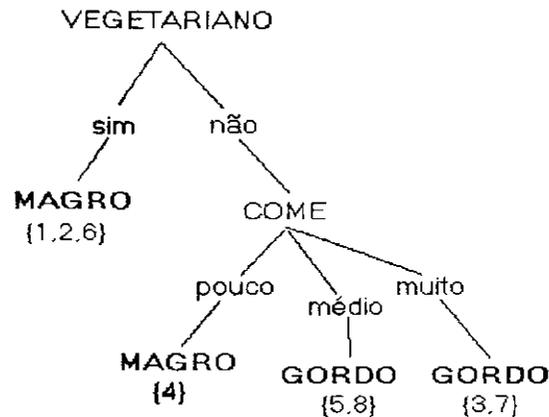


FIGURA 5.3 - \mathcal{AD} gerada pelo IDRT usando entropia normalizada

A \mathcal{AD} da FIG. 5.3, por sua vez, pode ser mapeada no seguinte conjunto de regras:

$R_1(\text{IDRT})$: se VEGETARIANO = sim então MAGRO

$R_2(\text{IDRT})$: se VEGETARIANO = não e COME = pouco então MAGRO

$R_3(\text{IDRT})$: se VEGETARIANO = não e COME = médio então GORDO

$R_4(\text{IDRT})$: se VEGETARIANO = não e COME = muito então GORDO

Analisando as regras fornecidas pelo IDRT e comparando-as com as do ID3 e do ADEX, concluímos que as primeiras são superiores às duas últimas, visto que:

- i) As regras do IDRT, apesar de ligeiramente maiores que as do ID3, em sua totalidade, são corretas do ponto de vista semântico.
- ii) Embora as regras do ADEX sejam todas úteis (i. e., todas têm condições relevantes na premissa) elas são bem maiores que as do IDRT.

O algoritmo IDRT, pela sua própria filosofia de construção, observando os aspectos sintático e semântico, gera uma árvore de decisão com um número reduzido de condições semanticamente irrelevantes, sendo esse número tanto

menor quanto menor for o fator de ponderação p . Entretanto, em sendo um elemento da família $TDIDT$, obviamente que o IDRT apresenta as inconveniências do problema sintático. Contudo, a negatividade desse problema poderia ser minimizada usando técnicas tradicionais de simplificação de ADs , lembrando, porém, que isso implica numa diminuição da acurácia da AD resultante. Embora essas técnicas só observam o aspecto sintático das ADs , elas poderiam ser reformuladas para englobar também o aspecto semântico. Assim, as condições irrelevantes, tanto as sintáticas como as semânticas (que por ventura ainda persistirem), poderiam ser eliminadas.

5.3 Uso da MRN na solução do problema semântico

Uma solução encontrada para o problema semântico foi usar MR (algoritmos ADEX e IDRT). Como a MRN é uma extensão da MR , vamos analisar o que alterar nesses algoritmos para que eles possam explorar a maior expressividade fornecida pela MRN .

Algoritmo ADEX

O algoritmo ADEX usa a MR para expandir uma árvore $TDIDT$. Na expansão o algoritmo consulta a MR para saber se, além das condições existentes na árvore, ainda há condições relevantes para uma dada classe. No caso da MRN , não temos o fato de uma condição ser ou não relevante para uma dada classe. O que temos é o quanto uma condição é relevante para uma classe, que é dado pelo valor da função de pertinência correspondente.

Para o ADEX, uma solução que nos parece adequada, é considerar como sendo relevantes aquelas condições que têm grau de relevância acima de um certo valor mínimo pré-fixado, por exemplo a partir de 0,5, representando o termo linguístico "mais ou menos relevante".

Algoritmo IDRT

Este algoritmo, na definição da função de avaliação pragmática (FAP), usa o conceito de intensidade de relevância, que no caso da MRN nada mais é do

que a Intensidade de Relevância de um Atributo a_i para o conjunto de classes E do domínio (IRa_iE) (veja capítulo 4). Portanto, praticamente nada mudará no IDRT para ele usar uma MRN .

5. 4 Uma solução abrangente para os problemas sintático e semântico

O algoritmo PRISM, descrito no capítulo 3, é uma solução para o problema sintático, mas não contempla o problema semântico. Já o IDRT, embora seja a melhor solução encontrada para o problema semântico, não resolve o problema sintático. Portanto, a saída encontrada foi buscar uma solução abrangente que observe ao mesmo tempo os dois aspectos, o sintático e o semântico.

A idéia da solução global é ter um algoritmo único que reúna as potencialidades do PRISM e do IDRT e que esteja, ao mesmo tempo, livre das deficiências de ambos. Com este objetivo, propomos o RPRISM ("Relevant PRISM") [Cirne 91a]. Ele induz regras modulares (como o PRISM) e usa conhecimento fornecido pelo especialista através de MRs (como o IDRT). A combinação destas características é o que permite solucionar simultaneamente os problemas sintático e semântico.

Com a extensão da MR para a MRN (capítulo 4), o RPRISM foi posteriormente estendido para o FRPRISM ("Fuzzy Relevant PRISM"). Assim, o FRPRISM, que usa conhecimento preliminar na forma de MRN em vez de MR , é simplesmente uma generalização do RPRISM.

5. 4. 1 O Algoritmo RPRISM

O RPRISM é bastante semelhante ao PRISM. A única diferença é que o RPRISM introduz o conceito de probabilidade relevante. Ela é utilizada sempre que for feito um cálculo de probabilidade condicional. Ou seja, o RPRISM difere do PRISM somente porque enquanto o PRISM calcula probabilidades condicionais, o RPRISM computa probabilidades condicionais relevantes. Definimos probabilidade condicional relevante $q(E_j|C)$ como:

$$q(E_j|C) = \begin{cases} p(E_j|C), & \text{se } C \text{ é relevante para } E_j \\ \min(1, n_{CE}) / n_C, & \text{caso contrário} \end{cases} \quad (5.1)$$

onde:

n_{CE} é o número de exemplos que têm a condição C e classe E_j

n_C é o número de exemplos que têm a condição C

A probabilidade relevante é, portanto, definida de forma que condições irrelevantes fiquem com os piores valores possíveis, resolvendo assim o problema semântico. De fato, a pior probabilidade que $p(E_j|C)$ pode ter é $1/n_C$, que define o valor da probabilidade relevante $q(E_j|C)$ quando C não é relevante para E_j .

Durante o processo de formação de uma regra, se todas as condições forem irrelevantes, a escolha da condição de maior probabilidade condicional relevante $q(E_j|C)$ será feita pela probabilidade condicional comum $p(E_j|C)$. Ou seja, já que todas as condições são semanticamente irrelevantes, escolhamos a de maior relevância sintática.

Note que, até nos casos extremos, a \mathcal{MR} completa ou vazia, o RPRISM ainda produz resultados coerentes. Em ambos os casos o mesmo conjunto de regras é gerado. Se a \mathcal{MR} for completa, concluímos que todas as regras são úteis (já que tudo é relevante). Ao passo que, se a \mathcal{MR} for vazia, obtemos que todas as regras são inúteis (como nada é relevante, tudo é inútil). Nas duas situações o processo é guiado pela relevância sintática.

O fato do algoritmo ser baseado no PRISM e produzir regras modulares, e não $\mathcal{AD}s$, soluciona também o problema sintático.

Para os mesmos dados de entrada utilizados com o PRISM e com o IDRT (exemplos do CT da FIG. 3.2 e \mathcal{MR} da FIG. 4.1), o RPRISM gerou as seguintes regras:

R_1 (RPRISM): se COME = pouco então MAGRO

R_2 (RPRISM): se VEGETARIANO = sim então MAGRO

$R_3(\text{RPRISM})$: se VEGETARIANO = não e DIABÉTICO = sim então GORDO

$R_4(\text{RPRISM})$: se COME = muito então GORDO

Veja que, agora, não há mais o problema semântico da regra $R_3(\text{PRISM})$ (substituída pela regra $R_3(\text{RPRISM})$) nem o problema sintático das regras $R_2(\text{IDRT})$ e $R_4(\text{IDRT})$ (substituídas pelas regras $R_1(\text{RPRISM})$ e $R_4(\text{RPRISM})$, respectivamente).

5.4.2 O Algoritmo FRPRISM

O algoritmo RPRISM usa o conceito de probabilidade condicional relevante, que foi definida pela equação (5.1). Essa equação pode ainda ser colocada na forma

$$q(E_j|C) = s * p(E_j|C) + (1-s) * \text{mín}(1, n_{CE})/n_C \quad (5.2)$$

onde,

$$s = \begin{cases} 1, & \text{se } C \text{ é relevante para } E_j \\ 0, & \text{caso contrário} \end{cases}$$

Na equação (5.2), a probabilidade condicional relevante é definida em função da relevância da condição C . Usando \mathcal{MR} , essa relevância é binária (i. e., C é ou não relevante para a classe E_j). Entretanto, com o uso da \mathcal{MRN} , a relevância de C pode assumir qualquer valor no intervalo $[0, 1]$, que é dado pela função de pertinência $\mathcal{F}r_{ij}(C)$, representando a intensidade de relevância da condição C (par atributo $a_i = \text{valor } v_{ik}$) para a classe E_j (vide capítulo 4). Portanto, a equação (5.2) deve ser generalizada de forma a permitir que o algoritmo RPRISM observe, agora, esse novo conjunto de valores da relevância de C .

A forma generalizada da probabilidade condicional relevante, dada pela equação (5.2), pode ser escrita como:

$$Q_f(E_j|C) = \mathcal{F}r_{ij}(C) * p(E_j|C) + (1 - \mathcal{F}r_{ij}(C)) * \text{mín}(1, n_{CE})/n_C \quad (5.3)$$

Observe que a probabilidade $q(E_j|C)$ de (5.1) é agora um caso particular da fornecida por (5.3). De fato, se em (5.3) $\mathcal{F}r_{ij}(C)$ for igual a 1 (um) (C é relevante para E_j) temos $Q_f(E_j|C) = p(E_j|C)$, ao passo que, se $\mathcal{F}r_{ij}(C)$ for igual a 0 (zero) (C é irrelevante para E_j) temos $Q_f(E_j|C) = \min(1, n_{CE})/n_C$.

Assim, o algoritmo FRPRISM nada mais é do que o algoritmo RPRISM, usando a probabilidade condicional relevante generalizada dada pela equação (5.3), e se usar uma \mathcal{MR} , em lugar de uma \mathcal{MRN} , produzirá o mesmo resultado do RPRISM.

5.5 Análise dos resultados parciais

Analisaremos os resultados dos algoritmos indutivos, até então descritos neste trabalho, aplicando-os ao domínio de brinquedo "magro-gordo", definido na FIG. 3.2, que o temos usado como exemplo para ilustrar as idéias apresentadas para resolver os problemas sintático e semântico. Uma análise com domínios reais será feita no capítulo 8.

Mesmo considerando que utilizamos um exemplo de brinquedo para ilustrar os algoritmos apresentados, podemos ter uma idéia da variação do tamanho e da qualidade das BCs geradas na TAB. 5.1. Nela, notamos a evolução sintática (através do número e tamanho médio das regras geradas) e semântica dos algoritmos discutidos neste trabalho. Os algoritmos ADEX, IDRT, RPRISM e FRPRISM, que usam matriz de relevância, geram regras imunes ao problema semântico, sendo que entre eles apenas o RPRISM e o FRPRISM estão isentos do problema sintático.

Como pode ser visto na TAB. 5.1, o RPRISM e o FRPRISM apresentam os mesmos resultados. Isto já era de se esperar, pois em usando uma \mathcal{MR} em lugar de uma \mathcal{MRN} , o FRPRISM se torna igual ao RPRISM.

É importante frisar que a conclusão acima não é definitiva. Primeiro, porque o exemplo utilizado representa um domínio de brinquedo e, como tal, não reflete as características dos domínios reais. Segundo, é que os algoritmos "semânticos" procuram, conforme a matriz de relevância fornecida, minimizar o problema semântico, não garantindo a sua eliminação.

TABELA 5.1 - Comparação entre os diversos algoritmos para o domínio "magro-gordo".

ALGORITMO	Regras gerads			Problema sintático
	Qtde	Tam. médio	Probl. semântico	
Aleatório	7	3,00	0	sim
ID3	4	1,50	2	sim
ADEX	4	2,00	0	sim
IDRT	4	1,75	0	sim
PRISM	4	1,25	1	não
RPRISM	4	1,25	0	não
FRPRISM	4	1,25	0	não

6 Indução semântica de regras modulares, econômicas e generalizadas

6.1 Introdução

Conforme vimos nos capítulos 2 e 4, a definição e utilização de diversos tipos de conhecimento preliminar tem como objetivo principal diminuir o empirismo da indução, procurando melhorar a compreensão da base de conhecimento gerada. Podendo esse conhecimento ser utilizado para simplificar e reformular uma base já gerada ou, preferencialmente, na geração direta dessa base. Vimos ainda, nesses capítulos, que as principais formas de conhecimento preliminar são: custo, generalização e relevância semântica.

No capítulo 3, mostramos que os métodos indutivos a partir de exemplos podem apresentar dois problemas, um de natureza estrutural (o sintático) e outro de natureza semântica (o semântico). Além dessas limitações, nenhum dos algoritmos analisados permite a utilização das principais formas de conhecimento preliminar de forma conjunta. Dessa forma, se existe a necessidade de se trabalhar com custo o usuário será levado a usar o EG2¹, ficando sujeito aos problemas sintático e semântico. Para tentar fugir desses problemas será necessário utilizar o FRPRISM, ficando impossibilitado do uso das informações de custo e/ou de generalizações. Se desejar ponderar entre qualidade e tamanho deverá utilizar o IDRT, que apresenta o problema sintático, além de não contemplar os aspectos de custo e generalização.

A proposição de um novo algoritmo, denominado ISREG (Indutor Semântico de Regras modulares Econômicas e Generalizadas), visa fornecer, de forma

¹ *Economic Generalizer 2, algoritmo da família TDIDT que gera regras econômicas e generalizadas*

conjunta, a possibilidade de utilização das principais formas de conhecimento preliminar, além de eliminar o problema sintático e procurar minimizar a possibilidade de ocorrência do problema semântico.

A idéia básica desse novo algoritmo consiste em combinar os aspectos de modularidade e de relevância semântica do algoritmo FRPRISM (desenvolvido para resolver simultaneamente os problemas sintático e semântico) com os aspectos de custo e generalização contemplados pelo algoritmo EG2. Além de fornecer mecanismos flexíveis que permitam ponderar os principais fatores envolvidos (custo, qualidade semântica, e tamanho), a exemplo do IDRT, que pondera os fatores qualidade semântica e tamanho.

A seguir, inicialmente apresentaremos o algoritmo EG2, de onde tiramos as idéias dos aspectos custo e generalização, e posteriormente apresentaremos o algoritmo ISREG, descrevendo a sua concepção, pseudocódigo e sua materialização através do ambiente A4.

6. 2 Indução de árvores de decisão econômicas e generalizadas

O algoritmo EG2, dependendo da calibragem de certos parâmetros, pode gerar árvores de decisão econômicas e generalizadas. Para isso, utiliza duas formas de conhecimento preliminar: custo e generalização. A primeira forma, para colocar nos nós das árvores os atributos de menor custo, e a segunda forma para produzir árvores compactas.

Para um melhor entendimento do EG2, antes de descrevê-lo, vamos apresentar uma descrição mais detalhada das duas formas de conhecimento preliminar que ele utiliza.

6. 2. 1 Custo

A maioria dos métodos indutivos não contempla a informação sobre o custo de cada atributo. Suponhamos então uma árvore de decisão que tenha como raiz um atributo cujo custo monetário envolvido na sua aquisição seja alto (por exemplo uma Tomografia Computadorizada, ou o teor de pureza de um minério). Toda a base de conhecimento proveniente desta árvore de decisão terá um custo muito alto, além do agravante da maioria dos elementos de

classificação subordinados à sub-árvore ligada ao valor negativo deste atributo de alto custo, poderem ser concluídos sem a sua aquisição.

O termo custo deve ser visto num sentido mais amplo do que o puramente monetário, ele pode representar outras grandezas tais como: grau do risco de vida envolvido, nível de dificuldade de aplicação, disponibilidade da informação, etc.. Podendo, inclusive, haver uma heterogeneidade entre as grandezas o que levará a necessidade de uma modelagem para transformá-las num valor uniforme.

6.2.2 Generalização

Os atributos de um domínio podem ser generalizados com o propósito de facilitar o aprendizado a partir de exemplos [Nuñez 91]. Uma rede de dependência *IS-A* entre os valores dos atributos auxilia a substituição de atributos apresentados nos exemplos por informações mais genéricas. Com esta informação é possível se definir o grau de abstração que o método indutivo deve trabalhar, em função da ocorrência dos valores da generalização definida. Ao fazer uma generalização o algoritmo estará, na realidade, introduzindo um teste de disjunção (*OR*) na premissa de uma regra, pois um valor generalizado reflete a ocorrência de pelo menos um dos valores a ele subordinado.

Para exemplificar os dois tipos de conhecimento preliminar descritos, consideraremos um domínio hipotético, denominado "brinquedo seguro", onde os casos exemplificam situações em que um brinquedo foi qualificado como Seguro (S) ou Perigoso (P) para o manuseio de uma criança, dependendo das combinações de algumas características apresentadas pelo produto. As tabelas TAB. 6.1 e TAB. 6.2 mostram respectivamente, para esse domínio, um possível conjunto de treinamento e uma estrutura de conhecimento preliminar que contempla as informações de custo monetário e de generalização.

Uma propriedade da generalização é que ela pode conter valores que não estão presentes no conjunto de treinamento, permitindo assim que seja induzido um conhecimento ainda não observado. Como é o caso do valor pentágono que, na TAB. 6.2, está generalizado em POLÍGONO, mas não está presente no conjunto de treinamento. É possível, também, usar de recursos lingüísticos para representar operadores lógicos, como: NÃO-PEQUENO, NÃO-

MÉDIO e NÃO-GRANDE, indicando a ocorrência de *NOT* pequeno, *NOT* médio e *NOT* grande.

TABELA 6.1 - Conjunto de Treinamento para o domínio "brinquedo seguro"

Caso	Forma	Cor	Tamanho	Material	Classe
1	quadrado	rosa	grande	couro	S
2	triângulo	rosa	grande	couro	S
3	elipse	azul	grande	couro	S
4	elipse	vermelho	médio	plástico	S
5	círculo	azul	grande	plástico	S
6	círculo	rosa	médio	plástico	S
7	quadrado	vermelho	pequeno	metal	P
8	quadrado	azul	pequeno	madeira	P
9	triângulo	azul	médio	madeira	P
10	triângulo	amarelo	médio	plástico	P
11	elipse	rosa	pequeno	metal	P
12	círculo	vermelho	pequeno	metal	P

TABELA 6.2 - Informações de custo e generalização, para o domínio "brinquedo seguro", adaptado de [Nuñez 91]

Atributo	Custo (\$)	Valor	Generalização
Forma	10	quadrado	IS-A POLÍGONO
		triângulo	IS-A POLÍGONO
		pentágono	IS-A POLÍGONO
		elipse	IS-A CÔNICA
		círculo	IS-A CÔNICA
Cor	30	vermelha	IS-A PRIMÁRIA
		azul	IS-A PRIMÁRIA
		amarela	IS-A PRIMÁRIA
Tamanho	140	grande	IS-A NÃO-PEQUENO
		médio	IS-A NÃO-PEQUENO
		grande	IS-A NÃO-MÉDIO
		pequeno	IS-A NÃO-MÉDIO
		pequeno	IS-A NÃO-GRANDE
		médio	IS-A NÃO-GRANDE
Material	300		

- f_e (π no original) que representa o fator de economia, no intervalo $[0,1]$, onde $f_e=0$ significa desprezar a informação de custo e $f_e=1$ solicita a busca da economia máxima;
- g (ct^3 no original) significando o limiar de generalização, indicando o percentual de aceitação de ocorrência dos valores da generalização, no intervalo $[0,1]$, onde $g=0$ significa desprezar a informação de generalização e $g=1$ só permite a generalização se todos os seus valores estiverem presentes no conjunto de treinamento.

Os valores $f_e=0$ e $g=0$ fazem com que o EG2 gere uma árvore igual a do ID3.

O Código 6.1 apresenta uma versão do algoritmo EG2, adaptada a partir de [Nuñez 91]. A principal modificação introduzida neste código foi a eliminação de uma tentativa de generalização automática existente no código original. Essa generalização automática buscava agrupar valores na seqüência em que se apresentavam na lista de valores. É uma tentativa falha pois somente uma combinação entre todos os valores presentes na lista poderia garantir a generalização. Ocorre que esta combinação pode levar a um grande esforço computacional, desnecessário, uma vez que a informação que ele forneceria pode ser eliciada de um especialista com muito mais precisão. Além do mais, no final do processo, o especialista teria que nomear as possíveis generalizações fornecidas automaticamente pelo algoritmo, tarefa essa que, na prática, nem sempre é viável.

O resultado da análise do conjunto de treinamento do domínio "brinquedo seguro", utilizando o conhecimento preliminar mostrado em TAB. 2.2, encontra-se na FIG. 6.1. O algoritmo foi executado com os parâmetros $f_e=1$ e $g=0.6$, visando o máximo de economia e buscando uma generalização que não exigisse a ocorrência de todos os valores. Desde que 60% dos valores da generalização estejam presentes no conjunto de treinamento o algoritmo irá generalizar a condição, dessa forma, o valor pentágono, presente na tabela de generalizações mas que não ocorreu no conjunto de treinamento, pode ser generalizado.

³ *Completeness threshold*

```

Entrada:  CT - Conjunto de Treinamento
          TG - Tabela de Generalizações
          fe - Fator de Economia
          g  - Limiar da Generalização

Saída:    AD - Árvore de Decisão

EG2 ( CT, fe, g )
  Crie uma AD vazia
  Se os exemplos possuem o mesmo elemento de classificação E
    Coloque o elemento de classificação E na raiz de AD
  Retorne AD
  Senão
    Calcule o ICF de cada atributo de CT utilizando fe
    Selecione o atributo que apresentou o menor ICF
    Crie Lista com as generalizações possíveis e os valores
      não generalizados do atributo selecionado
    Coloque o atributo selecionado na raiz da AD
    Enquanto Lista não for vazia
      vlr_ou_gen ← PrimeiroElementoDaLista
      Crie um subconjunto de CT que contenha apenas o par
        atributo=vlr_ou_gen
      Se o subconjunto tiver o g desejado
        Crie um ramo em AD associado ao valor
        EG2 ( subconjunto, fe, g )
        Retire de CT todos os exemplos mapeados pelo ramo
          construído
      Senão
        Decomponha a generalização em seus valores originais
        Reorganize Lista por ordem do valor da entropia
    fimEnquanto
fim EG2
    
```

CÓDIGO 6.1 - O algoritmo EG2

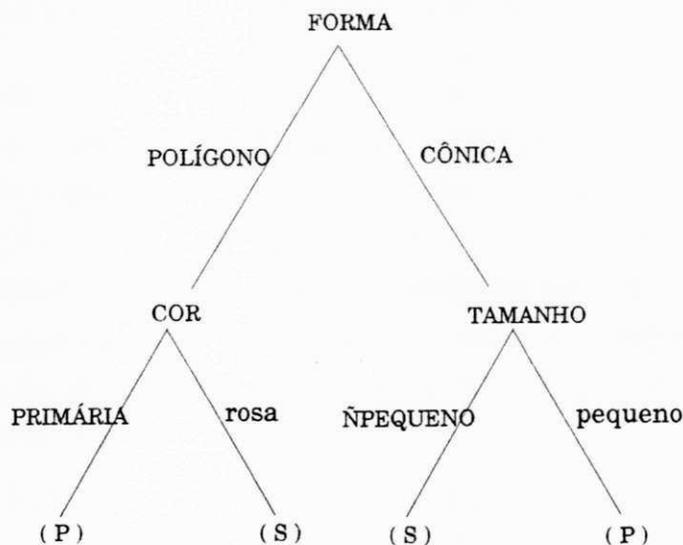


FIGURA 6.1 - Árvore de decisão gerada pelo EG2 para o domínio "brinquedo seguro".

Percorrendo a árvore de decisão gerada pelo EG2 podemos extrair as seguintes regras:

R1	Se Forma = POLÍGONO	& Cor = PRIMÁRIA	Então P	(7,8,9,10)
R2	Se Forma = POLÍGONO	& Cor = rosa	Então S	(1,2)
R3	Se Forma = CÔNICA	& Tamanho = NPEQUENO	Então S	(3,4,5,6)
R4	Se Forma = CÔNICA	& Tamanho = pequeno	Então P	(11,12)

Podemos observar que através da generalização o algoritmo consegue uma árvore extremamente compacta, com um custo médio (média aritmética do custo das condições) de \$50. No capítulo 8 faremos uma comparação entre o custo das bases de conhecimento geradas pelos algoritmos analisados.

6.3 O Algoritmo ISREG

Como vimos na introdução deste capítulo, a idéia básica deste novo algoritmo consiste em combinar os aspectos de modularidade e de relevância semântica do algoritmo FRPRISM com os aspectos de custo e generalização contemplados pelo algoritmo EG2. Além disso, deve fornecer mecanismos flexíveis que permitam ponderar os principais fatores envolvidos (custo, qualidade semântica e tamanho), a exemplo do IDRT, que pondera os fatores qualidade semântica e tamanho. Assim, o ISREG pode ser visto como a fusão desses três algoritmos (FRPRISM, EG2 e IDRT), herdando: do primeiro a indução semântica de regras modulares (contemplando os problemas sintático e semântico), do segundo as idéias de custo/benefício e da generalização, e do último a idéia da ponderação dos fatores envolvidos.

Para cada classe, a exemplo do FRPRISM, o ISREG constrói diretamente um conjunto de regras. A escolha de cada condição, que compõe a premissa de uma dessas regras, está baseada em uma função de avaliação. Essa função será deduzida de tal forma que permita fazer uma relação de custo/benefício, como na função *ICF* do EG2, utilizando as informações estatísticas e de relevância semântica, tiradas respectivamente do conjunto de treinamento e da matriz de relevância (nebulosa ou não), e permitindo a utilização de generalização de valores.

6.3.1 Função de avaliação global

A seleção de uma condição para a formação de uma regra no algoritmo ISREG será feita com base na função de avaliação global (*G_{EF}* - *Global Evaluation Function*). Essa função estabelece uma relação entre o custo da condição selecionada para compor uma regra e a qualidade que essa condição imporá à regra. Através de fatores externos, fornecidos pelo usuário, tanto o custo quanto a busca da qualidade podem ser desprezados ou priorizados. No caso da busca da qualidade ser desprezada, a função selecionará a condição que permitirá a construção de uma base de tamanho reduzido, dentro do limite de custo solicitado. Caso a informação de custo seja desprezada, a função se guiará somente pela qualidade semântica ou pelo tamanho conforme a solicitação. Se for solicitada a qualidade máxima, e não utilizar generalização, o ISREG gerará uma base de conhecimento igual à base gerada pelo FRPRISM. Se forem desprezadas as informações de custo, de qualidade e não utilizar generalização, o ISREG gerará uma base de conhecimento igual à base gerada pelo PRISM.

Para a dedução da *G_{EF}* algumas definições se fazem necessárias, assim sejam:

- **Condição C_m**

Lidando com generalização de valores de atributos, uma condição pode, agora, ser formada por um par "atributo = valor" ($C_m = (a_i = v_{ik})$) ou por um par "atributo = generalização" ($C_m = (a_i = g_{ik})$), onde g_{ik} representa a \mathcal{K} -ésima generalização do atributo a_i .

- **Função custo $FC(C_m)$**

A função custo de uma condição, geralmente igual à do atributo a_i relacionado com essa condição, pode ser de um par "atributo = valor" ($FC(a_i = v_{ik})$) ou de um par "atributo = generalização" ($FC(a_i = g_{ik})$).

- **Ganho de informação de uma condição $I(E_j | C_m)$**

Como vimos para o algoritmo PRISM, capítulo 3, para efeito de escolha da melhor condição, o ganho de informação $I(E_j | C_m)$ será representado por

$\mathcal{P}(\mathcal{E}_j | C_m)$, que representa a probabilidade de um exemplo, dado que contém a condição C_m , possuir a classe \mathcal{E}_j .

• **Probabilidade condicional relevante $Q(\mathcal{E}_j | C_m)$**

Como vimos para o algoritmo FRPRISM, capítulo 5, a probabilidade condicional relevante é uma função que engloba o ganho de informação $I(\mathcal{E}_j | C_m)$, representado por $\mathcal{P}(\mathcal{E}_j | C_m)$, com a relevância da condição C_m para classificar \mathcal{E}_j , é dada pela equação:

$$Q(\mathcal{E}_j | C_m) = g_r \cdot \mathcal{P}(\mathcal{E}_j | C_m) + (1 - g_r) \cdot \frac{\min(1, n_{C\mathcal{E}})}{n_C} \quad (6.2)$$

onde: $n_{C\mathcal{E}}$ é o número de exemplos que possuem a condição C_m e a classe \mathcal{E}_j ;

n_C é o número de exemplos que possuem a condição C_m ;

g_r é o grau de relevância da condição C_m para a classificação de \mathcal{E}_j .

Como g_r varia no intervalo $[0,1]$, podemos observar na equação 6.2 que quando a condição C_m é totalmente relevante para concluir \mathcal{E}_j ($g_r=1$), o valor de Q fica idêntico ao do ganho de informação. Quando ocorre da condição C_m ser totalmente irrelevante para concluir \mathcal{E}_j ($g_r=0$), o valor de Q é penalizado e, na maioria das vezes, é dado por $1/n_C$. Nos demais casos o ganho de informação sofre uma redução em função da relevância da condição C_m .

• **Função G^{EF}**

Seguindo a idéia da relação custo/benefício do algoritmo EG2, a função de avaliação global foi definida como a relação entre o aspecto custo e os aspectos tamanho e qualidade, em fórmulas temos:

$$G^{EF}(\mathcal{E}_j | C_m) = \frac{(\mathcal{F}C(C_m) + 1)^{fe}}{fq \cdot Q(\mathcal{E}_j | C_m) + (1 - fq) \cdot \mathcal{P}(\mathcal{E}_j | C_m)} \quad (6.3)$$

onde: fe é um fator de economia que pondera o custo e fica no intervalo $[0,1]$;

fq é um fator de qualidade semântica, atua no intervalo $[0,1]$;

$FC(C_m)$ é a função que fornece o custo da condição C_m .

O fator f_c determina quanto do custo da condição C_m será utilizado no cálculo da relação custo/benefício. Quando ele assume o valor 1 o valor integral do custo será utilizado, significando a busca da economia máxima. Quando assume o valor 0 a informação de custo é desprezada e a função passa a ser guiada pelo fator de qualidade. Nos demais casos apenas uma parcela do custo será utilizado na relação.

O fator f_q determina o nível de qualidade semântica desejado. Ele direciona o ganho de informação em busca da qualidade máxima ($f_q=1$), da quantidade mínima de regras ($f_q=0$) ou em busca de um equilíbrio entre os dois níveis.

A melhor condição será aquela que apresentar o menor valor de G_{EF} , significando, dentro dos parâmetros pré-definidos, a que apresenta maior qualidade global.

6.3.2 Procedimento geral do ISREG

O ISREG induz diretamente regras de produção, para cada elemento de classificação existente no conjunto de treinamento, sem criar uma árvore de decisão. A formação das regras é feita com base na seleção de condições que apresentem o menor valor para a função de avaliação global (G_{EF}). Os exemplos mapeados por cada regra gerada são retirados do conjunto de treinamento. A cada seleção de um novo elemento de classificação, o conjunto de treinamento é restaurado, *i.é.*, os exemplos retirados retornam, compondo a sua forma original. A condição de término do algoritmo exige que todos os elementos de classificação existentes no conjunto de treinamento tenham sido escolhidos e todos os exemplos tenham sido mapeados.

O código 6.2 apresenta um procedimento geral para o algoritmo ISREG.

Duas situações especiais podem acontecer durante a execução do algoritmo e merecem ser discutidas: um conflito de classificação e um empate entre duas condições durante o processo de seleção.

Existe uma rotina que trata a ocorrência de um conflito de classificação, que é identificado quando todas as condições já foram selecionadas e ainda restam

mais de um elemento de classificação no conjunto de treinamento. Na versão atual do algoritmo essa rotina apenas guarda a regra que naquele instante está em conflito para no final da execução fornecer uma referência cruzada das regras em conflito. Esse comportamento foi adotado por entendermos que somente o especialista pode determinar qual das regras em conflito é mais importante e deve permanecer na base de conhecimento.

```

Entradas: CT      - Conjunto de Treinamento
          TG      - Tabela de Generalizações
          MRN     - Matriz de Relevância Nebulosa
          g       - limiar de generalização
          fe      - fator de economia
          fq      - fator de qualidade

Saída:   BC      - Base de Conhecimento

ISREG (CT, TG, MRN, fe, fq, g)
  Para cada elemento de classificação E em CT
    Enquanto o elemento de classificação E existir em CT
      ListaCondições ← ∅
      CT_AUX ← CT
      TG_AUX ← TG
      Repita
        Calcule a GEF para cada condição Cm em CT_AUX
        Selecione a Cm com o menor valor GEF
        Retire de CT_AUX todos os exemplos que não possuam a
          Cm selecionada
        Inclua a Cm selecionada em ListaCondições
        Se Cm é uma generalização & não está compatível com g
          Retire a Cm selecionada de ListaCondições
          Restaure CT_AUX retornando os exemplos retirados
          Retire de TG_AUX a generalização selecionada
        Senão
          Retire de CT_AUX o atributo da Cm selecionada
      Até que CT_AUX contenha apenas o elemento de classificação E
        & não tenha ocorrido conflito
      Se ocorreu um conflito
        TrataConflito
      NovaRegra ← SE ListaCondições ENTÃO E
      Inclua NovaRegra em BC
      Retire de CT todos os exemplos mapeados por NovaRegra
      Restaure a TG original
    fimEnquanto
  Restaure o CT original
fimPara
Retorne ( BC )
Fim ISREG

```

CÓDIGO 6.2 - O algoritmo ISREG

Quando ocorre um empate nos valores encontradas pela função GEF na escolha da melhor condição para compor a regra, o algoritmo segue os seguintes critérios de desempates, por ordem de prioridade:

- Uma condição com generalização é sempre escolhida quando empata com uma condição formada apenas por um valor;
- Num empate entre condições com generalizações ou entre condições apenas com valores, o algoritmo estabelece uma ordem decrescente de escolha entre custo, relevância e tamanho, baseada nos valores informados nos parâmetros. Dessa forma se o usuário solicitou uma execução com os parâmetros $fe=0.8$ e $fq=1.0$, o critério de desempate será, na ordem, maior relevância semântica, menor custo, menor tamanho.
- Caso o desempate falhe nos dois itens anteriores, o que é praticamente impossível, a primeira condição dentre as que empataram será a escolhida.

6.3.3 O algoritmo ISREG no ambiente A4

O algoritmo ISREG foi implementado em C++ e pertence à classe ALGORITMOS do ambiente A4 (vide capítulo 9). O A4 é um ambiente que tem por finalidade auxiliar todo o processo de aquisição de conhecimento indutivo a partir de exemplos, desde a modelagem do mundo real em exemplos e conhecimento preliminar, até o tratamento das saídas geradas pelos métodos indutivos [Vasco 92a][Vasco 93d]. Na classe algoritmos estão os métodos indutivos que geram uma base de conhecimento.

Para uma boa implementação de um algoritmo é necessário que testes comparativos sejam realizados, principalmente se ele se propõe a melhorar os resultados de outros algoritmos, como é o caso do ISREG. Inicialmente no A4 estavam disponíveis os algoritmos ID3, PRISM, IDRT e FRPRISM, contudo, não haviam sido submetidos a testes mais rigorosos. Como o ISREG usa princípios do EG2 e desses dois últimos algoritmos, e objetivando a realização de testes comparativos, tivemos que ainda implementar o algoritmo EG2, além de rever, uniformizar e testar criteriosamente os algoritmos já implementados.

O A4, então em sua primeira versão, não dispunha de facilidades apropriadas para testar, de forma comparativa, os algoritmos com domínios reais. Além do mais, por ser um algoritmo parametrizado, o ISREG requer facilidades adicionais que possibilitem um estudo sistemático da variação desses parâmetros. Assim sendo, o ISREG provocou uma série de mudanças no ambiente A4, com o propósito de adequar suas ferramentas e uniformizar processos.

