Cláudio Reginaldo Alexandre

Aquisição indutiva de conhecimento contemplando os aspectos sintático, semântico, de generalização e custo

Campina Grande
1994

Cláudio Reginaldo Alexandre

Aquisição indutiva de conhecimento contemplando os aspectos sintático, semântico, de generalização e custo

Dissertação apresentada ao Curso de Mestrado em Informática da Universidade Federal da Paraíba, como requisito parcial à obtenção do título de Mestre em Informática.

Área de concentração: Inteligência Artificial

Orientador: Prof. Giuseppe Mongiovi
Universidade Federal da Paraíba

Campina Grande
Universidade Federal da Paraíba
1994

A381a Alexandre, Cláudio Reginaldo.
Aquisição indutiva de conhecimento contemplando os aspectos sintático, semântico, de generalização e custo / Cláudio Reginaldo Alexandre. - Campina Grande, 1994.
121 f.

Dissertação (Mestrado em Informática) - Universidade Federal da Paraíba, Centro de Ciências e Tecnologia, 1994.
"Orientação : Prof. M.Sc. Giuseppe Mongiovi".
Referências.

1. Inteligência Artificial. 2. Aquisição de Conhecimento. 3. Circuitos VLSI Standard-Cells. 4. Dissertação - Informática. I. Mongiovi, Giuseppe. II. Universidade Federal da Paraíba - Campina Grande (PB). III. Título

CDU 004.8(043)

Aquisição indutiva de conhecimento contemplando os aspectos sintático, semântico, de generalização e custo

Cláudio Reginaldo Alexandre

Dissertação apresentada e aprovada em 16 / 09 /1994

Giuseppe Mongiovi, M. Sc.
Orientador

Fernando Antonio de Carvalho Gomes, Dr.
Componente da Banca

Bernardo Lula Júnior, Dr.
Componente da Banca

Edilson Ferneda, Dr.
Componente da Banca

A meus pais Carneiro e Lucia,
meu irmão Cleton,
minha esposa Telma, e
meu filho Ilo.

AGRADECIMENTOS

"O que existe de mais interessante nos relacionamentos interpessoais é observar a troca que efetivamente existe neste processo. Poderíamos comparar as pessoas a uma bola de tinta: cada qual com sua cor específica, textura, brilho, tamanho e concentração e que, à medida que 'rolam' pelo caminho que a vida lhes traça, passam pelos caminhos uma das outras e vão deixando seus rastros coloridos. Desta forma, ao olharmos o caminho que já percorremos, veremos quantas marcas de cores diferentes já cruzaram por ele e como essas cores alteraram a nossa cor individual. O mais engraçado é que a natureza, sábia, diz que o branco é a mistura de todas as cores juntas."

Elson A. Teixeira e Andréa Monteiro de Barros

A todos que contribuíram com um pouco de sua cor, brilho e luz para que eu seja quem sou e como sou, meus mais sinceros agradecimentos. Contudo, gostaria de fazer um agradecimento especial a algumas pessoas que contribuiram na realização deste trabalho:

- Ao Prof. Giuseppe Mongiovi, pela competente orientação, disponibilidade e acima de tudo amizade que sempre prevaleceu em nosso relacionamento.
- Aos amigos do DETEC, especialmente Arisio Macena, Fernando Pinto, Fernando Canito, Paulo Jucá e Janete Pereira do Amaral, que além do afeto e amizade me deram um valoroso suporte técnico.
- Aos amigos do DETEC que se transformaram em colegas de Universidade, Francisco Araripe e José Bezerra pelo apoio e companheirismo e ao Haroldo César pela parceria nos artigos e comentários críticos feitos sobre esta dissertação.
- A todos os funcionários da agência do BNB de Campina Grande, em especial à Gonzaga, Sandra Maria, Célia, Luiz Inácio, Eduardo, João Amílcar e Ozias.
- A Lúcia Helena, pela dedicação e apoio.
- A Roberto Pinto, pela amizade, incentivo e troca de experiências.
- A todos os professores e funcionários da COPIN pelos valorosos ensinamentos e apoio, especialmente Hélio Menezes, Bernardo Lula, Edilson Ferneda, José Antão, Pedro S. Nicolleti, Jacques Sauvé e Aninha.
- Aos conterrâneos e amigos José Belo (parceiro de trabalhos) e Washington Luiz pela grande amizade, inestimável apoio e especialmente pelos memoráveis papos.
- A Dra. Sidneuma, enquanto prima pelo apoio e incentivo e enquanto médica pela valorosa contribuição na eliciação das matrizes de relevância.

Faço um agradecimento especial à instituição Banco do Nordeste do Brasil S/A, na figura de sua administração, que através do programa de Pós-Graduação permitiu minha liberação e forneceu todas as condições materiais para a realização deste trabalho.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE GRÁFICOS

LISTA DE CÓDIGOS

SUMÁRIO

RESUMO

A aquisição automática de conhecimento surgiu com o objetivo de resolver os problemas apresentados pelos métodos cognitivos e semi-automáticos, que envolvem desde o relacionamento do engenheiro do conhecimento com o especialista até a indisponibilidade de tempo dos especialista. No entanto, a avaliação da base de conhecimento gerada pelos métodos automáticos é um processo que ainda é feito exclusivamente pelo especialista. Os métodos indutivos de aquisição de conhecimento a partir de exemplos, uma das formas de aquisição automática mais utilizada, podem apresentar dois sérios problemas, um de natureza estrutural (problema sintático) e outro de natureza semântica (problema semântico). Esses métodos também são denominados empíricos enfatizando o fato de não requererem nenhum conhecimento preliminar sobre o domínio. A escolha de uma forma adequada de representação do conhecimento elimina o problema sintático e a utilização de uma das formas de conhecimento preliminar, denominada matriz de relevância, torna rara a ocorrência do problema semântico. As outras formas de conhecimento preliminar disponíveis (custo e generalização) auxiliam os métodos indutivos na busca de uma base de conhecimento de melhor qualidade. No entanto, nenhum método indutivo utiliza conjuntamente as principais formas de conhecimento preliminar disponíveis. Procurando preencher essa lacuna, neste trabalho propomos o algoritmo ISREG (Indutor Semântico de Regras modulares Econômicas e Generalizadas) que além disso resolve o problema sintático e minimiza a ocorrência do problema semântico. Propomos, também, um processo automático de avaliação da qualidade semântica das bases de conhecimento geradas pelos métodos indutivos.

1 Introdução

Artificial Intelligence is the science of making do things that would require intelligence if done by men.

Marvin Minsky

1.1 Inteligência Artificial

O reconhecimento inconteste da disciplina Inteligência Artificial (IA) como uma ciência vem sendo conquistado gradual e arduamente. Um dos motivos para esse lento reconhecimento pode residir nas incertezas que permeiam a área e tem aproximado mais a IA das ciências sociais ou da psicologia do que das ciências exatas como a física ou matemática.

Essa falta de definição leva a que cada grupo de pesquisadores faça uma definição própria da área, provocando a criação de tantas escolas quantos forem os grupos existentes. Isso tem gerado o ambiente propício para fortes confrontos e controvérsias dentro da disciplina.

Para [Firebaugh 89] uma forma de classificar os pesquisadores de IA é através do teste dos "meios vs. fins". Se o "fim" das pesquisas em IA é definido como o principal objetivo do estudo e o "meio" é definido como uma ferramenta básica ou modelo usado para auxiliar o estudo, então os pesquisadores de IA, geralmente, estão enquadrados em uma das duas categorias:

- **Mente como objeto e máquina como ferramenta.** Essa escola de pensamento inclui muitos psicólogos cognitivos e lingüísticos que visam entender o comportamento da mente humana como objetivo e os computadores como simples ferramentas para testar os modelos da mente.

- **Computador como objeto e mente como modelo.** Essa escola busca a criação de máquinas mais inteligentes como o objetivo da IA e o comportamento da mente humana como um modelo para auxiliar a simulação da inteligência.

Porém, Firebaugh reconhece que é impreciso classificar todos os pesquisadores como pertencentes claramente a uma das duas escolas. Muitos pesquisadores desenvolvem trabalhos que utilizam princípios das duas escolas.

Na realidade, a IA deve ser vista como uma ciência interdisciplinar que interage com áreas da psicologia, lingüística, filosofia, matemática, física, engenharia e ciências da computação. Contudo, é inegável que a robótica e a engenharia do conhecimento, duas especialidades da IA ligadas à segunda escola, têm apresentado trabalhos mais pragmáticos e com relativo sucesso.

1.2 Engenharia do conhecimento e Sistemas baseados em conhecimento

O termo Engenharia do Conhecimento, introduzido por Donald Michie em 1972, é utilizado para definir o processo de extração de conhecimento de um especialista humano e incorpora-lo ao computador. O indivíduo que executa essa atividade é chamado de Engenheiro do Conhecimento e os sistemas construídos por esses engenheiros são denominados Sistemas Baseados em Conhecimento (SBCs) [Michie 85].

Os SBCs são sistemas que fornecem meios para o manuseio e aplicação de conhecimento. A principal característica desses sistemas é a existência de uma separação explícita entre o conhecimento que possuem e as suas estratégias de controle. Dependendo do nível de habilidade dos SBCs eles podem ser conhecidos como assistentes inteligentes, tutores inteligentes ou sistemas especialistas.

1.3 Sistemas especialistas

Os sistemas especialistas procuram simular o comportamento do especialista humano na resolução de um problema em um determinado domínio. Eles possuem algumas características próprias, como uma grande concentração de sabedoria sobre uma estreita faixa do conhecimento humano específico de um domínio, possuem a capacidade de justificar seus conselhos, análises ou conclusões e se forem dotados de dados probabilísticos ou nebulosos podem lidar com incertezas [Rich 91]. A FIG. 1.1 mostra uma estrutura básica de um sistema especialista.

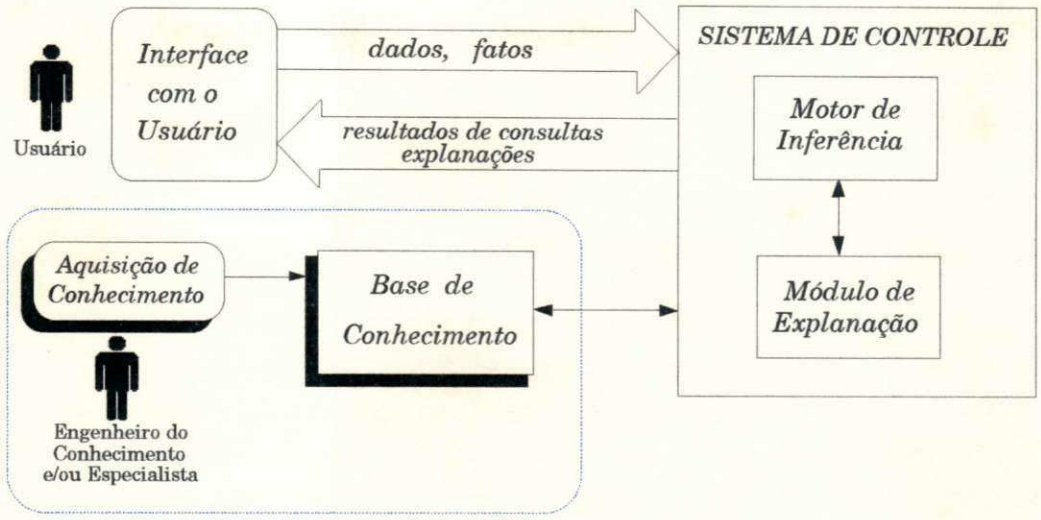

FIGURA 1.1 - Estrutura básica de um sistema especialista.

A criação da base de conhecimento se dá através da tarefa de aquisição de conhecimento que vem a ser a transformação de um conhecimento numa representação adequada para o computador [Kidd 87]. Várias classificações dos métodos de aquisição de conhecimento já foram sugeridas, uma dessas sugestões analisa a questão do ponto de vista do esforço empreendido pelo aprendiz na aquisição do conhecimento [Carbonell, Michalski and Mitchell 83]. Essa classificação estabelece que o aprendizado pode ser realizado por:

- **Um processo mecânico e implantação direta do novo conhecimento** - o aprendiz não realiza nenhuma inferência ou transformação sobre o conhecimento. O conhecimento é construído e modificado por uma entidade externa ou os dados e fatos adquiridos são simplesmente armazenados sem nenhum tipo de inferência.

- **Instrução** - o conhecimento é adquirido de um professor ou de uma fonte organizada de informação, livros por exemplo. Requer que o aprendiz transforme o conhecimento da linguagem externa para uma forma de representação interna, e que incorpore as novas informações ao conhecimento existente para uso futuro.

- **Analogia** - requer novos fatos ou um conhecimento histórico para transformar e argumentar com o conhecimento existente, o que pode levar a formulação de novos conceitos ou a aquisição de novas experiências que permitam lidar com a nova situação. Essa forma de aprendizagem requer mais inferência, por parte do aprendiz, do que as duas formas anteriores.

- **Exemplos** - dado um conjunto de exemplos e contra-exemplos de um conceito, o aprendiz induz a descrição de um conceito geral que descreve todos os exemplos positivos e nenhum dos contra-exemplos. A quantidade de inferência realizada pelo aprendiz é muito maior do que no aprendizado por analogia.

- **Observação e descoberta** - essa é a forma mais ampla de aprendizagem indutiva inclui descoberta de sistemas, formação de teorias, definição de critérios de classificação e outras tarefas similares sem o auxílio de um professor. Essa forma de aprendizagem não supervisionada requer que o aprendiz execute mais inferências do que qualquer uma outra forma apresentada até aqui.

Qualquer que seja o método escolhido, a aquisição de conhecimento foi definida por [Feigenbaum 81] como sendo o ponto de estrangulamento do processo de construção de um sistema especialista, essa definição ficou conhecida como o "gargalo" de Feigenbaum[1] (FIG. 1.2). Estruturar o domínio, identificar e formalizar os conceitos, ajustar o vocabulário utilizado, dentre outros aspectos, são dificuldades que o engenheiro do conhecimento encontra na tarefa de aquisição de conhecimento.

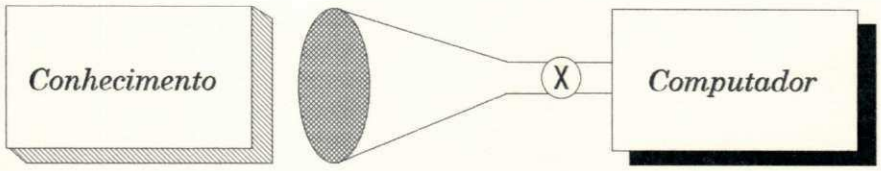

FIGURA 1.2 - O "gargalo" de Feigenbaum.