Dentre as mudanças e implementações realizadas no ambiente A4, podemos citar, como as mais relevantes, as seguintes:

- um método que fornece a documentação completa de um domínio catalogado no ambiente, informando as classes, atributos, valores, uma referência cruzada entre atributos e os valores que ocorrem para esse atributo, as formas de conhecimento preliminar disponíveis para o ambiente e a tabela de exemplos, se o usuário assim o desejar;
- uma interface única de passagem de parâmetros para os algoritmos, onde o algoritmo informa um valor padrão e os limites inferior e superior para cada parâmetro utilizado, ficando a cargo da interface a crítica de cada parâmetro;
- todos os métodos de avaliação quantitativa e qualitativa da base de conhecimento gerada;
- um método que permite selecionar, de forma aleatória, exemplos para o teste de acurácia, conforme um percentual informado pelo usuário;
- métodos genéricos de geração do arquivo de saída com as regras geradas, relação das regras inúteis e uma referência cruzadas das regras em conflito;

O algoritmo ISREG solicita a informação de três parâmetros (f_e , f_q e g) provocando a ocorrência de um grande número de combinações possíveis. Isso pode levar o usuário a executar o algoritmo diversas vezes até chegar num resultado aceitável. Procurando diminuir esse esforço gasto por parte do usuário, disponibilizamos no A4 a possibilidade desse usuário informar apenas um valor de variação para os parâmetros, e com base no valor

informado, as execuções necessárias são realizadas. No final desse processo é fornecido um relatório com as principais características de cada base gerada por cada uma das execuções realizadas, permitindo assim, que o usuário escolha a combinação de parâmetros que irá produzir a base de conhecimento de melhor desempenho para o seu domínio.

Os resultados do ISREG para o domínio "brinquedo seguro" e quatro domínios reais encontram-se no capítulo 8, onde também fazemos, para esses domínios, uma análise comparativa com os demais algoritmos do ambiente A4.

6.4 Conclusão

O algoritmo ISREG procura preencher uma lacuna existente entre os métodos indutivos a partir de exemplos salientada pela inexistência de um método que utilize, de forma conjunta, as principais formas de conhecimento preliminar disponíveis. Ao mesmo tempo elimina o problema sintático e busca minimizar a ocorrência do problema semântico que fragilizam as bases de conhecimento geradas pelos métodos indutivos. Através da variação de seus parâmetros é possível direcionar o resultado para um nível de qualidade desejado, em função do custo, do tamanho da base ou da qualidade semântica das regras geradas. Essa flexibilidade sugere uma maior possibilidade de apresentar melhores resultados que seus antecessores, apresentados nos capítulos anteriores.

7 Qualidade de uma base de conhecimento

7.1 Introdução

O resultado final de um processo de aquisição de conhecimento por métodos indutivos é uma base de conhecimento que será utilizada por um sistema baseado em conhecimento. O sucesso desse sistema dependerá fortemente da sua capacidade de atender satisfatoriamente às necessidades e às conveniências do usuário. Para isso, é necessário que a base de conhecimento gerada possua um conjunto de características que possibilitem ao sistema resolver problemas de uma forma simples e precisa e, ao mesmo tempo, apresentar, ao usuário, explicações claras e convincentes a respeito dessa solução. Portanto, teoricamente, para se avaliar a qualidade de uma base de conhecimento basta identificar e quantificar esse conjunto de características.

Embora essa idéia de avaliação seja simples, a sua materialização na prática não o é. A tarefa de indentificar e, principalmente, quantificar esse conjunto de características não é trivial. Isso porque, essas características não apenas envolvem aspectos objetivos, como tamanho e acurácia, que são facilmente mensuráveis, mas também, e principalmente, envolvem aspectos subjetivos, como compreensibilidade e credibilidade, que são de difícil mensuração.

Na avaliação das bases de conhecimento geradas pelos métodos indutivos, somente os aspectos objetivos têm sido observados, em particular, quase que exclusivamente, apenas os fatores acurácia e tamanho. Para essas bases não tem se apresentado uma forma, também automática, de avaliação dos aspectos subjetivos. Essa análise, quando é feita, sempre envolve a presença do especialista no domínio, que devido a sua indisponibilidade de tempo e custo da sua hora de trabalho, traz de volta o problema inicial que motivou o surgimento dos métodos automáticos.

Portanto, para manter o espírito automático do processo de aquisição de conhecimento, o que nos interessa na avaliação de uma base de conhecimento é dispor de um mecanismo que, de forma automática, permita avaliar tanto os aspectos objetivos como, e primordialmente, os aspectos subjetivos. Assim, mediríamos a base para saber a sua qualidade antes de utilizá-la em um sistema baseado em conhecimento, tendo dessa forma a opção de utilizar a melhor base visando um aumento da satisfação do usuário, que é o objetivo maior de um produto.

7.2 Formas de avaliação de uma base de conhecimento

Em princípio, a avaliação de uma base de conhecimento deve ser feita procurando-se atender aos interesses específicos do domínio. No entanto, podemos verificar que existem pontos em comum a qualquer domínio. A identificação dos problemas sintático e semântico ressaltaram deficiências nas bases de conhecimento, independentemente do domínio, *i.e.*, nenhuma área que seja usuária em potencial de um sistema baseado em conhecimento está imune da possibilidade de apresentar os referidos problemas. A eliminação desses problemas conduz à bases de conhecimentos de melhor qualidade, tanto quando examinadas sob a ótica dos aspectos objetivos como também dos aspectos subjetivos. Outro fator importante na construção de um sistema baseado em conhecimento é a relação custo/benefício envolvida na utilização desse sistema, que, no nosso entender, tanto pode ser um aspecto objetivo como subjetivo. Assim sendo, podemos dividir a avaliação de uma base de conhecimento sob os aspectos: objetivos, custo e subjetivos.

7.2.1 Aspectos objetivos

O problema sintático tem influência direta no tamanho das regras e conseqüentemente no tamanho da base de conhecimento gerada, sendo que a sua ausência, em geral, produz regras de menor tamanho. Por outro lado, independentemente do tamanho, essa base pode ter um bom nível de acerto nas classificações que lhe são submetidas. Dessa forma, na avaliação de uma base de conhecimento, basicamente temos dois aspectos que podem ser considerados objetivos: tamanho e acurácia.

Aspecto tamanho

Em uma análise pelo tamanho, vários têm sido os critérios adotados na avaliação da qualidade de uma base de conhecimento. No caso de árvores de decisão, [Pires 93] sugere usar o número de nós, o número de folhas, que correspondem ao número de regras, e o comprimento médio dos caminhos da árvore, assumindo-se que todas as folhas são equiprováveis. Elomaa e Holsti [Elomaa 89], usam o número de nós (inclusive as folhas) para as árvores de decisão e o número total de seletores (condições) para as listas de decisão. Já [Gams 89] usa o produto do número de regras (caminhos no caso de ADs) vezes o comprimento médio de uma regra (caminho) vezes o número de disjunções internas.

Neste trabalho, lidamos com bases de conhecimento representadas tanto na forma de regras como na de árvore de decisão, que, evidentemente podem facilmente ser transformadas em um conjunto equivalente de regras. Assim, para o aspecto tamanho consideraremos o número de regras geradas e o tamanho médio dessas regras, que é obtido pelo número médio de condições por regra. Obviamente, que, com respeito ao fator tamanho, quanto menores forem esses números, melhor será a qualidade da base gerada.

Aspecto acurácia

A análise pela acurácia, geralmente, é realizada dividindo-se a tabela de exemplos em duas partes. A partir da primeira parte é induzida a base de conhecimento, enquanto que a segunda parte (conjunto de teste) é utilizada posteriormente para verificação da base gerada. A frequência de acertos, ou erros, na classificação do conjunto de teste mede a qualidade da base. No nosso caso, consideraremos a frequência de acertos, que, evidentemente, quanto maior melhor será a base.

7.2.2 Aspecto custo

As informações sobre custo não são necessariamente monetárias, facilmente mensuráveis, mas podem ser vistas também em função de risco de vida, dificuldades de aplicação, urgência da informação, etc., que são de difícil mensuração. Assim, o aspecto custo tanto pode assumir um caráter objetivo

como subjetivo e, por isso, não o enquadraremos em nenhuma dessas duas classificações.

Como foi visto anteriormente, capítulo 5, o custo de cada atributo que integra o conjunto de treinamento, é uma forma de conhecimento preliminar que pode ser utilizado, ou não, na construção da base de conhecimento. Na análise da base de conhecimento sob o aspecto custo, utiliza-se este conhecimento preliminar para determinar o custo médio de uma classificação [Nuñez 91], *i.e.*, quanto custará, em média, ao usuário, realizar uma classificação através do sistema baseado em conhecimento que utilizará a base gerada. Sob este aspecto, obviamente, a melhor base será aquela que apresentar o menor custo.

Tratando-se de uma forma de conhecimento preliminar, o ideal é que esse custo seja utilizado na geração da base de conhecimento, auxiliando na seleção do atributo que irá compor a condição da regra. O algoritmo que assim proceder tende a apresentar uma base de melhor qualidade sob o ponto de vista do aspecto custo.

7.2.3 Aspecto subjetivos

A experiência tem nos mostrado que, em geral, para domínios reais, ao examinar uma base de conhecimento gerada por um algoritmo indutivo tradicional¹, o especialista, desse domínio, concorda parcialmente com os resultados que lhe são apresentados. Parcialmente, porque ele aceita uma boa parte das regras geradas, acha *incompreensíveis* e fica em dúvida com uma segunda parte e, em definitivo, não tem *credibilidade* nas regras da terceira parte restante, rejeitando-as por completo. Essa rejeição varia muito com a complexidade do domínio, bem como com o seu nível de modelagem. Contudo, o uso de conhecimento preliminar por parte dos algoritmos, principalmente na forma de relevância semântica, provoca uma diminuição acentuada no percentual de regras rejeitadas.

A *incompreensibilidade* das regras (aspecto subjetivo) tanto pode ser provocada pelo problema sintático como pelo problema semântico. O primeiro, porque pode induzir regras longas e com condições desnecessárias, enquanto que o segundo, porque causa o aparecimento de condições irrelevantes nas

¹algoritmo que não usa nenhuma forma de conhecimento preliminar e não embute simplificações.

premissas das regras, tornando-as, assim, incompreensíveis. O problema semântico também pode ser o responsável pelo outro aspecto subjetivo, que é a falta de *credibilidade* nas regras geradas. Pois, se essas regras apresentarem um baixo grau de relevância, o especialista poderá não acreditar nelas e, portanto, na prática, serão consideradas inúteis.

Do ponto de vista dos aspectos subjetivos, apesar de envolver os problemas sintáticos e semântico, a avaliação de uma base de conhecimento passa apenas pela solução do problema semântico. Isto porque, o problema sintático interfere apenas em parte do aspecto *compreensibilidade*, que é a parte referente ao tamanho das regras, e esse aspecto foi anteriormente contemplado na avaliação do aspecto objetivo "tamanho" (vide seção 7.2.1). Assim, na nossa abordagem, a avaliação de uma base, do ponto de vista dos aspectos subjetivos, consiste apenas em quantificar, através da relevância semântica, o significado das regras que compõem essa base, ou seja, avaliar a base sob a ótica do aspecto semântico.

Aspecto semântico

No nosso entender, uma base de conhecimento pode ser considerada semanticamente correta se todas as suas regras forem úteis na classificação do domínio, i.e., se as premissas de todas as regras forem totalmente relevantes.

Somente um especialista no domínio em questão, analisando cada uma das regras geradas, pode avaliar a base sob o aspecto semântico. Esse processo apresenta um resultado duvidoso devido às questões já levantadas de indisponibilidade de tempo e do custo da hora de trabalho do especialista, além da forte subjetividade presente neste tipo de avaliação, que pode levar a resultados diferentes entre especialistas. Contudo, esse aspecto é muito importante pois ele é um indicativo do grau de compreensibilidade e credibilidade que se tem em uma base gerada, devendo-se assim, dentro do possível, automatizar o processo de avaliação desse aspecto semântico.

7.3 Utilização da relevância semântica na avaliação dos aspectos subjetivos de uma base de conhecimento

Na nossa abordagem, a relevância semântica desempenha um papel fundamental na tentativa de minimizar a ocorrência do problema semântico, que é o causador principal da presença negativa dos aspectos subjetivos em uma base de conhecimento. A matriz de relevância como forma específica de representação de conhecimento preliminar é uma ferramenta simples e objetiva. Apesar da participação do especialista na sua construção, sua simplicidade evita que se recaia nos inconvenientes da aquisição de conhecimento cognitiva (vide capítulo 4).

Uma base de conhecimento é composta de regras que possuem a forma "se <premissa> então <elemento de classificação>". A premissa, por sua vez, é composta por uma conjunção de condições, onde cada condição é formada por um par "atributo=valor". A matriz de relevância possui valores no intervalo [0,1] para designar a relevância da condição para a conclusão de um determinado elemento de classificação. Dessa forma, podemos valorar a importância da condição, e conseqüentemente da premissa, para concluir o elemento de classificação presente na regra, conseguindo-se, assim, um grau de relevância para a regra.

Propagando essa medida de relevância da regra, podemos aplicá-la à base de conhecimento definindo-lhe um grau de relevância. Com base na matriz de relevância, a nova medida determinaria a qualidade semântica da base de conhecimento, indo da total irrelevância, pior base, até a total relevância, melhor base.

Com esse grau de relevância, passaríamos a ter uma medida da base de conhecimento sob o aspecto semântico, permitindo, assim, uma avaliação qualitativa dos aspectos subjetivos.

No caso da utilização do conhecimento preliminar denominado generalização, a condição pode assumir a forma "atributo=generalização", sendo, nesse caso, necessário definir como será calculada a relevância de uma generalização.

7.4 Definição do grau de relevância de uma base de conhecimento

O grau de relevância de uma base de conhecimento é um número, no intervalo $[0, 1]$, que mostra o nível de qualidade semântica dessa base, sendo obtido em função das informações contidas na matriz de relevância. Para definirmos a equação que deve ser utilizada no cálculo desse grau, é necessário, inicialmente, introduzirmos algumas definições dos elementos que compõem seu ambiente de atuação.

Assim sendo, sejam:

- $BC = \{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \dots, \mathcal{R}_{n_r}\}$ uma base de conhecimento formada por um conjunto de n_r regras;
- $\mathcal{R}_r = (P_r, E_j)$ uma regra de premissa P_r e de conclusão E_j ;
- $\mathcal{E} = \{E_1, E_2, E_3, \dots, E_{n_E}\}$ um conjunto de n_E elementos de classificação;
- $\mathcal{P}_r = \{C_1, C_2, C_3, \dots, C_{n_c}\}$ um conjunto de n_c condições representando a premissa da regra \mathcal{R}_r ;
- $C_m = (a_i = v_{i\kappa})$ uma condição em que a_i é um atributo e $v_{i\kappa}$ seu valor associado;
- $C_m = (a_i = g_{i\kappa})$ uma condição em que a_i é um atributo e $g_{i\kappa}$ uma generalização associada a esse atributo;
- $\mathcal{A} = \{a_1, a_2, a_3, \dots, a_{n_a}\}$ um conjunto de n_a atributos;
- $\mathcal{V}(a_i) = \{v_{i1}, v_{i2}, v_{i3}, \dots, v_{in_v_i}\}$ um conjunto de nv_i valores associados ao atributo a_i ;
- $\mathcal{G}(a_i) = \{g_{i1}, g_{i2}, g_{i3}, \dots, g_{in_g_i}\}$ um conjunto de ng_i generalizações definindo os valores associados ao atributo a_i , sendo $0 \leq ng_i < nv_i$;

- $g_{ik} = \{v_{i1}, v_{i2}, v_{i3}, \dots, v_{invg_{ik}}\}$ um conjunto de nvg_{ik} valores associados ao atributo a_i pertencentes ao conjunto de generalizações, sendo $0 \leq nvg_{ik} < nv_i$;
- $\mathcal{MR} = (mr_{ij})$ uma matriz de relevância nebulosa, com $1 \leq i \leq n_a$ e $1 \leq j \leq n_E$, onde cada elemento dessa matriz indica a relevância do atributo a_i para a classificação do elemento de classificação E_j e pode ser definido por: $mr_{ij} = \{(\rho / v_{ik}) \mid \rho \in \mathbb{R}, 0 \leq \rho \leq 1 \text{ e } v_{ik} \in \mathcal{V}(a_i)\}$ sendo \mathbb{R} o conjunto dos números reais;
- \mathcal{F}_{ijk} uma função que fornece a relevância do atributo a_i para o elemento de classificação E_j no valor v_{ik} ;
- $\mathcal{FC}(C_m)$ uma função que fornece o custo da condição C_m .

Estabelecendo ainda, que o valor que indica qualquer medida de relevância é sempre um número real no intervalo $[0,1]$, onde o valor 0 representa a total irrelevância e 1 indica a total relevância do elemento que estiver sendo analisado, podemos então definir:

- **grau de relevância de uma condição** como sendo a relevância de um par "atributo=valor" ou "atributo=generalização" para a conclusão do elemento de classificação indicado. Essa equação possui duas formas dependendo do formato da condição que está sendo analisada, no caso de uma condição com generalização, foi realizado um estudo para verificar o comportamento dos resultados obtidos pela equação proposta com o propósito de validar sua definição (APÊNDICE A)[Alexandre 94a]. Essa relevância é definida por:

$$\mathcal{R}(C_m, E_j) = \begin{cases} \frac{1}{nvg_{ik}} \sum_{k=1}^{nvg_{ik}} \mathcal{F}_{ijk} & \text{se } C_m = (a_i = g_{ik}) \\ \mathcal{F}_{ijk} & \text{se } C_m = (a_i = v_{ik}) \end{cases} \quad (7.1)$$

sendo σ^2 a variância da relevância dos valores que compõem a generalização g_{ik} .

- **grau de relevância de uma regra** como sendo a relevância da premissa para concluir o elemento de classificação constante da regra. Quando esse grau tem valor zero identifica-se uma regra totalmente inútil, sem nenhum valor prático. Na realidade, a identificação de uma regra inútil deve se basear num valor que denominamos limiar de utilidade de uma regra. O limiar de utilidade da regra determina o grau de relevância a partir do qual uma regra pode ser considerada útil. O grau de relevância de uma regra é obtido pela aplicação da equação (7.2)

$$GRR(\mathcal{R}_r) = \frac{1}{|\mathcal{P}_r|} \frac{\sum_{C_c \in \mathcal{P}_r} \mathcal{R}(C_c, \mathcal{E}_j)}{1 + \sigma^2} \quad (7.2)$$

onde σ^2 é a variância da relevância das condições que compõem a regra \mathcal{R}_r .

- **grau de relevância de uma base de conhecimento** como sendo a média corrigida do grau de relevância das regras que compõem a base. Uma base composta somente por regras totalmente inúteis, apresentará um grau de relevância igual a zero. O cálculo desse grau é determinado por:

$$GRBC(BC) = \frac{\frac{1}{n_r} \sum_{r=1}^{n_r} GRR(\mathcal{R}_r)}{1 + \sigma^2} \quad (7.3)$$

com σ^2 representando a variância da relevância das regras que compõem a base de conhecimento BC .

A variância foi utilizada para evitar distorções nos resultados, observadas quando existe um desvio padrão alto entre os valores utilizados no cálculo. Essas distorções foram verificadas após a primeira definição do cálculo da relevância de uma base de conhecimento [Mongiovi 93c].

A necessidade dessa correção deve-se ao fato de que, na nossa avaliação, as distorções apresentadas nos resultados que utilizavam somente a média no cálculo, poderiam comprometer a análise de uma base de conhecimento. Para exemplificar, suponhamos uma regra \mathcal{R}_1 com uma premissa composta pelas

condições C_1 e C_2 concluindo \mathcal{E} , sendo que uma das condições é totalmente relevante (1,0) e a outra é totalmente irrelevante (0,0) para concluir \mathcal{E} . E uma outra regra \mathcal{R}_2 com uma premissa composta pelas condições C_3 e C_4 também concluindo \mathcal{E} , sendo que agora essas duas condições apresentam o mesmo valor médio de relevância (0,5). O grau de relevância das duas regras apresentam o mesmo valor quando calculados pela média (0,5). Com as equações apresentadas o resultado para a regra \mathcal{R}_2 [$GRR(\mathcal{R}_2) = 0,5$] se manteria, mas para a regra \mathcal{R}_1 o resultado seria outro [$GRR(\mathcal{R}_1) = 0,4$], ou seja, a regra \mathcal{R}_1 é penalizada por possuir uma condição completamente irrelevante.

A TAB. 7.1 apresenta alguns exemplos que mostram o comportamento dos valores calculados pela média e dos mesmos valores ajustados pela variância (as parcelas tanto podem representar as relevâncias das condições de uma regra, como as relevâncias das regras de uma base de conhecimento).

TABELA 7.1 - Comportamento da média versus o valor ajustado no cálculo da relevância.

Parcelas					Média	Valor Ajustado
1,00	0,00				0,50	0,40
1,00	0,00	0,00			0,33	0,27
1,00	0,00	0,00	0,00		0,25	0,21
1,00	0,00	0,00	0,00	0,00	0,20	0,17
0,00	1,00				0,50	0,40
0,00	1,00	1,00			0,67	0,54
0,00	1,00	1,00	1,00		0,75	0,63
0,00	1,00	1,00	1,00	1,00	0,80	0,69
0,50	0,50				0,50	0,50
0,90	0,10				0,50	0,43
0,50	0,50	0,75	0,75	1,00	0,70	0,68

Um outro comportamento desejável e que é atendido pelas equações (7.2) e (7.3), diz respeito a convergência. Essa característica se faz necessária porque, no cálculo pela média, a presença de valores altos tendem atrair a média final, dessa forma, a existência de valores com relevância alta, mesmo existindo elementos irrelevantes, podem oferecer como resultado final um grau de relevância acima do desejável. Se ocorrer uma rápida convergência poderíamos ter, por exemplo, uma base de conhecimento que embora contendo regras irrelevantes, apresentasse um alto grau de relevância devido

a uma presença maior de regras totalmente relevantes. Um comportamento aceitável seria o de convergir mais lentamente para 1 quando fosse mais forte a presença de elementos relevantes e convergisse mais rapidamente para zero quando fosse maior a presença de elementos irrelevantes.

Para verificarmos esse comportamento admitamos os seguintes casos extremos:

- Caso 1 - Uma regra com uma condição totalmente irrelevante e as demais totalmente relevantes.
- Caso 2 - Uma regra com uma condição totalmente relevante e as demais totalmente irrelevantes.

O GRAF. 7.1 mostra a convergência dos valores calculados como grau de relevância pela média e pelas equações definidas, aqui chamados de valores ajustados.

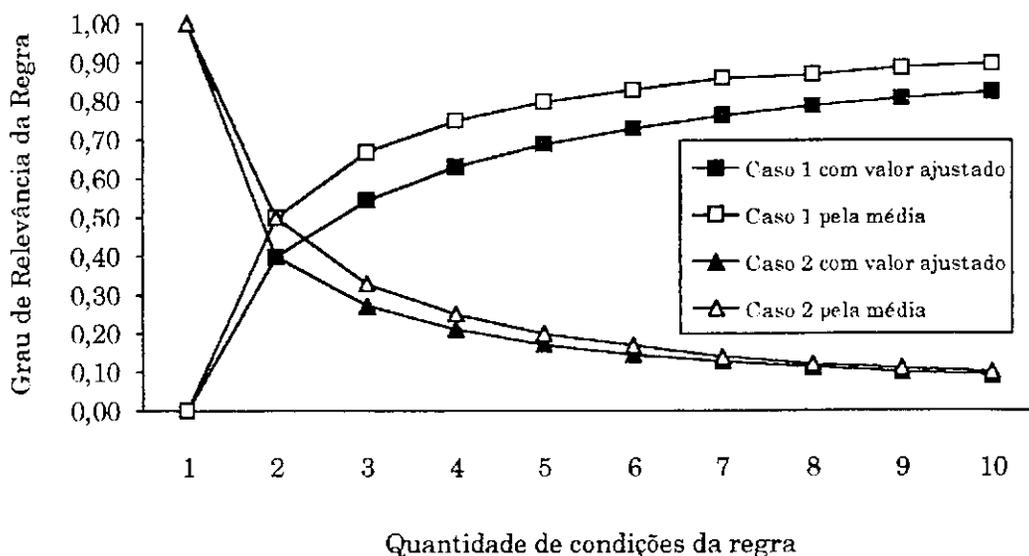


GRÁFICO 7.1 - Convergência da relevância de uma regra

7.5 Utilização das formas de avaliação de uma base de conhecimento

Anteriormente apresentamos, de uma forma empírica, algumas formas de avaliação dos aspectos objetivos e de custo de uma base conhecimento.

Utilizando o mesmo ambiente apresentado na seção anterior, podemos definir as formas de cálculo desses tipos de avaliação como:

- **quantidade de regras** - é o número de regras que compõem a base de conhecimento. É um bom indicativo sobre a base quando a busca é o menor tamanho, sua obtenção é simples e está representado por n_r .
- **tamanho médio das regras** - é um valor obtido através do cálculo da média aritmética do número de condições existentes em todas as regras da base de conhecimento. Pode ser calculado por:

$$\overline{TR}(BC) = \frac{\sum_{n_r} \sum_{n_c} P_r}{n_r} \quad (7.4)$$

- **custo médio de uma classificação** - representa o custo de uma base de conhecimento, e vem a ser a média ponderada dos custos de todas as classificações usadas na geração dessa base. Entende-se por custo de uma classificação, como sendo a soma dos custos dos atributos que compõem uma regra vezes o número de exemplos que ela mapeia [Nuñez 91]. Assim, o custo médio de uma classificação pode ser obtido pela seguinte equação.

$$\overline{CR}(BC) = \frac{\sum_{n_r} (\sum_{n_c} FC(C_m)) \cdot n_{exmr}}{\sum_{n_r} n_{exmr}} \quad (7.5)$$

sendo:

$FC(C_m)$ uma função que fornece o custo da condição C_m da premissa P_r ;

n_{exmr} o número de exemplos mapeados pela regra r que está sendo analisada;

7.6 Avaliação de uma base de conhecimento para o domínio "brinquedo seguro"

Para exemplificar as diversas formas de avaliação de uma base de conhecimento, vamos utilizar o domínio "brinquedo seguro", que foi definido,

no capítulo 6, pelo conjunto de treinamento da TAB. 6.1 e pelo conhecimento preliminar, na forma de custo e generalização, na TAB 6.2. Uma possível matriz de relevância para esse domínio é a que está descrita na TAB. 7.2.

TABELA 7.2 - Matriz de relevância nebulosa para o domínio "brinquedo seguro"

Atributo	Classe	
	Seguro	Perigoso
Forma	{0.50/quadrado + 0.50/triângulo + 0.75/elipse + 0.75/círculo}	{0.75/quadrado + 0.75/triângulo + 0.50/elipse + 0.50/círculo}
Cor	{0.00/vermelho + 0.00/azul + 0.00/amarela + 0.00/rosa}	{0.00/vermelho + 0.00/azul + 0.00/amarela + 0.00/rosa}
Tamanho	{0.75/grande + 0.75/médio + 0.25/pequeno}	{0.75/grande + 1.00/médio + 1.00/pequeno}
Material	{0.00/metal + 0.75/plástico + 1.00/couro + 1.00/madeira}	{1.00/metal + 0.25/plástico + 0.00/couro + 0.00/madeira}

Com base nas definições do "domínio brinquedo seguro", apresentamos na TAB. 7.3 os valores encontrados para as formas de avaliação das bases de conhecimento geradas pelos algoritmos estudados. Para os valores do aspecto objetivo "tamanho" utilizamos a equação (7.4) e para os do custo a equação (7.5). Apresentamos, também, como análise da qualidade dos aspectos subjetivos, o grau de relevância da base de conhecimento utilizando a equação (7.3) e o número de regras totalmente inúteis, ou seja, as regras que apresentaram um valor zero para a equação (7.2).

TABELA 7.3 - Resultados da avaliação das bases de conhecimento geradas, pelos algoritmos estudados, para o domínio "brinquedo seguro".

Algoritmos	Aspectos objetivos		Custo Médio de Classificação	Aspectos subjetivos	
	Quant. de Regras	Tam.Médio de Regra		Regras Inúteis ²	Grau de Relevância
ID3	6	1.50	303.33	0.17%	0.59
IDRT	5	1.60	143.33	-	0.81
PRISM	6	1.33	150.00	0.33%	0.41
FRPRISM	7	1.29	199.17	0.28%	0.52
EG2 $f_e=1.0$ $g=0.6$	4	2.00	95.00	-	0.48
ISREG $f_e=0.3$ $f_q=1$	4	1.5	200.00	-	0.89
ISREG $f_e=1.0$ $f_q=0$	5	2.00	93.33	-	0.40

² Foram consideradas as regras totalmente inúteis, *i.e.*, as regras que não apresentam nenhuma condição relevante.

Podemos observar que os algoritmos que utilizam a relevância semântica (IDRT, FRPRISM e ISREG com $f_q = 1$) tiveram, numa análise geral, um bom desempenho no que concerne aos aspectos subjetivos, pois não geram regras inúteis e apresentam um elevado grau de relevância, principalmente o ISREG. O conhecimento preliminar, na forma de custo, mostra toda sua utilidade, pois os algoritmos que o utilizam, EG2 e ISREG com $f_e = 1$, apresentam graus de relevância e tamanhos compatíveis com os demais, além de não gerarem regras inúteis e apresentarem os menores custos. Evidentemente, que esse exemplo serve apenas como ilustração das questões levantadas, havendo a necessidade de um teste mais significativo com uma modelagem de domínios reais e utilizando um conjunto de treinamento com um bom número de exemplos, é exatamente disso o que trata o próximo capítulo.

8 Análise dos resultados obtidos com os algoritmos semânticos

8.1 Introdução

O produto final de um algoritmo indutivo é uma base de conhecimento, que posteriormente será utilizada por um sistema baseado em conhecimento. Portanto, avaliar um algoritmo indutivo consiste em analisar os fatores mais importantes que estão relacionados com a base por ele gerada. Ou seja, o importante é a qualidade, sob os diversos aspectos, dos resultados apresentados.

Alguns dos algoritmos estudados nesta tese, em particular o ISREG, possuem parâmetros que permitem priorizar ou desprezar certas características de uma base de conhecimento. Dessa forma, seus resultados devem ser analisados em função dos aspectos que influenciam a geração da base de conhecimento. Para auxiliar esta análise vamos utilizar as seguintes grandezas, que medem os diversos aspectos da base gerada:

- **Quantidade de regras (Qtde)** - o número de regras geradas pelo algoritmo;
- **Tamanho médio das regras ($\overline{T_{am}}$)** - a média aritmética da quantidade de condições presentes em cada regra;
- **Regras inúteis (Inut.)** - o percentual de regras que possuem um grau de relevância inferior ao limiar de utilidade da regra. Este limiar foi definido como sendo igual a 0.25, numa analogia com o valor que

representa a pouca relevância de uma condição, utilizada na matriz de relevância. Foi utilizado o mesmo valor para todos os domínios;

- **Grau de relevância da base** ($GRBC$) - a média aritmética corrigida do grau de relevância semântica das regras que compõem a base de conhecimento, obtida através da aplicação da equação (7.3);
- **Custo médio de classificação** ($\overline{\text{Custo}}$) - a média ponderada do custo de todas as classificações feitas para gerar a base de conhecimento (equação 7.5);
- **Frequência de acerto** (FreqAcerto) - o percentual de acerto realizado na classificação dos exemplos que compõem o conjunto de teste. Essa grandeza só será utilizada nos domínios que tenham uma quantidade significativa de exemplos, suficiente para a divisão da tabela de exemplos em um conjunto de treinamento e um conjunto de teste.

Ao longo deste trabalho utilizamos dois pequenos domínios ("magro-gordo" e "brinquedo-seguro") apenas para exemplificar os problemas sintático e semântico, além de mostrar a estrutura das principais formas de conhecimento preliminar disponíveis e a utilização que os algoritmos podem fazer dessas estruturas. Embora tenham nos dado uma idéia dos novos algoritmos propostos, eles não possibilitam uma avaliação mais consistente das vantagens e desvantagens desses algoritmos. Em função disso, e para efeito de uma análise comparativa, apresentaremos, para quatro domínios reais, os resultados encontrados por todos os algoritmos estudados. Para ilustrar os tipos de relatórios usados na análise, mostraremos também os resultados para o domínio "brinquedo seguro".

Entre os novos algoritmos propostos (IDRT, FRPRISM e ISREG), o último é o mais abrangente e, portanto, a avaliação de seus resultados merecerá uma atenção toda especial. Assim, para cada algoritmo estudado nesta tese buscaremos uma combinação de parâmetros do ISREG que permite fazer uma comparação entre os resultados obtidos. Essa comparação é mostrada nas tabelas de resultados apresentadas nas seções seguintes. Os resultados do ISREG que sejam superiores ou iguais àqueles apresentados pelo algoritmo com o qual ele está sendo comparado, serão ressaltados por um destaque na célula correspondente. Nas comparações com o algoritmo PRISM e ID3, os destaques referem-se ao ID3 pois o ISREG, por definição, deve apresentar um resultado igual ao PRISM quando seus parâmetros

forem iguais a zero e não usar generalização. A linha que apresenta os resultados do FRPRISM e do ISREG foi colocada apenas para comprovar que esses algoritmos apresentam resultados iguais quando os parâmetros do ISREG desprezam as informações de custo, de generalização e solicitam a busca da qualidade máxima. Os melhores resultados obtidos pelo ISREG, levando em consideração um balanceamento entre os fatores analisados, serão mostrados nas últimas linhas de cada tabela.

O tipo de análise descrita anteriormente baseia-se numa inspeção visual de uma tabela de resultados, ressaltando as grandezas em que o ISREG é igual ou superior à aqueles com que está sendo comparado. Apesar de válida, essa análise permite apenas observar as grandezas de uma forma individualizada, não fornecendo uma idéia mais completa da diferença do comportamento global entre dois algoritmos. Assim, além da análise "visual", realizaremos uma outra análise, baseada em uma única métrica, que permite avaliar, de uma forma mais objetiva, essa diferença globalizante. À essa métrica chamaremos de *índice de qualidade global*

Portanto, na análise dos resultados dos algoritmos propostos, adotaremos o seguinte procedimento. Para cada domínio, apresentaremos uma tabela de resultados de todos os algoritmos estudados. Em seguida, faremos, conforme descrito anteriormente, uma análise subjetiva desses resultados. Subsequentemente, faremos uma análise mais objetiva baseada no índice de qualidade global entre dois algoritmos. Finalmente, considerando todos os domínios conjuntamente, apresentaremos uma análise comparativa entre os algoritmos que usam relevância semântica (IDRT, FRPRISM e ISREG), aqui chamados de semânticos, e os que não usam (ID3, PRISM e EG2), referenciados como não semânticos. Mais uma vez, atenção especial será dada ao algoritmo ISREG.

8.2 Definição de um índice de qualidade global entre dois algoritmos

O *índice de qualidade global* (I_{QG}) é uma medida comparativa e, para um mesmo domínio, representa a diferença de qualidade global das bases de conhecimento geradas por dois algoritmos indutivos. Será definido como sendo a média ponderada das diferenças relativas de todas as grandezas envolvidas na avaliação de uma base.

Sejam os seguintes conjuntos:

- $\mathcal{A} = \{\text{ID3, PRISM, EG2, IDRT, FRPRISM, ISREG}\}$ o conjunto dos algoritmos estudados;
- $\mathcal{D} = \{\text{Qtde, Tam., Inut, GRBC, Custo, FreqAcerto}\}$ o conjunto das grandezas que medem os aspectos qualitativos de uma base de conhecimento;
- $\mathcal{D}_d = \{\text{GRBC, FreqAcerto}\}$ o subconjunto de \mathcal{D} , formado pelas grandezas que são diretamente proporcionais;
- $\mathcal{D}_i = \{\text{Qtde, Tam., Inut, Custo}\}$ o subconjunto de \mathcal{D} , formado pelas grandezas que são inversamente proporcionais;

Assim, para uma dado domínio, o *índice de qualidade global* (I_{QG}) entre dois algoritmos \mathcal{A}_i e \mathcal{A}_j será definido como:

$$I_{QG}(\mathcal{A}_i, \mathcal{A}_j) = \left(\sum_{\kappa \in \mathcal{D}} s_{\kappa} \cdot f_{\kappa} \cdot \frac{p_{\kappa i} - p_{\kappa j}}{\max(p_{\kappa \mathcal{A}}) - \min(p_{\kappa \mathcal{A}})} \right) / \left(\sum_{\kappa \in \mathcal{D}} f_{\kappa} \right) \quad (8.1)$$

onde:

f_{κ} - fator de importância relativa da grandeza κ

$p_{\kappa i}$ e $p_{\kappa j}$ - valores das grandezas κ para os algoritmos i e j respectivamente;

s_{κ} - fator indicativo de proporcionalidade da grandeza κ ($s_{\kappa} = +1$ se $\kappa \in \mathcal{D}_d$) e

($s_{\kappa} = -1$ se $\kappa \in \mathcal{D}_i$);

$\max(p_{\kappa \mathcal{A}})$ e $\min(p_{\kappa \mathcal{A}})$ - representam respectivamente o maior e menor valor da grandeza κ , entre os valores fornecidos pelos algoritmos de \mathcal{A} para essa grandeza. Se $\max(p_{\kappa \mathcal{A}}) = \min(p_{\kappa \mathcal{A}})$, praticamente impossível de ocorrer, então a parcela do somatório da equação (8.1) correspondente à grandeza κ será nula.