Além dos fatores já citados que dificultam o processo de aprendizado, dois outros fatores têm motivado a busca de novas soluções: a indisponibilidade de tempo e o alto custo da hora de trabalho de um especialista humano. Nos processos de aprendizagem que envolvem entrevistas, observações, experimentos, etc., a presença do especialista é indispensável e pode consumir muito tempo e dinheiro até que se consiga um resultado satisfatório. O aprendizado por máquina ou automático[2] [Michalski 86] surgiu como uma tentativa de minimizar, ou até eliminar, os problemas enfrentados pelas formas de aquisição de conhecimento utilizadas até aquele instante.

1.4 Aprendizado automático

Mais uma vez uma grande variedade de definições e classificações dificultam o entendimento do que seja aprendizado automático e deixam seus objetivos confusos. Uma tentativa de agrupar as diversas visões das diferentes correntes científicas foi feita por [Carbonell 90] ao dividir o aprendizado automático em quatro paradigmas:

- Conexionista - modelos baseados em redes neurais artificiais.
- Genético - sistemas classificadores.
- Analítico - baseado em explanações e algumas formas de analogias.

---

1 *The Feigenbaum bottleneck*
2 *Machine Learning*

- Indutivo - adquire conceitos a partir de conjuntos de exemplos e contra-exemplos.

Desses paradigmas o mais amplamente estudado e aplicado a sistemas baseados em conhecimento é o indutivo. Nele procura-se inferir uma descrição geral de um certo conceito a partir de um conjunto de exemplos e contra-exemplos extraídos do especialista, do mundo real, ou de uma base de conhecimento apreendida pelo próprio sistema. Se a fonte dos exemplos é o mundo real, o aprendiz pode ter a opção de realizar experimentos dos quais ele recebe um retorno e reformula sua descrição geral [Firebaugh 89].

1.5 Aprendizado indutivo a partir de exemplos

O aprendizado indutivo a partir de exemplos requer a existência de um algoritmo que através da análise das instâncias, casos reais coletados pelo engenheiro do conhecimento, tentará induzir as regras que constituirão a base de conhecimento. A FIG. 1.3 mostra uma representação do aprendizado indutivo a partir de exemplos.

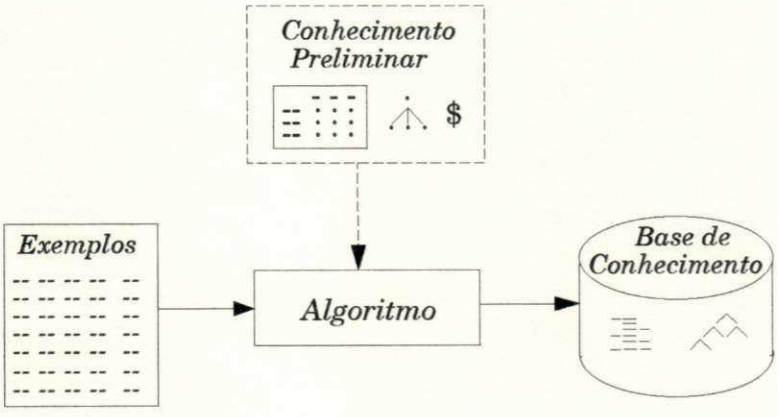

FIGURA 1.3 - Aprendizado indutivo a partir de exemplos.

Os algoritmos indutivos diferem entre si na maneira como realizam a indução, na forma de representar a base de conhecimento e na utilização ou não de um conhecimento adicional, chamado preliminar por ser eliciado a priori, que auxilia na geração da base de conhecimento.

Podemos relacionar como vantagens dos métodos indutivos a partir de exemplos, os seguintes pontos:

- Minimizam ou até tornam desnecessária a presença do especialista no processo de aquisição de conhecimento.

- Podem captar conhecimento através de generalizações, atividade natural do processo de aprendizado humano.

- Produzem bases de conhecimento coerentes, *i.e.*, bases livres de inconsistências, circularidades, etc.

Apesar dessas vantagens, de uma forma geral, os algoritmos indutivos apresentam pontos negativos nos seguintes casos:

C1. Os que geram árvores de decisão fornecem um conhecimento pouco inteligível, pois essas árvores, normalmente, provocam a introdução de condições artificiais nas conclusões [Cendrowska 88] [Gaines 93].

C2. A maioria utiliza somente as informações estruturais contidas nos exemplos, desconhecendo uma possível relação entre as características desses exemplos e a importância que elas podem ter para a conclusão ou diagnóstico. Esse desconhecimento pode fazer com que a ocorrência de casos raros sejam generalizados como conhecimento [Mongiovi 90] [Cirne Filho 92].

C3. Poucos algoritmos utilizam conhecimento preliminar e nenhum utiliza as principais formas disponíveis desse conhecimento de forma conjunta. Isso faz com que a seleção do algoritmo se faça em função dos benefícios que a utilização de um determinado tipo de conhecimento preliminar possa trazer, deixando a base gerada sujeita aos problemas provenientes da utilização desse algoritmo.

A maioria desses pontos negativos apresentados foram descobertos através da análise dos resultados apresentados pelo algoritmo ID3 [Quinlan 83], por ter sido um dos primeiros métodos propostos. Alguns algoritmos descritos a seguir procuram eliminar esses pontos negativos apresentados. Para o caso **C1** foi proposta uma solução através do algoritmo PRISM [Cendrowska 88], já para o caso **C2** existem as soluções propostas pelos algoritmos IDRT [Mongiovi 91] e RPRISM [Cirne Filho 91], sendo que este último algoritmo procura solucionar também o caso **C1**. O algoritmo EG2 [Nuñez 91] utiliza duas formas de conhecimento preliminar mas está sujeito aos problemas **C1** e **C2**. No entanto, não existe nenhum algoritmo que cubra, ao mesmo tempo, o caso **C3** e que também procure evitar a ocorrência dos casos **C1** e **C2**.

A escolha de um algoritmo indutivo para gerar uma base de conhecimento, pode ser feita com base na análise da forma de execução da indução, no tipo de conhecimento preliminar utilizado ou na forma de representação do conhecimento, entre outros fatores. Contudo, isso não garante que a melhor base de conhecimento será gerada. E como saber qual a melhor base gerada ? É necessário então que existam formas de avaliar a base de conhecimento gerada.

1.6 Avaliação de uma base de conhecimento

Apesar de todo o sucesso dos métodos de aprendizado automático, conseguindo diminuir drasticamente, e por vezes eliminando, a participação do especialista na elaboração de um sistema especialista. Uma tarefa, no entanto, ainda continua sendo desempenhada exclusivamente pelo especialista: a avaliação da base de conhecimento gerada.

Essa dependência do especialista humano, traz de volta todos os inconvenientes que motivaram o surgimento do aprendizado automático, uma vez que somente ele, analisando cada regra gerada, pode determinar a sua importância ou aplicabilidade no domínio modelado. Essa avaliação pode ser feita sob os aspectos quantitativo (tamanho da base, comprimento das regras e acurácia), de custo e qualitativo (compreensibilidade e aplicabilidade das regras).

Supondo que o engenheiro do conhecimento possua a oportunidade de executar vários algoritmos indutivos gerando, conseqüentemente, várias bases de conhecimento. Qual delas ele deve apresentar para o especialista avaliar ? E no caso dessas bases serem grandes, pode ficar inviável avaliar mais de uma e selecionar as melhores regras.

Uma solução para esse problema seria automatizar um processo de pré-avaliação, tornando possível selecionar uma base de conhecimento sob o aspecto desejado, ficando para o especialista, quando necessário, apenas a tarefa de validação da base escolhida.

A automatização do primeiro aspecto é relativamente simples, uma vez que ele envolve apenas dados provenientes de uma análise quantitativa da base gerada. Uma forma de avaliação sob o aspecto custo foi definida por [Nuñez 91], não apresentando dificuldades para sua automatização. Para o aspecto qualitativo (semântico) não dispomos de nenhuma grandeza definida que possibilite sua automatização, contudo, através da utilização de um conhecimento preliminar que determine a relevância semântica de cada característica do domínio para todos os seus elementos de classificação, seja possível definir e automatizar essa grandeza.