Em termos globais, considerando todos as grandezas envolvidas, e dentro de uma certa relação de ordem de qualidade, que chamaremos de "melhor", o valor do *índice de qualidade global* indica a superioridade ou não de um dado algoritmo em relação a outro. Ou seja:

$$I_{QG}(\mathcal{A}_i, \mathcal{A}_j) > 0 \text{ significa } \mathcal{A}_i \text{ "melhor" que } \mathcal{A}_j$$

$I_{QG}(\mathcal{A}_i, \mathcal{A}_j) < 0$ significa \mathcal{A}_j "melhor" que \mathcal{A}_i

$I_{QG}(\mathcal{A}_i, \mathcal{A}_j) = 0$ significa \mathcal{A}_i "igual" a \mathcal{A}_j

Queremos deixar claro que o índice definido pela equação (8.1) não é uma medida absoluta de aferir a qualidade entre dois algoritmos, ele é, isto sim, apenas uma forma, que a achamos razoavelmente representativa, de avaliar a diferença de qualidade das bases de conhecimento geradas por dois algoritmos indutivos.

8.3 Procedimento de análise dos resultados

Na análise dos resultados dos algoritmos propostos, adotaremos o seguinte procedimento: para cada domínio, apresentaremos uma tabela de resultados de todos os algoritmos estudados; faremos então, conforme descrito anteriormente (seção 8.1), uma análise subjetiva desses resultados; e, em seguida, faremos uma análise mais objetiva baseada no *índice de qualidade global* entre dois algoritmos. Finalmente, considerando todos os domínios conjuntamente, apresentaremos uma análise comparativa entre os algoritmos que usam relevância semântica (IDRT, FRPRISM e ISREG), os semânticos, e os que não usam (ID3, PRISM e EG2), os não semânticos. Mais uma vez, atenção especial será dada ao algoritmo ISREG.

Análise objetiva

Neste tipo de análise, usando o *índice de qualidade global*, faremos uma comparação dos algoritmos dois a dois. Para cada domínio, apresentaremos uma tabela contendo os resultados da comparação de cada algoritmo semântico com os não semânticos, exceto para o ISREG, que o compararemos apenas com o EG2 e o IDRT. As razões dessa exceção serão explicadas mais adiante.

O algoritmo semântico IDRT apresenta um fator de ponderação (fq) entre os aspectos semântico, representado pelas grandezas $Inut$ e $GRBC$, e tamanho, representado pelas grandezas $Qtde$ e \overline{Tam} . Vários experimentos, com diversos domínios, têm nos mostrado que esse algoritmo, em geral, atinge o seu ponto de equilíbrio para $fq = 0.5$. Isto porque, para valores muito abaixo de 0.5 o IDRT tende a se comportar como o ID3, perdendo, assim, muito em qualidade semântica, e para valores muito acima de 0.5 a tendência é ele

apresentar um aumento no tamanho médio das regras ($\overline{\text{Tam.}}$), no custo médio e o $GRBC$ fica mais ou menos estável, com um ligeiro declínio devido ao aumento do $\overline{\text{Tam.}}$. Por exemplo, para o domínio Zoo, vide seção 8.5.2, numa avaliação global encontramos $I_{QG}(\text{IDRT}(fq = 0.5), \text{IDRT}(fq = 1.0)) = 67.7\%$, ou seja, o IDRT com ($fq = 0.5$) apresenta um resultado geral 67.7% "melhor" que o IDRT com ($fq = 1.0$). Portanto, neste estudo comparativo consideraremos o IDRT com ($fq = 0.5$).

Conforme já havíamos mencionado, o algoritmo ISREG será comparado apenas com o EG2 e IDRT, isto porque:

a) os algoritmos PRISM e FRPRISM são casos particulares do ISREG. O primeiro é o ISREG para os valores de seus parâmetros ($fe = 0$, $fq = 0$ e $g = 0$) e o segundo para os valores ($fe = 0$, $fq = 1$ e $g = 0$). Portanto, uma comparação do ISREG com esses algoritmos traria pouca informação útil;

b) o algoritmo ID3 pode ser visto como sendo o IDRT com ($fq = 0$) e também como sendo o EG2 com ($fe = 0$ e $g = 0$), ou seja, ele é um caso particular tanto do IDRT como do EG2. Assim, comparando o ISREG com esses dois últimos algoritmos, indiretamente também o estamos comparando com o ID3.

Na análise objetiva, compararemos o ISREG duas vezes com cada um dos algoritmos EG2 e IDRT. Na primeira vez, os valores dos parâmetros do ISREG serão iguais aos correspondentes tanto do EG2 como aos do IDRT, i.e., o ISREG terá os valores ($fe = 1.0$ e $fq = 0.0$) na comparação com o EG2; e os valores ($fe = 0.0$ e $fq = 0.5$) na comparação com o IDRT. Na segunda comparação, para cada um desses dois algoritmos, procuraremos uma combinação de valores dos parâmetros do ISREG para o qual ele se mostre "melhor".

8.4 O domínio "brinquedo seguro"

Conforme mencionamos anteriormente, este domínio foi incluído na análise dos resultados por ser pequeno, de "brinquedo", permitindo, assim, uma visualização completa das bases de conhecimento geradas. Além do mais, é o único para o qual dispomos das três formas de conhecimento preliminar (relevância, custo e generalização).

A escolha do melhor resultado apresentado pelo algoritmo ISREG para esse domínio, e também para os demais domínios, foi feita com base no relatório que fornece os resultados das execuções realizadas em função da variação dos valores dos parâmetros do algoritmo. Foi solicitada uma variação de 0.2, o que resulta em 36 execuções. A FIG. 8.1 mostra trechos desse relatório.

Domínio	- BrinqSeguro							
Matriz de Relevância	- BrinqSeguro.REL							
Tabela de Generalização	- BrinqSeguro.HRQ						Limiar de Generalização (g) - 0.60	
N.Exemp.do Conj.Treinamento	- 12							
fe => Fator de Economia = [0,1] onde: 0=perdulário ... 1=economia máxima								
fp => Fator de Qualidade = [0,1] onde: 0=tamanho mínimo de regra ... 1=qualidade máxima								
			----- Regras -----			Nível	Freq.	Custo
		Qtde	Tam.Med	Conflito	Inúteis	Qualidade	Acerto	Médio
fe=0.00	fp=0.00	6	1.50	0	1	0.52	---	305.000
	fp=0.20	6	1.50	0	1	0.52	---	305.000
	fp=0.40	6	1.50	0	1	0.52	---	305.000
	fp=0.60	6	1.50	0	1	0.52	---	305.000
	fp=0.80	5	1.60	0	0	0.74	---	266.667
	fp=1.00	5	1.60	0	0	0.74	---	266.667
fe=0.20	fp=0.00	5	2.00	0	0	0.43	---	129.167
	fp=0.20	4	1.50	0	0	0.72	---	163.333
			.	.	.			
			.	.	.			
			.	.	.			
fe=0.80	fp=0.80	5	2.20	0	0	0.40	---	120.000
	fp=1.00	5	2.20	0	0	0.49	---	156.667
fe=1.00	fp=0.00	5	2.20	0	0	0.40	---	93.333
	fp=0.20	5	2.20	0	0	0.40	---	93.333
	fp=0.40	5	2.20	0	0	0.40	---	93.333
	fp=0.60	5	2.20	0	0	0.40	---	93.333
	fp=0.80	5	2.20	0	0	0.40	---	93.333
	fp=1.00	5	2.20	0	0	0.40	---	93.333

FIGURA 8.1 - Trechos do relatório de análise da variação dos parâmetros do algoritmo ISREG.

As regras que compõem a base de conhecimento gerada por cada algoritmo foram retiradas de um relatório padrão, cujo formato pode ser observado na FIG. 8.2.

Nome: BrinqSeguro.OUT

1 Se	Forma = Conica & Tamanho = NaoPequeno	Então S	(4)
2 Se	Material = couro	Então S	(3)
3 Se	Forma = Poligona & Tamanho = NaoGrande	Então N	(4)
4 Se	Material = metal	Então N	(3)

(n) - representa o número de exemplos mapeados pela regra.

Resumo (ISREG):

Domínio	- BrinqSeguro
Tabela Hierarquia	- BrinqSeguro.HIRQ
Matriz de Relevância	- BrinqSeguro.RRL
N.Ex.do Conj.Treinamento	- 12
Fator de Economia (fe)	- 0.30
Fator de Qualidade (fq)	- 1.00
Limiar da Generalização (g)	- 0.60
No. Total de Regras	- 4
Tamanho Médio de Regra	- 1.50
Nível de Qualidade da Base	- 0.89
Custo Médio de Classificação	- 200.00

FIGURA 8.2 - Relatório padrão de saída dos algoritmos.

TABELA 8.1 - Resultados dos algoritmos para o domínio "brinquedo seguro"

Algoritmos	Parâmetros	Regras			GRBC	Custo
		Qtde	Tam.	Inut.		
ID3	-	6	1.50	17%	0.59	303.33
PRISM	-	6	1.33	33%	0.41	150.00
ISREG	fe=0.0 fq=0.0 g=0.0	6	1.33	33%	0.41	150.00
IDRT	p=0.5	5	1.60	0%	0.81	143.33
ISREG	fe=0.0 fq=0.5 g=0.0	7	1.29	28%	0.47	190.00
FRPRISM	-	7	1.29	28%	0.52	199.17
ISREG	fe=0.0 fq=1.0 g=0.0	7	1.29	28%	0.52	199.17
EG2	fe=1.0 g=0.6	4	2.00	0%	0.48	95.00
ISREG	fe=1.0 fq=0.0 g=0.6	5	2.00	0%	0.40	93.33
ISREG	fe=0.3 fq=1.0 g=0.6	4	1.50	0%	0.89	200.00

Em TAB. 8.1 foi mostrado um resumo de todos os resultados apresentados pelos algoritmos estudados e os resultados mais significativos do algoritmo ISREG.

Ao executar o ISREG desprezando as informações de custo, qualidade semântica e generalização, *i.é.*, utilizar o algoritmo na sua forma mais simples, obtemos uma base de conhecimento ainda superior àquela gerada pelo ID3 e igual à base gerada pelo PRISM. O que nos dá um custo médio de classificação ($\overline{\text{Custo}}=150,00$) inferior, e um tamanho médio de regras também inferior ($\overline{\text{Tam.}}=1,33$) àqueles apresentados pelo ID3. As regras obtidas para esse caso foram as seguintes:

R1	Se Tamanho = grande	Então S	(1,2,3,5)
R2	Se Material = plástico	& Forma = elipse	Então S (4)
R3	Se Forma = círculo	& Cor = rosa	Então S (6)
R4	Se Tamanho = pequeno	Então P	(7,8,11,12)
R5	Se Cor = amarelo	Então P	(10)
R6	Se Material = madeira	Então P	(8,9)

O EG2 havia gerado a base de menor custo médio de classificação ($\overline{\text{Custo}}=95,00$), no entanto, o ISREG conseguiu baixar ainda mais o custo ($\overline{\text{Custo}}=93,33$). Porém, houve uma perda significativa de qualidade ($\mathcal{GRBC}=0,40$) se comparada com a base apresentada anteriormente ($\mathcal{GRBC}=0,89$). As regras obtidas para esse caso foram as seguintes:

R1	Se Forma = CÔNICA	& Cor = PRIMÁRIA	Então S (3,4,5)
R2	Se Forma = POLÍGONO	& Cor = rosa	Então S (1,2)
R3	Se Forma = círculo	& Cor = rosa	Então S (6)
R4	Se Forma = POLÍGONO	& Cor = PRIMÁRIA	Então P (7,8,9,10)
R5	Se Forma = CÔNICA	& Tamanho = pequeno	Então P (11,12)

O ISREG conseguiu gerar uma base de conhecimento que apresentou o melhor nível de qualidade sob o aspecto da relevância semântica ($\mathcal{GRBC}=0,89$), mantendo o tamanho de 4 regras que tinha sido o menor tamanho obtido entre os outros algoritmos (EG2) e diminuindo o tamanho médio das regras. Contudo, nesse caso, o custo ficou muito alto, embora menor que o custo apresentado pelo ID3. As regras geradas para formação dessa base foram:

R1	Se Forma = CÔNICA	& Tamanho = NÃO-PEQUENO	Então S (3,4,5,6)
R2	Se Material = couro	Então S	(1,2,3)
R3	Se Forma = POLÍGONO	& Tamanho = NÃO-GRANDE	Então P (7,8,9,10)
R4	Se Material = metal	Então P	(7,11,12)

Podemos observar que o ISREG mantém a característica apresentada pelos algoritmos PRISM e FRPRISM de mapear o mesmo exemplo em mais de uma regra (exemplos 3 e 7).

O ISREG gerou regras inúteis somente em alguns casos onde as informações de qualidade semântica foram desprezadas (parâmetro $f_q=0.0$). Nos demais casos as regras inúteis foram eliminadas, cumprindo assim seu objetivo de minimizar, e em alguns casos eliminar, a ocorrência do problema semântico. Esses resultados sinalizam que o algoritmo pode apresentar resultados de melhor qualidade que os demais algoritmos estudados.

Análise objetiva

A TAB. 8.2 mostra os resultados da comparação objetiva entre os algoritmos semânticos e não semânticos, e também do ISREG com o IDRT e, excepcionalmente para este domínio, com o FRPRISM.

TABELA 8.2 - Comparação objetiva entre os algoritmos semânticos e não semânticos para o domínio "Brinquedo seguro".

Domínio: "Brinquedo seguro"		$(s_k \cdot f_k \cdot \frac{P_{ki} - P_{kj}}{\max(P_{kA}) - \min(P_{kA})}) \cdot 100$						IOG (%)
Algoritmos		Regras			GRBC $s_k = +1$	Custo $s_k = -1$	Freq.	
Ai	Aj	Qtde $s_k = -1$	Tam. $s_k = -1$	Inut $s_k = -1$			Acerto $s_k = +1$	
IDRT	ID3	33	14	52	45	76	-	44.0
IDRT	PRISM	33	-38	100	81	3	-	35.8
IDRT	EG2 (fe=1.0)	-33	56	0	67	-23	-	13.4
FRPRISM	ID3	-33	29	-33	-14	49	-	-0.4
FRPRISM	PRISM	-33	5	-15	22	-23	-	-8.8
FRPRISM	EG2 (fe=1.0)	-33	100	-85	8	-44	-	-10.8
ISREG (fe=1.0, fq=0.0)	EG2 (fe=1.0)	-33	0	0	-16	1	-	-9.6
ISREG (fe=0.3, fq=1.0)	EG2 (fe=1.0)	0	70	0	84	-5	-	20.8
ISREG (fe=0.0, fq=0.5)	IDRT (fq=0.5)	-66	44	85	69	20	-	30.4
ISREG (fe=0.3, fq=1.0)	FRPRISM	100	-29	85	75	0	-	38.5

Neste caso, em se tratando de um domínio de "brinquedo", foi possível fazer uma análise mais minuciosa dos resultados apresentados pelo ISREG.

Portanto, aqui nos limitaremos a uma avaliação bem geral dos resultados da TAB. 8.2. Nela podemos observar que:

- a) o algoritmo IDRT apresenta um *índice de qualidade global* bastante positivo, mostrando toda sua "força semântica" através da superioridade que obteve para as grandezas *Inut* e *GRBC*;
- b) o FRPRISM, excepcionalmente neste caso, apresenta um *índice de qualidade global* negativo, inclusive até em comparação com o ID3, se bem que por uma minúscula margem de diferença (-0.4%).
- c) o ISREG, a exemplo do IDRT, apresenta um *índice de qualidade global* bastante positivo, inclusive até mesmo em relação ao próprio IDRT. A exceção ficou na primeira comparação com o EG2, onde pelo fato de desprezar o aspecto qualidade semântica ($f_q = 0$), o ISREG obteve, em relação ao EG2, um valor negativo (-16) para a grandeza *GRBC*.

Por se tratar de um domínio de "brinquedo", essas observações, embora valiosas, não são necessariamente verdadeiras para domínios reais.

8.5 Domínios reais

Com o objetivo de verificar o comportamento dos algoritmos semânticos diante de domínios reais e com um número significativo de exemplos, repetimos o estudo apresentado no item anterior para outros domínios. Com exceção do domínio que cobre casos de amenorréia, todos os demais foram importados via "ftp" (*file transfer program*) de um diretório público (*pub/machine-learning-databases*) que está disponibilizado no endereço "ics.uci.edu" que pertence ao *Department of Information and Computer Science* da Universidade da Califórnia. O conhecimento preliminar dos domínios o obtivemos junto a especialistas específicos. Esses especialistas, também, nos auxiliaram na adaptação dos dados importados. A documentação dos domínios utilizados nos testes encontra-se no apêndice B.

A TAB. 8.3 apresenta um resumo das principais características de cada domínio utilizado nos testes cujos resultados serão mostrados a seguir. As informações contidas nessa tabela têm os seguintes significados:

- nome - nome do domínio.
- #casos - número de casos que compõem a tabela de exemplos.
- #ct - número de exemplos do conjunto de treinamento.
- #ctes - número de exemplos do conjunto de teste.
- #ecl - número de elementos de classificação
- #atrib - número de atributos.
- #nomi - número de atributos que possuem valores nominais.
- #cont - número de atributos que possuem valores contínuos.

TABELA 8.3 - Principais características dos domínios utilizados nos testes.

nome	#casos	#ct	#ctes	#ecl	#atrib	#nomi	#cont
Amenorréia	91	70	21	6	6	6	0
Zoo	101	77	24	7	16	15	1
Heart-disease-Cleveland	303	228	75	5	13	8	5
Pima-indians-diabetes	768	576	192	2	8	0	8

8.5.1 Amenorréia

Este é um domínio real no campo da medicina, cobre casos de amenorréia (atraso menstrual), extraído de [Nuñez 88]. Tem como característica peculiar o fato de que um dos seus atributos (perfil hormonal), apesar de ter um custo monetário muito elevado, é suficiente para classificar os elementos de classificação presentes no conjunto de treinamento. A matriz de relevância nebulosa utilizada foi eliciada de um ginecologista (Dr. Renato Andrade Brasileiro), que confirmou a total relevância do atributo perfil hormonal. Não existem generalizações. Os resultados encontrados estão mostrados em TAB. 8.4.

Apesar da boa qualidade das bases geradas pelos algoritmos ID3 e IDRT, essas bases possuem um custo médio de classificação muito alto. Isso ocorreu devido o fato do atributo perfil hormonal ter sido o escolhido para ser colocado na raiz da árvore de decisão. Os algoritmos PRISM e FRPRISM conseguiram diminuir os custos mantendo o tamanho e a qualidade da base. Tanto o EG2 quanto o ISREG ao tentarem gerar uma base de custo mínimo diminuíram a frequência de acerto e a base gerada resultou muito grande. Porém somente o ISREG conseguiu gerar uma base que busca um equilíbrio entre os fatores de qualidade, tamanho e custo, mantendo a frequência de acerto em 100%, conforme mostra a última linha da TAB. 8.4.

TABELA 8.4 - Resultados dos algoritmos para o domínio "amenorréia".

Algoritmos	Parâmetros	Regras			GRBC	Custo	Freq. Acerto
		Qtde	Tam.	Inut.			
ID3	-	5	1.00	0%	0.82	9842.56	100%
PRISM	-	5	1.00	0%	0.82	7614.04	100%
ISREG	fe=0.0 fq=0.0	5	1.00	0%	0.82	7614.04	100%
IDRT	p=0.5	5	1.00	0%	0.82	9842.56	100%
ISREG	fe=0.0 fq=0.5	5	1.00	0%	0.82	7614.04	100%
FRPRISM	-	5	1.00	0%	0.82	7614.04	100%
ISREG	fe=0.0 fq=1.0	5	1.00	0%	0.82	7614.04	100%
EG2	fe=1.0	41	4.90	0%	0.53	5087.72	62%
ISREG	fe=1.0 fq=0.0	41	5.05	0%	0.53	5403.51	62%
ISREG	fe=0.2 fq=0.4	25	3.64	0%	0.60	4666.67	100%

Análise objetiva

Os resultados da comparação entre os algoritmos semânticos e não semânticos são os da TAB. 8.5.

TABELA 8.5 - Comparação objetiva entre os algoritmos semânticos e não semânticos para o domínio Amenorréia.

Domínio: Amenorréia		$\left(s_k \cdot f_k \cdot \frac{P_{ki} - P_{kj}}{\max(P_{kA}) - \min(P_{kA})} \right) \cdot 100$						Log (%)
Algoritmos		Regras			GRBC $s_k = +1$	Custo	Freq.	
Ai	Aj	Qtde	Tam.	Inut.			Acerto $s_k = +1$	
IDRT	ID3	0	0	0	0	0	0	0.0
IDRT	PRISM	0	0	0	0	-43	0	-7.2
IDRT	EG2 (fe=1.0)	100	96	0	100	-93	94	49.5
FRPRISM	ID3	0	0	0	0	43	0	7.2
FRPRISM	PRISM	0	0	0	0	0	0	0.0
FRPRISM	EG2 (fe=1.0)	100	96	0	100	-49	100	58.0
ISREG (fe=1.0, fq=0.0)	EG2 (fe=1.0)	0	-4	0	0	-61	0	-10.8
ISREG (fe=0.2, fq=0.4)	EG2 (fe=1.0)	44	31	0	24	8	100	34.5
ISREG (fe=0.0, fq=0.5)	IDRT (fq=0.5)	0	0	0	0	43	0	7.2
ISREG (fe=0.0, fq=1.0)	IDRT (fq=0.5)	0	0	0	0	24	0	4.0

Como o domínio amenorréia possui um atributo (perfil hormonal) com um grande poder de classificação, não obstante de ter um custo muito elevado, os algoritmos apresentam muitos valores iguais para as diversas grandezas, exceto evidentemente para o custo. Mesmo assim, de um modo geral, quando os comparamos com os algoritmos não semânticos, os semânticos apresentam um *índice de qualidade global* bastante positivo, isto porque, com exceção da grandeza custo, as diferenças de qualidade para as demais grandezas são todas positivas ou nulas. Conforme pode ser visto em TAB. 8.5, os dois únicos índices negativos dos algoritmos semânticos são $I_{QG}(\text{IDRT}, \text{PRISM})$ e $I_{QG}(\text{ISREG}(f_e=1.0, f_q=0.0), \text{EG2})$, valendo -43 e -61 respectivamente, ambos causados exatamente pela grandeza custo. Além do mais, o segundo índice envolve o algoritmo EG2, que é o especialista em custos. Entretanto, após uma variação de seus parâmetros, buscando-se mais qualidade semântica e menos custo, o ISREG superou o EG2 em todas as grandezas, inclusive no aspecto custo, tendo-se, conforme mostra a TAB.8.5, $I_{QG}(\text{ISREG}(f_e=0.2, f_q=0.4), \text{EG2}) = 34.5\%$.

8.5.2 Zoo

Esta base de dados foi criada por Richard Forsyth, apresentada no *Forsyth's PC/Beagle User's Guide*, e tem por objetivo classificar vários tipos de animais agrupados em 7 conjuntos. Cinco dos sete grupos são facilmente identificados, por exemplo o grupo dos mamíferos, das aves, dos peixes, etc. No entanto, dois grupos não permitem uma classificação precisa, uma vez que, por exemplo, minhoca e lagosta estão no mesmo grupo. A matriz de relevância utilizada foi montada com base em informações encontradas em enciclopédias e revistas especializadas. A TAB. 8.6 contém os resultados apresentados pelos algoritmos.

O domínio Zoo possui atributos bem definidos que permitem classificar cada grupo de animais o que levou os algoritmos a apresentarem resultados semelhantes. Podemos observar que o IDRT conseguiu gerar a base com melhor nível de qualidade semântica (0.81%). No entanto essa base possui o maior custo (7.03) dentre todas aquelas apresentadas na tabela. O EG2 e o ISREG atingiram seus objetivos ao buscarem o menor custo possível gerando bases com custo zero, porém nessa comparação, o ISREG foi superior nas demais grandezas, exceto na acurácia. As duas últimas linhas da tabela mostram resultados quando houve uma tentativa de equilibrar todos os índices envolvidos; elas apresentam um nível de qualidade semântica bem

próximo do máximo obtido, baixos custos, além de oferecer uma frequência de acerto dentro dos níveis apresentados pelos demais algoritmos.

TABELA 8.6 - Resultados dos algoritmos para o domínio "Zoo".

Algoritmos	Parâmetros	Regras			$GRBC$	$\overline{\text{Custo}}$	Freq. Acerto
		Qtde	Tam.	Inut.			
ID3	-	13	2.31	0%	0.61	2.12	96%
PRISM	-	14	2.07	7%	0.57	2.03	96%
ISREG	$fe=0.0$ $fq=0.0$	14	2.07	7%	0.57	2.03	96%
IDRT	$p=0.5$	15	4.53	0%	0.81	7.03	96%
ISREG	$fe=0.0$ $fq=0.5$	15	2.07	6%	0.59	2.12	96%
FRPRISM	-	16	2.38	0%	0.69	3.39	100%
ISREG	$fe=0.0$ $fq=1.0$	16	2.38	0%	0.69	3.39	100%
EG2	$fe=1.0$	15	2.67	7%	0.58	0.00	96%
ISREG	$fe=1.0$ $fq=0.0$	13	2.23	0%	0.62	0.00	92%
ISREG	$fe=0.3$ $fq=1.0$	17	3.18	0%	0.71	1.52	100%
ISREG	$fe=1.0$ $fq=0.7$	15	3.00	0%	0.70	0.00	96%

Análise objetiva

Para o domínio Zoo, os resultados da comparação entre os algoritmos semânticos e não semânticos são os mostrados na TAB. 8.7.

Da análise dessa tabela podemos concluir que:

a) O IDRT não obteve um bom desempenho para o domínio Zoo. Ao ponderar tamanho e qualidade semântica, o IDRT conseguiu ser muito eficiente no segundo aspecto, isso mostrado através das grandezas Inut e $GRBC$, onde obteve excelentes índices em relação aos algoritmos não semânticos. Entretanto, essa eficiência foi obtida em detrimento das grandezas Qtde e $\overline{\text{Tam.}}$, onde as diferenças para essas grandezas são todas negativas.

TABELA 8.7 - Comparação objetiva entre os algoritmos semânticos e não semânticos para o domínio "Zoo".

Domínio: Zoo		$(s_k \cdot f_k \cdot \frac{P_{ki} - P_{kj}}{\max(P_{kA}) - \min(P_{kA})}) \cdot 100$						I _{OG} (%)
Algoritmos		Regras			G _{RBC} $s_k = +1$	Custo $s_k = -1$	Freq.	
Ai	Aj	Qtde $s_k = -1$	T _{am.} $s_k = -1$	Inut $s_k = -1$			Acerto $s_k = +1$	
IDRT	ID3	-50	-89	0	83	-71	0	-21.2
IDRT	PRISM	-25	-100	100	100	-71	0	1.0
IDRT	EG2 (f _e =1.0)	0	-75	100	96	-100	0	3.5
FRPRISM	ID3	-75	-3	0	33	-18	100	6.2
FRPRISM	PRISM	-50	-12	100	79	-19	100	33.0
FRPRISM	EG2 (f _e =1.0)	-25	12	100	46	-48	100	30.8
ISREG (f _e =1.0, f _q =0.0)	EG2 (f _e =1.0)	0	13	100	83	0	0	32.6
ISREG (f _e =0.0, f _q =1.0)	EG2 (f _e =1.0)	-25	12	100	46	-48	100	31.0
ISREG (f _e =0.0, f _q =0.5)	IDRT (f _q =0.5)	0	100	-86	-91	70	0	-1.2
ISREG (f _e =1.0, f _q =0.7)	IDRT (f _q =0.7)	-50	15	0	-12	75	0	4.7

b) O FRPRISM, embora tenha apresentado diferenças totalmente negativas para as grandezas custo e Qtde, no geral possui um *índice de qualidade global* bastante positivo. Isto porque, além de apresentar uma alta qualidade semântica, que é a sua especialidade, ele ainda conseguiu as diferenças qualitativas máximas para a grandeza Freq.Acerto.

c) Na comparação com o EG2, o ISREG apresenta os dois *índices de qualidade global* bastante positivos, no valor de 32.6% e 31.0% respectivamente. As duas únicas diferenças negativas verificaram-se para as grandezas Qtde e custo, isto porque o ISREG priorizou o aspecto qualidade semântica (f_q=1.0), e ignorou o aspecto custo (f_q=0.0), tendo a priorização do primeiro aspecto resultado num aumento da grandeza Qtde. Na comparação do ISREG com o IDRT, no geral, verificamos um certo equilíbrio, com o IDRT obtendo melhores resultados para as diferenças qualitativas das grandezas Inut e G_{RBC} e o ISREG para as grandezas T_{am.} e custo. Um aspecto importante a ser considerado é que nas quatro comparações, o ISREG obteve diferenças positivas para a grandeza T_{am.}. Isto porque, enquanto o ISREG gera diretamente regras modulares, os algoritmos IDRT e EG2 geram árvores de decisão, que, em geral, apresentam um tamanho médio superior ao das regras.

8.5.3 Heart-disease-Cleveland

O Dr. Robert Detrano selecionou os casos apresentados neste domínio entre os pacientes da *Cleveland Clinic Foundation* (Cleveland/EUA). A base de dados original possui 76 atributos, porém, somente os 13 utilizados neste trabalho são efetivamente manipulados nos trabalhos de aprendizagem automático. Os diagnósticos apontam, ou não, a presença de doenças do coração, com base em um exame denominado *angiographic disease status*. A matriz de relevância utilizada foi eliciada de um cardiologista (Dr. Demóstenes Gonçalves Ribeiro - Prof. da Universidade Federal do Ceará), que também forneceu as informações de custo de cada atributo. Em TAB. 8.8 relacionamos os resultados apresentados pelos algoritmos para esse domínio.

TABELA 8.8 - Resultados dos algoritmos para o domínio "heart-disease-Cleveland".

Algoritmos	Parâmetros	Regras			GRBC	Custo	Freq. Acerto
		Qtde	Tam.	Inut.			
ID3	-	139	4.64	10%	0.38	41.52	46%
PRISM	-	120	3.23	12%	0.38	19.97	55%
ISREG	$fe=0.0$ $fq=0.0$	120	3.23	12%	0.38	19.97	55%
IDRT	$p=0.5$	142	5.22	0%	0.50	50.41	59%
ISREG	$fe=0.0$ $fq=0.5$	181	3.56	9%	0.46	29.25	59%
FRPRISM	-	196	3.82	3%	0.56	33.52	62%
ISREG	$fe=0.0$ $fq=1.0$	196	3.82	3%	0.56	33.52	62%
EG2	$fe=1.0$	143	5.66	15%	0.31	8.27	51%
ISREG	$fe=1.0$ $fq=0.0$	151	5.98	22%	0.30	8.16	51%
ISREG	$fe=1.0$ $fq=0.2$	156	5.95	18%	0.32	7.95	53%

Este conjunto de treinamento possui alguns atributos numéricos o que força a realização de uma discretização provocando a ocorrência de 2 exemplos ambíguos levando à geração de 2 regras em conflito. A matriz de relevância fornecida pelo especialista possui muitos valores irrelevantes o que justifica o baixo valor do grau de relevância das bases de conhecimento geradas, provocando, também, a geração de regras inúteis. Contudo, o ISREG apresentou alguns resultados significativos, como por exemplo uma base com um custo médio de classificação menor do que a base gerada pelo EG2 (7.95 x 8.27) que é o algoritmo especialista em custo. Conseguiu também gerar a

base com o melhor nível de qualidade semântica e a melhor acurácia (resultado igual ao do FRPRISM, que é um caso particular do ISREG).

Análise objetiva

A TAB. 8.9 mostra os resultados da comparação entre os algoritmos semânticos e não semânticos e do ISREG com o IDRT.

TABELA 8.9 - Comparação objetiva entre os algoritmos semânticos e não semânticos para o domínio "Heart-disease-Cleveland".

Domínio: Heart-disease		$(s_k \cdot f_k \cdot \frac{P_{ki} - P_{kj}}{\max(P_{kA}) - \min(P_{kA})}) \cdot 100$						IDG (%)
Algoritmos		Regras			\overline{GRBC} $s_k = +1$	\overline{Custo} $s_k = -1$	Freq.	
Ai	Aj	Qtde $s_k = -1$	Tam. $s_k = -1$	Inut $s_k = -1$			Acerto $s_k = +1$	
IDRT	ID3	-4	-21	45	46	-20	81	21.0
IDRT	PRISM	-28	-72	45	46	-70	25	-9.0
IDRT	EG2 (fe=1.0)	10	16	68	73	-99	50	18.2
FRPRISM	ID3	-75	30	32	69	19	100	29.2
FRPRISM	PRISM	-100	-21	41	69	-32	44	0.2
FRPRISM	EG2 (fe=1.0)	-69	67	55	96	-59	69	26.5
ISREG (fe=1.0, fq=0.0)	EG2 (fe=1.0)	-10	-12	-32	-4	1	0	-9.5
ISREG (fe=0.0, fq=1.0)	EG2 (fe=1.0)	-69	67	55	96	-59	69	26.5
ISREG (fe=0.0, fq=0.5)	IDRT(fq=0.5)	-51	60	-41	-15	50	0	0.5
ISREG (fe=0.0, fq=1.0)	IDRT(fq=0.5)	-71	51	-13	23	40	19	8.1

Analisando os resultados da TAB. 8.9 podemos concluir que:

a) Em comparação com os algoritmos não semânticos, em média, o IDRT apresenta um *índice de qualidade global* positivo. Cumpre muito bem o seu papel de algoritmo semântico, pois apresenta todas as diferenças positivas para as grandezas *Inut* e \overline{GRBC} , que são as que quantificam o aspecto semântico. No aspecto tamanho, o IDRT, quando comparado com os algoritmos ID3 e PRISM, apresenta diferenças negativas para as grandezas *Qtde* e $\overline{Tam.}$, sendo que para a segunda grandeza, em relação ao PRISM, essa diferença é bem acentuada (-72). Dois motivos são os responsáveis pela ocorrência dessa diferença acentuada. O primeiro, porque o IDRT gera árvores de decisão e, conseqüentemente, em média, apresenta um $\overline{Tam.}$

superior aos demais algoritmos que geram diretamente regras, como é o caso do PRISM. O segundo motivo é que, enquanto o PRISM procura a menor regra possível, o IDRT, por ser semântico, busca um equilíbrio entre o menor tamanho e a melhor qualidade semântica.

b) O FRPRISM apresenta *índices de qualidade global* bastante positivos em relação aos algoritmos não semânticos, à exceção da comparação com o PRISM onde se verifica um certo equilíbrio. Conforme mostra a TAB. 8.9, as diferenças nas grandezas correspondentes aos aspectos semântico e acurácia são bem significativas, principalmente em relação ao aspecto acurácia.

c) Na comparação com o EG2, o ISREG apresenta dois *índices de qualidade global* distintos: um negativo (-9.5%), quando o algoritmo prioriza o aspecto custo ($f_e=1.0$) e despreza a qualidade semântica ($f_q=0.0$); e, o outro positivo (26.5%), quando o ISREG faz exatamente ao contrário, ignora o custo ($f_e=0.0$) e prioriza a qualidade semântica ($f_q=1.0$). A negatividade do primeiro índice deve-se exatamente ao fato do ISREG desprezar o aspecto semântico, fazendo com que, na comparação, isso cause diferenças negativas nas grandezas relacionadas com esse aspecto, que são $Inut$ e $GRBC$. Na comparação com o IDRT, o ISREG apresentou valores positivos para os dois *índices de qualidade global*, que são (0.5%) e (8.1%) respectivamente. A insignificância do valor do primeiro índice deve-se ao fato do ISREG ter dado uma importância média ao aspecto qualidade semântica ($f_q=0.5$) o que, em geral, isso é muito pouco na comparação com o IDRT, que é especialista no aspecto semântico. Como consequência, nesse caso, as duas diferenças das grandezas relacionadas com o aspecto semântico, $Inut$ e $GRBC$, são negativas. A exemplo do domínio Zoo, um aspecto importante é que nas quatro comparações, o ISREG obteve três diferenças bastante positivas para a grandeza $\overline{Tam.}$.

8.5.4 Pima-indians-diabetes

O *National Institute of Diabetes and Digestive and Kidney Diseases* (Maryland/EUA) realizou um estudo com as mulheres índias, maiores de 21 anos, da tribo Pima que residem próximo de Phoenix no Arizona, desse estudo o Dr. Vincent Sigillito selecionou os casos que compõem a tabela de exemplos deste domínio. Os diagnósticos acusam ou não a existência de sinais de diabetes nas pacientes observadas. Uma endocrinologista (Dra. Maria Sidneuma Melo) forneceu as informações de custo e os valores da

matriz de relevância utilizada para este domínio. A TAB. 8.10 sumariza os resultados apresentados pelos algoritmos para o domínio Pima-indians.

TABELA 8.10 - Resultados dos algoritmos para o domínio "pima-indians-diabetes".

Algoritmos	Parâmetros	Regras			GRBC	Custo	Freq. Acerto
		Qtde	Tam .	Inut.			
ID3	-	213	5.69	19%	0.29	25.17	57%
PRISM	-	250	5.14	25%	0.31	45.08	73%
ISREG	$fe=0.0$ $fq=0.0$	250	5.14	25%	0.31	45.08	73%
IDRT	$p=0.5$	227	5.73	10%	0.36	33.71	55%
ISREG	$fe=0.0$ $fq=0.5$	340	5.12	14%	0.36	50.41	75%
FRPRISM	-	369	5.25	9%	0.39	52.93	76%
ISREG	$fe=0.0$ $fq=1.0$	369	5.25	9%	0.39	52.93	76%
EG2	$fe=1.0$	223	5.61	28%	0.27	20.19	52%
ISREG	$fe=1.0$ $fq=0.0$	301	5.96	24%	0.28	42.15	66%

Todos os valores deste domínio são numéricos, o que novamente levou à necessidade de realizar uma discretização desses valores e, como resultado, provocou um elevado número de exemplos ambíguos (57). Esse fato tem conseqüências distintas para as duas famílias de algoritmos, a dos que geram árvore de decisão e não lidam com exemplos ambíguos (ID3, IDRT, EG2) e a dos que aceitam exemplos ambíguos e, conseqüentemente, geram regras em conflito (PRISM, FRPRISM e ISREG). Quando comparamos os resultados das duas famílias, os da primeira geram um número menor de regras, que é um ponto positivo, mas também apresentam uma menor frequência de acertos (acurácia), que é um ponto bastante negativo. Já os da segunda família, embora gerem um número maior de regras, conseqüência da aceitação de exemplos ambíguos, apresentam um índice de acurácia maior, o que é um ponto bastante positivo.

A matriz de relevância eliciada para este domínio possui muitos valores irrelevantes o que, novamente, leva a um baixo nível de qualidade semântica da base gerada e a um elevado número de regras inúteis, uma vez que os graus de qualidade das regras geradas estão muito próximos do limiar de utilidade da regra (0.25), utilizado para caracterizar uma regra com inútil. Mesmo assim, os algoritmos FRPRISM e ISREG apresentaram uma boa frequência de acerto. O EG2 gerou a base de menor custo, porém,

apresentou os menores valores para a acurácia (52%) e para o $GRBC$ (0.27), e também apresentou o maior número de regras inúteis(28%). Já o ISREG, embora apresente o maior números de regras geradas (369), conseqüência do elevado número de regras em conflito (66), apresentou os menores valores para tamanho médio de regras (5.14) e para o percentual de regras inúteis (9%) e os maiores valores para acurácia (76%) e para o $GRBC$ (76%).

Análise objetiva

Os resultados da comparação entre os algoritmos semânticos e não semânticos e do ISREG com o IDRT são os mostrados na TAB. 8.11.

TABELA 8.11 - Comparação objetiva entre os algoritmos semânticos e não semânticos para o domínio "Pima-indians-diabetes".

Domínio: Pima-diabetes		$(s_k \cdot f_k \cdot \frac{P_{ki} - P_{kj}}{\max(P_{kA}) - \min(P_{kA})}) \cdot 100$						Freq.	IQG (%)
Algoritmos		Regras			$GRBC$	\overline{Custo}	Acerto		
Ai	Aj	Qtde $s_k = -1$	Tam. $s_k = -1$	Inut $s_k = -1$	$s_k = +1$	$s_k = -1$	$s_k = +1$		
IDRT	ID3	-9	-5	47	58	-26	-8	9.5	
IDRT	PRISM	15	-70	79	41	34	-7	15.3	
IDRT	EG2 (fe=1.0)	-2	-13	95	75	-41	13	21.2	
FRPRISM	ID3	-100	52	52	83	-84	79	13.7	
FRPRISM	PRISM	-76	-13	84	67	-24	13	8.5	
FRPRISM	EG2 (fe=1.0)	-93	43	100	100	-100	100	25.0	
ISREG (fe=1.0, fq=0.0)	EG2 (fe=1.0)	-50	-41	21	8	-67	58	-11.8	
ISREG (fe=0.0, fq=1.0)	EG2 (fe=0.0)	-100	52	52	83	-84	79	13.7	
ISREG (fe=0.0, fq=0.5)	IDRT (fq=0.5)	-72	73	-21	0	-51	8	-10.5	
ISREG (fe=0.0, fq=1.0)	IDRT (fq=0.5)	-91	57	5	25	-58	87	4.2	

Da análise dos dados dessa tabela podemos concluir que:

a) Na comparação com os algoritmos não semânticos, o IDRT apresenta três *índices de qualidade global* bastante positivos. As duas únicas diferenças negativas apreciáveis ocorrem na grandeza custo (-41% em relação ao EG2) e na grandeza $\overline{Tam.}$ (-70% em relação ao PRISM). A primeira é perfeitamente justificável visto que o EG2 é um especialista em custo. Para a segunda, a

explicação é igual a que apresentamos anteriormente (item (a) da seção 8.5.3) para o caso do domínio Heart-disease-Cleveland.

b) Igualmente ao IDRT, o FRPRISM também apresenta *índices de qualidade global* bastante positivos. Um destaque especial é para o aspecto semântico, onde o FRPRISM, em comparação com os algoritmos não semânticos, apresenta diferenças positivas bastante acentuadas nas grandezas *Inut* e *GRBC*. Outro aspecto com diferenças também bastante positivas é o da acurácia.

c) Na comparação do ISREG com os algoritmos EG2 e IDRT observamos um certo equilíbrio. Para o primeiro, enquanto o ISREG predomina nas grandezas relacionadas ao aspecto semântico, que são *Inut* e *GRBC*, o EG2 confirma a sua superioridade no aspecto custo. Com relação ao IDRT, como se trata também de um algoritmo semântico, o ISREG só consegue um melhor resultado para o aspecto semântico, e também para o *índice de qualidade global*, quando prioriza totalmente esse aspecto, i.e., usa ($f_q=1.0$). Na verdade, para o domínio "Pima-diabetes" o ISREG não apresenta um resultado satisfatório quando comparado com o IDRT. Novamente, a exemplo dos domínios Zoo e Heart-disease-Cleveland, um aspecto muito importante é que nas quatro comparações, o ISREG obteve três diferenças bastante positivas para a grandeza $\overline{Tam.}$. Outro fato igualmente importante é que, nas comparações, todas as diferenças qualitativas no aspecto acurácia são também bastante significativas.

8.6 Análise final

Na seção anterior realizamos, para cada domínio isoladamente, um estudo do comportamento dos algoritmos semânticos, com especial atenção para o ISREG. Cada um desses estudos nos levou a um conjunto de conclusões parciais, que são de caráter localizado para cada domínio. Procedendo a uma análise dessas conclusões parciais, chegamos a um conjunto de conclusões de caráter geral, que, pelo menos para os domínios estudados, poderemos considerá-las válidas.

8. 6. 1 Conclusões de caráter geral

Em linhas gerais, para esses domínios essas conclusões são:

- a) O número de regras (Qtde) produzidas pelos algoritmos que geram regras modulares (PRISM, FRPRISM e ISREG), em geral, é maior do que daqueles que geram árvores decisão (ID3, IDRT e EG2). Isso acontece porque os algoritmos da primeira família (os "modulares"), além de serem mais robustos, um exemplo pode ser mapeado por mais de uma regra, também aceitam exemplos ambíguos, o que evidentemente leva à geração de um número maior de regras.
- b) Os algoritmos semânticos apresentam uma frequência de acerto superior a dos não semânticos, sendo que essa superioridade é mais acentuada entre os semânticos "modulares" e os algoritmos não semânticos que geram árvores de decisão. Mesmo na comparação de dois algoritmos semânticos, por exemplo ISREG e IDRT, o "modular" ISREG apresenta uma acurácia superior a do não "modular" IDRT. No nosso entender, a explicação para a superioridade, no aspecto acurácia, apresentada pelos algoritmos semânticos, em particular os "modulares", é a mesma que usamos anteriormente (item (a) da seção 8.5.4) para justificar o porque da geração de um número maior de regras por parte desses algoritmos.
- c) Em geral, para todos os domínios, o tamanho médio das regras ($\overline{\text{Tam.}}$) geradas pelos algoritmos "modulares" é inferior ao $\overline{\text{Tam.}}$ das regras produzidas pelos algoritmos não "modulares", que são os que geram árvore de decisão. Isto porque, enquanto os algoritmos "modulares" constroem as regras de uma forma individualizada, buscando as "melhores" condições (pares atributo-valor) para cada uma dessas regras, os algoritmos não "modulares" constroem as regras de uma forma coletiva, procurando sempre, de uma forma gradativa, o "melhor" atributo, objetivando, assim, uma otimização global do conjunto de regras. Assim, em geral, enquanto os algoritmos "modulares" geram um número maior de regras com um $\overline{\text{Tam.}}$ menor, os não "modulares" apresentam um comportamento inverso, geram um número menor de regras com um $\overline{\text{Tam.}}$ maior.
- d) Quando comparados com os não semânticos, como era de se esperar, os algoritmos semânticos apresentam uma superioridade inquestionável no aspecto semântico, que é representado pelo percentual do número de regras inúteis (Inut) e pelo grau de relevância da base de conhecimento ($GRBC$). A

única exceção ocorreu com o domínio Zoo, na comparação do ISREG com o EG2, mas mesmo assim, nesse caso, o ISREG desprezou o fator de qualidade semântica, considerando ($f_q=0.0$). Considerando todos os domínios estudados, a comparação do ISREG com o IDRT mostra que esses algoritmos apresentam um certo equilíbrio para o aspecto semântico, o que é bastante razoável visto que ambos são semânticos, embora o primeiro seja "modular" e o segundo não.

8.6.2 Quadro resumo

Objetivando se ter uma visão macroscópica do comportamento dos algoritmos semânticos diante dos não semânticos e, em particular, do ISREG em relação ao EG2 e IDRT, apresentamos na TAB. 8.12 os resultados das comparações desses algoritmos considerando todos os domínios conjuntamente.

TABELA 8.12 - Comparação objetiva entre os algoritmos semânticos e não semânticos para todos os domínios conjuntamente (média dos *índices de qualidade global*).

Algoritmos		Domínios					Média entre domínios	
A_i	A_j	Ame-norréia	Zoo	Heart	Pima	Brinq. Seguro	com BrinqSeg	sem BrinqSeg
IDRT	$\overline{\text{Ñsemânticos}}$	14.1	-5.6	10.1	15.3	31.1	13.0	8.5
FRPRISM	$\overline{\text{Ñsemânticos}}$	21.7	23.3	18.6	8.2	-6.5	13.2	18.0
ISREG	$\overline{\text{EG2}}$	11.9	31.8	8.5	1.0	5.6	11.8	13.3
ISREG	$\overline{\text{IDRT}}$	5.6	1.8	4.3	-3.2	30.4	7.8	2.1

A TAB. 8.12 mostra um quadro resumo dos índices de qualidade global entre os algoritmos comparados. Cada elemento de uma linha dessa tabela representa a média dos *índices de qualidade global*, resultantes da comparação entre um algoritmo semântico e os algoritmos não semânticos (vide seção 8.5). A última linha refere-se à comparação do ISREG com o IDRT. As duas últimas colunas fornecem o *índice de qualidade global* médio entre todos os domínios estudados, considerando e não considerando respectivamente o domínio "brinquedo seguro".

Conforme mostra a TAB 8.12, em média, considerando todos os domínios estudados (com e sem o domínio "brinquedo seguro"), os algoritmos

semânticos apresentam um nível de qualidade global superior aos não semânticos. O algoritmo ISREG, quando comparado com o EG2, que é seu "concorrente" no aspecto custo, também apresenta um *índice de qualidade global* médio positivo. Já em relação ao IDRT, notamos um certo equilíbrio dos dois algoritmos, mas mesmo assim, ainda com uma ligeira predominância do ISREG.

8. 6. 3 O Algoritmo ISREG

Conforme já mencionamos neste trabalho, o objetivo principal do ISREG é permitir a utilização, de forma conjunta, das principais formas de conhecimento preliminar. Procura também reunir o que de melhor propuseram os algoritmos que o antecederam, em particular os algoritmos IDRT, FRPRISM e EG2. Dessa forma, era esperado do ISREG um desempenho relacionado com a priorização dada a cada um dos conhecimentos preliminar utilizados. Assim sendo, o algoritmo deveria apresentar resultados idênticos àqueles obtidos pelos algoritmos originais que utilizem o mesmo conhecimento preliminar.

Além de atender às expectativas iniciais, o ISREG, em alguns casos, conseguiu produzir resultados superiores àqueles apresentados pelos algoritmos mais especializados em um determinado tipo de conhecimento preliminar. Como foi o caso da geração de bases de conhecimento com um custo médio de classificação menor do que o apresentado pela base gerada pelo EG2 que é o algoritmo especialista em custo.

Os resultados salientaram uma outra característica importante do ISREG: a possibilidade de buscar um equilíbrio entre os fatores de custo, qualidade semântica e tamanho da base gerada, permitindo a escolha de uma base que melhor atenda às necessidades do usuário.

9 A4: Ambiente de apoio à aquisição automática de conhecimento utilizando algoritmos indutivos

Nos últimos anos surgiram vários algoritmos generalizadores, bem como algumas técnicas de simplificação dos resultados desses algoritmos. Cada um desses algoritmos e técnicas de simplificação apresenta suas peculiaridades em termos de entrada, processo de indução e saída. Na entrada é necessário saber se o algoritmo impõe alguma restrição ao conjunto de treinamento e, se for o caso, que tipo de conhecimento preliminar utiliza. É necessário saber se o processo de indução embute algum tipo de simplificação. Se embute, qual? Se não embute, qual seria a técnica de simplificação mais adequada a ser utilizada? Para efeito de análise dos resultados, é necessário ainda saber qual o tipo de saída produzida pelo algoritmo? É árvore de decisão, conjunto de regras ou grafo de conhecimento? Além do mais, para um dado problema, ainda temos questões gerais tais como: Como e em que modelar o domínio? Como comparar ou mesmo fundir os resultados de dois ou mais algoritmos? Como analisar os resultados de algoritmos que usam informações semânticas?

Para dar resposta a todo esse questionamento, a forma que nos pareceu mais natural foi desenvolver um ambiente de apoio à aquisição de conhecimento. Esse ambiente é o A4 (Ambiente de Apoio à Aquisição Automática de Conhecimento Utilizando Algoritmos Indutivos) [Vasco 92a].

A idéia que permeia a definição do A4 é prover um ambiente integrado para auxiliar a tarefa de aquisição automática de conhecimento. O ambiente fundamenta-se em funcionalidades e técnicas de aprendizado automático, mas envolve também algumas características dos métodos baseados em entrevistas (semi-automáticos). Com o A4, objetiva-se fazer uso de vantagens

dos dois enfoques. A FIG. 9.1 mostra as camadas que compõem a arquitetura deste ambiente: a camada de modelagem do domínio, a de indução e a de tratamento das saídas.

Nesta tese especificaremos os componentes do A4, bem como a funcionalidade que cada um deles deve prover. O projeto e a especificação do ambiente seguem a filosofia de orientação a objetos. Objetiva-se atingir os benefícios que o paradigma traz, como: reusabilidade, confiabilidade e adaptabilidade.

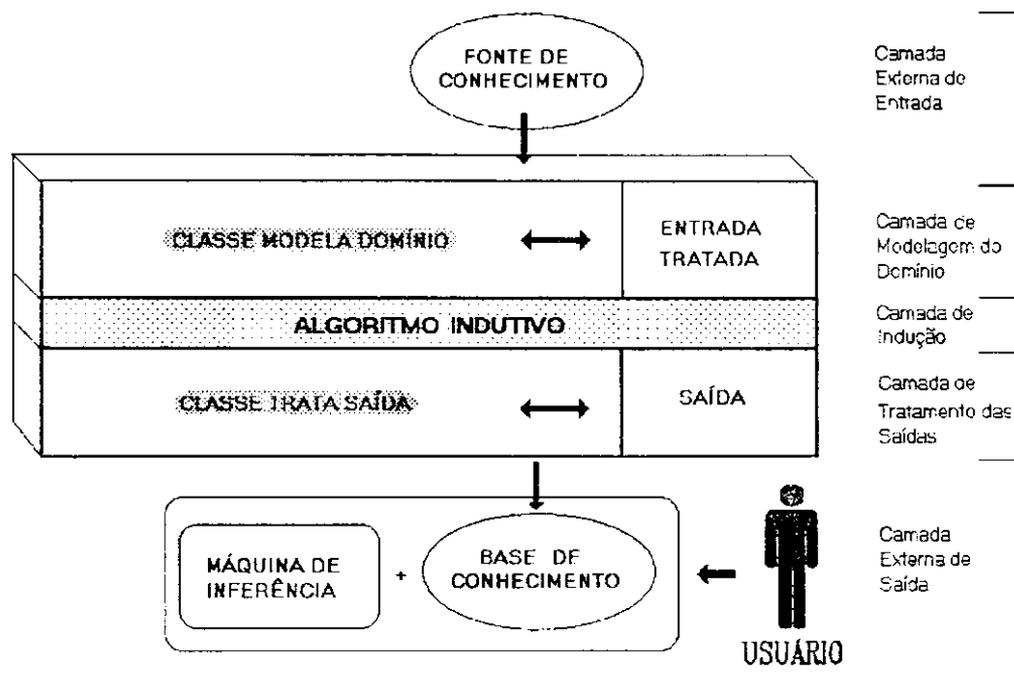


FIGURA 9.1 - Camadas da arquitetura do A4

9.1 Objetivos e características do A4

O A4 tem a finalidade de integrar os enfoques das técnicas de aquisição de conhecimento analisadas no capítulo 1 deste trabalho (técnicas semi-automáticas e automáticas), tentando preservar suas vantagens e minimizar suas desvantagens. O enfoque principal é centrado nos métodos de aprendizado automático indutivo. Os métodos baseados em entrevistas auxiliam a modelagem do domínio [Vasco 92b].

Os objetivos do A4 são:

1. Preencher o vazio existente na maioria dos métodos indutivos, com respeito à obtenção e utilização de conhecimento preliminar.
2. Prover um ambiente flexível, que possa abrigar diversos tipos de algoritmos generalizadores para avaliações comparativas entre eles e, conseqüentemente, permitir o uso do mais adequado a cada problema.
3. Facilitar a tarefa de construção da tabela de exemplos para, assim, tornar mais simples e eficiente o processo de aquisição de conhecimento.
4. Prover um ambiente que permita fácil extensão pela inclusão de novas ferramentas.
5. Minimizar a presença do especialista e do engenheiro de conhecimento no processo de aquisição de conhecimento.

Para atingir os objetivos anteriormente relacionados, o ambiente provê as seguintes funcionalidades:

- f1) Dispor de diferentes tipos de entrevistas focadas para a eliciação dos diversos tipos de conhecimento preliminar. O conhecimento preliminar não constitui uma forma "completa" de representar o conhecimento sobre um dado domínio. Sua intenção é captar apenas alguns aspectos objetivos do conhecimento. Essa objetividade é responsável por facilitar sobremaneira a sua eliciação, o que permite uma melhoria significativa na qualidade da indução automática, sem recair na exaustão inerente às entrevistas semi-automáticas.
- f2) Auxiliar o especialista/engenheiro de conhecimento na identificação de atributos em diferentes níveis de abstração, a partir de conhecimento preliminar, e da análise da correlação entre os atributos.
- f3) Identificar problemas na composição da tabela de exemplos que exigem tratamento anterior à aplicação de um determinado algoritmo generalizador, como: preencher "buracos" ("missing values"), identificar e tratar ambigüidades da tabela de exemplos, evitar a interdependência dos atributos, garantir a exclusão mútua de valores dos atributos, etc.

- f4) Prover mecanismos de validação dos resultados obtidos pelos algoritmos generalizadores.
- f5) Tornar as saídas geradas pelos algoritmos mais claras e compreensíveis, através da utilização de informações semânticas do especialista. Pretende-se, assim, aumentar a inteligibilidade do conhecimento gerado.
- f6) Prover mecanismos de simplificação, para tornar mais simples e eficientes as saídas geradas pelos algoritmos de indução.
- f7) Auxiliar na combinação de resultados gerados por diferentes algoritmos.

9.2 Arquitetura do A4

A FIG. 9.2 mostra a arquitetura do ambiente A4 com seus componentes.

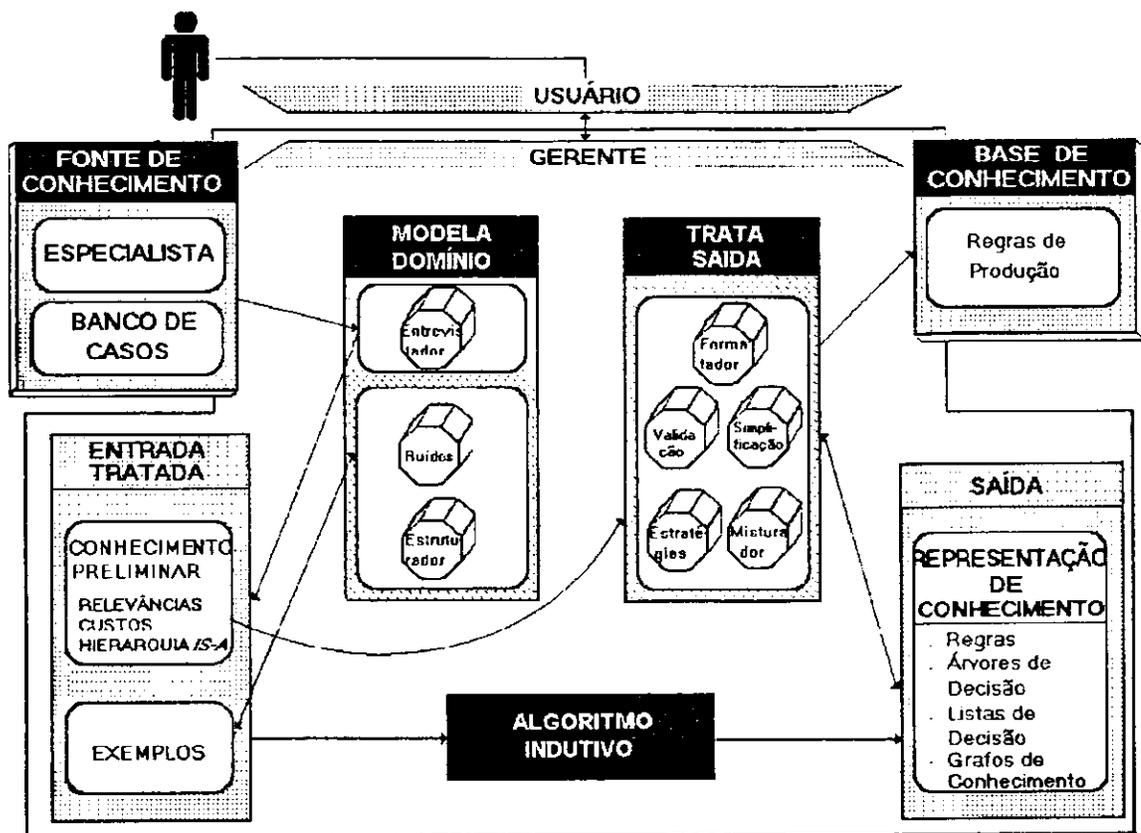


FIGURA 9.2 - Arquitetura do A4

O ambiente A4 não tem uma estrutura seqüencial de atuação. O usuário, através do GERENTE, pode determinar uma ordem própria de ativação dos objetos. Por exemplo, após a geração de uma saída por um determinado algoritmo generalizador, o usuário pode ativar as opções de simplificação e validação desta saída. Mas, em qualquer momento pode decidir remodelar a entrada. A tarefa de aquisição de conhecimento será realizada em várias e possivelmente repetidas etapas. Esta característica iterativa reflete uma propriedade importante do processo de aquisição de conhecimento que é a incrementabilidade. Ou seja, o conhecimento é adquirido aos poucos, com sucessivos refinamentos e incrementos de novas informações.

Dois componentes da FIG. 9.2 são externos ao A4: a FONTE DE CONHECIMENTO e a BASE DE CONHECIMENTO. A arquitetura contempla as informações relativas e estes componentes através das classes ENTRADA TRATADA e SAÍDA, respectivamente. A ENTRADA TRATADA serve como área de trabalho à modelagem do domínio. Já a idéia de separar a BASE DE CONHECIMENTO da SAÍDA objetiva compatibilizar as mais diversas formas de conhecimento que podem ser geradas pelos algoritmos. A BASE DE CONHECIMENTO tem um formato padrão, baseado em regras de produção, o que facilita o uso dos diversos "Shells" do mercado.

A seguir serão definidos os componentes do A4. Uma pequena descrição de suas funcionalidades acompanha as definições.

9.2.1 Gerente

O GERENTE tem como função básica integrar todos os componentes do ambiente e controlar sua utilização. A infra-estrutura básica para o funcionamento do ambiente, bem como as funcionalidades para criação e manutenção dos objetos também cabem ao GERENTE. A FIG. 9.3 mostra suas principais funções.

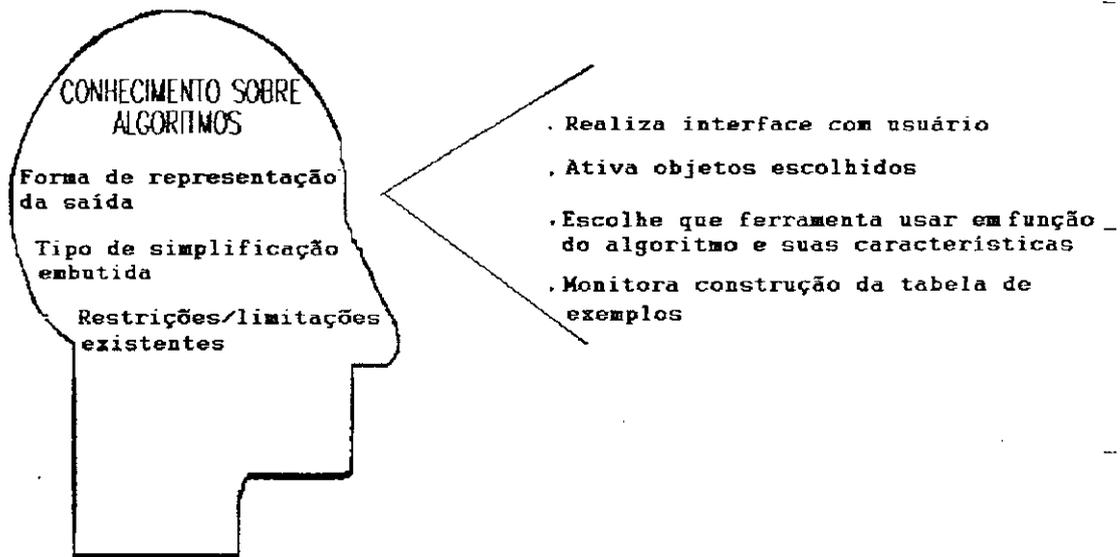


FIGURA 9.3 - As atribuições e conhecimento do gerente

É o GERENTE que se responsabiliza pela interface com o usuário. Há a necessidade dele estar sempre a receber algum evento, já que o usuário pode escolher opções de processamento em diferentes sequências.

Além da interface com o usuário, o GERENTE deve conter informações sobre os algoritmos generalizadores (por exemplo, que tipos de problemas resolvem, quais as restrições que impõem à modelagem do mundo real e que tratamentos podem ser aplicados às suas saídas).

9.2.2 Algoritmo indutivo

A classe ALGORITMO INDUTIVO é fundamental para o ambiente, mas a arquitetura não provê nenhum algoritmo novo. Na verdade, os algoritmos são tratados como caixas pretas. Há somente mecanismos para incluir e retirar algoritmos. É necessário que, ao se incluir um determinado algoritmo, ele se adeqüe à estrutura básica da arquitetura. Ou seja, os algoritmos têm que estar preparados para acessar as entradas e gerar as saídas dentro do formato definido pelo A4. Além disso, é necessário que se acrescente conhecimento ao GERENTE sobre as características básicas do novo algoritmo, como: tipo de representação de conhecimento usada, se realiza algum tipo de simplificação, quais limitações exige da entrada, se já trata "buracos", etc.

9.2.3 Entrada tratada

A ENTRADA TRATADA é um modelo da fonte de conhecimento. Ela pode ser manuseada pelos objetos da classe MODELA DOMÍNIO para se adequar ao algoritmo desejado. As informações podem estar na ENTRADA TRATADA sob duas formas: exemplos e conhecimento preliminar. Inicialmente, o A4 está preparado para trabalhar com três tipos diferentes de conhecimento preliminar: relevância semântica, custos e generalizações de valores de atributos. Contudo, o tratamento de outros tipos pode ser adicionado ao ambiente.

9.2.4 Modela domínio

A camada pré-indução é constituída da classe MODELA DOMÍNIO. Ela possui subclasses com objetos que atuam na ENTRADA TRATADA, antes da mesma ser utilizada pelos algoritmos generalizadores. Estes objetos têm a finalidade de auxiliar a modelagem do mundo real, facilitando a criação da tabela de exemplos, eliciando conhecimento preliminar sobre o domínio e adequando as entradas à aplicação do algoritmo desejado. A classe MODELA DOMÍNIO é composta por três subclasses: a classe de objetos que trata da estruturação do exemplo, a classe de objetos que trata ruídos e a classe entrevistadora (no capítulo 10 apresentaremos uma descrição mais detalhada desta classe).

A classe Estruturador auxilia a modelagem do domínio com objetos que identificam e refinam os atributos que compõem a estrutura do exemplo. Estes objetos propõem agrupamentos de atributos, realizam discretizações de valores contínuos, identificam interdependência de valores entre atributos e adequam a estrutura do exemplo às restrições pelo algoritmo generalizador.

A classe Entrevistador responsabiliza-se por eliciar do especialista os exemplos, bem como o conhecimento preliminar sobre o domínio. O processo de entrevista pode ser guiado pelo Entrevistador para que se selecione exemplos representativos do domínio em questão.

A classe Trata-ruído visa identificar e, se possível, tratar os ruídos que podem aparecer na formação da tabela de exemplos. Ruídos, como "buracos" e exemplos ambíguos, são examinados.

9. 2. 5 Trata saída

A classe TRATA SAÍDA possui objetos que atuam nas saídas dos algoritmos. Além do que pode ser acrescentado, estão previstos os seguintes objetos:

Validador

O Validador terá a função de testar a saída gerada pelo algoritmo generalizador quanto a sua eficiência e eficácia. Esta validação pode necessitar de mais semântica do especialista, o que pode forçar a comunicação do Validador com o especialista. Outra possibilidade de adicionar semântica na validação é utilizar as informações preliminares, como relevância semântica, adquirida pela classe MODELA DOMÍNIO junto ao especialista.

Além desta validação semântica pode-se submeter casos diferentes aos do conjunto de treinamento, para identificar o grau de acurácia da saída gerada. Para esta validação ser realizada é preciso um mecanismo de inferência nas saídas geradas pelo algoritmo.

Simplificador

Após a execução do algoritmo generalizador, obtém-se uma saída em alguma forma de representação de conhecimento. Essa saída pode ser aperfeiçoada através da diminuição do número de condições que levam à classificação dos exemplos. Isto pode acontecer principalmente em domínios em que se encontram ruídos. Nesses domínios, como os algoritmos tendem a gerar saídas que reflitam o domínio, verifica-se uma grande quantidade de conceitos destinados a cobrir casos específicos e de pouca importância, obtidos por causa dos ruídos. Eliminar estas partes pouco importantes ou até irrelevantes é função do Simplificador. Esta tarefa, também conhecida como poda, objetiva tornar a tarefa de classificação mais simples e eficiente. O que se observa é que a poda de uma saída, embora implique em perda de conteúdo sobre algumas classificações, no geral torna esta tarefa melhor realizada. Existem duas alternativas de realizar poda. De uma maneira geral, a poda depois do algoritmo ter gerado as saídas é mais aconselhável [Niblett 87], pois o algoritmo Simplificador tem uma visão global da base

gerada, mas algumas implementações são feitas no próprio algoritmo generalizador, como é o caso do algoritmo C4.5 [Quinlan 89].

Outro ponto importante quando se realiza simplificações nas saídas é que, em geral, após a simplificação, o conhecimento obtido está mais compreensível. Prover mais compreensão é fundamental para se dizer que um sistema de aprendizagem realmente adquiriu conhecimento. A clareza dos resultados auxilia, por exemplo, a prover boas explicações sobre classificações realizadas.

A tarefa de simplificação pode ser realizada, por exemplo, pela eliminação de arcos (ligação entre dois nós) de uma árvore de decisão ou de condições de uma regra. O Simplificador pode também receber informações semânticas vindas do especialista ou fazer uso do conhecimento já adquirido sobre o domínio, como relevância semântica.

Misturador

O Misturador é o objeto que integra as várias soluções geradas por algoritmos diferentes para um mesmo domínio. Suponha que um conjunto de regras de produção é gerado de uma árvore de decisão após a aplicação de um algoritmo da família *TDIDT*. O Misturador poderá comparar (com auxílio de conhecimento preliminar ou do especialista, através de entrevistas) que regras equivalem ou são contrastantes em relação a um outro conjunto de regras advindo, por exemplo, de um algoritmo tipo PRISM [Cendrowska 88]. Estas comparações podem gerar um conjunto de regras mais robusto e de melhor qualidade.

Estrategista

As estratégias são conhecimento sobre que ação tomar a seguir em uma dada situação. Estas decisões são tomadas considerando-se fatores que influenciam a qualidade da escolha.

Um exemplo do uso de estratégias pode ser visto num domínio médico. A escolha de que teste um paciente deve realizar para identificar uma certa disfunção é povoada de estratégia. O médico não solicita logo ao paciente aquele teste que pode dar o resultado com total precisão, pois fatores outros

podem estar envolvidos, como custo do teste, fator de risco do teste e o estado do paciente. Por exemplo, uma tomografia computadorizada, devido a seu custo, só é sugerida após a tentativa de descoberta do problema por outras formas. O exemplo em questão contempla a estratégia ligada a custo. Ou seja, se o teste é potencialmente conclusivo, mas o custo é muito alto, não o realize inicialmente. Outros tipos de estratégias podem ser vislumbrados, sendo que de uma forma geral relacionam-se à escolhas de que ações tomar a seguir.

Dentre as diversas formas de representar estratégias [Gruber 89], duas serão usadas no A4: meta-regras que controlam a ativação de regras de produção da base de conhecimento, e modificações na topologia ou ordem dos atributos da própria representação de conhecimento gerada pelo algoritmo indutivo.

Formatador

A classe de objetos que fazem formatação se responsabiliza por transformar a SAÍDA (representação de conhecimento intermediária) no formato da base de conhecimento. Aqui é contemplada a geração de regras de produção a partir de árvores de decisão, grafos de conhecimento e outras representações.

9. 2. 6 Saída

A classe SAÍDA contém objetos que armazenam as saídas advindas dos algoritmos. Ela deve aceitar formas de representação de conhecimento, que serão utilizadas dependendo do algoritmo que gerará estes conhecimentos. As formas de representação previstas são: regras de produção (PRISM [Cendrowska 88], RPRISM [Cirne 91a]), árvores de decisão (ID3 [Quinlan 86], APREND [Gomes 89], C4 [Quinlan 87]), listas de decisão (CN2 [Elomaa 89]) e grafos de conhecimento [Machado 90], mas nada impede que outras sejam agregadas.

O A4 está sendo desenvolvido em C++ em estações de trabalho SPARC/SUN 2. A interface do ambiente foi desenvolvida a partir de rotinas básicas do Xview/SUN e do Open Developments Windows Guide, também da SUN.

10 Modelagem do domínio

Os métodos automáticos indutivos consideram o conjunto de exemplos e até o conhecimento preliminar sobre o domínio (quando usado), como já disponíveis [Quinlan 87] [Cendrowska 88] [Cirne 91a] [Nuñez 91]. Na verdade, há um lacuna entre como estas entradas se apresentam no mundo real e como elas se tornam operacionais aos algoritmos indutivos. Por exemplo, a identificação dos atributos mais representativos que formam os exemplos e a conseqüente construção da tabela de exemplos tem se mostrado uma tarefa não trivial [Hart 89]. Essa tarefa, se não for devidamente informatizada diminui acentuadamente o caráter automático da aquisição de conhecimento.

Para preencher a lacuna acima citada, neste capítulo propomos formas de auxiliar a modelagem do mundo real em formas tratáveis pelos métodos indutivos de aquisição de conhecimento. A modelagem resultará em exemplos devidamente estruturados e em conhecimento preliminar sobre o domínio, como relevância semântica [Mongiovi 90a], custos e generalizações de valores de atributos [Nuñez 91]. O processo desta modelagem é realizado pela classe de objetos MODELA DOMÍNIO, que está inserida no ambiente A4 (descrito no capítulo 9).

10.1 Em que modelar o domínio

Os trabalhos realizados em aquisição indutiva de conhecimento têm focado quase que exclusivamente o processo de generalização em si. Entretanto, para que os métodos indutivos de aquisição de conhecimento apresentem resultados satisfatórios, é necessário que as entradas dos algoritmos generalizadores reflitam ao máximo o domínio do problema a ser resolvido.

Pode-se aqui reafirmar a tradicional máxima de processamento de dados de que "se entra lixo, sai lixo".

No processo de aquisição de conhecimento via técnicas indutivas, o domínio tem sido modelado na forma de exemplos e conhecimento preliminar. Sendo que a primeira é a única forma de entrada para a maioria dos métodos indutivos. A seguir, apresentamos uma descrição conceitual dessas duas formas.

10.1.1 Os Exemplos

A definição mais comumente encontrada na literatura é que um exemplo ou caso é um conjunto de pares atributo/valor e um elemento de classificação. Tabela de exemplos é uma seqüência de exemplos advindos diretamente do especialista ou de uma outra fonte de conhecimento (banco de dados, fichários, etc.).