1.7 Objetivos da dissertação

Este trabalho atua no âmbito dos métodos indutivos, procurando contornar alguns dos seus pontos negativos, citados na seção 1.5, e buscando uma solução para o problema da avaliação semântica de uma base de conhecimento (seção 1.6). Para atingir esses objetivos este trabalho apresenta duas finalidades básicas:

- Definir uma grandeza que permita avaliar, de forma automatizada, a qualidade semântica de uma base de conhecimento, *i.e.*, determinar o grau de aplicabilidade e clareza das regras geradas.

- Propor um algoritmo que utilize, de forma conjunta, as principais formas de conhecimento preliminar, e procure minimizar a ocorrência dos problemas estruturais e de natureza semântica apresentados pelos algoritmos indutivos citados na seção 1.5.

Esperamos dessa forma contribuir para preencher as lacunas identificadas no processo de aquisição indutiva de conhecimento. Este trabalho se enquadra numa linha de pesquisa que busca operacionalizar um ambiente que auxilie todo o processo de aquisição de conhecimento. No estágio atual esse ambiente, denominado A4 (Ambiente de Apoio a Aquisição Automática de Conhecimento)[Vasco 93], encontra-se parcialmente implementado e foi no seu contexto que as propostas desta dissertação foram implementadas e os experimentos foram realizados.

1.8 Organização dos capítulos

Para cumprir os objetivos definidos anteriormente, organizamos este trabalho em sete capítulos, incluindo esta introdução.

Capítulo 2 - Utilização do conhecimento preliminar na aprendizagem indutiva, onde abordaremos o método indutivo de aprendizado e apresentaremos algumas formas de conhecimento preliminar utilizadas por esses métodos. Introduziremos, também, um domínio base que será referenciado ao longo do trabalho.

Capítulo 3 - Análise crítica de alguns algoritmos indutivos, onde os algoritmos indutivos mais representativos serão analisados e comparados, para isso selecionamos aqueles que utilizam alguma forma de conhecimento preliminar além do ID3 que serve de referência inicial.

Capítulo 4 - Qualidade de uma base de conhecimento, no qual apresentaremos um processo automático de avaliação da qualidade semântica de uma base de conhecimento.

Capítulo 5 - Indução semântica de regras modulares, econômicas e generalizadas, onde propomos um algoritmo, o ISREG, que contempla as principais formas de conhecimento preliminar e procura minimizar a possibilidade de ocorrência dos problemas apresentados pelos algoritmos indutivos analisados no capítulo 3.

Capítulo 6 - Análise dos resultados obtidos com o algoritmo ISREG, no qual faremos uma análise dos resultados obtidos pelo novo algoritmo. Essa análise utiliza, além do domínio descrito no capítulo dois, domínios reais com um número maior de exemplos.

Capítulo 7 - Conclusões, onde apresentaremos as conclusões obtidas com a elaboração deste trabalho e proporemos alguns trabalhos futuros com o propósito de dar continuidade às pesquisas nessa área.

Apêndice A - Relevância semântica de uma generalização: definição e um estudo de caso, no qual apresentaremos a definição de uma equação para o cálculo da relevância semântica de uma generalização, e mostraremos, também, os resultados de um estudo realizado para verificar o comportamento da equação definida diante dos valores reais fornecidos por especialistas.

Apêndice B - O algoritmo ISREG, onde o código fonte do algoritmo proposto será mostrado.

Apêndice C - Documentação dos domínios utilizados nos testes, onde apresentaremos todas as características e formas de conhecimento preliminardos dos domínios utilizados neste trabalhos.

Os principais termos utilizados pela área de aquisição de conhecimento, e os novos aqui introduzidos, foram definidos ao longo de todo o trabalho. Para facilitar sua localização apresentamos, no final deste trabalho, um índice remissivo dos termos definidos.

2 Utilização do conhecimento preliminar na aprendizagem indutiva

A indução é um dos métodos de inferência utilizado nos processos de aprendizagem, sua diversidade é determinada em função da utilização de um conhecimento adicional, eliciado preliminarmente, sobre o domínio. Neste capítulo explicaremos o mecanismo do método indutivo e sua relação com a utilização do conhecimento preliminar, apresentaremos os tipos de conhecimento preliminar mais utilizados e introduziremos, como exemplo, um domínio que possui as formas de conhecimento preliminar apresentadas e que será referenciado nos capítulos seguintes.

2.1 Introdução

Classificar as técnicas de aprendizagem tem sido tarefa de inúmeros pesquisadores que sempre buscam a melhor forma de modelar esse processo, conseqüentemente várias classificações já foram propostas representando as principais linhas de pesquisas.

Uma dessas classificações foi defendida por [Michalski 90] e apresentada em [Pires 93], ela divide o processo de aprendizagem em sintético e analítico. A aprendizagem sintética busca criar um conhecimento novo, ou ampliar o já apreendido, enquanto que na aprendizagem analítica o objetivo principal é reformular o conhecimento existente buscando sua melhor utilização. Dois métodos são utilizados para proceder a inferência, a indução utilizada pela aprendizagem sintética e a dedução utilizada pela analítica.

É no âmbito do método indutivo que se situa este trabalho, assim sendo, entendemos ser importante uma compreensão mais detalhada do processo desse método.

2.2 Princípios gerais da indução

O aprendizado por dedução consiste em utilizar uma regra geral verdadeira e verificar se ela se aplica em casos específicos.

Suponhamos a seguinte afirmação:

> Os empregados da empresa XYZ que trabalham no setor K a mais de 8 anos recebem um adicional de insalubridade.

Se agora for encontrado um indivíduo sobre o qual se pode afirmar:

> João é funcionário da empresa XYZ e trabalha no setor K a 10 anos,

então é possível se concluir que

> João recebe um adicional de insalubridade.

Se a regra geral é verdadeira, então a dedução também será [Hart 87].

A inferência indutiva segue um caminho oposto ao do aprendizado dedutivo: dado um conjunto de exemplos específicos, procura-se induzir uma regra geral. Suponha que não se conhece qual o setor da empresa XYZ que paga o adicional de insalubridade, mas existe um fichário com os dados de todos os funcionários desta empresa, então será possível conjecturar sobre a regra geral.

Considerando que existem funcionáios que:

- trabalham no setor K a 9, 11, 14 e 15 anos e recebem o adicional;
- não trabalham no setor K e não recebem o adicional;
- trabalham no setor K a 1, 3, 5 e 7 anos e não recebem o adicional.

É possível então induzir que os funcionários que trabalham no setor K a 9 anos ou mais têm direito ao adicional. Observa-se que a indução não tem a mesma precisão apresentada pela dedução, no exemplo o valor correto é 8 anos. Pode-se imaginar, também, que para um conjunto de situações observadas é possível gerar, potencialmente, um número infinito de hipóteses capazes de justificar esses fatos. Dessa forma, um conhecimento adicional àquele encontrado nos fatos é necessário para estabelecer um critério de preferência, ou um conjunto de restrições que reduzam o conjunto de hipóteses, esse conhecimento adicional é denominado conhecimento preliminar[1].

A utilização do conhecimento preliminar determina a definição de um tipo de indução, denominada indução empírica [Pires 93]. O termo empírica é utilizado para ressaltar a pouca, ou quase nenhuma, utilização do conhecimento preliminar nesse processo. A maioria dos trabalhos de indução desenvolvidos, particularmente aqueles que utilizam

[1] Originalmente designado *background knowledge* traduzido conhecimento preliminar por representar um conhecimento prévio sobre o domínio [Donato Júnior 94].

como forma de representação do conhecimento as árvores de decisão e os sistemas de regras de classificação são englobados por essa definição.