Os atributos que compõem um exemplo podem ter valores lineares (ordinais ou contínuos), nominais ou estruturados hierarquicamente [MacDonald 89]. A FIG. 10.1 mostra exemplos dos três tipos de pares atributo/valor.

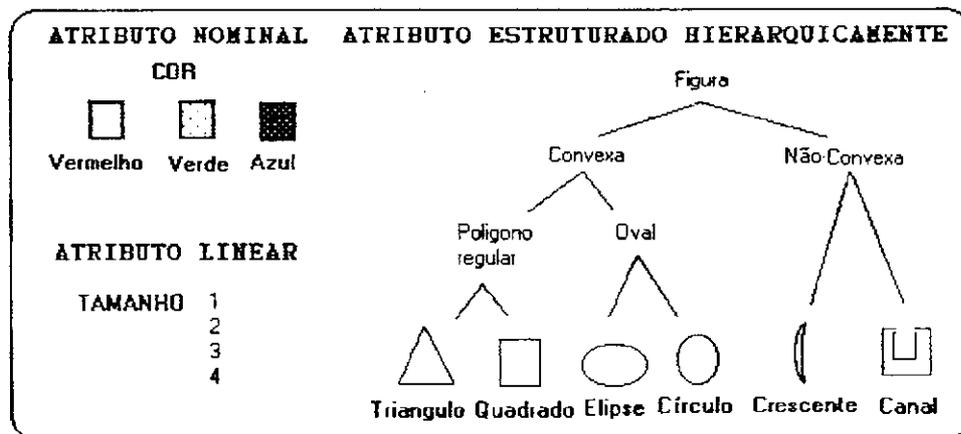


FIGURA 10.1 - Tipos de valores de atributos

A seleção de quais atributos e valores devem compor a estrutura do exemplo tem que ser feita de forma criteriosa, pois eles devem ser representativos do domínio em questão. Além disto, se persegue uma racionalização nos valores dos atributos, evitando, por exemplo, valores contínuos.

Na modelagem dos exemplos deve-se levar em consideração as restrições impostas pelos métodos de aprendizagem automática que os utilizam. Algumas destas limitações são em decorrência de limitações intrínsecas aos próprios algoritmos. Por isto, elas podem variar de algoritmo para algoritmo, como a aceitação ou não de "buracos", onde divergem o ID3 (que não os aceita) [Quinlan 86] e o C4 (que os trata) [Quinlan 89].

Contudo, a mais generalizada limitação dos métodos de aprendizagem automática reside na possibilidade de crescimento do espaço de soluções do problema. Esta situação ocorre, principalmente, em domínios não determinísticos e que apresentam um grande número de atributos. Nestes casos, além do grande esforço computacional exigido, as saídas podem se tornar por demais complexas e, conseqüentemente, incompreensíveis ao especialista. Em virtude disto, busca-se, na modelagem dos exemplos, atingir uma "quantidade tratável" de atributos. É importante ressaltar que "quantidade tratável" de atributos, em sendo um conceito contextual, deve ser definido em função de cada situação.

10.2.2 Conhecimento preliminar

O conhecimento preliminar ou informações semânticas sobre o domínio constituem conhecimento do especialista sobre o problema a ser resolvido. Estas informações devem complementar o conhecimento que é capturado somente através de informações sintáticas, como os exemplos. Na semântica do especialista é que está embutido o conhecimento heurístico.

Relevância semântica

A relevância semântica relaciona, em um dado domínio, elementos de classificação com os atributos e seus respectivos valores. Relevância, de um modo geral, indica a importância que tem um determinado par atributo/valor para concluir um elemento de classificação. Os algoritmos generalizadores têm, na sua maioria, condições de descobrir relevância sintática através de estatísticas. Como esta relevância nem sempre reflete totalmente a realidade, dá-se a necessidade do uso de relevância semântica.

Uma forma de representar a relevância semântica é através de uma matriz de relevância ou matriz de relevância nebulosa (vide capítulo 4).

Custo

Informações sobre o custo de cada atributo é outro tipo de conhecimento que não é contemplado pela maioria dos métodos indutivos. As informações sobre custos não são necessariamente monetárias, mas podem ser vistas também em função de risco de vida, dificuldade de aplicação, urgência da informação, etc.

Os métodos indutivos de aquisição de conhecimento que geram árvores de decisão podem fazer uso das informações sobre custo dos atributos, no momento de determinar que atributo deve ser eleito para a raiz da árvore. Deve-se levar em consideração não apenas a quantidade de informação do atributo [Quinlan 89], mas uma relação entre custo e benefício do atributo.

Generalização de valores de atributos

Através de modelos que reflitam a organização hierárquica dos valores dos atributos do domínio, pode-se facilitar a aprendizagem a partir de exemplos. Uma possibilidade disto pode ser vista em [Nuñez 91], no qual uma rede de dependência IS-A entre os valores dos atributos auxilia a substituição de valores de atributos apresentados nos exemplos por informações mais genéricas. Informações hierárquicas são úteis para definir o grau de generalização, ou o nível de abstração que se deseja trabalhar. Este tipo de informação influencia fortemente a quantidade de pesquisa que o algoritmo tem de realizar. A FIG. 10.2 mostra um exemplo de como informações hierárquicas podem ser usadas por algoritmos indutivos para gerar árvores menores. Este exemplo contempla um tipo de generalização chamada "climbing generalization tree" [Michalski 83].

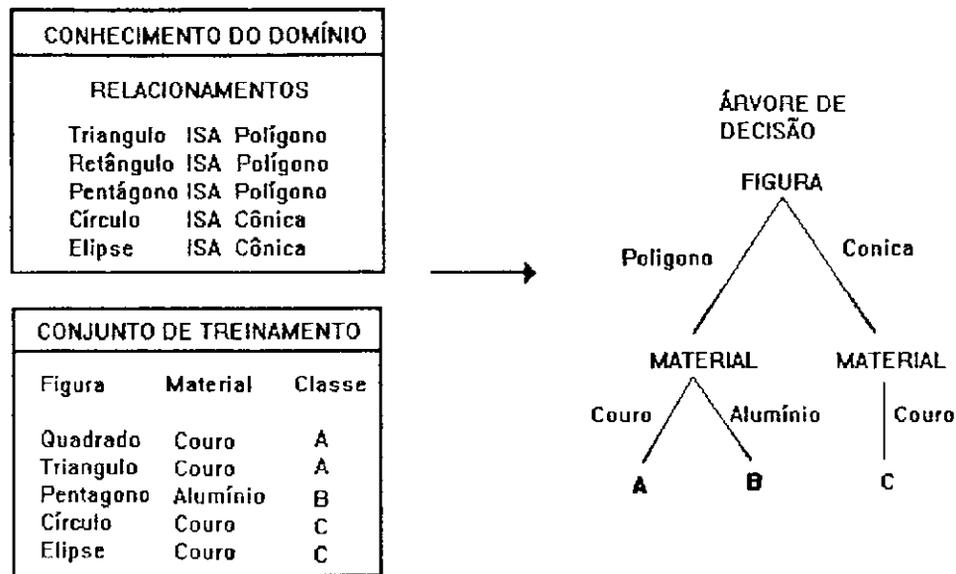


FIGURA 10.2 - Utilização de conhecimento hierárquico em ADs

10.2 A Classe modela domínio

A classe MODELA DOMÍNIO possui subclasses com objetos que representam o mundo real em entradas tratáveis pelos algoritmos generalizadores. Estes objetos facilitam a criação da tabela de exemplos e a eliciação de conhecimento preliminar, adequando-os à aplicação de um determinado algoritmo. Conforme pode ser visto na FIG. 10.3, a classe MODELA DOMÍNIO é composta por três subclasses: a classe de objetos que trata de "entrevistar" o especialista, a de estruturar os exemplos e a classe que trata ruídos. A seguir apresentaremos uma descrição de cada uma delas.

10.2.1 Entrevistador

A classe Entrevistador é responsável por entrevistar o especialista para identificar quais são os atributos, valores e o elemento de classificação que compõem a estrutura de um exemplo. Nesta entrevista são feitas sugestões de quais exemplos devem ser selecionados e de como estruturá-los. Além disto, cabe à classe Entrevistador a eliciação de conhecimento preliminar sobre o domínio.

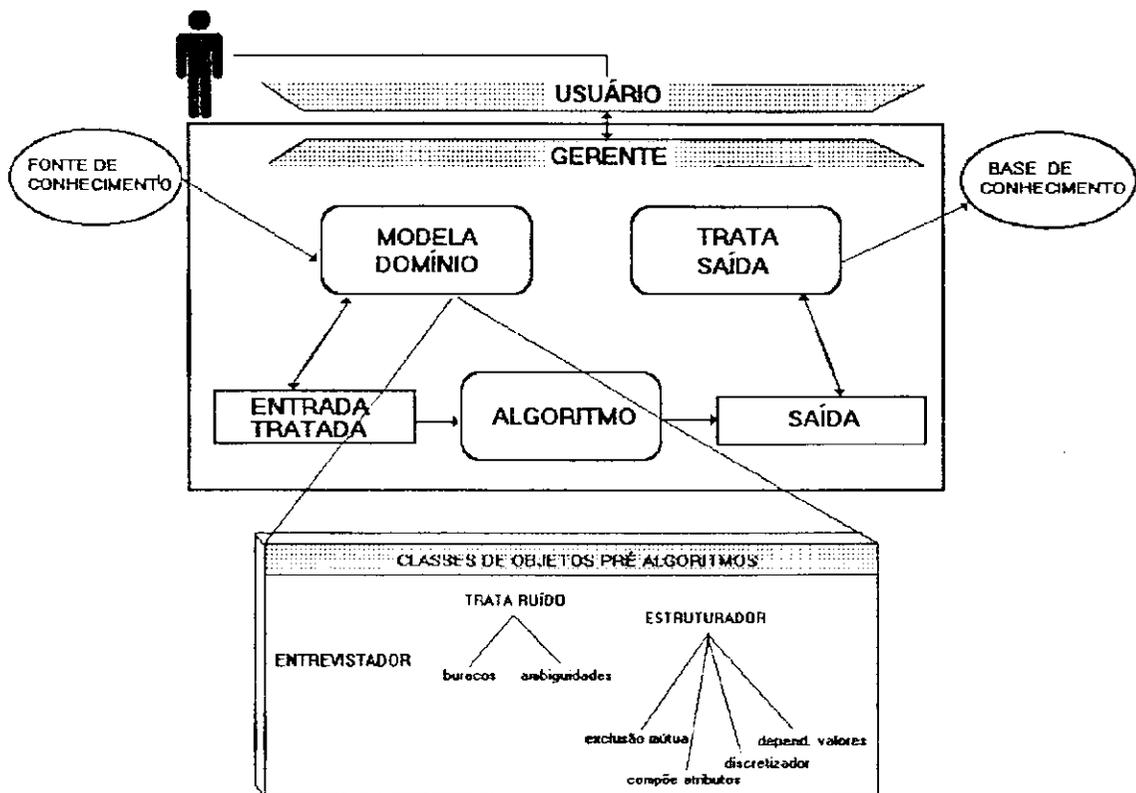


FIGURA 10.3 - A classe MODELA DOMÍNIO no contexto do A4

10.2.2 Estruturador

A classe Estruturador tem a tarefa de auxiliar a criação da tabela de exemplos, através da identificação e refinamento da estrutura desses exemplos. Isto é realizado através das subclasses: Compõe-atributo, Trata-interdependência-de-valor, Trata-exclusão-mútua e Discretizador.

Compõe-atributo

Em princípio, a modelagem de certos domínios pode requerer um grande número de atributos. Conforme visto na seção 10.1, dentro do possível, isto deve ser evitado. Uma estratégia possível para trabalhar com um número menor de atributos é compor (fundir) dois ou mais atributos em um só. Isto pode ser feito com atributos cujos valores não ocorram simultaneamente na conceituação de um exemplo. Caso os valores dos atributos ocorram simultaneamente, pode-se então tentar fundi-los e combinar esses valores. Esta estratégia está ilustrada na FIG. 10.4.

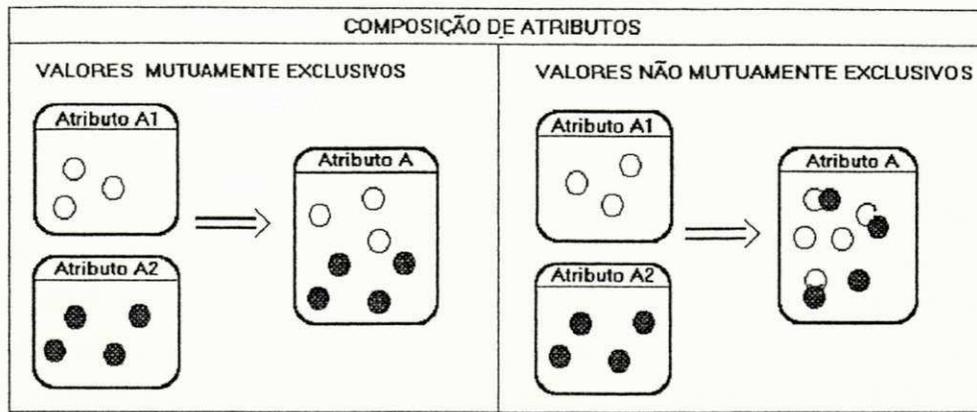


FIGURA 10.4 - Composição de valores de atributos

Algumas heurísticas são usadas para identificar atributos que podem ser agrupados. Por exemplo, atributos que aparecem pouco na tabela de exemplos (provavelmente têm baixo grau de importância), são candidatos em potencial ao agrupamento. Outra heurística possível é agrupar dois atributos que contenham poucos valores (dois ou três). Com esta exigência evita-se uma explosão combinatória no número de valores do atributo resultante. Contudo, mesmo que a quantidade de combinações seja grande, pode-se tentar identificar se muitas delas são impossíveis de ocorrer na prática. Caso isto seja verdadeiro, a combinação também pode ser considerada boa.

Trata-interdependência-de-valor

Também perseguindo diminuir o número de atributos que compõem a estrutura do exemplo, pode-se buscar a identificação de atributos que mantenham uma certa dependência entre si. Isto acontece quando, numa tabela de exemplos de um certo domínio, percebe-se que toda vez que um atributo A possui valor "a" um outro atributo B possui valor "b". Trata-interdependência-de-valor identifica este relacionamento e os atributos envolvidos são, então, candidatos naturais ao agrupamento. Este tipo de auxílio serve também para identificar níveis hierárquicos diferentes entre os atributos, pois os atributos que estão se mostrando dependentes podem ser substituídos por um outro de nível de abstração maior.

Discretizador

Outra característica fundamental de vários métodos indutivos é a necessidade de discretizar valores contínuos (transformar valores lineares em nominais). Isto é fundamental no desempenho dos algoritmos. O discretizador examinará os valores dos atributos para classificá-los dentro de uma determinada escala. Na prática, nem sempre é trivial identificar a priori em quantas faixas os valores de um certo atributo devem ser divididos. Observe que a importância de um mesmo atributo é contextual. Por exemplo, num contexto legal geralmente é suficiente discretizar o atributo idade em MENOR (< 18 anos) e MAIOR (≥ 18 anos). Já num contexto desportivo, poderíamos ter os seguintes valores INFANTIL (< 12), JUVENIL (de 12 a 18), ADULTO (de 18 a 35) e SENIOR (> 35).

Trata-exclusão-mútua

Existem restrições e limitações impostas à modelagem do mundo real que são específicas de cada algoritmo. Uma restrição da maioria dos algoritmos é que os valores dos atributos sejam mutuamente exclusivos. Uma solução possível para esta exigência é dividir o atributo em dois e redistribuir seus valores, se a quantidade de atributos for tratável. Caso contrário, os valores não mutuamente exclusivos são combinados. Identificar estes problemas e auxiliar as adaptações necessárias são tarefas do Trata-exclusão-mútua.

10.2.3 Trata ruído

Os ruídos são informações danosas à execução dos algoritmos. Eles acontecem porque na grande maioria dos domínios do mundo real não há uma garantia de que, num exemplo, a correlação entre os pares atributo/valor e os respectivos elementos de classificação é fidedigna. A tabela de exemplos pode conter exemplos ambíguos, o que pode prejudicar ou até impossibilitar a execução de muitos algoritmos generalizadores (por exemplo, alguns da família *TDIDT*). Alguns destes ruídos resultam de falhas em medições, valores ausentes, limites mal definidos (ex: Quanto uma pessoa é alta?) e interpretações subjetivas (ex: Que critérios são usados para descrever uma pessoa como atlética?). Há uma enorme subjetividade envolvida na identificação de ruídos, dificultando assim um tratamento genérico.

Entretanto, alguns tipos de ruídos podem ser identificados e tratados. Dois dos ruídos frequentes são: "buracos" e ambigüidades.

Trata-buraco

Os "buracos" ("missing values", "unknwon values") são encontrados quando há atributos não valorados na tabela de exemplos. Este problema é extremamente sério e deve ser contornado, pois informações faltando podem levar a resultados incorretos. Uma das dificuldades ao se tratar os "buracos" é que eles podem acontecer por razões diversas. Uma razão é devido a impossibilidade de se obter o valor do atributo. A outra alternativa é porque o atributo não é necessário para o exemplo. A terceira é porque a presença do atributo é indiferente na conclusão do exemplo.

Trata-buraco é o responsável em contornar esses problemas, considerando essas diversas perspectivas e as características específicas de cada algoritmo generalizador.

Trata-ambigüidade

Em algumas situações, a tabela de exemplos pode apresentar casos com um mesmo conjunto de pares atributo/valor concluindo classes diferentes. Esta situação pode caracterizar um ruído na tabela de exemplos, um caso intencional de exemplos ambíguos ou mesmo uma falha na estrutura do exemplo (situação em que outro atributo pode distinguir os exemplos, desfazendo, assim, a ambigüidade). Nestas ocasiões, Trata-ambigüidade alerta o usuário, auxiliando-o, quando necessário, na solução do problema.

10.3 Como realizar a modelagem

Em situações reais, observa-se que a obtenção de uma estrutura que identifique com exatidão os exemplos, com seus atributos e valores, tais quais se apresentam no mundo real, apresenta alguma complexidade. A seleção de quais atributos e valores devem compor um exemplo, tem que ser feita levando-se em consideração as limitações impostas pelos métodos de aprendizado automático que os utilizam.

No ambiente A4, a modelagem do mundo real é fundamentada em interações do especialista e/ou engenheiro do conhecimento com o ambiente. A filosofia deste processo interativo é exigir o mínimo do especialista. Ele é necessário somente para construir a estrutura da tabela de exemplos, fornecer algumas características e, posteriormente, validar os resultados obtidos. Esta participação, considerada pequena em relação aos métodos semi-automáticos, enriquece os resultados que os algoritmos generalizadores provêm.

10.3.1 Alternativas de eliciação

No ambiente A4, a aquisição dos elementos necessários à formação da estrutura de um exemplo e do conhecimento preliminar tem duas alternativas distintas de ser realizada.

1) É feita a alimentação do conjunto de casos sem a formação de uma estrutura prévia dos exemplos. A medida que os casos vão sendo fornecidos, os atributos e seus respectivos valores vão sendo cadastrados.

2) Dividi-se em duas fases explícitas.

2a) Criação de uma estrutura básica para os exemplos. Nesta fase, os objetos que tratam as entradas estão ativos e podem propor modificações na estrutura dos exemplos. Estes objetos também guiam o processo de alimentação dos casos, auxiliando para que sejam selecionados casos mais representativos e, provavelmente, com atributos mais significativos.

2b) Alimentação do grande contingente de casos.

Outras alternativas poderiam existir, mas na realidade seriam uma variação das duas citadas. Por exemplo, se o especialista, desejar (puder) alimentar todos os exemplos, seria como se a fase de estruturação (2a) da alternativa 2 se prolongasse ao longo de todo o processo de modelagem do domínio.

Em ambas as alternativas anteriores, ao final da alimentação de todos os exemplos, é feita uma nova crítica nas entradas e é gerada uma versão para validação pelo especialista.

A escolha da alternativa mais adequada de realizar o processo de criação da tabela de exemplos é circunstancial. O que se verifica como ponto positivo na

alternativa 2 é que ela possibilita uma maior liberação do especialista, que é necessário somente na fase de estruturação e ao final na validação. Nesta opção, a fase de validação provoca poucas alterações na estrutura inicial, visto que na formação da estrutura do exemplo foram usados casos selecionados e mais representativos. Já a alternativa 1 deve ser escolhida quando se trabalha em domínios onde os exemplos se apresentam com uma certa organização. O processo de alimentação poderá então ser realizado por um usuário com pouco conhecimento do domínio, não necessariamente o especialista.

O conhecimento preliminar sobre o domínio segue as mesmas alternativas dos exemplos, em virtude dele ser adquirido ao mesmo tempo que os exemplos.

10.3.2 O Processo de entrevista

As fases do processo de eliciação de exemplos e conhecimento preliminar são descritas na FIG. 10.5. A seguir faremos algumas observações sobre estas fases (de 1 a 5).

- 1) A informação sobre o domínio (médico, fiscal, educacional, etc) e a tarefa a se realizar (classificação, interpretação, diagnóstico, etc), indicam a forma de interagir do sistema, visto que diferentes terminologias podem ser usadas em diferentes tipos de problemas.
- 2) Os problemas de classificação que compõem os exemplos são logo apresentados pelo usuário, pois auxiliam o processo de entrevista e da obtenção de conhecimento preliminar.
- 3) O Entrevistador passa a solicitar casos que concluam os elementos de classificação apresentados. Ele poderá sugerir características dos casos a serem apresentados, solicitando, por exemplo, casos que se destaquem por frequência (alta ou baixa) de ocorrência, para identificar tanto os atributos mais comuns, como os mais raros. O auxílio do Entrevistador pode levar à solicitação de exemplos que representem pares atributo/valor ainda não presentes na tabela de exemplos. Devido à boa representatividade que possuem, os exemplos selecionados podem levar à formação de conceitos mais acurados e que podem classificar todo o contingente da tabela de exemplos mais rapidamente.

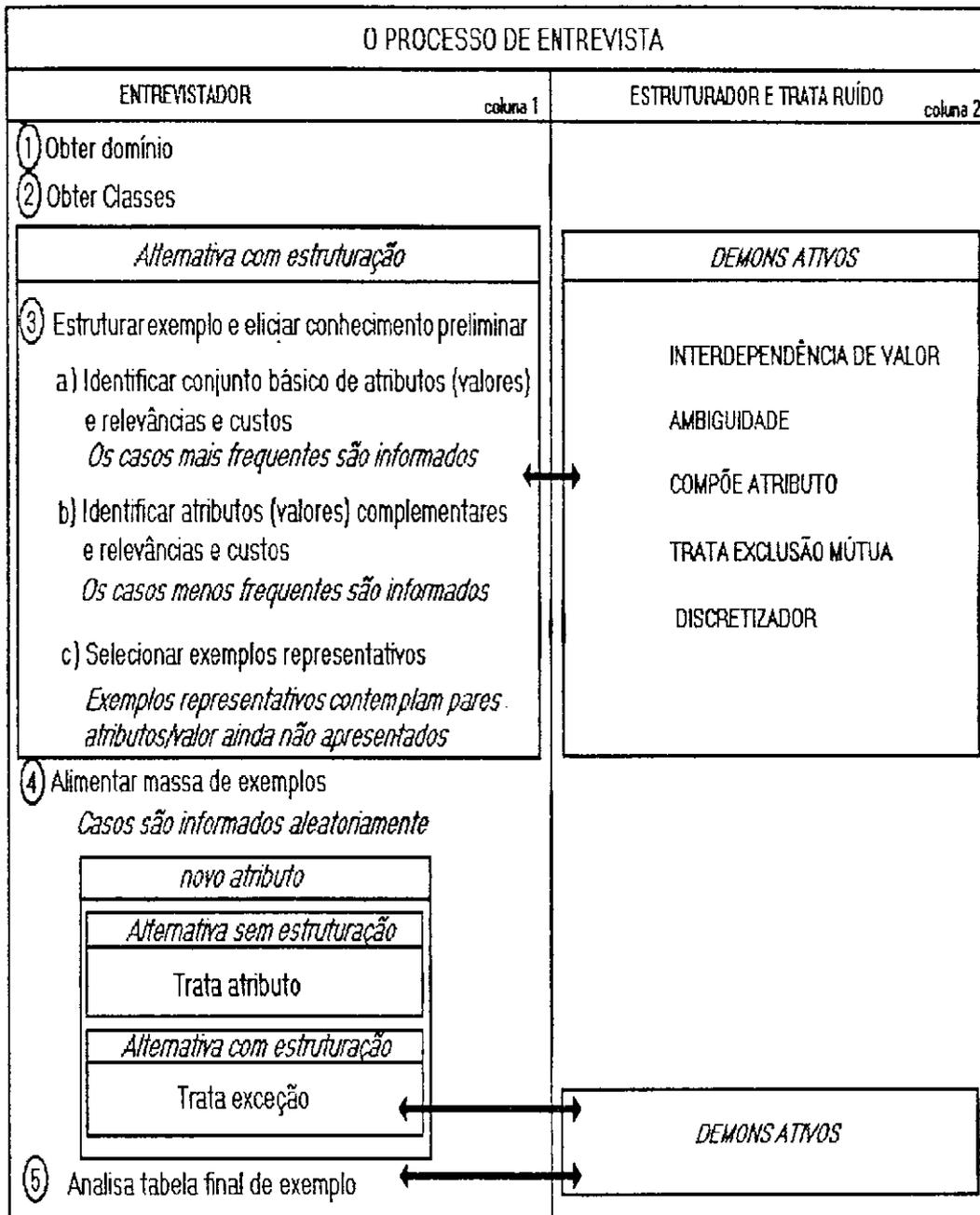


FIGURA 10.5 - Fases da entrevista da classe modela domínio

A aquisição de relevância semântica e de custo dos atributos é realizada nesta fase, juntamente com a identificação dos atributos e valores que compõem um exemplo. Isto é feito com entrevistas focadas, nas quais as perguntas não exigem um processo grande de raciocínio e as respostas não são muito amplas.

A coluna 2, referente às classes Estruturador e Trata-ruído, cita os objetos destas classes ("DEMONS"), podendo sugerir modificações na estrutura de exemplo. O usuário pode definir um número mínimo de exemplos

incluídos, que os objetos destas classes devem observar antes de se ativarem. Por exemplo, somente a partir de 10 exemplos incluídos e de um intervalo de 10 em 10 é que os objetos podem atuar.

- 4) Na fase exclusiva de alimentação de exemplos, são informados todos os casos que devem compor a tabela de exemplos. Nesta fase, se aparecerem novos atributos, o tratamento dado é em função da alternativa de eliciação escolhida. Se houve uma fase de estruturação de exemplos, a inclusão de novos atributos é tratada como uma exceção. Neste tratamento se controla autorizações para modificar a estrutura da tabela de exemplos criada pelo especialista. Se a opção de alimentação não segue a uma estruturação prévia de exemplos, a entrada de novos atributos obedece somente às críticas padrões de entrada de dados.

- 5) Ao se analisar a tabela de exemplos, independentemente da forma pela qual o processo de geração e digitação da mesma tenha sido realizado, todos os objetos da classe modela domínio são ativados para uma análise final. No A4, a tarefa de discretização é feita após uma análise da distribuição dos valores dos atributos na tabela de exemplos. A partir desta análise, pode-se propor intervalos que reflitam os valores ordinais. A discretização e a determinação da escala de valores resultantes, é feita interativamente com o engenheiro do conhecimento e/ou especialista. Também o tratamento de "buracos" é feito nesta fase, utilizando, por exemplo, a estratégia de preencher os "buracos" com o valor que mais ocorreu na tabela de exemplos para o atributo correspondente [Clark 89].

11 Uso de semântica na melhoria do aprendizado em um modelo simbólico-conexionista

11.1 Introdução

Uma boa forma de se investigar os diferentes paradigmas de Aprendizado Automático ("*Machine Learning*"), aproveitando-se das melhores características de cada um deles, é trabalhar em uma plataforma híbrida, que sustente vários paradigmas. O Modelo Neural Combinatório (MNC) é uma dessas plataformas, representando uma ponte importante entre os campos das redes neurais subsimbólicas e o de sistemas especialistas (SEs). Esse modelo é uma rede neural de ordem superior adequada para tarefas de classificação [Machado 89], que possibilita a construção de SEs conexionistas [Gallant 88].

O aprendizado no modelo é composto de duas fases, como pode ser visto na FIG. 11.1: o treinamento de uma rede inicial, que gera uma rede intermediária, com acumuladores e dados de controle; e a poda e normalização da rede intermediária, que gera finalmente a rede que pode ser efetivamente usada para inferência. O algoritmo de treinamento usa uma versão simples do mecanismo de retropropagação ("*backpropagation*") para captar as regularidades do domínio a partir de uma massa de casos reais. Tanto os dados de treinamento quanto os nós da rede gerada podem ser booleanos [Machado 89] ou difusos [Denis 91].

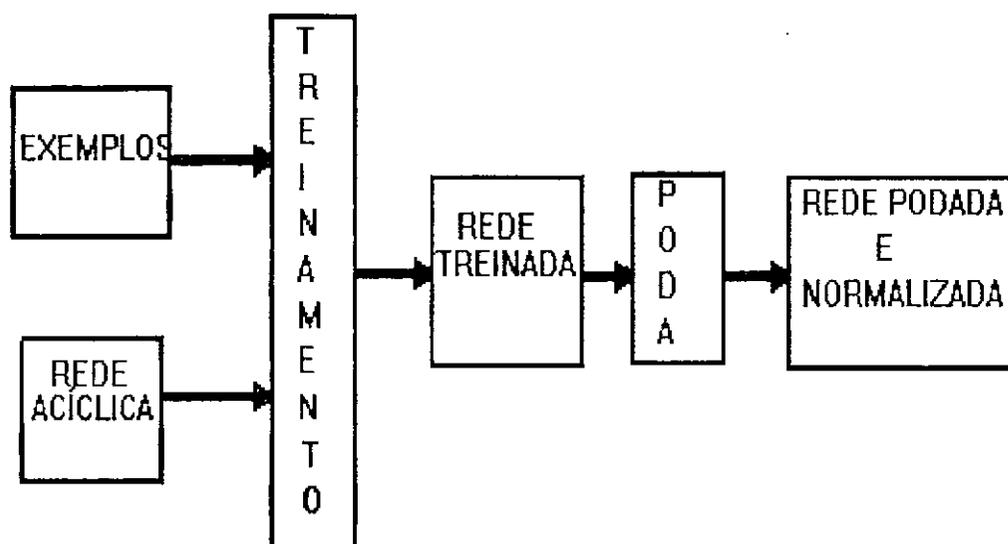


FIGURA 11.1 - Fases do aprendizado no Modelo Neural Combinatório.

Na primeira fase, são ajustados os pesos de uma rede acíclica, através de um mecanismo de punição das "asserções" (caminhos ou aglomerados da rede) contraditas pelos exemplos e recompensa daquelas reforçadas pelos mesmos. Isso lhe permite identificar, dentre os aglomerados da rede, aqueles que estão de acordo com os exemplos do conjunto de treinamento.

O algoritmo de treinamento não cria novas vias de raciocínio. Ele apenas treina as vias existentes na topologia que lhe é apresentada. É por isso que a topologia inicial de trabalho desempenha um papel tão importante para o aprendizado no modelo. Fazendo uma analogia com o paradigma simbolista, a topologia inicial funciona como um conjunto com todas as possíveis asserções que se possa fazer sobre o domínio. Percebe-se então que a escolha da topologia de trabalho assume importância crucial para a eficiência computacional do algoritmo e para a qualidade do aprendizado desempenhado por ele. Quanto maior a topologia, maior o esforço computacional gasto pelo algoritmo; quanto menor, mais pobre o leque de conceitos a treinar e aprender.

Abordamos aqui uma forma de melhorar o resultado da aprendizagem, não através de mudanças nos algoritmos de treinamento e poda, mas sim através de alterações na topologia gerada para ser treinada. Essas alterações decorrem do uso da relevância semântica, uma forma de conhecimento preliminar eliciada de especialistas. A relevância semântica permite eliminar

de antemão diversos aglomerados que representam conceitos sabidamente irrelevantes para a classe e que terminariam por ser descartados num estágio posterior. Assim, podemos partir de uma topologia maior do que aquela que usaríamos normalmente, aplicar sobre ela a poda promovida pela relevância semântica e, só então, submetê-la para treinamento. O uso de uma topologia menor e de melhor qualidade permite que o aprendizado se torne mais eficiente e eficaz.

Neste capítulo, apresentaremos algumas topologias, mostrando sua origem e as vantagens que oferecem em relação às demais. Iniciaremos, na seção 11.2, pela análise de duas topologias previstas pelos autores do modelo. Na seção 11.3, apresentamos a relevância semântica e a topologia criada a partir do seu uso. A seção 11.4 mostra um análise comparativa entre a topologia criada com o uso da relevância semântica e a topologia original. As conclusões são apresentadas na seção 11.5.

11.2 Topologias previstas pelo MNC

11.2.1 Topologia Combinatorial de Ordem Superior (TCOS)

Esta é a topologia originalmente usada pelo MNC. Como vimos, o aprendizado é realizado através do treinamento de uma topologia pré-concebida. A topologia "mais segura" para esse aprendizado seria aquela que contivesse todas as combinações possíveis das evidências do domínio. Assim, nenhuma combinação (ou asserção), por mais esdrúxula que fosse, escaparia ao crivo do algoritmo. Uma topologia com essa característica teria, num domínio com ' n ' evidências (pares atributo/valor), um total de

$$\mathcal{N}C(n,n) = \sum_{i=1}^n C(n,i) = 2^n - 1$$

combinações, que seria infactível para problemas reais, porque o tamanho da topologia cresceria exponencialmente com o aumento do número de evidências. A alternativa adotada para diminuir este problema foi limitar o tamanho das combinações da topologia, definindo-se a Topologia Combinatorial de Ordem Superior (TCOS), que comporta apenas as

combinações com tamanho máximo igual à ordem da topologia. A TCOS apresenta células na camada combinatória, que é igual ao número de caminhos, aglomerados ou combinações da rede.

$$\mathcal{NC}(n, m) = \sum_{i=1}^m C(n, i)$$

Passemos agora à análise de alguns trabalhos anteriores que indiquem que valores 'm' deve assumir: Norusis & Jacques [Norusis 75] trabalharam em um ambiente de classificação bayesiano poderoso e flexível e recomendam que os aglomerados formados tenham ordem igual à metade ou mais do número de atributos para permitir um bom desempenho discriminatório, especialmente quando as classes diagnósticas não são facilmente diferenciáveis.

Machado et al. [Machado 89] sugerem uma ordem bem mais praticável. Baseado: (a) no número mágico de Miller (7 ± 2) [Miller 56], que denota o limite de entidades mentais que os seres humanos podem processar ao mesmo tempo; (b) no trabalho de Leão [Leão 88], onde os grafos de conhecimento eliciados de uma população de cardiologistas apresentam aglomerados com uma média de 2,61 elementos e desvio-padrão de 0,068, não ultrapassando o tamanho máximo de 8; e (c) no tamanho do antecedente de regras de sistemas especialistas (por exemplo, SE de previsão meteorológica, média de 3,94 e desvio-padrão de 0,038); Machado et al. recomendam o uso da ordem entre 5 e 9 (números mágicos de Miller).

Contudo, apesar da diminuição no crescimento da rede devido à limitação imposta à ordem usada, o número de combinações continua a crescer rapidamente com o aumento da quantidade de evidências identificadas pela modelagem do domínio. Veja na TAB. 11.1 que a TCOS ainda assume tamanho muito grande nos casos reais, apesar da limitação no tamanho dos aglomerados. Isto significa que a limitação imposta pela ordem apenas diminui um pouco o crescimento da topologia, permitindo o uso de valores um pouco maiores para n .

TABELA 11.1. A tabela mostra o tamanho da TCOS de ordem 5, 7 e 9, para diversos números de evidências. Nela podemos ver que, mesmo não tendo um crescimento exponencial, a TCOS assume tamanho muito grande nos casos reais.

n	NC(n,5)	NC(n,7)	NC(n,9)
3	$7,0 \cdot 10^00$	$7,0 \cdot 10^00$	$7,0 \cdot 10^00$
5	$3,1 \cdot 10^01$	$3,1 \cdot 10^01$	$3,1 \cdot 10^01$
10	$6,4 \cdot 10^02$	$9,7 \cdot 10^02$	$1,0 \cdot 10^03$
19	$1,7 \cdot 10^04$	$9,4 \cdot 10^04$	$2,6 \cdot 10^05$
44	$1,2 \cdot 10^06$	$4,7 \cdot 10^07$	$9,3 \cdot 10^08$
50	$2,4 \cdot 10^06$	$1,2 \cdot 10^08$	$3,2 \cdot 10^09$
100	$7,9 \cdot 10^07$	$1,7 \cdot 10^{10}$	$2,1 \cdot 10^{12}$
158	$8,0 \cdot 10^08$	$4,5 \cdot 10^{11}$	$1,4 \cdot 10^{14}$
200	$2,6 \cdot 10^09$	$2,4 \cdot 10^{12}$	$1,2 \cdot 10^{15}$

↔ casos reais

11.2.2 Topologia eliciada

O grafo de conhecimento é um grafo E/OU acíclico, com três camadas, a saber: a camada de entrada, onde os nós representam evidências sobre o domínio; a camada de saída, formada por nós OU que representam as classes ou hipóteses do domínio; e a camada intermediária, que fica entre as duas anteriores e pode ter um ou mais níveis. A camada intermediária é formada por nós E ("and") que associam diferentes combinações ou padrões de entrada, representando abstrações que podem ser estruturadas hierarquicamente, em diversos níveis. Os aspectos importantes de um grafo de conhecimento são: (a) sua topologia, identificada pela disposição dos nós nas diversas camadas e pelas ligações entre os mesmos, sendo que os nós da camada inferior representam as evidências importantes para a classe representada pelo nó da camada superior, (b) o grau de importância de cada evidência no aglomerado, de acordo com a ordem de disposição da mesma na camada de entrada e (c) o peso de cada arco [Leão 88]. Veja na FIG. 11.2 um exemplo de grafo de conhecimento.