2.3 Principais tipos de conhecimento preliminar

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. O conhecimento preliminar também pode ser utilizado para simplificações e reformulações da base já gerada ou pode ser, preferencialmente, utilizado diretamente na geração desta base.

Um estudo mais completo sobre o papel do conhecimento preliminar no aprendizado pode ser encontrado em [Donato Júnior 94], nos deteremos nos tipos de conhecimento preliminar mais conhecidos e que serão referenciados e utilizados ao longo deste trabalho.

2.3.1 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 transforma-las num valor uniforme.

2.3.2 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, consideremos 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. A TAB. 2.1 mostra um possível conjunto de treinamento e a TAB. 2.2 apresenta uma estrutura de conhecimento preliminar que contempla as informações de custo monetário e de generalização para esse domínio. Este é utilizado apenas de forma didática para exemplificar as questões que serão levantadas a seguir, não sendo representativo, tendo em vista o pequeno número de exemplos apresentados.

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 está generalizado em POLÍGONO, mas não esta 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 2.1 - 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 2.2 - 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		

2.3.3 Relevância semântica

A inexistência de informações que auxiliem o algoritmo na geração da base de conhecimento, pode levar esse algoritmo a elaborar regras que, embora corretamente induzidas, não possuem nenhum significado prático ou são incompreensíveis para o especialista. Isso ocorre porque os algoritmos indutivos consideram todas as condições potencialmente iguais para uma determinada classificação.

Suponha que em um determinado conjunto de treinamento, num domínio de mecânica de automóveis, existam apenas dois exemplos classificando *Falta de Combustível* ocorridos em dois carros de cor azul. É provável que a regra "Se Cor = azul Então Falta de Combustível" seja gerada por um algoritmo indutivo que não utilize nenhum conhecimento adicional para informar que o atributo Cor, apesar de presente no conjunto de treinamento, é irrelevante para classificar um defeito mecânico. Evidentemente, uma regra desse tipo não tem nenhum significado prático. A ocorrência dessa situação foi denominado de problema semântico [Mongiovi 91][Cirne Filho 91].

Visando minimizar a ocorrência do problema semântico, foi definida a relevância semântica como sendo um tipo de conhecimento preliminar que relaciona, em um dado domínio, elementos de classificação com os atributos e seus respectivos valores

[Mongiovi 90]. A importância que tem um determinado par atributo=valor para realizar uma classificação define a relevância. A seleção do par atributo=valor realizada pelos algoritmos indutivos (por entropia, probabilidade de ocorrência, etc.) caracterizam a relevância sintática. Mas como em determinadas situações ela é insuficiente para refletir o conhecimento do especialista, faz-se necessária a utilização da relevância semântica. É na semântica do especialista que está embutido o conhecimento heurístico [Vasco 93].

Uma matriz foi a forma mais natural encontrada para representar a relevância semântica. Nas colunas ficam os elementos de classificação, enquanto que nas linhas colocam-se os atributos. Cada célula que relaciona as linhas com as colunas contém o conjunto de valores do atributo que são relevantes para a classificação do elemento existente na coluna. Essa matriz foi denominada Matriz de Relevância (MR) [Mongiovi 90]. A TAB. 2.3 apresenta uma MR para o conjunto de treinamento mostrado na TAB. 2.1.

TABELA 2.3 - Matriz de Relevância para o domínio "brinquedo seguro"

Classe / Atributo	Seguro	Perigoso
Forma	{elipse + círculo}	{quadrado + triângulo}
Cor	∅	∅
Tamanho	{grande}	{pequeno}
Material	{couro}	{metal}

Termos lingüísticos mais precisos, tipo: pouco relevante, mais ou menos relevante, muito relevante, etc., podem ser representados através de um conjunto nebuloso, transformando a matriz de relevância numa Matriz de Relevância Nebulosa (MRN) [Mongiovi 93a][Mongiovi 93b]. Um possível conjunto de valores designando os termos lingüísticos utilizados em uma MRN pode ser representado por:

$$\mathcal{TL} = \left\{ \begin{array}{c} 0,00 \text{ / irrelevante}, \quad 0,25 \text{ / pouco relevante}, \\ 0,50 \text{ / mais ou menos relevante}, \\ 0,75 \text{ / muito relevante}, \quad 1,00 \text{ / totalmente relevante} \end{array} \right\}$$

A TAB. 2.4 apresenta uma matriz de relevância nebulosa para o domínio "brinquedo seguro". Verifica-se facilmente que a MR é um caso particular de uma MRN, onde os valores presentes são totalmente relevantes, isso torna a MRN muito mais abrangente e precisa.

TABELA 2.4 - 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}

2.4 Conclusão

Os métodos indutivos de aprendizagem automática utilizam um conjunto de treinamento sobre o qual procedem uma inferência visando o aprendizado. O conjunto de treinamento possui informações que permitem estruturar uma base de conhecimento, mas não fornece subsídios que permitam determinar a importância prática ou o nível de preferência dos atributos dentro de uma classificação.

O conhecimento preliminar é uma forma de complementar as informações contidas no conjunto de treinamento, sendo esse conhecimento eliciado de um especialista. A forma de participação do perito humano deve ser bem definida, de modo a ser a mais objetiva possível [Donato Júnior 94]. Essa objetividade visa evitar que o processo de aprendizado automatizado recaia nos inconvenientes da aquisição de conhecimento cognitiva.

No caso da matriz de relevância, as experiências realizadas têm demonstrado que a sua eliciação desenvolve-se de forma fácil, exigindo apenas um pouco do tempo do especialista. No caso dos domínios médicos que serão utilizados neste trabalho a eliciação de cada matriz de relevância foi feita num tempo semelhante ao de uma simples consulta médica.

O mesmo comportamento foi observado na eliciação dos outros tipos de conhecimento preliminar o que evidencia que a utilização do conhecimento preliminar não põe em risco o caráter automático da aquisição de conhecimento por métodos indutivos.

3 Análise crítica de alguns algoritmos indutivos

Neste capítulo analisaremos o comportamento dos algoritmos indutivos mais representativos, utilizando como entrada o domínio definido no capítulo anterior. Demonstraremos que as bases de conhecimento geradas por esses algoritmos são vulneráveis aos problemas sintático e semântico e apresentaremos as principais soluções para esses problemas. A seqüência da análise dos algoritmos vai desde a detecção de problemas nas bases de conhecimento geradas pelo ID3, até as soluções propostas pelos algoritmos IDRT, PRISM, FRPRISM e EG2. Estas soluções foram escolhidas para o estudo por se basearem, principalmente, no uso de algumas formas de conhecimento preliminar. No final do capítulo apresentaremos um estudo comparativo entre os resultados encontrados.

3.1 Introdução

Os algoritmos indutivos realizam uma aprendizagem a partir de exemplos gerando uma base de conhecimento que, posteriormente, poderá ser utilizada por um sistema especialista. São ditos empíricos por utilizarem pouco, ou quase nenhum, conhecimento preliminar o que os torna passíveis de apresentarem problemas estruturais e de compreensão da base de conhecimento gerada [Michalski 90][Mongiovi 93a].