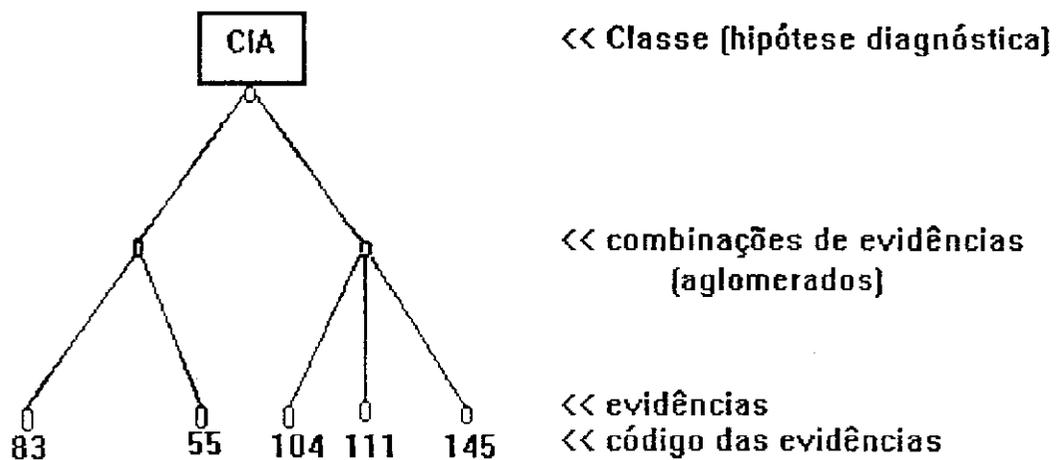


FIGURA 11.2 - Grafo de conhecimento sobre Comunicação Interatrial (CIA), retirado de [Leão 88].

Os grafos de conhecimento eliciados de especialistas podem ser transformados em redes pré-treinadas através do achatamento da camada intermediária para que passe a ter apenas um nível. Dessa forma, ela fica semelhante à TCOS treinada e pode ser manipulada pela versão incremental do algoritmo de aprendizado [Machado 92]. Essa versão recebe como entrada uma rede pré-treinada e um conjunto de treinamento, que é usado para refinar a rede.

Contudo, o uso de topologias construídas a partir de grafos de conhecimento eliciados de especialistas não é muito promissor. Denis argumenta que "sempre existe o risco do conhecimento do especialista ser falho ou incompleto, ou de o mesmo não conseguir expressá-lo adequadamente nos grafos de conhecimento. Nesse caso, o MNC (algoritmo de treinamento) seria incapaz de criar as vias de raciocínio faltantes" [Denis 91].

Como vimos, o grafo de conhecimento possui muito mais informação que aquela expressa em sua topologia. Disto, podemos concluir que a eliciação de um grafo de conhecimento com o único objetivo de usar sua topologia como base para a execução do algoritmo de treinamento é um desperdício do tempo de especialista e um erro de estratégia. O desperdício de tempo ocorre porque o especialista é solicitado a dar informações que não são posteriormente usadas pelo algoritmo. O erro de estratégia ocorre porque, durante a construção de um grafo de conhecimento, o especialista é levado a colocar de forma sucinta a sua maneira particular de pensar. Dessa forma,

O uso da matriz de relevância para podar os caminhos irrelevantes da Topologia Combinatorial de Ordem Superior apresenta dois grandes benefícios. O primeiro deles diz respeito à qualidade do conhecimento, na medida em que este é livrado de caminhos (asserções) sem importância, que equivaleriam às regras inúteis identificadas por Mongiovi & Cirne [Mongiovi 90a]. O segundo benefício diz respeito à eficiência computacional do algoritmo, que é um fator importante, do qual depende a sua utilidade. A poda antecipada de caminhos irrelevantes pode diminuir o "overhead" na geração da topologia e a quantidade de espaço necessária para armazenar a rede em memória secundária. Contudo, a existência desse segundo benefício depende de alguns parâmetros, como o grau de preenchimento da matriz de relevância, o número de classes do domínio e o tipo de implementação que se faça do algoritmo.

Em comparação aos grafos de conhecimento, a matriz de relevância é, decididamente, uma forma mais segura e produtiva de "eliciar uma topologia inicial". Primeiro, a sua construção não requer a eliciação de informação que não seja usada. Segundo, ela permite ao especialista especificar de forma mais sucinta uma topologia maior. Isto ocorre porque o especialista não precisa relacionar todas as combinações interessantes para a topologia. Ele apenas indica quais as evidências são relevantes para cada classe. Cabe ao algoritmo gerar as combinações pertinentes àquelas evidências indicadas. Assim, a inclusão de algumas evidências corresponde à inclusão implícita de todas as combinações (aglomerados) possíveis de serem formadas com aquelas evidências. Essas características tornam a descrição da topologia a ser treinada uma tarefa mais rápida e fácil, permitindo ainda ao especialista incluir na topologia a ser treinada todas as evidências que apresentam alguma relevância, mesmo que parcial.

A diminuição do esforço computacional despendido pelo algoritmo devido à diminuição nas dimensões da topologia a ser treinada, se dá através da adoção da Topologia Combinatorial Relevante de Ordem Superior como base para o treinamento.

11.3.2 A Topologia Combinatorial Relevante de Ordem Superior (TCROS)

Os algoritmos de aprendizado são muito dependentes de uma atividade conhecida como busca. Na verdade, a busca, inerente tanto ao aprendizado indutivo quanto ao dedutivo, torna-se intratável a não ser que seja fortemente restringida ou podada através do uso de tendências ("*bias*"), devido ao enorme tamanho do espaço de possibilidades [MacDonald 89]. O conhecimento preliminar pode ser um forte instrumento na poda do espaço de busca. E, de fato, a relevância semântica cumpre muito bem este papel. Conheceremos, a seguir, a Topologia Combinatorial Relevante de Ordem Superior (TCROS).

A TCROS é formada pelos aglomerados (combinações) relevantes que compõem a Topologia Combinatorial de Ordem Superior. Diz-se que um aglomerado é relevante quando todas as suas evidências (células de entrada, que representam pares atributo-valor) são relevantes. Ou seja, a TCROS pode ser obtida a partir da TCOS através da supressão dos aglomerados não relevantes (i.e., onde pelo menos uma das evidências que o compõem seja formada por um par atributo-valor que não pertença à matriz de relevância eliciada do especialista).

Numa análise inicial, as perspectivas que se abrem com o uso da TCROS são impressionantes: um estudo de caso em ginecologia [Mongiovi 90b] mostrou que das 158 evidências e 18 classes do domínio, a classe com o maior número de evidências relevantes possui 44, enquanto que a de menor número possui 3 evidências relevantes, fazendo uma média de cerca de 19 evidências relevantes por classe, com um desvio-padrão de 10,78. Vê-se que a quantidade de evidências a combinar caiu de forma impressionante, pelo menos nesse estudo de caso, passando de 158 para no máximo 44. Veja na TAB. 11.1 os tamanhos das topologias de ordem 5, para as quantidades de evidências citadas. Como $\mathcal{N}C(n, m)$, que mede o tamanho da TCOS de ordem '*m*' para '*n*' evidências, varia exponencialmente com a variação de '*n*', qualquer acréscimo ou decréscimo no parâmetro '*n*' repercute significativamente no tamanho da topologia.

Contudo, há uma característica da TCROS que se coloca como forte obstáculo ao usufruto desta diminuição da topologia, que é a sua diferenciação por classe (heterogeneidade). Quando o algoritmo de

aprendizado usa a TCOS, ele se beneficia da homogeneidade dessa topologia entre as diversas classes. Durante o aprendizado, as combinações são geradas e testadas contra os exemplos da seguinte forma: para cada combinação gerada, todos os exemplos são testados, à procura daqueles que satisfazem a combinação em questão. Um exemplo satisfaz a uma combinação quando contém todas as evidências que formam a combinação. Na terminologia de redes neurais, isto equivale a dizer que o exemplo consegue propagar-se pelo caminho formado pela combinação em questão.

Assim, quando é encontrado um exemplo que satisfaz a combinação em questão, os pesos daquela combinação correspondentes a todas as classes são alterados da seguinte forma: a classe a qual pertence o exemplo tem o peso daquela combinação de sua topologia aumentado (recompensa) e as demais classes têm aquele peso diminuído (punição). É por isso que dizemos que as redes relativas a todas as classes são geradas "simultaneamente", numa única consulta à topologia padrão, que serve de guia para o aprendizado. Dessa forma, economiza-se o esforço de gerar várias vezes as mesmas combinações e testá-las contra todos os exemplos, para todas as classes, visto que a topologia a ser treinada é idêntica para todas as classes.

O uso da TCROS, contudo, inviabiliza esse "paralelismo". Como cada classe possui uma TCROS diferente, as diferentes topologias devem ser geradas, cada uma para o treinamento de sua classe específica. Cada topologia deve ser submetida a todos os exemplos do conjunto de treinamento para calcular os pesos da rede relativa a uma única classe. Em seguida, outra topologia é gerada para o treinamento da rede relativa a outra classe e assim sucessivamente.

11.4 Análise comparativa entre as topologias TCROS e TCOS

A partir do que foi exposto até aqui, não se torna óbvio que o uso da TCROS proporcione uma diminuição no espaço de busca como um todo, pois, apesar do tamanho da versão da topologia referente a cada classe ser menor ou igual ao da TCOS, a heterogeneidade entre as versões da TCROS obriga que as mesmas sejam geradas separadamente. O espaço de busca total a ser pesquisado pelo algoritmo é dado pela soma das diversas versões de TCROS. Assim, para que o uso da TCROS seja favorável, a diminuição do tamanho de cada versão tem que ser suficiente para que a soma delas seja menor que a TCOS original.

A seguir, faremos uma análise sobre o comportamento do tamanho das topologias, medido em quantidade de aglomerados ou combinações, que é igual ao número de nós da camada intermediária. Isto ocorre porque as topologias em questão são todas "achatadas", i.e., são grafos com apenas três camadas, sendo que a intermediária é a camada combinatória e possui apenas um nível. O esforço computacional despendido pelo algoritmo de treinamento tem correspondência direta com o tamanho da topologia treinada.

O tamanho de cada TCROS é diretamente relacionado ao grau de preenchimento da matriz de relevância. Mais especificamente, o tamanho da TCROS relativa à classe ' c_i ' é $\mathcal{N}C(n_{erc_i}, m)$ células na camada intermediária, onde ' n_{erc_i} ' é o número de evidências relevantes para a classe ' c_i ', ou seja, o número de elementos (representando pares atributo-valor) presentes na coluna ' i ' da matriz de relevância.

Para uma matriz de relevância completa, as diversas versões de TCROS são iguais entre si e iguais à TCOS. Nesse caso, que na prática é irreal, o uso da TCOS é mais conveniente porque a soma dos tamanhos das versões de TCROS dá muito maior que o tamanho da TCOS. À medida que se altera a matriz de relevância, diminuindo-se seu grau de preenchimento, as TCROS vão diminuindo de tamanho e se diferenciando. O ponto em que o uso de TCROS e TCOS apresentam esforço igual para o aprendizado pode ser descrito pela equação:

$$\sum_{i=1}^{n_c} \mathcal{N}CR(i, m) = \mathcal{N}C(n, m) \cdot k \quad (11.1)$$

onde:

' n ' é o número de evidências do domínio;

' m ' é a ordem da topologia (tamanho máximo dos aglomerados);

' n_c ' é o número de classes do domínio;

$\mathcal{N}C(n, m)$ é o tamanho da TCOS de ordem ' m ' para um domínio de ' n ' evidências, dado em quantidade de aglomerados;

$\mathcal{NCR}(i,m)$ é o tamanho da TCROS de ordem ' m ' para a classe ' c_i ', com ' n_{erc_i} ' evidências relevantes. Esse tamanho, em número de aglomerados, é dado por $\mathcal{NCR}(i,m) = \mathcal{NC}(n_{erc_i}, m)$;

' \mathcal{K} ' é uma constante que pode assumir valores entre 0 e ' n_c ', dependendo da implementação.

Quando o primeiro membro da equação 11.1 passar a ser menor que o segundo membro, o uso da TCROS passará a ser preferível ao uso da TCOS, com relação ao esforço computacional despendido. Convém deixar claro aqui que o ponto exato em que essa transição ocorre depende também de características próprias da implementação e do ambiente em que o algoritmo é executado. Isto é expresso pela constante ' \mathcal{K} ', cujo cálculo depende da relação entre o esforço despendido para gerar uma combinação e procurar pelos exemplos que a satisfazem, contra o esforço de gerenciar o ajuste de pesos para todas as classes. Como esse gerenciamento é bastante simples, espera-se que essa constante assumira valores um pouco maiores, porém muito próximos de 1. Há ainda a possibilidade de termos constantes somadas a qualquer dos lados de 11.1. Isto tudo depende de particularidades da implementação e da arquitetura do ambiente em que a mesma é executada.

11.4.1 Um caso real (O Ginecol)

Voltando ao exemplo do Ginecol, se trabalharmos com aglomerados de tamanho máximo 5, temos:

$$\sum_{i=1}^{18} \mathcal{NCR}(i,5) = \mathcal{NC}(3,5) + \dots + \mathcal{NC}(39,5) + \mathcal{NC}(44,5)$$

$$= 3.350.066 \quad (\mathcal{N1})$$

$$\mathcal{NC}(158,5) = 795.404.548 \quad (\mathcal{N2})$$

Comparando o número $\mathcal{N1}$, que representa a soma dos tamanhos das versões de TCROS, criadas de acordo com a matriz de relevância eliciada do especialista, com o número $\mathcal{N2}$, que representa o tamanho da topologia

TCOS que seria treinada para cada classe do domínio, concluímos que o uso da matriz de relevância e da TCROS apresenta resultados melhores com muito menos esforço computacional, pelo menos no caso do Ginecol.

Esta afirmação é corroborada pela TAB. 11.2, que mostra a razão entre o tamanho da TCROS e da TCOS para redes com ordens entre 1 e 7, referentes ao domínio desse estudo de caso. O tamanho considerado para a TCROS é calculado como a soma dos tamanhos das versões referentes a todas as 18 classes levantadas pelo estudo de caso em ginecologia. De acordo com a tabela, a diferença de tamanho entre as topologias cresce à medida que a ordem aumenta.

TABELA 11.2. Mostra a razão entre o tamanho da TCROS e da TCOS para redes com ordens entre 1 e 7, referentes ao domínio do estudo de caso em ginecologia.

Ordem m	tam[TCROS] / tam[TCOS]
1	$2,127 * 10^0$
2	$3,848 * 10^{-1}$
3	$7,925 * 10^{-2}$
4	$1,781 * 10^{-2}$
5	$4,212 * 10^{-3}$
6	$1,026 * 10^{-3}$
7	$2,542 * 10^{-4}$

Até aqui nos detivemos em um aspecto da contribuição proporcionada pelo uso da relevância semântica, que é a diminuição do tamanho da topologia a ser treinada. O segundo aspecto diz respeito à melhoria na qualidade dos resultados, através da eliminação de conceitos semanticamente irrelevantes. A importância da eliminação desses conceitos é apresentada por Cirne & Mongiovi [Cirne 91a].

11.4.2 Generalizando os resultados

O resultado acima diz respeito a um estudo de caso. Mas será que em outras situações a vantagem da TCROS se apresenta sempre e com tanta clareza?

A essa questão não cabe uma resposta absoluta. A vantagem da TCROS depende fundamentalmente de três pontos:

- (a) A variação de $\mathcal{N}(n,m)$ com relação à variação de ' n ', que determina a diminuição do tamanho de cada versão da TCROS com relação à diminuição da quantidade de evidências a combinar;
- (b) A proporção de evidências relevantes com relação ao total de evidências, que depende do grau de preenchimento das colunas da matriz de relevância;
- (c) A quantidade de classes do domínio, que determina o número de versões da TCROS que devem ser somadas para a obtenção da topologia final.

Com relação a essas preocupações, podemos afirmar o seguinte:

- (a) Como pode ser visto na TAB. 11.1, pequenas variações no valor de ' n ' causam grandes variações no tamanho de $\mathcal{N}(n,m)$, principalmente nos casos reais ($n \gg m$);
- (b) A experiência tem mostrado que as matrizes de relevância tendem a ser esparsas. Quanto à disposição da matriz de relevância, temos observado a existência de dois tipos de atributos, que poderíamos chamar de classificadores e particulares. Os classificadores são relevantes para uma grande quantidade de classes, porém com valores diferentes para cada classe ou grupo de classes. Dessa forma, os valores acabam se distribuindo pelas colunas da matriz de relevância. Os atributos particulares são aqueles relevantes para poucas classes. Sua utilidade está em ajudar a qualificar uma classe ou distingui-las em seu grupo. Ambos os tipos de atributos contribuem para diminuir o grau de preenchimento da matriz de relevância.
- (c) O número de classes de cada domínio não é geralmente muito grande, tendendo a ser menor que dez. Se por acaso esse número for ultrapassado, a tendência é dividir o domínio em subdomínios. Devemos lembrar que quanto maior o número de classes, maior deve ser a quantidade de exemplos oferecidos ao algoritmo para que ele possa identificar as características que determinam cada classe.

Assim, apesar de não podermos dar uma resposta absoluta em favor da TCROS, podemos sustentar que seu uso é vantajoso na esmagadora maioria dos casos. Para comprovar essa afirmação, vamos, a seguir, proceder uma análise dos parâmetros que influenciam o tamanho das topologias. Esta análise numérica é baseada no cálculo sistemático do tamanho das duas topologias para um determinado domínio e pequenas variações dos parâmetros que o caracterizam.

11.4.2.1 Identificação e variação dos parâmetros

O tamanho da topologia TCOS é determinado pelos seguintes parâmetros: número de evidências, número de classes e ordem da rede. Para o tamanho da TCROS, além desses parâmetros, temos ainda os relacionados com a relevância semântica, que são a intensidade de relevância da MR e a distribuição da relevância na MR. Para cada um desses parâmetros consideraremos: o tipo dos valores que assumem, a faixa de valores válidos e razoáveis e o valor típico (valor mais comumente assumido pelo parâmetro).

Segundo [Donato 94], as características desses parâmetros são:

Número de evidências: assume valores numéricos inteiros, maiores que 1. A faixa de valores razoáveis está, grosseiramente, entre 30 e 200. Com o objetivo de facilitar a manipulação dos domínios experimentais e o cálculo do tamanho das topologias, consideraremos domínios pequenos, com 40 evidências (corresponderia, por exemplo, a um domínio com 11 atributos e uma média de 3.6 valores/atributo). Este é, portanto, o valor típico deste parâmetro.

Número de classes: assume valores numéricos inteiros, maiores que 1. A faixa de valores razoáveis situa-se entre 2 e 15. Nosso domínio experimental possui 6 classes.

Ordem da rede: assume valores numéricos inteiros, iguais ou maiores que 1, sendo razoáveis aqueles entre 3 e 9. Escolhemos como valor típico para este estudo a ordem 6.

Intensidade de relevância da MR: determinada pelo grau de preenchimento da matriz de relevância. Assume valores numéricos contínuos, entre 0 e 1. O valor deste parâmetro é muito importante para o

tamanho da TCROS, sendo decisivo para a existência ou não de vantagem ao se usar a relevância semântica. Para o estudo ora desenvolvido, consideraremos como típico o valor de 0.25, ou seja 25%.

Distribuição da relevância: reflete a forma como as evidências relevantes se distribuem entre as classes. É tão importante para o tamanho da TCROS como o parâmetro anterior. Se, por exemplo, todas as evidências forem relevantes para uma determinada classe, a subrede referente àquela iguala-se em tamanho à subrede padrão da TCOS. A FIG. 11.3 mostra os três cenários de distribuição de relevância considerados neste trabalho, que partem de uma distribuição uniforme (A) até uma distribuição concentrada (C). Considerando típica a distribuição dispersa (B).

Cenários	Classes					
	C1	C2	C3	C4	C5	C6
A	10	10	10	10	10	10
B	20	16	10	5	5	4
C	35	20	5	0	0	0

FIGURA 11.3 - Distribuição da relevância semântica entre classes.

11. 4. 2. 2 Os Resultados do estudo

Considerando o número relativamente grande de parâmetros envolvidos, neste estudo optamos pela estratégia de fixar os parâmetros em seus valores típicos e variar apenas um deles de cada vez. A apresentação dos estudos relativos aos parâmetros escolhidos será sucinta, baseada nos gráficos que mostram o tamanho das topologias para diferentes valores do parâmetro. Apresentamos apenas os resultados referentes aos parâmetros que exercem maior influência no tamanho das topologias.

Variação da ordem da rede

O GRAF. 11.1 mostra o gráfico para a variação do parâmetro ordem da rede. Os tamanhos das duas topologias apresentam comportamento exponencial, sendo que a TCROS, definida neste trabalho, apresenta uma curva bem mais suave. Na prática, isso viabiliza o uso da TCROS com valores da ordem da rede próximos ao do seu valor típico, definido como sendo 6.

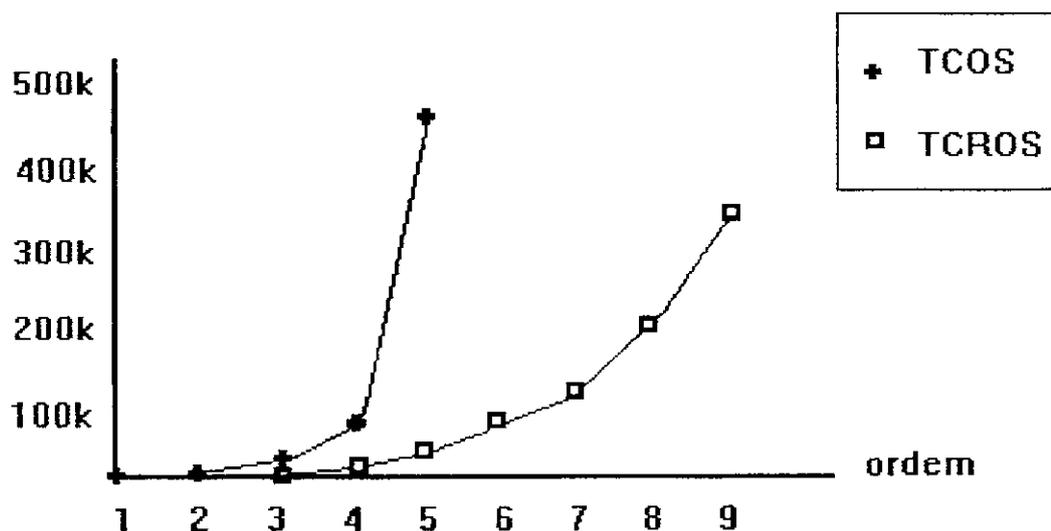


GRÁFICO 11.1 - Variação do parâmetro ordem da rede.

Variação da distribuição da relevância entre classes

Como vimos anteriormente, a relevância semântica provoca a diferenciação das subredes da topologia porque as evidências relevantes são diferentes para cada classe. O GRAF. 11.2 mostra o comportamento do tamanho das topologias em função da distribuição da relevância entre as classes. Os cenários que aparecem nesse gráfico são aqueles definidos na FIG. 11.3. Nessa figura, no cenário A a distribuição é homogênea, no cenário B a distribuição já é heterogênea, mas ainda dispersa, enquanto no cenário C as evidências relevantes estão concentradas em poucas classes.

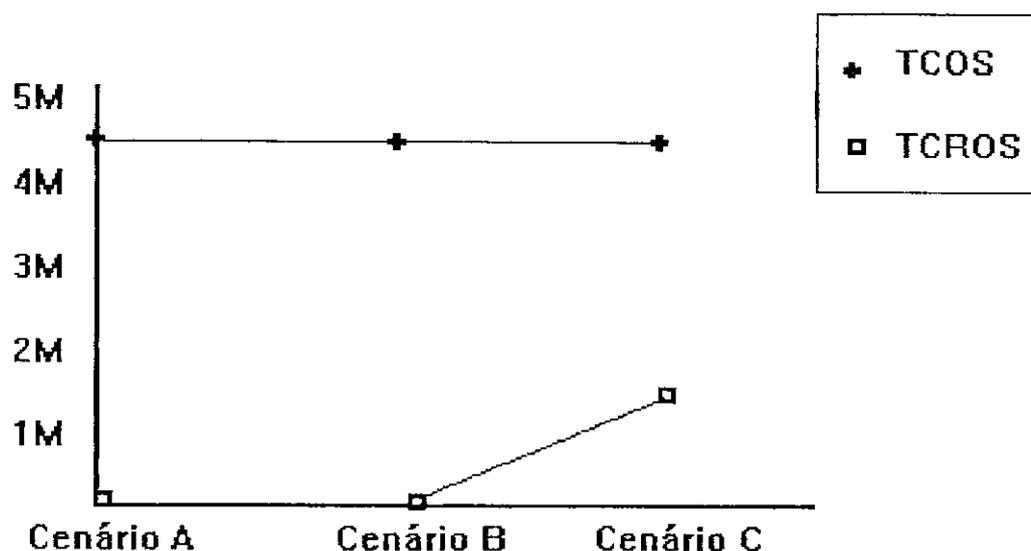


GRÁFICO 11.2 - Variação da distribuição da relevância entre as classes, mantendo a intensidade de relevância no valor típico de 25%.

Como podemos ver no GRAF. 11.2, o tamanho da TCOS, que não considera a relevância semântica, não é afetado pela variação do parâmetro distribuição da relevância. Já a TCROS, embora bem menor em tamanho do que a TCOS, apresenta uma expressiva variação de tamanho, sendo essa variação tanto maior quanto maior for a concentração da relevância em apenas algumas classes.

Variação da intensidade de relevância

Em uma MR, a intensidade de relevância das evidências do domínio é, praticamente, determinada pelo grau de preenchimento da matriz. Nos GRAFs. 11.3-A e 11.3-B, apresentamos o tamanho das topologias para diferentes valores deste parâmetro, considerando duas circunstâncias que diferem quanto ao parâmetro distribuição da relevância. No primeiro gráfico, a distribuição de relevância adotada é a uniforme, o que significa que as evidências relevantes estão distribuídas igualmente por todas as classes do domínio. Já no segundo, adotamos a distribuição de relevância dispersa, definida como sendo típica na FIG. 11.3.

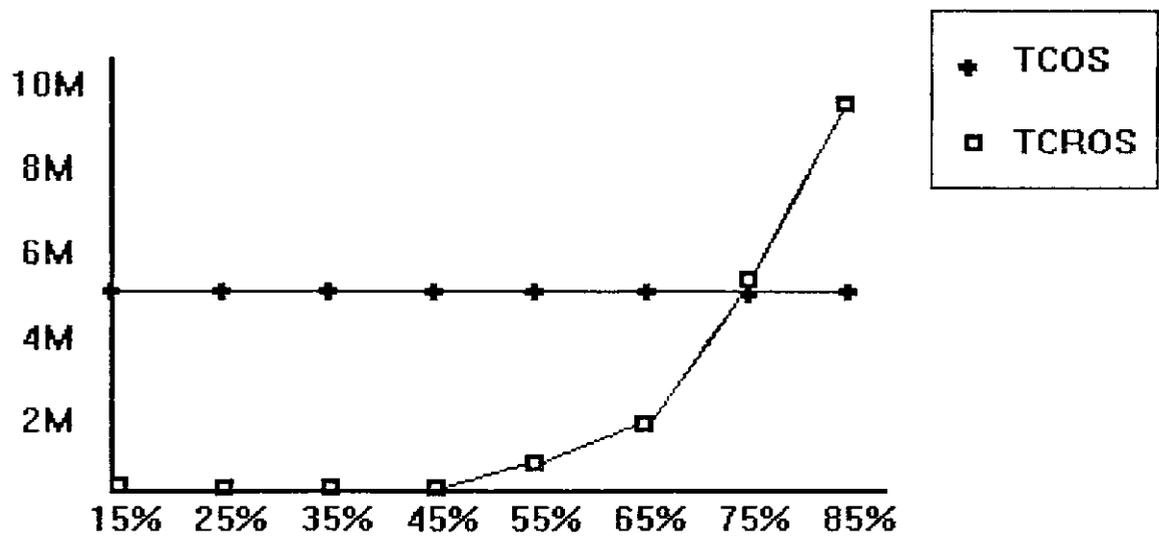


GRÁFICO 11.3-A - Variação da intensidade de relevância, mantendo a distribuição uniforme (proporcional a do cenário A da FIG. 11.3).

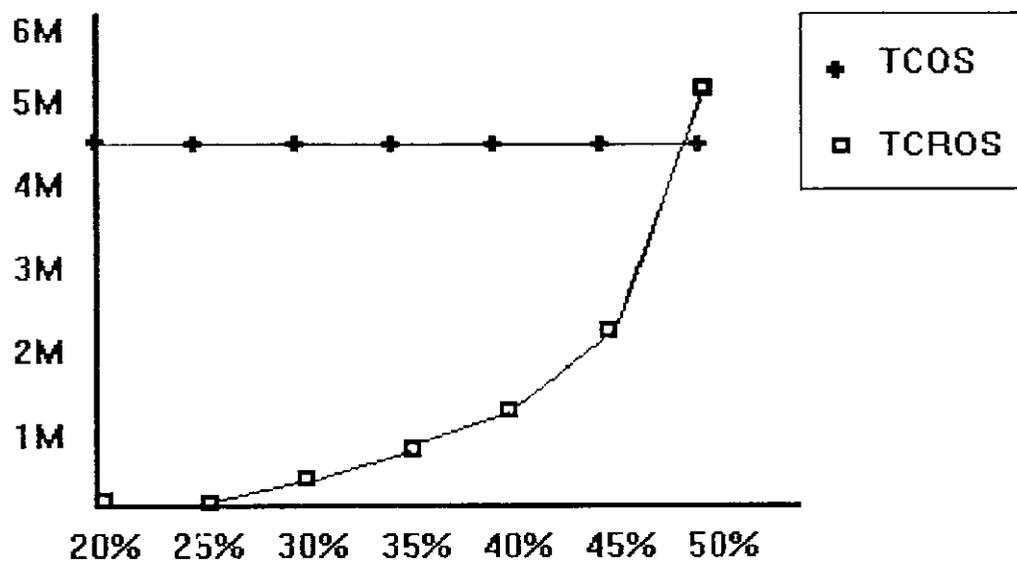


GRÁFICO 11.3-B - Variação da intensidade de relevância mantendo a distribuição dispersa (proporcional a do cenário B da FIG. 11.3).

Conforme mostram os GRAFs. 11.3-A e 11.3-B, a variação do parâmetro intensidade de relevância produz grande reflexo no tamanho da TCROS, fazendo-o comportar-se como uma curva exponencial, que parte de valores muito inferiores ao do tamanho da TCOS, chegando a igualá-lo e até ultrapassá-lo (no valores 75% e 47% para as distribuições uniforme e dispersa respectivamente), sendo esse ponto de ultrapassagem tanto menor quanto maior for a concentração de evidências relevantes entre as classes.

Como vimos, os GRAFs. 11.3-A e 11.3-B indicam que o valor assumido pelo parâmetro intensidade de relevância pode até invalidar o uso da topologia TCROS, dependendo do valor que assume. Entretanto, claro também está que, se esse parâmetro se mantiver próximo ao seu valor típico, o que freqüentemente acontece na prática, o uso da TCROS será vantajoso, visto que, neste caso, essa topologia apresenta tamanho bem inferior ao da TCOS.

Contudo, apesar da importância da intensidade de relevância para o tamanho das topologias, a escolha pelo uso da relevância semântica não pode se basear somente neste parâmetro. Comparando as figuras, percebemos a influência do parâmetro distribuição de relevância, que proporciona grande variação no tamanho da TCROS para um mesmo valor do parâmetro intensidade de relevância, principalmente quando esse valor se situa acima de seu valor típico, o que felizmente, na prática, raramente acontece.

11.5 Conclusão

A topologia a ser usada no Modelo Neural Combinatório exerce grande influência sobre o desempenho do algoritmo de aprendizado e sobre a qualidade dos resultados. Neste trabalho, abordamos formas de melhorar o resultado do aprendizado, não através de mudanças nos algoritmos de treinamento e poda, mas sim através de alterações na topologia originalmente gerada para ser treinada. Essas alterações decorrem do uso de um tipo de conhecimento preliminar: a relevância semântica. As topologias obtidas dessa forma foram comparadas entre si e com a original, levando-se em consideração as melhorias decorrentes na qualidade do aprendizado e no desempenho do algoritmo.

A Topologia Combinatorial Relevante de Ordem Superior proporciona uma melhoria no resultado gerado pelo algoritmo e mantém as vantagens

originais do MNC, como: robustez (admissão de exemplos ambíguos, buracos e atributos multivalorados no conjunto de treinamento), conhecimento expresso numa forma inteligível e verificável diretamente por um especialista humano, e facilidade na modelagem do domínio. Contudo, alguns dos problemas originais são mantidos, como: a não captura de conhecimento hierárquico (estruturado), que se deve ao "achatamento" da camada intermediária das topologias estudadas; e a inexistência de garantia de que o conhecimento gerado classifique todos os exemplos do conjunto de treinamento. Esta última "desvantagem" está diretamente relacionada à possibilidade de aceitar exemplos ambíguos, visto ser impossível que a rede gerada classifique corretamente um exemplo e os exemplos ambíguos a ele correspondentes.

A aprendizagem automática não pode se restringir ao uso de uma única fonte de conhecimento; deve sim lançar mão de toda ajuda que esteja à sua disposição, representada nas mais diversas formas possíveis (exemplos, matrizes de relevância, características dos atributos do domínio, etc). A fidelidade a um paradigma não lhe garante bons resultados, mas o aproveitamento das melhores características de vários paradigmas pode lhe garantir as melhores armas disponíveis para que possa realizar seu trabalho, que é fixar conhecimento de qualidade, da melhor forma possível. As alterações propostas neste trabalho contribuem para a melhoria dos resultados obtidos pelo MNC, e se encaixam perfeitamente às demais extensões que vêm sendo propostas ao modelo.

A TCOS assume que todos os atributos do domínio são multivalorados. Contudo, uma rápida análise de vários domínios modelados nos mostra que, na realidade, a esmagadora maioria dos atributos encontrados são monovalorados. Visando tirar proveito dessa informação, [Donato 94] definiu a TCOSSES: Topologia Combinatorial de Ordem Superior com Exclusividade Seletiva, de ordem m , que é formada por todas as combinações possíveis de pares atributo-valor, com tamanho máximo igual a m , exceto aquelas em que o(s) atributo(s) monovalorado(s) aparece(m) com mais de um valor simultaneamente, garantindo, assim, que os atributos monovalorados assumam valores mutuamente exclusivos

Unindo as contribuições dadas pela TCROS (Relevante) com a TCOSSES (Exclusividade Seletiva), [Donato 94] definiu ainda a TCROSES: Topologia Combinatorial Relevante de Ordem Superior com Exclusividade Seletiva. A TCROSES mostra que essas contribuições podem ser unidas em uma só

topologia, proporcionando um resultado bem melhor que aquele esperado a partir do estudo de suas contribuições individuais.

Por entendermos que uma descrição mais detalhada das topologias TCOSES e TCROSES foge um pouco do nosso objetivo, que é o de mostrar a importância do uso da matriz de relevância para o MNC, neste capítulo, nos limitamos apenas a apresentar as idéias básicas que norteiam essas topologias. Um estudo completo do assunto poderá ser encontrado em [Donato 94].

12 Conclusões e trabalhos futuros

*O importante é não parar de questionar.
A curiosidade tem a sua própria razão de existir.*

Albert Einstein

Neste capítulo apresentaremos nossas conclusões com base na avaliação global dos resultados obtidos na solução dos problemas identificados e, também, relacionaremos algumas sugestões de futuros trabalhos que poderão ser desenvolvidos a partir deste.

12.1 Conclusões

Poderemos considerar o trabalho, apresentado nesta tese, com sendo composto de quatro grandes temas:

- identificação dos problemas relacionados com os métodos indutivos de aquisição de conhecimento a partir de exemplos;
- importância do conhecimento preliminar, em particular a relevância semântica, na aprendizagem indutiva a partir de exemplos;
- concepção e implementação de algoritmos indutivos para contornar os problemas identificados;
- definição e implementação de um ambiente de apoio à aquisição automática de conhecimento.

Desses temas, demos uma ênfase maior ao dos algoritmos indutivos.