Um conjunto de dados que descrevem situações do mundo real, é a principal fonte de informações dos algoritmos indutivos. Esse conjunto de dados é denominado tabela de exemplos. A tabela de exemplos pode ter diferentes origens. Ela pode ser oriunda de uma base de dados existente (por exemplo, um fichário médico com o quadro histórico de vários pacientes), como também pode ser um conjunto tutorial preparado por um especialista (por exemplo, os casos clássicos de um determinado domínio). A tabela de exemplos normalmente é dividida em dois subconjuntos: um denominado conjunto de treinamento e o outro denominado conjunto de teste. O conjunto de treinamento é

utilizado pelos algoritmos indutivos para gerar uma base de conhecimento, enquanto o conjunto de teste é utilizado posteriormente para validar a base gerada.

Um exemplo é uma descrição de uma situação ocorrida em uma área da atividade humana. Cada exemplo é composto por um conjunto de condições e um elemento de classificação. As condições constituem-se de um par atributo=valor, sendo o atributo o fato concreto observado no mundo real e o valor um dos estados, situações, medidas, etc., que o atributo pode assumir. As condições também podem assumir a forma atributo=generalização, sendo a generalização a representação de um agrupamento de valores. O elemento de classificação, também chamado de classe, pode ser um diagnóstico, uma ação, ou qualquer conclusão atingida com a análise das condições. Um exemplo de uma tabela de exemplos foi apresentado em TAB. 2.1.

A estrutura de dados mais utilizada pelos algoritmos indutivos é a árvore de decisão. A árvore de decisão é uma estrutura tradicional de árvore, sendo que, neste caso, as folhas são os elementos de classificação, os nós não terminais representam os atributos e os ramos denotam os valores ou generalizações desses atributos. Ela é construída de forma que, ao percorrermos o caminho da raiz para uma das folhas, identificamos as condições suficientes para classificar a folha, o conjunto dessas condições é denominado premissa e a sua composição com a conclusão forma uma regra.

3.2 Família TDIDT

Os algoritmos da família TDIDT[1] [Quinlan 86] têm sido os mais estudados e aplicados na aquisição indutiva de conhecimento.

O objetivo principal desses algoritmos é mapear um conjunto de exemplos em uma árvore de decisão de tamanho mínimo (altura e largura).

Os algoritmos TDIDT constroem uma árvore de decisão a partir do nó raiz, colocando nesse nó um atributo selecionado no conjunto de treinamento através de uma função de avaliação (qui-quadrado, estatística G, índice de diversidade GINI e medida proporcional de ganho são algumas funções de avaliação propostas, no entanto, a mais utilizada é o cálculo da entropia [Klir 88]). Com o nó raiz definido, o conjunto de treinamento é dividido em subconjuntos (um para cada valor do atributo selecionado) e arcos são ligados a este nó, na mesma proporção. Recursivamente o processo é aplicado a cada subconjunto, provocando o aprofundamento dos ramos da árvore, até que uma condição de parada seja detectada, quando então, o elemento de classificação presente no subconjunto é devolvido pelo algoritmo para o ponto da chamada recursiva ou o algoritmo se encerra. A condição de parada pode ser definida tanto para construir árvores de decisão que classificam todos os elementos do conjunto de treinamento em domínios determinísticos, quanto para decidir pela não expansão da

[1] *Top Down Induction of Decision Trees*

árvore quando os exemplos fornecidos forem insuficientes. O Código 3.1 mostra uma descrição geral dos algoritmos TDIDT.

```
Entrada:  CT - Conjunto de Treinamento
Saída:    AD - Árvore de Decisão

TDIDT ( CT )
      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 valor da função de avaliação para cada atributo
                  presente em CT
         Selecione o atributo que apresentou o melhor valor
         Coloque o atributo selecionado na raiz de AD
         Para cada valor do atributo selecionado
            Crie um ramo em AD associado ao valor
            Crie um subconjunto CT onde só ocorra o par atributo=valor
            TDIDT ( subconjunto )
            Retire de CT todos os exemplos mapeados pelo ramo construído
         fimPara
fimTDIDT
```

CÓDIGO 3.1 - Algoritmos TDIDT

O algoritmo da família TDIDT mais conhecido e amplamente divulgado é o ID3 que serviu de base para quase todos os estudos realizados nesta área e desde seu surgimento várias melhorias tem sido propostas para aprimorar seu processo.

3.3 O algoritmo ID3

O ID3[2] utiliza como função de avaliação o cálculo da entropia [Quinlan 83]. Sua condição de retorno da recursão é que exista somente um elemento de classificação no subconjunto que está sendo trabalhado e a condição de parada é que todos os exemplos já tenham sido classificados. Para o sucesso de sua execução é necessário que as seguintes condições, com relação ao conjunto de treinamento, sejam atendidas:

- não existam contra-exemplos, também chamados conflitos de classificação, isto é, exemplos que possuem as mesmas características mas apresentam elementos de classificação distintos;
- não existam ruídos, ou seja, cada exemplo é completo e correto;
- os valores dos atributos devem ser discretos e mutuamente exclusivos.

[2] *Iterative Dichotomizer 3*

O Código 3.2 apresenta uma descrição para o ID3.

```
Entrada:  CT - Conjunto de Treinamento
Saída:    AD - Árvore de Decisão

ID3 ( CT )
     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 a entropia para cada atributo presente em CT
          Selecione o atributo que apresentou a menor entropia
          Coloque o atributo selecionado na raiz de AD
          Para cada valor do atributo selecionado
             Crie um ramo em AD associado ao valor
             Crie um subconjunto de CT onde só ocorra o par atributo=valor
             ID3 ( subconjunto )
             Retire de CT todos os exemplos mapeados pelo ramo construído
          fimPara
fimID3
```

CÓDIGO 3.2 - O algoritmo ID3

A FIG. 3.1 apresenta a árvore de decisão gerada pelo algoritmo ID3 após analisar o conjunto de treinamento do domínio "brinquedo seguro" (TAB. 2.1).

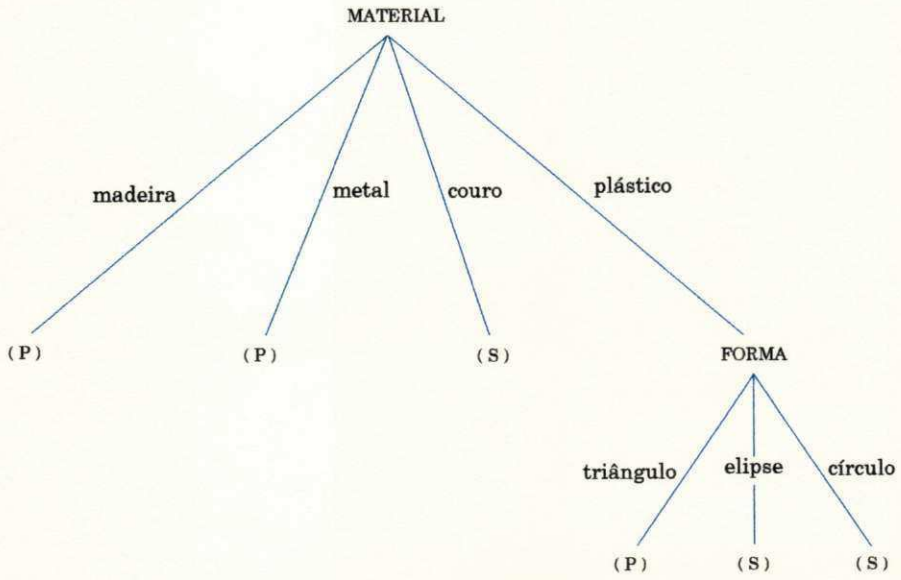

FIGURA 3.1 - Árvore de decisão gerada pelo ID3 para o domínio "brinquedo seguro".