No uso dos métodos de aprendizagem indutiva no processo de aquisição de conhecimento a partir de exemplos, identificamos um conjunto de problemas, que se não devidamente solucionados tornam esses métodos de pouca ou nenhuma utilidade prática.

Entre os métodos indutivos mais utilizados estão os da família TDIDT, que geram árvores de decisão, e, em menor escala, os da família AQ, que geram diretamente regras de produção. Apesar do seu sucesso, os métodos TDIDT apresentam os problemas sintático e semântico. Embora os métodos da família AQ apresentem apenas o problema semântico, eles requerem um maior esforço computacional. Além dessas limitações, a maioria desses métodos não contempla o aspecto custo e não procura generalizar soluções.

O problema sintático aparece devido ao fato das árvores de decisão geradas pelos algoritmos TDIDT não serem equivalentes a um conjunto de regras modulares. Isto pode resultar em anomalias, principalmente se houver atributos com um grande número de valores, uma vez que nem todos esses valores serão relevantes na classificação. O problema semântico advém dos algoritmos das duas famílias, TDIDT e AQ, não levarem em consideração, durante o processo de indução, as relações de relevância existentes entre as classes e as condições de um dado domínio. Isto em geral, implica na geração de estruturas despidas de semântica, que resulta no aparecimento de regras sem significado prático, ou seja regras inúteis.

A utilização dos algoritmos indutivos em casos reais e para domínios distintos, tais como as áreas médica, financeira e educacional, chamou a nossa atenção para um outro problema, que é o da modelagem do domínio. Isto é, como a partir do mundo real obter um conjunto de treinamento, e possivelmente um conhecimento preliminar, que espelhe fielmente esse mundo. É um problema complexo e que, praticamente, pouco tem sido abordado.

Além dos problemas sintático, semântico e da modelagem do domínio, a diversidade de algoritmos indutivos existentes nos levou a uma série de questões de cunho prático. Por exemplo, como comparar ou mesmo fundir os resultados de dois ou mais algoritmos? Se o algoritmo não embute algum tipo de simplificação, que técnicas de simplificação utilizar? O algoritmo usa algum tipo de conhecimento preliminar?

O uso do conhecimento preliminar diminui o empirismo da indução e, juntamente com os exemplos do conjunto de treinamento, permite aos algoritmos indutivos gerar bases de conhecimento de melhor qualidade e, o que é mais importante, mais próximas da realidade. Isto aumenta a confiabilidade do usuário nos resultados gerados e permite a redução dos esforços pós-algoritmos.

O conhecimento preliminar pode se apresentar em diversas formas diferentes. Cada uma delas, tratando de aspectos específicos do domínio e podendo auxiliar, a sua maneira, na melhoria da qualidade e/ou da eficiência da aprendizagem. Cabe salientar que esses diversos tipos de conhecimento preliminar podem ser usados de forma isolada ou simultaneamente.

Até o final da década de 80, praticamente, os algoritmos indutivos não usavam conhecimento preliminar, a preocupação era gerar bases de conhecimento de tamanho reduzido e com uma boa acurácia. Entretanto, a partir do início da década de 90, assistimos ao surgimento dos primeiros algoritmos indutivos utilizando algum tipo de conhecimento preliminar, basicamente na forma de custo e generalização, onde, com a primeira forma objetiva-se a geração de bases de conhecimento econômicas e com a segunda visa-se produzir bases compactas e hierarquicamente estruturadas.

Dentro dessa preocupação de gerar bases de conhecimento mais próximas da realidade, visando uma maior satisfação do usuário, propusemos um tipo de conhecimento preliminar, que o denominamos de relevância semântica. Esse conhecimento relaciona a importância que os diversos pares atributo-valor têm na conclusão das classes, possibilitando assim, aos algoritmos que o utilizam gerar bases de conhecimento com um mínimo de regras inúteis e, evidentemente, com um máximo grau de confiabilidade.

Inicialmente representamos a relevância semântica por meio de uma matriz, denominada matriz de relevância, onde as linhas representam os atributos e as colunas as classes. Cada célula que relaciona as linhas com as colunas contém o conjunto de valores do atributo naquela linha que são relevantes para a classificação do elemento existente na coluna. Posteriormente, para possibilitar a fusão de duas ou mais matrizes de relevância e permitir a representação de certos termos linguísticos, tais como "pouco relevante" ou "mais ou menos relevantes", estendemos essa matriz para uma estrutura mais geral que denominamos de matriz de relevância nebulosa. Cada célula dessa nova estrutura é representada por um conjunto nebuloso.

Dentro da nossa concepção de abordagem dos problemas dos métodos indutivos, o conceito de relevância semântica desempenhou um papel fundamental. Embora inicialmente o tenhamos concebido para auxiliar na solução do problema semântico, ele ainda mostrou-se bastante útil em mais três situações, que são: avaliação da qualidade semântica de uma base de conhecimento, apoio ao processo da modelagem do domínio e enriquecimento do Modelo Neural Combinatório.

A inexistência de um processo automatizado que permitisse uma avaliação qualitativa das bases de conhecimento geradas, era uma lacuna existente nos métodos de aprendizagem automática. Essa avaliação era feita, quando possível, com uma forte participação do especialista, o que prejudicava o caráter automático do processo. A definição do conhecimento preliminar relevância semântica, representado na forma de matriz de relevância, nos possibilitou automatizar o processo de avaliação da qualidade semântica de uma base de conhecimento, gerada pelos métodos indutivos. Isso foi feito através da definição do grau de relevância de uma base de conhecimento, que pondera o grau de relevância das regras que a compõem.

A definição de uma matriz de relevância para um dado domínio também mostrou-se útil na modelagem do próprio domínio. Por exemplo, se a intensidade de relevância de uma condição (par atributo-valor) for nula, ou quase nula, significa que o valor desse atributo é irrelevante na classificação, sendo assim um forte indicativo de que esse valor deveria ser suprimido do atributo, ou então deveria se processar um redefinição dos valores desse atributo. Da mesma forma, se a intensidade de relevância de um atributo for nula ou quase nula, significa que esse atributo é irrelevante para o domínio, devendo assim ser eliminado da tabela de exemplos.

Usando a relevância semântica, na forma de matriz de relevância, conseguimos reduzir drasticamente o grande esforço computacional requerido pelo Modelo Neural Combinatório, viabilizando-o, assim, para ser aplicado em domínios reais. Além disso, o uso dessa relevância resultou numa melhoria significativa do nível de qualidade semântica das bases geradas pelo modelo.

Um outro ponto muito importante com relação à matriz de relevância, nebulosa ou não, é que a sua eliciação mostrou-se ser um processo relativamente simples. A obtenção das matrizes de relevância para os

domínios reais apresentados no capítulo 8 e os dois experimentos realizados neste trabalho, o das linguagens de programação no capítulo 4 e o da relevância semântica de uma generalização do apêndice A, serviram para comprovar a facilidade de eliciação da matriz de relevância junto aos especialistas, minimizando o problema da falta de disponibilidade de tempo desses especialistas. Uma vez modelado o domínio, essa eliciação, inclusive, pode ser feita à distância, como aconteceu no caso da matriz de relevância obtida, via FAX, para o domínio "heart-disease-Cleveland" utilizado nos testes dos algoritmos estudados nesta tese.

Para solucionar o problema sintático, foi proposto, por Cendrowska, o algoritmo PRISM. Esse algoritmo não mais utiliza árvores de decisão como saída. O conhecimento por ele induzido é representado por um conjunto de regras de produção. Os conjuntos de regras são superiores às árvores de decisão porque podem representar o conhecimento modularmente, enquanto as árvores de decisão não o podem. Entretanto, o PRISM ainda sofre do problema semântico.

Para a solução do problema semântico, fizemos a proposta de estender os algoritmos TDIDT para possibilitar o uso das informações sobre as relações de relevância do domínio. Estas informações são obtidas do especialista através da matriz de relevância. A primeira solução sugerida, denominada algoritmo ADEX, foi a expansão da árvore de decisão gerada por um algoritmo TDIDT, de forma a incluir condições relevantes nos caminhos que não as tivessem. Devido ao aumento de tamanho alcançado pela árvore expandida, esta solução não foi considerada ideal e, portanto, a abandonamos.

Alternativamente, em relação ao ADEX, propusemos o algoritmo IDRT, que constrói diretamente a árvore de decisão, mesclando informações sintáticas (encontradas através das funções de avaliação de atributos) com semânticas (representadas na matriz de relevância). Com domínios reais, quando comparado com os algoritmos não semânticos, o IDRT obteve índices de qualidade bastante positivos, principalmente nas grandezas relacionadas com o aspecto semântico, onde na maioria das vezes obteve os valores máximos. Embora esta solução tenha sido satisfatória para o problema semântico, ela não resolveu o problema sintático, justamente por continuar usando árvores de decisão como saída.

O algoritmo RPRISM, que induz regras modulares (como o PRISM), usando informações semânticas obtidas da matriz de relevância (como o IDRT), foi

proposto para resolver ambos os problemas simultaneamente. O RPRISM foi concebido para evitar o problema sintático e minimizar os efeitos negativos do problema semântico. Posteriormente, estendemos o RPRISM para o FRPRISM, que nada mais é do que o RPRISM usando uma matriz de relevância nebulosa. O FRPRISM, a exemplo do IDRT, também apresentou excelentes resultados quando comparado com os algoritmos não semânticos, principalmente, mais uma vez, nas grandezas relacionadas com o aspecto semântico. Com uma vantagem, diferentemente do IDRT, o FRPRISM é imune às negatividades do problema sintático.

Finalmente, procurando eliminar o problema sintático, minimizar a ocorrência do problema semântico e, também, possibilitar a utilização de forma conjunta dos principais tipos de conhecimento preliminar disponíveis, desenvolvemos o algoritmo ISREG.

O principal objetivo do algoritmo ISREG é permitir ao usuário uma priorização dos aspectos presentes nos tipos de conhecimento preliminar utilizados. Dessa forma, o usuário pode priorizar ou procurar um equilíbrio entre fatores tais como: tamanho da base de conhecimento gerada, custo e qualidade semântica. Os resultados obtidos com o algoritmo ISREG, comparados com outros algoritmos indutivos, utilizando domínios reais com um número de exemplos significativos, mostram que ele atingiu seu objetivo principal além de, em alguns casos, apresentar resultados superiores àqueles encontrados pelos seus antecessores. Devido à utilização de dois parâmetros básicos que são utilizados para priorizar os fatores envolvidos na indução, o ISREG pode fornecer um relatório que apresenta um resumo dos principais indicadores de avaliação de uma base de conhecimento, esse relatório mostrou-se bastante útil na fase de testes.

Um enfoque mais detalhado em torno do mundo real, com vistas à organização de dados de treinamento a serem submetidos aos algoritmos indutivos a partir de exemplos, mostrou que é necessária uma boa modelagem do domínio, para então se obter resultados eficazes e confiáveis a partir desses algoritmos.

Para contornar esse problema da modelagem do domínio e os demais problemas inerentes às especificidades dos diversos algoritmos generalizadores, propusemos o Ambiente de Apoio à Aquisição de Conhecimento A4. Esse ambiente atenua as dificuldades que envolvem a tarefa de aquisição automática de conhecimento, mais especificamente a

baseada em indução a partir de exemplos. Esta atenuação de dificuldades acontece em função da integração de informações semânticas aos algoritmos generalizadores, suporte a diferentes algoritmos e mecanização de procedimentos de refinamento das entradas e avaliação das saídas.

Apesar do A4 possuir um conjunto de facilidades que nos permitiram implementar e avaliar, com domínios reais, os algoritmos estudados nesta tese, ele ainda é um protótipo de pesquisa e, como tal, sujeito a modificações, principalmente no tratamento das entradas e saídas. Na sua utilização ficou claro a necessidade de dar continuidade nas pesquisas que envolvem a modelagem de um domínio, principalmente no que diz respeito a domínios que apresentam atributos com valores contínuos, que leva à necessidade de uma discretização desses valores.

Por último, como conclusão final, salientamos mais uma vez que a aprendizagem automática, como um método de aquisição de conhecimento, é uma forma eficaz de desenvolvimento de sistemas baseados em conhecimento, particularmente para domínios em que o conhecimento não está bem estruturado. Destacamos também a importância da utilização de múltiplas estratégias de aprendizagem, como, por exemplo, nesta tese onde criamos uma abordagem híbrida entre os métodos automáticos (via os paradigmas indutivo e conexionista) e os cognitivos (via a matriz de relevância). Outro aspecto importante é que, nos algoritmos de aprendizagem, as pesquisas devem caminhar no sentido em que o protocolo, entre o mestre e o aprendiz, seja um diálogo, em oposição à situação de monólogo, praticamente utilizado nos algoritmos dos diversos paradigmas, inclusive nos algoritmos de aprendizagem por indução empírica. No nosso entender, acreditamos que o uso da abordagem híbrida e o diálogo entre agentes racionais deve ser o paradigma predominante da aprendizagem neste restante de milênio.

12.2 Trabalhos futuros

Imre Lakatos [Lakatos 84] afirma que:

"um trabalho de pesquisa começa e termina com problemas".

Portanto, seguindo a afirmação de Lakatos, apesar dos vários resultados positivos alcançados, este trabalho está longe do ponto final das pesquisas sobre os problemas encontrados no processo da aprendizagem automática, em

particular a aprendizagem indutiva a partir de exemplos, que foi o objetivo do nosso estudo. Assim, apresentaremos, a seguir, algumas sugestões de trabalhos futuros, que poderão ser desenvolvidos a partir das observações feitas ao longo desta tese. São elas:

- Propor um modelo para a escolha do paradigma de aprendizagem automática (ou combinação de paradigmas) mais adequado na solução de uma dada família de problemas;
- Estudar a viabilidade do uso da relevância semântica em outras áreas da aprendizagem automática, tais como algoritmos genéticos e ILP (Inductive Logic Programming);
- Realizar um estudo comparativo experimental entre o processo automático de avaliação qualitativa de uma base de conhecimento, proposto nesta tese, e a avaliação subjetiva dos especialistas;
- Verificar, através de uma quantidade maior de testes, se não existe uma faixa de valores dos parâmetros utilizados pelo algoritmo ISREG onde se concentram os possíveis melhores resultados do algoritmo. Diminuindo, dessa forma, a quantidade de execuções do algoritmo realizadas em busca desses melhores resultados;
- Dotar o algoritmo ISREG da capacidade de manipular atributos que possuam valores contínuos, ou então, preferencialmente, possibilitar o ambiente A4 de realizar discretizações precisas e confiáveis. Permitindo, inclusive, a discretização dos elementos de classificação;
- Definir um procedimento para determinar, em função da matriz de relevância, a grandeza limiar de utilidade de uma regra, utilizada para caracterizar uma regra inútil;
- Tornar, a nível de ambiente, a visão da tabela de exemplos como sendo um conjunto de atributos e valores, permitindo que o elemento de classificação seja selecionado entre qualquer um dos atributos. Essa característica foi encontrada em algumas das bases disponibilizadas no ambiente, descritas no capítulo 8;
- Implementar objetos que realizem a transformação das diferentes formas possíveis de representação de conhecimento, utilizadas no A4,

para alguns importantes *shells* do mercado, como, por exemplo, o G2 e o NEXPERT;

- Incorporar ao A4 algoritmos indutivos com características diferentes dos atuais e realizar um estudo comparativo entre esses algoritmos e os algoritmos semânticos propostos nesta tese. Entre esses algoritmos poderíamos incluir: os incrementais IDL e AQ15, o C4.5 (que embute simplificação) e alguns que geram listas de decisão como, por exemplo, CN2 e SDL (que usa processos estocásticos). Evidentemente que, neste caso, novas técnicas de comparação terão que ser desenvolvidas;
- Transformar o gerente do A4 em um sistema baseado em conhecimento, semelhante ao *hipercom* do ambiente integrado de aquisição de conhecimento HOLOS [Oliveira 90]. Entre outras facilidades, para um dado domínio, esse novo gerente poderia auxiliar o usuário no seguinte: escolha do algoritmo indutivo mais adequado, modelagem do domínio (específica para o algoritmo selecionado) e escolha da melhor combinação de fatores de avaliação da base de conhecimento gerada;
- Desenvolver, e incorporar ao A4, técnicas de simplificação de resultados gerados pelos algoritmos indutivos, bem como técnicas de fusão de resultados de diferentes algoritmos generalizadores. Essas técnicas devem ser gerais no sentido que possibilitem contemplar as diversas formas de representação de conhecimento aceitas pelo A4;
- Utilizando domínios reais, comparar o desempenho geral entre a aprendizagem no Modelo Neural Combinatório estendido pelas alterações propostas nesta tese e a dos algoritmos indutivos, fazendo modelagens específicas para cada paradigma de aprendizagem. Na comparação, além das grandezas qualitativas utilizadas neste trabalho, deveríamos ainda considerar a facilidade de modelagem e o aspecto incremental dos paradigmas envolvidos;
- Estudar a viabilidade do uso da generalização de valores de atributos na formação do conjunto de evidências, que compõem a camada de entrada, para as topologias TCROS e TECROSES do Modelo Neural Combinatório estendido;

APÊNDICE A - Relevância semântica de uma generalização: definição e um estudo de caso

O conhecimento preliminar denominado relevância semântica, operacionalizado através da matriz de relevância [Mongioli 90a], possui a relevância de um par "atributo=valor" para a conclusão de cada elemento de classificação. Um outro tipo de conhecimento preliminar denominado generalização, agrupa valores possibilitando a utilização de um conceito mais abrangente na caracterização de uma condição. Por exemplo, o nível de instrução primário, ginásio e científico, pode ser generalizado para nível médio passando a ser o valor genérico que será testado na condição.

É necessário, portanto, definir a forma de cálculo da relevância semântica de uma generalização. Em virtude das distorções apresentadas pela utilização da média aritmética, principalmente quando existe um desvio padrão alto, e aproveitando as observações realizadas quando da definição da fórmula do grau de relevância semântica de uma base de conhecimento, definimos a relevância semântica de uma generalização como sendo:

$$\mathfrak{R}(C_m, E_j) = \frac{\frac{1}{nvg_{ik}} \sum_{k=1}^{nvg_{ik}} F_{ijk}}{1 + \sigma^2} \quad (\text{A.1})$$

onde:

$C_m = (a_i = g_{ik})$ em que a_i é um atributo e g_{ik} uma generalização associada a esse atributo e pertencente ao conjunto $\mathcal{G}(a_i) = \{g_{i1}, g_{i2}, g_{i3}, \dots, g_{in_g}\}$, sendo $0 \leq n_g < n_{v_i}$;

E_j é um elemento de classificação pertencente ao conjunto de elementos de classificação $\mathcal{E} = \{E_1, E_2, E_3, \dots, E_{n_E}\}$;

n_{v_i} é o número de valores do atributo a_i ;

F_{ijk} é uma função que fornece a relevância do atributo a , para o elemento de classificação E_j no valor k ;

σ^2 é a variância da média das relevâncias dos valores que compõem a generalização.

A matriz de relevância possui dados fornecidos por um ou mais especialistas no domínio em estudo, o mesmo ocorrendo com a generalização. Assim sendo, ocorreu-nos a seguinte questão: será que o comportamento da relevância de uma generalização realmente segue a fórmula definida? Ou seja, o especialista preenche a matriz de relevância com a sua crença da importância de cada par "atributo=valor" para a conclusão de cada elemento de classificação, mas se solicitarmos desse especialista a sua crença na generalização ela terá o mesmo valor que encontraremos se utilizarmos a fórmula A.1?

Para verificarmos esse comportamento, realizamos um experimento tomando como domínio uma área do conhecimento de todos, programação de televisão. Idealizamos então dois formulários. Um onde solicitávamos a classificação da importância de alguns atributos para a escolha dos programas de televisão relacionados. Outro onde generalizávamos alguns valores. Selecionamos 12 (doze) pessoas, que passaremos a chamar de especialistas, dividindo-os em dois grupos de 6 com ambiente, cultura e local de trabalho diferentes. Cada especialista preencheu os dois formulários em momentos distintos, com um intervalo de tempo de aproximadamente duas semanas entre eles, com o propósito de evitar qualquer tipo de associação entre as relevâncias informadas para os pares "atributo=valor", presentes no formulário sem generalizações e as relevâncias informadas para os pares "atributo=generalização" presentes no outro formulário.

Escolhemos como atributos a idade, o nível de instrução e o sexo, e como elementos de classificação os programas desenho animado, esporte, novela, noticioso, entrevista e filme. A solicitação para um enquadramento desses programas, colocada no próprio formulário, tinha a intenção de evitar que cada elemento de classificação tivesse claramente um par "atributo=valor" como prioritário para sua classificação.

Um modelo dos formulários utilizados estão apresentados nas figuras FIG. A1 e FIG. A2.

Na sua opinião qual a relevância (importância) dos atributos (características) na preferência de cada um dos programas de televisão relacionados ?						
Atributos	Programas de Televisão					
	Desenho Animado	Esportes	Novela	Noticioso	Entrevista	Filme
Idade Infantil (01-06)	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Infanto-juvenil (07-11)	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Adolescente (12-18)	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Adulto (19-39)	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Sênior (40-59)	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Velho (60-79)	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Ancião (> 80)	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Nível-Instrução Analfabeto	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Primário (1a.Fase 1o.Grau)	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Ginásio (2a.Fase 1o.Grau)	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Científico (2o. Grau)	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Graduado	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Pós-Graduado	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Sexo Masculino	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Feminino	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10

A relevância deve ser atribuída no intervalo [0,10] onde o zero significa irrelevância do atributo e o 10 significa total relevância do atributo.
A matriz deve ser preenchida na sua totalidade, isto é, deve ser informada a relevância de todos os atributos para todas as classes.
Para cada programa devem ser observados os seguintes pontos:
Desenho Animado - considere desenhos tipo Os Flintstones, Ducktales, Pica-Pau, etc.;
desconsidere desenhos tipo Desenhos Bíblicos, Comandos em Ação, etc.
Esportes - considere todas as transmissões esportivas de eventos como futebol, basquete e vôlei;
desconsidere os programas de notícias esportivas.
Novela - observe apenas as novelas das 19 horas, que apresentam um enredo mais leve, tendendo ao humor, sem maior profundidade em temas polêmicos
Noticioso - leve em consideração os noticiosos mais gerais tipo os jornais Nacional, do SBT, da Manchete, etc.;
não considere noticiosos específicos tipo Aqui e Agora, Manchete Esportiva, etc.
Entrevista - enquadre apenas os programas tipo Jô Onze-Meia, Cara-a-Cara, Gente de Expressão, Franzine, etc.;
não analise os demais 'Talk-Shows' tipo Clodovil Abre o Jogo, Hebe Camargo.
Filme - considere apenas os filmes classificados como comédia leve (apenas fugindo ao gênero pastelão), tipo Um Dia a Casa Cai, Juguem Minha Mãe do trem, Corra que Polícia Vem Ai, Apertem Os Cintos! O Piloto Sumiu... etc.;
desconsidere os humoristas mais refinados tipo Wood Allen, Steve Martin e filmes de humor-negro tipo A Família Addams, A Morte Lhe Cai Bem, etc.

FIGURA A.1 - Formulário de avaliação do grau de relevância de uma generalização (valores não generalizados)

Na sua opinião qual a relevância (importância) dos atributos (características) na preferência de cada um dos programas de televisão relacionados ?						
Atributos	Programas de Televisão					
	Desenho Animado	Esportes	Novela	Noticioso	Entrevista	Filme
Idade Infantil (01-11)	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Adolescente (12-18)	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Adulto (19-39)	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Maduro (40-79)	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Ancião (> 80)	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Nível-Instrução Analfabeto	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Médio	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Superior	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Sexo Masculino	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
Feminino	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10

A relevância deve ser atribuída no intervalo [0,10] onde o zero significa irrelevância do atributo e o 10 significa total relevância do atributo.
A matriz deve ser preenchida na sua totalidade, isto é, deve ser informada a relevância de todos os atributos para todas as classes.
Para o atributo Nível-Instrução adotamos a seguinte classificação:
Médio - englobando o primário (1a.Fase do 1o.Grau), ginásio (2a.Fase do 1o.Grau) e científico (2o.Grau).
Superior - abrange os indivíduos graduados e pós-graduados, inclusive doutores.
Para cada programa devem ser observados os seguintes pontos:
Desenho Animado - considere desenhos tipo Os Flintstones, Ducktales, Pica-Pau, etc.;
desconsidere desenhos tipo Desenhos Bíblicos, Comandos em Ação, etc.
Esportes - considere todas as transmissões esportivas de eventos como futebol, basquete e vôlei;
desconsidere os programas de notícias esportivas.
Novela - observe apenas as novelas das 19 horas, que apresentam um enredo mais leve, tendendo ao humor, sem maior profundidade em temas polêmicos
Noticioso - leve em consideração os noticiosos mais gerais tipo os jornais Nacional, do SBT, da Manchete, etc.;
não considere noticiosos específicos tipo Aqui e Agora, Manchete Esportiva, etc.
Entrevista - enquadre apenas os programas tipo Jô Onze-Meia, Cara-a-Cara, Gente de Expressão, Franzine, etc.;
não analise os demais "Talk-Shows" tipo Clodovil Abre o Jogo, Hebe Camargo.
Filme - considere apenas os filmes classificados como comédia leve (apenas fugindo ao gênero pastelão), tipo Um Dia a Casa Cai, Joguem Minha Mãe do Trem, Corra que Polícia Vem Ai, Apertem Os Cintos! O Piloto Sumiu... etc.;
desconsidere os humoristas mais refinados tipo Wood Allen, Steve Martin e filmes de humor-negro tipo A Família Addams, A Morte Lhe Cai Bem, etc.

FIGURA A.2 - Formulário de avaliação do grau de relevância de uma generalização (valores generalizados)

Nosso objetivo principal, como já foi mencionado, era verificar o comportamento do grau de relevância de uma generalização fornecido por um especialista, em comparação com o valor conseguido pela aplicação da fórmula (A.1), que fornece o mesmo grau de relevância a partir da relevância dos valores que compõem a generalização.

Após a aplicação dos formulários, adotamos a seguinte metodologia para análise dos resultados:

- dividimos os valores por 10 para colocá-los dentro do intervalo [0,1] que é o intervalo de trabalho utilizado pelo grau de relevância;
- aplicamos a fórmula (A.1) aos valores do formulário que não apresentava generalizações;
- para cada especialista confrontamos o valor generalizado encontrado pela fórmula com o valor informado no formulário que continha a generalização;
- como a representação gráfica da confrontação dos valores encontrados para cada especialista já mostravam uma grande proximidade, calculamos a média dos valores encontrados pela fórmula (A.1) e a comparamos com a média dos valores informados pelo especialista.

Os resultados obtidos serão mostrados na ordem inversa da metodologia utilizada com o propósito de tornar a leitura gradual quanto ao nível de interesse do leitor. Mostraremos sempre uma tabela e um gráfico para representar os valores encontrados.

Para simplificar a construção das tabelas e dos gráficos, utilizados para mostrar os resultados da pesquisa, definimos algumas convenções que estão apresentadas em TAB. A.1 e TAB. A.2.

TABELA A.1 - Codificação das classes utilizadas no experimento

Elemento de classificação		Elemento de classificação		Elemento de classificação	
Código	Descrição	Código	Descrição	Código	Descrição
C1	Desenho Animado	C3	Novela	C5	Entrevista
C2	Esporte	C4	Noticioso	C6	Filme

TABELA A.2 - Codificação das generalizações utilizadas no experimento

Atributos	Valores informados não generalizados	Generalização pela fórmula	Generalização do especialista
Idade	Infantil (01-06)	G1f	G1e
	Infanto-juvenil (07-11)		
	Senoir (40-59)	G2f	G2e
	Velho (60-79)		
Nível de instrução	Primário	G3f	G3e
	Ginásio		
	Científico		
	Graduado	G4f	G4e
	Pós-graduado		

A codificação da generalização segue um formato GnX onde n é o número da generalização e X pode assumir os valores **f** (fórmula) ou **e** (especialista), sendo que as generalizações com **f** foram obtidas aplicando a equação (A.1) sobre os valores informados pelos especialistas e as generalizações com **e** foram informadas pelos especialistas. O resultado final encontrado na pesquisa está mostrado em TAB. A.3 e GRAF. A.1. Esse resultado representa, para cada classe, a média dos valores das quatro generalizações, valores esses obtidos tanto pela equação A.1 como dos especialistas. Nesse último caso, consideramos uma média dos valores das generalizações fornecidas pelos especialistas.

TABELA A.3 - Resultado final do experimento

Generalização	Elemento de Classificação (Classe)					
	C1	C2	C3	C4	C5	C6
Fórmula	0.46	0.58	0.46	0.60	0.54	0.61
Especialista	0.48	0.63	0.51	0.64	0.56	0.63

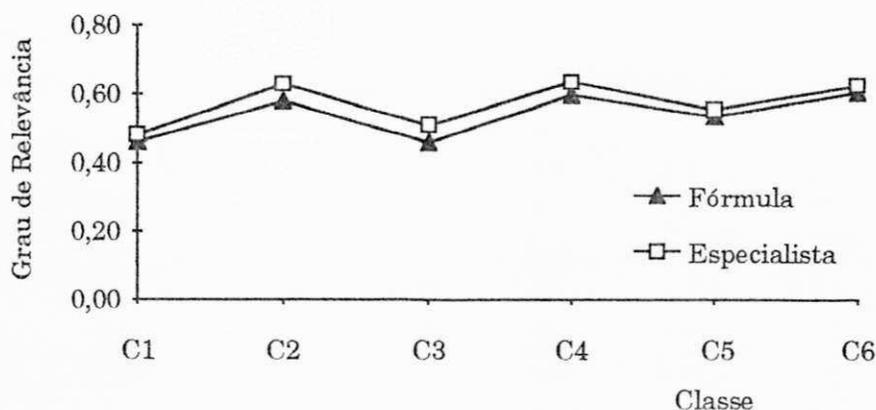


GRÁFICO A.1 - Resultado final do experimento

Os resultados finais do experimento permite-nos concluir que a fórmula (A.1) pode ser utilizada para calcular a relevância de uma generalização tendo em vista as seguintes considerações:

- em termos percentuais, as diferenças encontradas entre as médias finais não são apreciáveis, variam entre 3.1% e 9.3%, com uma média de 5.7%;
- apesar do domínio escolhido ser de conhecimento público, na realidade os indivíduos pesquisados não são especialistas em comunicação ou comportamento social o que torna a pesquisa uma tomada de opinião e não um posicionamento técnico sobre o assunto;
- os formulários foram aplicados em momentos diferentes, por motivos já explicados, os "especialistas" podem mudar de opinião sobre um posicionamento tomado antes uma vez que eles não usaram parâmetros técnicos, ou se basearam em experiências passadas para preencher os formulários;
- numa aplicação real, a diferença observada tende a ser menor, ou nem existir, pois o especialista sempre usará os mesmos parâmetros para informar a relevância de um atributo para um determinado elemento de classificação.

Para aprofundarmos mais a análise dos resultados podemos observar em TAB. A.4 e GRAF. A.2 o comportamento de cada generalização. Aqui foi calculada a média das relevâncias encontradas pela fórmula (A.1) por generalização e a comparamos com a média das relevâncias informadas pelos especialistas também por generalização.

TABELA A.4 - Resultados por generalização.

Generalização	Elemento de Classificação (Classe)					
	C1	C2	C3	C4	C5	C6
G1f	0.87	0.39	0.22	0.13	0.09	0.47
G1e	0.91	0.41	0.23	0.13	0.05	0.45
G2f	0.20	0.61	0.63	0.81	0.75	0.64
G2e	0.17	0.69	0.70	0.88	0.82	0.72
G3f	0.49	0.68	0.58	0.61	0.51	0.61
G3e	0.47	0.69	0.66	0.68	0.55	0.67
G4f	0.28	0.63	0.41	0.85	0.80	0.72
G4e	0.38	0.73	0.44	0.88	0.83	0.69

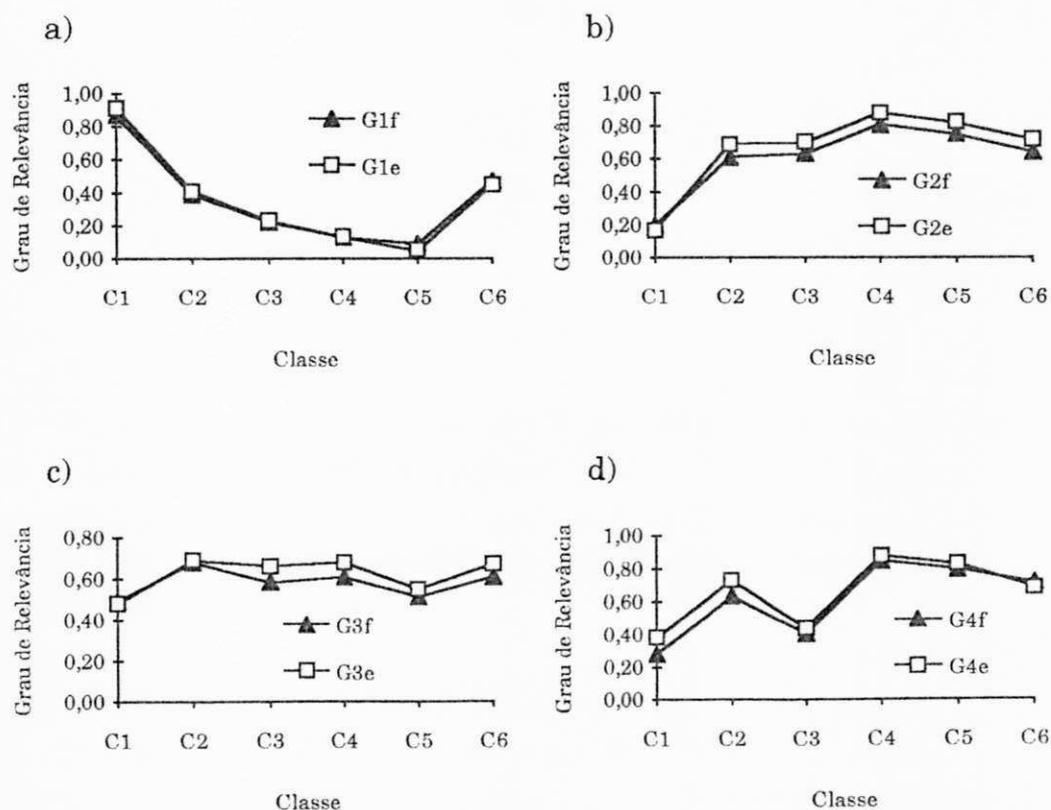


GRÁFICO A.2 - Resultados por generalização.

Os dados de cada especialista, mostrando os valores em uma tabela e em um gráfico para cada generalização analisada, podem ser encontrados em [Alexandre94b].

APÊNDICE B - Documentação dos domínios utilizados nos testes

Apresentaremos a seguir os relatórios produzidos pelo ambiente A4 como documentação dos domínios utilizados para testes neste trabalho.

A4 - Ambiente de Apoio a Aquisição Automática de Conhecimento

Documentação do Domínio - BringSeguro

2 classes
4 atributos
12 exemplos

----- Classes -----

Cod Nome	#Exs.
000 S	6
001 P	6

----- Atributos -----

Cod Nome	Custo
000 Forma	10
001 Cor	30
002 Tamanho	140

----- Valores -----

Cod Nome	Cod.Generalização
000 quadrado	000
001 triangulo	000
002 elipse	001
003 circulo	001
004 pentagono	000
005 vermelho	002
006 azul	002
007 amarelo	002
015 rosa	
008 grande	003
009 medio	003 004
010 pequeno	004

```

003 Material          300  011 metal
                       012 plastico
                       013 couro
                       014 madeira
    
```

Tabela de Generalização = BrinqSeguro.HRQ

```

-----
Cod Nome
-----
    
```

```

000 Poligona
001 Conica
002 Primaria
003 NaoPequeno
004 NaoGrande
    
```

Matriz de Relevância = BrinqSeguro.REL

```

-----
Atributo      Valor      Classes      S      P
-----
Forma         quadrado   0.50 0.75
              triangulo  0.50 0.75
              elipse    0.75 0.50
              circulo   0.75 0.50
              pentagono 0.50 0.75

Cor           vermelho   0.00 0.00
              azul      0.00 0.00
              amarelo  0.00 0.00
              rosa    0.00 0.00

Tamanho       grande    0.75 0.75
              medio    0.75 1.00
              pequeno  0.25 1.00

Material      metal     0.00 1.00
              plastico  0.75 0.25
              couro   1.00 0.00
              madeira  0.00 0.00
    
```

A4 - Ambiente de Apoio a Aquisição Automática de Conhecimento

Documentação do Domínio - Amenorreia

5 classes
6 atributos
91 exemplos

----- Classes -----

Cod Nome	#Exs.
000 GRAVIDEZ	27
001 MENOPAUSA	30
002 DISFUNCAO_HORMONAL	16
003 OUTRAS_PATOLOGIAS	2
004 SAUДАVEL	16

----- Atributos -----

Cod Nome	Custo
000 Idade	0
001 Vomitos_Enjoos	0
002 Sexo_Periodo	0
003 BHCG	2000
004 Perfil_Hormonal	8000
005 Atraso_Menstrual	0

----- Valores -----

Cod Nome	Cod.Generalização
000 jovem	
001 madura	
002 senil	
003 sim	
004 nao	
003 sim	
004 nao	
005 positivo	
006 negativo	
007 e	
008 pe	
009 p	
010 n	
011 x	
012 i20	
013 20a40	
014 s40	

Matriz de Relevância = Amenorreia.REL

Atributo	Valor	Classes	0	1	2	3	4
Idade	jovem		1.00	0.25	0.75	0.50	0.50
	madura		0.25	1.00	0.75	0.75	0.25
	senil		0.00	0.25	0.25	0.50	0.00
Vomitos_Enjoos	sim		0.75	0.25	0.75	0.50	0.00
	nao		0.25	0.25	0.25	0.25	1.00

Sexo_Periodo	sim	1.00	0.00	0.25	0.50	0.00
	nao	0.00	0.00	0.25	0.25	1.00
BHCG	positivo	1.00	0.25	0.25	0.50	0.00
	negativo	0.00	0.25	0.25	0.25	1.00
Perfil_Hormonal	e	1.00	1.00	1.00	0.75	0.50
	pe	1.00	1.00	1.00	0.75	1.00
	p	1.00	1.00	1.00	0.75	0.50
	n	1.00	1.00	1.00	0.75	0.50
	x	1.00	1.00	1.00	0.75	0.50
Atraso_Menstrual	i20	0.75	0.25	0.75	0.50	0.50
	20a40	1.00	0.50	0.75	0.75	0.25
	s40	1.00	1.00	1.00	0.75	0.00

A4 - Ambiente de Apoio a Aquisição Automática de Conhecimento

Documentação do Domínio - HD-Cleveland

5 classes
 13 atributos
 303 exemplos (com contra-exemplos)

----- Classes -----

Cod Nome	#Exs.
000 0	164
001 1	55
002 2	36
003 3	35
004 4	13

----- Atributos -----		----- Valores -----	
Cod Nome	Custo	Cod Nome	Cod.Generalização
000 Idade	0	020 20.00_a_40.00	
		021 40.00_a_50.00	
		022 50.00_a_60.00	
		023 60.00_a_70.00	
		024 70.00_a_80.00	
001 Sexo	0	000 masculino	
		001 feminino	
002 Dor_no_peito	0	002 angina_tipico	
		003 angina_atipico	
		004 sem_dor_anginal	
		005 assintomatico	
003 Pressao_sanguinea	15	025 94.00_a_120.00	
		026 120.00_a_150.00	
		027 150.00_a_175.00	
		028 175.00_a_200.00	

004 Colesterol	12	029 126.00_a_235.00 030 235.00_a_250.00 031 250.00_a_350.00 032 350.00_a_450.00 033 450.00_a_570.00
005 Acucar_no_sangue	10	006 sim 007 nao
006 Eletrocardiograma	15	008 normal 009 curva_ST-T_anormal 010 problema
007 Thalach	0	034 71.00_a_87.00 035 87.00_a_96.00 036 96.00_a_113.00 037 113.00_a_203.00
008 Exerc_induzio_angina	0	006 sim 007 nao
009 Oldpeak	20	038 0.00_a_0.70 039 0.70_a_1.80 040 1.80_a_3.00 041 3.00_a_6.20
010 Exercicio_rampa	0	011 aclave 012 plana 013 declive
011 Numero_max_veias	20	014 0 015 1 016 2 017 3
012 Thal	15	008 normal 018 defeito_permanente 019 defeito_reversivel

Matriz de Relevância = HD-Cleveland.REL

Atributo	Valor	Classes	0	1	2	3	4
Idade	20.00_a_40.00		0.75	0.00	0.00	0.00	0.00
	40.00_a_50.00		0.25	0.00	0.00	0.00	0.00
	50.00_a_60.00		0.25	0.25	0.25	0.50	0.50
	60.00_a_70.00		0.50	0.50	0.50	0.50	0.50
	70.00_a_80.00		0.00	0.50	0.50	0.50	0.75
Sexo	masculino		0.00	0.00	0.00	0.50	0.50
	feminino		0.00	0.00	0.00	0.00	0.00
Dor_no_peito	angina_tipico		0.00	0.25	0.25	0.75	1.00
	angina_atipico		0.00	0.25	0.25	0.50	0.75
	sem_dor_anginal		0.75	0.00	0.00	0.00	0.00
	assintomatico		1.00	0.00	0.00	0.00	0.00

Pressao_sanguinea	94.00_a_120.00	0.00	0.00	0.00	0.00	0.00
	120.00_a_150.00	0.00	0.00	0.00	0.00	0.00
	150.00_a_175.00	0.00	0.00	0.00	0.50	0.50
	175.00_a_200.00	0.00	0.25	0.50	0.50	0.75
Colesterol	126.00_a_235.00	0.50	0.50	0.50	0.50	0.50
	235.00_a_250.00	0.50	0.50	0.50	0.50	0.50
	250.00_a_350.00	0.50	0.50	0.50	0.50	0.50
	350.00_a_450.00	0.50	0.50	0.50	0.50	0.50
	450.00_a_570.00	0.50	0.50	0.50	0.50	0.50
Acucar_no_sangue	sim	0.00	0.00	0.00	0.00	0.00
	nao	0.00	0.00	0.00	0.00	0.00
Eletrocardiograma	normal	1.00	0.00	0.00	0.00	0.00
	curva_ST-T_anormal	0.25	0.50	0.50	0.75	0.75
	problema	0.00	0.75	0.75	1.00	1.00
Thalach	71.00_a_87.00	0.00	0.00	1.00	0.75	0.50
	87.00_a_96.00	0.00	1.00	0.75	0.50	0.50
	96.00_a_113.00	0.75	0.50	0.50	0.75	0.50
	113.00_a_203.00	0.25	0.25	0.50	0.50	0.75
Exerc_induz_angina	sim	0.00	0.50	0.50	0.75	1.00
	nao	1.00	0.00	0.00	0.00	0.00
Oldpeak	0.00_a_0.70	1.00	0.00	0.00	0.00	0.00
	0.70_a_1.80	0.25	1.00	1.00	0.50	0.50
	1.80_a_3.00	0.00	0.50	0.50	0.75	0.75
	3.00_a_6.20	0.00	0.50	0.50	1.00	1.00
Exercicio_rampa	aclive	0.50	0.50	0.50	0.50	0.50
	plana	0.50	0.50	0.50	0.50	0.50
	declive	0.50	0.50	0.50	0.50	0.50
Numero_max_veias	0	1.00	0.50	0.50	0.50	0.50
	1	0.25	0.50	0.50	0.50	0.50
	2	0.50	0.50	0.50	0.75	0.50
	3	0.50	0.50	0.50	0.50	0.50
Thal	normal	1.00	0.50	0.50	0.50	0.50
	defeito_permanente	0.50	0.50	0.50	0.50	0.50
	defeito_reversivel	0.50	0.50	0.50	0.50	0.50

A4 - Ambiente de Apoio a Aquisição Automática de Conhecimento

Documentação do Domínio - Pima-Indians-diabetes

2 classes
8 atributos
768 exemplos (com contra-exemplos)

----- Classes -----

Cod Nome	#Exs.
000 NoDiabetes	500
001 Diabetes	268

----- Atributos -----		----- Valores -----	
Cod Nome	Custo	Cod Nome	Cod.Generalização
000 Num_pregnant	0	000 0.00_a_4.00	
		001 4.00_a_6.00	
		002 6.00_a_14.00	
		003 14.00_a_17.00	
001 Plasma_glucose	6	004 0.00_a_77.00	
		005 77.00_a_137.00	
		006 137.00_a_146.50	
		007 146.50_a_199.00	
002 Diastolic_blood	15	008 24.00_a_71.85	
		009 71.85_a_72.06	
		010 72.06_a_76.27	
		011 76.27_a_122.00	
003 Triceps_skin	15	012 7.00_a_20.00	
		013 20.00_a_30.00	
		014 30.00_a_60.00	
		015 60.00_a_99.00	
004 2Hour_serum_insulin	28	016 14.00_a_85.00	
		017 85.00_a_126.00	
		018 126.00_a_744.00	
		019 744.00_a_846.00	
005 Body_mass_index	15	020 10.00_a_229.00	
		021 229.00_a_350.00	
		022 350.00_a_573.00	
		023 573.00_a_671.00	
006 Diabetes_pedegree	0	024 7.00_a_46.57	
		025 46.57_a_49.08	
		026 49.08_a_60.13	
		027 60.13_a_242.00	
007 Age	0	028 21.00_a_32.00	
		029 32.00_a_40.00	
		030 40.00_a_70.00	
		031 70.00_a_81.00	

Matriz de Relevância = Pima-Indians-diabetes.REI.

Atributo	Valor	Classes	0	1
Num_pregnant	0.00_a_4.00		0.00	0.00
	4.00_a_6.00		0.00	0.00
	6.00_a_14.00		0.00	0.25
	14.00_a_17.00		0.00	0.75
Plasma_glucose	0.00_a_77.00		0.75	0.25
	77.00_a_137.00		0.75	0.25
	137.00_a_146.50		0.25	0.75
	146.50_a_199.00		0.00	1.00

Diastolic_blood	24.00_a_71.85	0.75	0.25
	71.85_a_72.06	0.00	0.25
	72.06_a_76.27	0.00	0.25
	76.27_a_122.00	0.00	0.25
Triceps_skin	7.00_a_20.00	0.50	0.00
	20.00_a_30.00	0.25	0.25
	30.00_a_60.00	0.00	1.00
	60.00_a_99.00	0.00	1.00
2Hour_serum_insulin	14.00_a_85.00	0.75	0.25
	85.00_a_126.00	0.25	0.25
	126.00_a_744.00	0.00	0.00
	744.00_a_846.00	0.00	0.25
Body_mass_index	10.00_a_229.00	0.25	0.00
	229.00_a_350.00	0.50	0.50
	350.00_a_573.00	0.00	0.75
	573.00_a_671.00	0.00	1.00
Diabetes_pedegree	7.00_a_46.57	0.75	0.00
	46.57_a_49.08	0.25	0.25
	49.08_a_60.13	0.25	0.25
	60.13_a_242.00	0.75	0.25
Age	21.00_a_32.00	0.25	0.00
	32.00_a_40.00	0.25	0.00
	40.00_a_70.00	0.00	0.25
	70.00_a_81.00	0.00	0.25

A4 - Ambiente de Apoio a Aquisição Automática de Conhecimento

Documentação do Domínio - Zoo

7 classes
16 atributos
101 exemplos

----- Classes -----

Cod Nome	#Exs.
000 Mamiferos	41
001 Aves	20
002 Repteis	5
003 Peixes	13
004 Anfibios	4
005 Insetos	8
006 Mariscos??	10

----- Atributos -----		----- Valores -----	
Cod Nome	Custo	Cod Nome	Cod.Generalização
000 hair	0	000 no	
		001 yes	
001 feathers	0	000 no	
		001 yes	
002 eggs	0	000 no	
		001 yes	
003 milk	0	000 no	
		001 yes	
004 airborne	0	000 no	
		001 yes	
005 aquatic	0	000 no	
		001 yes	
006 predator	5	000 no	
		001 yes	
007 toothed	5	000 no	
		001 yes	
008 backbone	10	000 no	
		001 yes	
009 breathes	10	000 no	
		001 yes	
010 venomous	5	000 no	
		001 yes	
011 fins	5	000 no	
		001 yes	
012 legs	0	002 0	
		003 2	
		004 4	
		005 5	
		006 6	
		007 8	
013 tail	0	000 no	
		001 yes	
014 domestic	0	000 no	
		001 yes	
015 catsize	0	000 no	
		001 yes	

Abstract

Generally speaking, inductive algorithms for acquiring knowledge from examples have an empirical nature. They don't use background knowledge, just observing the quantitative aspects involved in the generation of a knowledge base. They try to generate knowledge bases, having a minimum size and a maximum accuracy, without paying any attention to the semantic quality of the generated results. In this thesis, while preserving the automatic nature of the acquisition process, we decrease the empirical aspect of the process by proposing a kind of background knowledge, in the semantic relevance form, which is ease to represent and to elicit. In order to improve the understanding of the knowledge base produced, the semantic relevance, either in an isolated form or combined with other types of background knowledge, such cost and generalization, was incorporated to the inductive methods. Besides that, we use the semantic relevance in order to: a) define an automated process for evaluating the semantic quality of knowledge bases generated by inductive methods; b) decrease the heavy process load presented by the Combinatorial Neural Model. In this work, we also propose the A4 (Knowledge Acquisition Environment Support) that helps the inductive knowledge process, modeling the domain into examples and background knowledge and treating the algorithms results.

Referências bibliográficas

- [Abrett 89] Abrett, H. e Burstein, M. H. *The KREME Knowledge Editing Environment*. AAAI Knowledge Acquisition for Knowledge Based System Workshop, Banff/Canadá, novembro/89.
- [Alexandre 94a] Alexandre, C. R., Mongiovi, G. e Bezerra, H. C. F. - *Relevância Semântica de uma Generalização: Definição e um Estudo de Caso*. Anais da XX Conferência Latinoamericana de Informática. México, setembro/1994
- [Alexandre 94b] Alexandre, C. R. - *Aquisição Indutiva de Conhecimento Contemplando os Aspectos Sintático, Semântico, de Generalização e de Custo*. Dissertação de Mestrado, Curso de Pós-graduação em Informática da UFPB, Campina Grande - PB - setembro/94.
- [Alexandre 94c] Alexandre, C. R., Mongiovi, G. e Moura, J. A. B. *Semantic Induction of Generalized, Cost Effective and Modular Rules* - Proceedings of the XI Brazilian Symposium on Artificial Intelligence - Fortaleza - outubro/1994
- [Austin 90] Austin, S. - *An Introduction to Genetic Algorithms* - AI Expert, v. p. 48-53, 1990.
- [Bezerra 94a] Bezerra, H. C. F., Mongiovi, G. e Alexandre, C. R. *Uma Análise Comparativa entre Algoritmos Incrementais e não Incrementais Utilizados para Geração de Árvores de Decisão no Processo de Aquisição de Conhecimento* - Anais do II Congreso y Exposición Internacional de Informática - Mendoza/Argentina - junho/1994;
- [Bezerra 94b] Bezerra, H. C. F., Mongiovi, G. e Silva, H. M. - *Incremental Inductive Algorithms: When Should they be Used ?* - Proceedings of the XI

- Brazilian Symposium on Artificial Intelligence - Fortaleza - outubro/1994
- [Boose 84] Boose, J. H. *Personal Construct Theory and the Transfer of Human Expertise*. Anais do 3o AAAI, 1984.
- [Boose 87] Boose, J. H. *Expertise Transfer and Complex Problems: Usign AQUINAS as a Knowledge. Aquisition Workbench for Knowledge-Based Systems*, II Man-Machine Studies. Academic Press, 1987.
- [Boose 90] Boose, J. *Knowledge Acquisition for Knowledge-Based Systems*. IOS press, Tokyo, 1990.
- [Boucherom 88] Boucherom, S. *Apprentissage et calculus* - Tese de doutorado, Universite Montpellier II, Montpellier, França, dezembro/1988.
- [Boucherom 92] Boucherom, S. *Théorie de l'apprentissage - de l'approche formelle aux enjeux cognitifs* - Editora Hermès, Paris/1992.
- [Boy 86] Boy, G. *An Expert System for Fault Diagnosis in Orbital Refueling Operations*. AIAA 24th Aerospace Science Meeting. Reno, Nevada, 1986.
- [Boy 87] Boy, G.; Faller, B. e Sallantin, J. *Acquisition et Ratification de Connaissances*. Relatório Técnico. CRIM, Montpellier, França, 1987.
- [Breiman 84] Breiman, L.; Friedman, J. H.; Olshen, R. A. e Stone, C. J. *Classification and Regression Trees*. Wadsworth International Group, 1984.
- [Carbonell 89] Carbonell, J. G. *Introduction: Paradigms for Machine Learning*. Artificial Intellegence, vol. 40, 1989.
- [Castiñeira 90] Castiñeira, M. I. e Monard, M. C. *Análise de um Método de Poda para Árvores Indutivas de Decisão*. Anais do 7o Simpósio Brasileiro de Inteligência Artificial. Campina Grande, PB, Brasil, 1990.

- [Cendrowska 88] Cendrowska, J. *PRISM: An Algorithm for Inducing Modular Rules*. Knowledge-Based Systems, vol. 1, Academic Press, 1988.
- [Cirne 90] Cirne Filho, W. C. e Mongiovi, G. *Expansão Semântica para os Métodos de Aquisição de Conhecimento da Família TDIDT*. Anais do 1o Simpósio de Inteligencia Artificial y Robotica. -Luján, Argentina, 1990.
- [Cirne 91a] Cirne Filho, W. C. e Mongiovi, G. *Indução Semântica de Regras Modulares*. Anais do 8o Simpósio Brasileiro de Inteligência Artificial. Brasília, DF, Brasil, 1991.
- [Cirne 91b] Cirne Filho, W. C. e Mongiovi, G. *Aquisição de Conhecimento através de Aprendizado Automático*. Anais do 2o Simpósio de Inteligencia Artificial y Robotica. Luján, Argentina, 1991.
- [Cirne 92] Cirne Filho, W. C. *Uso de Semântica na Melhoria dos Métodos Indutivos de Aquisição de Conhecimento*. Dissertação de Mestrado, Curso de Pós-graduação em Informática da UFPB, Campina Grande - PB - dezembro/1992.
- [Clark 89] Clark, P. e Niblet, T. *The CN2 Induction Algorithm*. Machine Learning, vol 3, Morgan Kaufmann, 1989.
- [Copi 78] Copi, I. M. *Introdução à Lógica*. 2ª. ed. - São Paulo. Editora Mestre Jou, 1978.
- [Davis 82] Davis, R., Shrobe, H., Hamcher, W., Wieckert, K., Shirley, M. e Palit, S. *Diagnosis Based on Description of Structure and Function*. Anais do 1o AAAI, 1982.
- [Denis 91] Denis, F.A.R.M & Machado, R.J. "O Modelo Conexionista Evolutivo". Relatório Técnico, IBM, Centro Científico Rio, CCR-128, Setembro, 1991.
- [Diederich 88] Diederich, J., Ruhmann, I. e May, M. *KRITON: A Knowledge Acquisition Tool for Expert Systems*. Academic Press, vol. 2 - 1988.

- [Dietterich 81] Dietterich, T. G. e Michalski, R. S. - *Inductive Learning of Structural Descriptions: Evaluation Criteria and Comparative Review of Selected Methods*. Artificial Intelligence, v. 16, 1981.
- [Dietterich 86] Dietterich, T. G. e Michalski, R. S. - *Learning to Predict Sequencies*. Machine Learning: An Artificial Intelligence Approach, v. 2 - Michalski, Carbonel & Mitchell (eds) Morgan Kaufmann, 1986.
- [Donato 93] Donato, E. T., Mongiovi, G. e Vasco, J. J. P. *Uso de Semântica para Melhorar o Aprendizado em um Modelo Simbólico-Conexionista - X Simpósio Brasileiro de Inteligência Artificial - Porto Alegre - outubro/93;*
- [Donato 94] Donato , E. T. *Uso de Conhecimento Preliminar na Melhoria do Aprendizado em um Modelo Simbólico-Conexionista - Dissertação de Mestrado, Curso de Pós-graduação em Informática da UFPB, Campina Grande - PB - março/94.*
- [El-Khomi 88] El-khomi, B. M. *Inductive Algorithms and Tools: A Survey*. Relatório Técnico do SYSLAB, Universidade de Estocolmo, Suécia, 1988.
- [Elomaa 89] Elomaa, T. e Holsti, N. *An Experimental Comparison of Induction Decision Trees and Decision Lists in Noisy Domains*. Proceedings of the Fourth European Working Session on Learning 89, Montpellier/França.
- [Eshelman 88] Eshelman, L. *MOLE: A Knowledge Acquisition Tools that Burries Certainty Factors*. Knowledge-Based Systems, vol. 4, Academic Press, 1988.
- [Feigenbaum 81] Feigenbaum, E. A. *Expert Systems in 1980's. The State of the Art Report on Machine Intelligence*. A. Bond (ed). Pergamon-Infotech, 1981.
- [Firebaugh 88] Firebaugh, M. W., *Artificial Imtelligence: A Knowledge-based Approach*. PWS-KENT Publishing Company, Boston, 1988.
- [Ferneda 92] Ferneda, E. *Conception d'un Agent Rationnel et examen de son raisonnement en géométrie - Tese de doutorado, Universite Montpellier II, Montpellier, França, dezembro/1992.*

- [Fisher 91] Fisher, G., Lemke, A. C., Mastaglio, T. e Morch, A. I. *The role of critiquing in cooperative problem solving* - ACM Transation on Information System, v. 9, n. 3, pp 123-151 - abril/1991.
- [Gams 89] Gams, M., *New Measurements Highlight the Importance of Redundant Knowledge*. Proceedings of the Fourth European Working Session on Learning 89, Montpellier/França.
- [Gaines 93] Gaines, B. R. e Shaw, M. L. G. *Eliciting Knowledge and Transferring it Effectively to a Knowledge-based System*. IEEE Transactions on Knowledge and Data Engineering, v.5,n.1,p.4-13, fevereiro/1993.
- [Gallant 88] Gallant, S.I., "*Conexionist Expert Systems*", Communications of the ACM, v.31, n.2, p. 152-168, Feb, 1988.
- [Gascuel 88] Gascuel, O. *Critères pour Élaguer la Recherche Lorsque la Complétude et la Cohérence ne Sont pas Requises*. Relatório Técnico, CRIM, Montpellier, França, 1988.
- [Goldberg 89] Goldberg, D. E. - *Genetic Algorithms in Search, Optimization and Machine Learning* - Reading Masachussets: Addison Wesley Publishing, 1989.
- [Gomes 88] Gomes, F. A. C., Mongiovi, G. e Silva, H. M. *APREND - Um Sistema de Aquisição Automática de Conhecimento a Partir de Exemplos*; 14a Conferência Latinoamericana de Informática. Buenos Aires, Argentina, outubro/1988.
- [Gomes 89] Gomes, F. A. C. *APREND: Um Sistema de Aquisição Automática de Conhecimento a Partir de Exemplos*. Dissertação de mestrado - Curso de Pós-graduação em Informática da UFPB - Campina Grande, Paraíba, abril/1989.
- [Gomes 92] Gomes, F. A. C. *Utilisation d'Algorithmes Stochastiques en Apprentissage* - Tese de Doutorado, Universite Montpellier II, Montpellier, França, dezembro/1992.

- [Gomes 93] Gomes, F. A. C. e Gascuel, O. *Learning Stochastic Decision Lists with Limited Complexity*. X Simpósio Brasileiro de Inteligência Artificial - Porto Alegre - outubro/93;
- [Gruber 89] Gruber, T. *The Acquisition of Strategic Knowledge*. Academic Press, 1989.
- [Hart 89] Hart, A. *Induction and Knowledge Elicitation*. In *Knowledge Acquisition for Expert Systems*. Plenum Press, 1989.
- [Holland 75] Holland, J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.
- [Jackson 86] Jackson, P. *Information to Expert Systems*. Addison- Wesley Publishing, 1986.
- [Kelly 55] Kelly, G. *The Psychology of Personal Constructs*. New York- Norton, 1955.
- [Knguyen 87] Knguyen, T. A., Perkins, W. A., Laffey, T. J. e Pecora, D. *Knowledge Base Verification*. AI Magazine, vol. 8, no 2, 1987.
- [Kietz 93] Kietz, J. U., *A Comparative Study of Structural Most Specific Generalization Used in Machine Learning*. Proceedings of the Third International Workshop on Inductive Logic Programming - Bled/Slovenia, abril/1993.
- [Klir 88] Klir, G. J. e Folger T. A. *Fuzzy Sets, Uncertainty and Information*. Prentice-Hall, 1988.
- [Lakatos 84] Lakatos, I., *Preuves et Refutations: Essai sur la logique de la découverte mathématique*. Editora Hermann, Paris/1984.
- [Lavrac 93] Lavrac, N. e Dzeroski, S. *Tutorial on Inductive Logic Programming*. Proceedings of the Scandinavian Conference on Artificial Intelligence. Estocolmo/Suécia, maio/1993.

- [Lavrac 94] Lavrac, N. *Inductive concept learning using background knowledge*. Proceedings of the XI Brazilian Symposium on Artificial Intelligence - Fortaleza - outubro/1994
- [Leão 88] Leão, B.F., "Construção da Base de Conhecimento de um Sistema Especialista de Apoio ao Diagnóstico de Cardiopatias Congênitas". Tese de Doutorado em Cardiologia da Escola Paulista de Medicina, São Paulo, Brasil, 1988.
- [MacDonald 89] MacDonald, B. A. e Witten, I. H. *A Framework for Knowledge Acquisition Through Techniques of Concept Learning*. IEEE Transactions on Systems, Man and Cybernetics, vol 19, No. 3, May/Jun 1989.
- [Machado 89] Machado, R.J. & Rocha, A.F., "Redes Neurais Combinatórias -- Um Modelo Conexionista para Sistemas Baseados em Conhecimento", Anais do VI Simpósio Brasileiro de Inteligência Artificial, p. 344-358, 1989.
- [Machado 90] Machado, R. J., Duarte, V., Denis, F. M., da Rocha, A. F., Ramos, M. P. e Guilherme, I. R. *Heuristic Learning Expert System*. Anais do 7o Simpósio Brasileiro de Inteligência Artificial. Campina Grande, PB, Brasil, 1990.
- [Machado 92] Machado, R.J., Ferlin, C., Rocha, A.F. & Erthal, G.J., "Incremental Learning in Fuzzy Neural Networks". Proceedings of IPMU'92 Spain, Palma de Mallorca, July, 1992.
- [Marcus 87] Marcus, S. *Taking Backtracking with a Grain of SALT*. Knowledge-Based Systems, vol. 2, Academic Press, 1987.
- [Michalski 83] Michalski, R. *Theory and Methodology of Inductive Learning*. Machine Learning, vol 1, Morgan Kaufmann, 1983.
- [Michalski 86] Michalski, R. S. *Understanding the Nature of Learning: Issues and Research Directions*. Machine Learning an Artificial Intelligence Approach, Vol. II, pp. 348, Morgan Kaufmann Publidshers, Los Altos, CA, 1986.

- [Michalski 87] Michalski, R. S. *Learning Strategies and Automated Knowledge Acquisition: An Overview. Computational Models of Learning*, Springer-Verlag, 1987.
- [Miller 56] Miller, G.A., "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information", *Psychological Review*, 63, p. 81 - 97, 1956.
- [Mingers 89a] Mingers, J. *An Empirical Comparison of Selection Measures for Decision-Tree Induction*. Machine Learning 3. Kluwer Academic Publishers, 1989.
- [Mingers 89b] Mingers, J. *An Empirical Comparison of Pruning Methods for Decision-Tree Induction*. Machine Learning 4. Kluwer Academic Publishers, 1989.
- [Minsky 75] Minsky, M. *A Framework for Representing Knowledge*. *Psychology of Computer Vision*. pp. 211-217, McGraw-Hill, New York. 1975.
- [Minton 90] Minton, S., Carbonell, J. G., Knoblock, C. A., Kuokka, D. R., Etzione, O. e Gil, Y. - *EBL: A Problem Solving Perspective*. *Artificial Intelligence*, special vol. on Machine Learning, Carbonell, J. G. (ed), v.40, n. 1-3, p. 63-118, 1990.
- [Mitchell 86] Mitchell, T. M. *Generalization as search*. *Artificial Intelligence*, v. 18-2, p. 203-226, 1986.
- [Mizoguchi 91] Mizoguchi, R. *Classification Knowledge Acquisition System with an Attribute Elicitation Facility*. Proc. Australian Workshop on Knowledge Acquisition for Knowledge Based Systems. Pokolin, agosto/91.
- [Mongiovi 90a] Mongiovi, G. e Cirne Filho, W. C. *Um Algoritmo Baseado em Conhecimento para Ampliar a Potencialidade do ID3*. *Anais do 7o Simpósio Brasileiro de Inteligência Artificial*. Campina Grande, PB, Brasil, novembro/1990.

- [Mongiovi 90b] Mongiovi, G., ALENCAR, W. & BAENY, X. "*Ginecol: Um Sistema de Diagnóstico Ginecológico*". Relatório Técnico, Universidade Federal da Paraíba, Depto de Sistemas e Computação, 1990.
- [Mongiovi 91] Mongiovi, G. e Cirne Filho, W. C. *O Uso de Semântica na Construção e Expansão de Árvores de Decisão Indutivas*. Anais da XVII Conferencia Latinoamericana de Informática. Caracas, Venezuela, julho/1991.
- [Mongiovi 93a] Mongiovi, G. *Aquisição de Conhecimento a Partir de Exemplos : Uma Abordagem Pragmática* - Tese aprovada no Concurso Público para Professor Titular do DSC/CCT/UFPB - Campina Grande, PB - março/93;
- [Mongiovi 93b] Mongiovi, G. e Vasco, J. J. P. *Relevância Semântica Nebulosa para os Métodos Indutivos de Aquisição de Conhecimento* - X Simpósio Brasileiro de Inteligência Artificial - Porto Alegre - outubro/93;
- [Mongiovi 93c] Mongiovi, G. e Alexandre, C. R. *Qualidade Semântica de Bases de Conhecimento Geradas por Métodos Indutivos* - Third Argentine Symposium on Artificial Intelligence - Buenos Aires - novembro/93;
- [Muggleton 92] Muggleton, S. *Inductive Logic Programming*. London Academic Press, 1992.
- [Nguifo 93] Nguifo, E. M. - *Concevoir une Abstraction a Partir de Ressemblances* - Tese de doutorado, Universite Montpellier II, Montpellier, França, maio/1993.
- [Niblett 87] Niblett, T. *Constructing Decision Trees in Noisy Domains*. Proceedings of the Second European Working Session on Learning , Belgrado/Iugoslávia, 1987.
- [Norusis 75] Norusis, M.J., Jacques, J.A., "*Diagnosis II. Diagnostic Models Based on Attribute Clusters: A Proposal and Comparisons*", Computers and Biomedical Research, 8, p.173, 1975.

- [Nuñez 88] Nuñez, M. *El metodo de aprendizaje EG2: Una aplicacion de conocimiento de bases a exemplos estructurados*. Tese de mestrado, Facultad de Informática, Universidad Politecnica de Madrid, 1988.
- [Nuñez 91] Nuñez, M. *The Use of Background Knowledge in Decision Tree Induction*. *Machine Learning*, 6 pp:231-250, Kluwer Academics Publishers, Boston, 1991.
- [Oliveira 90] Oliveira, J. G. e Mongiovi, G. *Holos: Um Ambiente Integrado de Aquisição Automática de Conhecimento*. Anais do 1o Simpósio de Inteligencia Artificial y Robotica. Luján, Argentina, novemro/1990.
- [Oliveira 93] Oliveira, J. G. e Mongiovi, G. *Hipercom : Módulo Administrador de um Ambiente de Aquisição de Conhecimento com Características de Hipertexto - XIX Conferência Latinoamericana de Informática - Buenos Aires, Argentina - agosto/93*.
- [Oliveira 94] Oliveira, J. G. e Mongiovi, G. - *Uma Abordagem Pragmática Unindo Tecnologias de Inteligência Artificial, Hipermeios, Projeto e Programção Orientada a Objetos, B.D.O.O. e Ferramentas num Ambiente de Aquisição Automática de Conhecimento - Anais do II Congreso y Exposición Internacional de Informática - Mendoza/Argentina - junho/1994*;
- [Pires 93] Pires, F. J. G. M. - *Aprendizagem por Indução Empírica* Tese de doutorado, Universidade Nova de Lisboa, Lisboa, Portugal, setembro/1993.
- [Py 92] Py, M. - *Un Agent Rationnel pour Reasoner en Analogie* - Tese de doutorado, Universite Montpellier II, Montpellier, França, novembro/1992.
- [Quinlan 83] Quinlan, J. R. *Learning Efficient Classification Procedures and Their Application to Chess And Games*. *Machine Learning: An Artificial Intelligence Approach*. Michalski, (eds) - Morgan Kaufman - 1983.
- [Quinlan 86] Quinlan, J. R. *Induction of Decision Tree*. *Machine Learning Journal I*. Kluwer Academic Publisher, 1986.

- [Quinlan 87] Quinlan, J. R. *Generating Production Rules from Decision Trees*. Anais do IJCAI. Milão, Itália, 1987.
- [Quinlan 89] Quinlan, J. R. *Simplifying Decision Trees*. International Journal of Man-Machine Studies 27, 3, 1989.
- [Reitz 92] Reitz, P. *Contribution à l'étude des environnements d'apprentissage*. Tese de doutorado, Université Montpellier II, Montpellier, França, fevereiro/191992.
- [Rich 83] Rich, E. *Artificial Intelligence*. New York: McGraw-Hill, 1983. 553p.
- [Rich 91] Rich, E. e Knight, K. *Artificial Intelligence*. 2.ed. New York: McGraw-Hill Inc., 1991. 621p.
- [Sallantin 86] Sallantin, J. *Logiques et comportements des systèmes rationnelles: une esquisse d'épistémologie*, in: J. L. Moigne (coord.) "Intelligence des Mécanismes, Mécanismes de L'intelligence", Fayard/Fondation Diderot, Paris, 1986.
- [Schlimmer 86] Schlimmer, J. C. e Fisher, D. *A Case Study of Incremental Concept Induction*. Anais do Fifth National Conference on Artificial Intelligence. Morgan Kaufmann, 1986.
- [Shinohara 91] Shinohara, A. e Miyano, S. *Teachability in Computational Learning* - New Generation Computing Journal, v. 8, n. 4, pp 337-347 - 1991.
- [Simon 83] Simon, H. A., "Why should machine learn?". Machine Learning: An Artificial Intelligence Approach, pp. 25-27, Tioga Publishing Company, Palo Alto, CA, 1983.
- [Shapiro 87] Shapiro, A. D. *Structured Induction in Expert Systems*. Addison Wesley, 1987.
- [Shaw 88] Shaw, L. G. e Gaines, B. R. *KITTEN: Knowledge Initiation and Transfer Tools for Experts and Novices*. Knowledge-Based Systems, vol. 2, Academic Press, 1988.

- [Soucek 89] Soucek, B. e Soucek, M. - *Neural and Massive Parallel Computers, the Sixth Generation* - John Willey and Sons, 1989.
- [Utgoff 88] Utgoff, P. E. *ID5: An Incremental ID3*. Proceeding of the Fifth International Conference on Machine Learning. Morgan Kaufmann, 1988.
- [Utgoff 89] Utgoff, P. E. *Improved Training via Incremental Learning*. Proceeding of the Sixth International Workshop on Machine Learning, 1989.
- [Valiant 84] Valiant, L. G. *A Theory of the Learnable*. Communication of the ACM Journal, v. 27, n. 11, pp 1134-1142 - 1984.
- [Van de Vilde 89] Van de Vilde, W. *IDL, or Taming the Multiplexer*. Proceeding of the Fourth European Working Session on Learning. Montpellier, França, 1989.
- [Varejão 91] Varejão, F. M., Machado, R. J. e Milidiu, R. L. - *ALBA: Um Ambiente para Classificadores Bayesianos por Aglomeração*. Relatório Técnico do Rio Científico Center, IBM, Brasil, 1991.
- [Vasco 92a] Vasco, J. J. F.; Mongiovi, G.; Cirne, W. C. e Donato Jr, E. T. *A4 : Ambiente de Apoio à Aquisição Automática de Conhecimento*. Anais do IX Simpósio Brasileiro de Inteligência Artificial, Rio de Janeiro, outubro/92.
- [Vasco 92b] Vasco, J. J. F.; Mongiovi, G.; Cirne, W. C. e Donato Jr, E. T. *Modelagem do Domínio para os Métodos Indutivos de Aquisição de Conhecimento*. Anais do IX Simpósio Brasileiro de Inteligência Artificial, Rio de Janeiro, outubro/92.
- [Vasco 93a] Vasco, J. J. P., Mongiovi, G., Cirne, W. C. e Donato Jr, E. T. - *Modelling the Real World for Knowledge Acquisition Inductive Methods* - Computer Science International Congress - Mendoza/Argentina - junho/93;

- [Vasco 93b] Vasco, J. J. P., Mongiovi, G., Cirne, W. C. e Donato, E. T. - *Um Ambiente para Aquisição Automática de Conhecimento a Partir de Exemplos* - XIX Conferência Latinoamericana de Informática - Buenos Aires, Argentina - agosto/93
- [Vasco 93c] Vasco, J. J. F., Mongiovi, G. e Sucupira, M. F. *Development of a Knowledge Acquisition Environment Following the Object Oriented Paradigm* - Third Argentine Symposium on Artificial Intelligence - Buenos Aires- novembro/93;
- [Vasco 93d] Vasco, J. J. F. - *Um Ambiente de Apoio à Aquisição Automática de Conhecimento* - Dissertação de Mestrado, Curso de Pós-graduação em Informática da UFPB, Campina Grande - PB, junho/94.
- [Warfield 71] Warfield, J. N. - *Societal Systems: Planning, Policy and Complexity*, John Willey and Sons, 1971.
- [Winston 87] Winston, P. H. *Artificial Intelligence*. Eddison-Wesley, 1987.