Da árvore de decisão gerada pelo ID3 podemos formar as seguintes regras:

R1	Se	Material = madeira	Então	P	(8,9)
R2	Se	Material = metal	Então	P	(7,11,12)
R3	Se	Material = couro	Então	S	(1,2,3)
R4	Se	Material = plástico & Forma = triângulo	Então	P	(10)
R5	Se	Material = plástico & Forma = elipse	Então	S	(4)
R6	Se	Material = plástico & Forma = círculo	Então	S	(5,6)

Os valores entre parênteses indicam os números dos exemplos (casos) mapeados pela regra, esses números correspondem à seqüência de casos mostrados em TAB. 2.1. Deste ponto em diante sempre que nos referenciarmos a uma regra gerada por um algoritmo específico, o faremos na forma *Rn(nome_do_algoritmo)*, onde *Rn* representa a identificação da regra (por exemplo, R5(ID3)).

3.4 O problema sintático

Como foi visto anteriormente, as árvores de decisão podem ser facilmente transformadas em um conjunto de regras, no entanto, segundo [Cendrowska 88] [Mongiovi 93a] [Gaines 93], existem casos em que um conjunto de regras provenientes de uma árvore de decisão não é eficaz devido a fatores tais como:

- o conjunto de regras gerado é muito grande e complexo;
- o conjunto de regras testa condições desnecessárias levando o sistema especialista a fazer perguntas sem significado prático ou redundantes;
- as classificações realizadas pelo conjunto de regras são mutuamente exclusivas, tornando essas regras pouco "robustas";
- o conhecimento fica pulverizado pela árvore fazendo com que informações relevantes e suficientes para uma classificação fiquem subordinadas a condições menos relevantes, e por vezes irrelevantes.

Para exemplificar o último ponto descrito, consideremos as seguintes regras:

R1	Se	Tamanho = pequeno	Então	P	(7,8,11,12)
R2	Se	Material = madeira	Então	P	(8,9)
R3	Se	Cor = amarelo	Então	P	(10)

Essas regras são suficientes para classificar "P", em relação ao conjunto de treinamento apresentado na TAB. 2.1. Entretanto, não existe uma árvore de decisão

correspondente, porque elas não compartilham um atributo comum que possa ser colocado na raiz da árvore.

As questões levantadas sobre a utilização da árvore de decisão são de natureza estruturais, e exatamente por isso foram denominadas de problema sintático [Cirne Filho 92] [Mongiovi 93a].

O problema sintático já foi alvo de várias tentativas de solução. Uma possibilidade é a simplificação, executada a posteriori, que busca partes comuns de ramos distintos da árvore, isso porém pode levar, em caso de árvores grandes, a uma explosão combinatórial do número de comparações envolvidas no processo. Várias formas de simplificação podem se apresentar para uma mesma árvore, surgindo então uma questão: qual a melhor simplificação a ser feita ? Somente um especialista no domínio envolvido poderia responder a essa questão. Outra solução é induzir regras sem criar uma árvore de decisão, ou seja, gerar regras modulares a partir do conjunto de treinamento.

Os algoritmos da família AQ [Michalski 83] e o algoritmo PRISM [Cendrowska 88] geram regras modulares, a principal diferença entre eles é que enquanto os AQs buscam melhores combinações de condições (melhores complexos), o PRISM busca melhores condições. Isso permite ao PRISM usar técnicas de indução semelhantes aos TDIDT, diferindo apenas na estratégia de indução. Por esse motivo o PRISM será utilizado neste trabalho como base para solução do problema sintático.

3.5 O algoritmo PRISM

O PRISM[3] [Cendrowska 88] induz regras diretamente sem criar uma árvore de decisão. Seu processo se inicia escolhendo, de forma aleatória, um elemento de classificação do conjunto de treinamento, o passo seguinte é selecionar a maior probabilidade de ocorrência entre os pares atributo=valor para o elemento de classificação escolhido. É então criado um subconjunto de exemplos caracterizado pelo par atributo=valor selecionado. O processo de seleção do par atributo=valor e formação de subconjuntos, se repete até que um subconjunto formado contenha apenas o elemento de classificação escolhido. Uma regra é então estruturada pela conjunção dos pares atributo=valor selecionados para geração dos subconjuntos, concluindo o elemento de classificação que está sendo trabalhado. Os exemplos mapeados são então retirados do conjunto de treinamento, o processo se repete até que todos os exemplos contendo o elemento de classificação escolhido tenham sido retirados do conjunto de treinamento. Ao escolher o próximo elemento de classificação o conjunto de treinamento é restaurado, *i.e.*, 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

[3] A autora não deu nenhuma justificativa para o nome deste algoritmo.

[Cendrowska 88] [Gaines 93]. O Código 3.3 apresenta um procedimento geral para o algoritmo PRISM.

```
Entrada:  CT - Conjunto de Treinamento
Saída:    BC - Base de Conhecimento

PRISM ( CT )
    Para cada elemento de classificação E existente em CT
       Enquanto o elemento de classificação E existir em CT
          ListaCondições ← ∅
          CT_AUX ← CT
          Repita
             Calcule a probabilidade de ocorrência para cada par
                     atributo=valor
             Selecione o par com maior probabilidade
             Inclua o par selecionado em ListaCondições
             Crie um subconjunto de CT contendo todas as ocorrências do
                  par selecionado
             Até que no subconjunto criado exista apenas o elemento de
                     classificação E
             Regra ← SE ListaCondições ENTÃO E
             Retire de CT todos os exemplos mapeados por Regra
       fimEnquanto
       Restaure o CT original (CT ← CT_AUX)
    fimPara
fimPRISM
```

CÓDIGO 3.3 - O algoritmo PRISM

A maior diferença entre os algoritmos PRISM e ID3 reside no fato de que o primeiro busca encontrar os pares atributo=valor relevantes, enquanto o segundo concentra-se na busca de atributos relevantes, mesmo que alguns valores sejam irrelevantes para a classificação.

Analisando o conjunto de treinamento da TAB. 2.1, o algoritmo PRISM gerou as seguintes regras:

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)

Observando essas regras podemos verificar que não existe uma árvore de decisão correspondente a essas regras. Como não existe o compromisso de manter um atributo comum a todas as regras, novas condições foram formadas permitindo que o algoritmo mapeasse todos os exemplos mantendo a mesma quantidade de regras, se comparado com o ID3. Outro fato que deve ser observado é a capacidade do algoritmo de gerar

bases de conhecimento "robustas", ou seja, regras diferentes que mapeiam o mesmo exemplo (as regras R4(PRISM) e R6(PRISM) mapeiam o exemplo 8). Isso ocorre devido a restauração do conjunto de treinamento que é feita no final do processo de classificação de todos os exemplos de um determinado elemento de classificação.

3.6 O problema semântico

A inexistência de um conhecimento preliminar que auxilie o algoritmo na geração da base de conhecimento, pode levar este algoritmo a elaborar regras que, embora corretamente induzidas, não possuem nenhum significado prática ou são incompreensíveis para o especialista. Esse problema foi denominado de problema semântico [Mongiovi 90]. Ele ocorre, principalmente, porque os algoritmos indutivos consideram todas as condições potencialmente iguais para uma determinada classificação [Mongiovi 93a].
Dentro do domínio "brinquedo seguro" e analisando a matriz de relevância nebulosa apresentada na TAB. 2.4, podemos identificar os seguintes problemas semânticos existentes nas bases geradas pelos algoritmos analisados até o momento:

R1(ID3) :	Se	Material = madeira	Então P
R5(PRISM) :	Se	Cor = amarelo	Então P
R6(PRISM) :	Se	Material = madeira	Então P

Conforme a MRN utilizada, em tese fornecida pelo especialista do domínio, a regra R5(PRISM) apresenta o atributo Cor, que é totalmente irrelevante para qualquer conclusão no domínio estudado. Já as regras R1(ID3) e R6(PRISM) indicam que o valor madeira do atributo Material é irrelevante para realizar conclusões no referido domínio.

Isso significa que, embora os algoritmos tenham se comportado corretamente no seu processamento, as regras citadas não têm nenhuma utilidade prática para o especialista, pois elas não contém nenhuma condição que o especialista considere relevante para a conclusão de "P". Devido a isso essas regras são denominadas de regras inúteis. O principal motivo que levou a geração dessas regras foi o desconhecimento, por parte dos algoritmos, de um conhecimento preliminar que os informasse do grau de relevância da condição selecionada.

Problema semelhante foi levantado por [Uthurusamy 93] que identificou uma falta de evidência[4] nas regras geradas pelo ID3 e apresentou como solução o algoritmo *Inferule*. Esse algoritmo gera uma árvore binária e utiliza para seleção do atributo uma função que mede o poder de detalhamento desse atributo em relação a um determinado elemento de classificação. Embora que com essa função possam ser

[4] *Inconclusiveness* no original

resolvidos alguns casos, o problema semântico persiste, visto que a informação semântica não pode ser encontrada no conjunto de treinamento, por tratar-se de uma estrutura despida dessa característica. A informação semântica tem que ser fornecida pelo especialista e a matriz de relevância se apresenta como uma solução viável.

A tentativa inicial de solucionar esse problema, via matriz de relevância, foi feita através da definição do ID3X[5] [Mongiovi 90], um algoritmo que recebe como entrada uma árvore gerada pelo ID3 e com base nas informações contidas na matriz de relevância expande a árvore, colocando atributos relevantes nos ramos que apresentavam o problema semântico. Posteriormente, o ID3X foi generalizado para expandir árvores de decisão geradas por qualquer algoritmo TDIDT e recebeu o nome de ADEX[6] [Cirne Filho 92]. Essa solução apresenta como inconveniente a possibilidade de aumentar muito a profundidade da árvore de decisão, o que leva a formação de regras com um grande número de condições.

O passo seguinte foi definir um algoritmo que construísse a árvore de decisão contemplando as informações semânticas disponíveis na matriz de relevância. Esse algoritmo foi denominado IDRT [Cirne Filho 92].

Como indutor semântico de regras modulares, foi proposto o algoritmo RPRISM[7] [Cirne Filho 91][Cirne Filho 92], que utiliza uma função de avaliação definida como probabilidade condicional relevante, fazendo uso da matriz de relevância. A seguir foi proposta uma modificação [Mongiovi 93a][Mongiovi 93b] na fórmula da função de avaliação permitindo que o algoritmo passasse a utilizar a matriz de relevância nebulosa. Tal modificação foi implementada gerando o algoritmo FRPRISM.

Neste trabalho, devido ao fato da matriz de relevância ser um caso particular de uma matriz de relevância nebulosa, analisaremos apenas os algoritmos que apresentam uma solução para o problema semântico através da utilização dessa última. Dessa forma serão abordados apenas os algoritmos IDRT e FRPRISM.

3.7 O algoritmo IDRT

O IDRT[8] tem a estrutura de um algoritmo TDIDT, porém, utiliza uma Função de Avaliação Pragmática (FAP) cujo objetivo principal é ponderar a informação retirada do conjunto de treinamento (aspecto sintático) com a informação obtida na matriz de relevância (aspecto semântico) [Cirne Filho 92] . Esta ponderação é variável e permite que se direcione o resultado dando ênfase a um dos dois aspectos. A FAP normaliza a função de avaliação tradicional com base na entropia e a compõe com uma outra função denominada intensidade de relevância, esta composição é feita de tal forma que

[5] *Iterative Dichotomizer 3 Extented*
[6] Árvore de Decisão Expandida
[7] *Relevant PRISM*
[8] *Induction of Decision Relevant Trees*

os melhores atributos obtenham os maiores valores. A FAP é obtida pela aplicação da equação 3.1.

$$\mathcal{FAP}(a_i) = p.\mathcal{N}(a_i) + (1-p).I\mathcal{R}(a_i) \qquad (3.1)$$

sendo:

p um fator de ponderação;

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

$I\mathcal{R}(a_i)$ a intensidade de relevância do atributo.

A variável de ponderação, que é informada ao algoritmo, é quem determina a prioridade de utilização destas duas funções. Um fator de ponderação (p) igual a 0 (zero) prioriza o aspecto semântico enquanto que o valor 1 (um) enfatiza o aspecto sintático, gerando uma árvore semelhante à do ID3. Originalmente o IDRT foi definido utilizando a matriz de relevância normal, posteriormente foi proposta uma alteração na FAP de tal forma que ela pudesse utilizar a matriz de relevância nebulosa [Mongiovi 93a][Mongiovi 93b], alteração esta incorporada na implementação utilizada neste trabalho. O Código 3.4 descreve o algoritmo IDRT.

```
Entrada:  CT  - Conjunto de Treinamento
          MRN - Matriz de Relevância Nebulosa
          p   - Fator de Ponderação
Saída:    AD  - Árvore de Decisão

IDRT ( CT, MRN, p )
     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 a FAP para cada atributo presente em CT
          Selecione o atributo que apresentou a maior FAP
          Coloque o atributo selecionado na raiz de AD
          Para cada valor do atributo selecionado
             Crie um ramo em AD associado ao valor
             Crie um subconjunto de CT onde só ocorra o par atributo=valor
             IDRT ( subconjunto, MRN, p )
             Retire de CT todos os exemplos mapeados pelo ramo construído
          fimPara
fimIDRT
```

CÓDIGO 3.4 - O algoritmo IDRT

A FIG. 3.2 mostra a árvore de decisão gerada pelo IDRT ao analisar o conjunto de treinamento do domínio "brinquedo seguro", utilizando a matriz de relevância apresentada em TAB. 2.4. Como função de ponderação foi informado p=0,5 que solicita um equilíbrio entre os aspectos sintático e semântico.

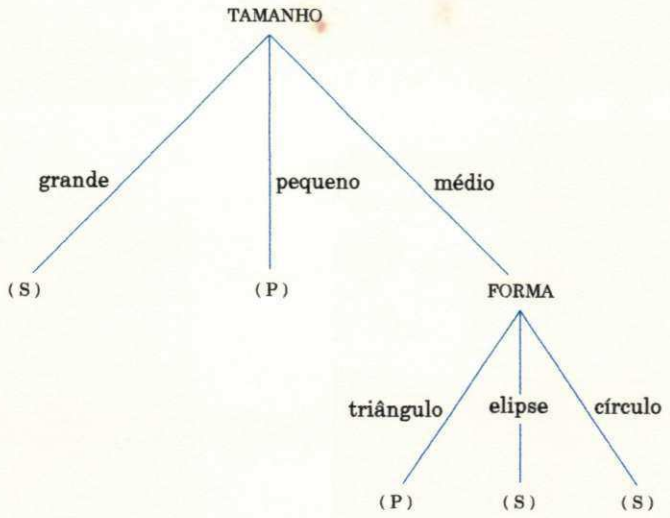

FIGURA 3.2 - Árvore de decisão gerada pelo IDRT para o domínio "brinquedo seguro".

Transformando a árvore gerada pelo IDRT em regras, temos:

R1	Se Tamanho = grande	Então S		(1,2,3,5)
R2	Se Tamanho = pequeno	Então P		(7,8,11,12)
R3	Se Tamanho = médio	& Forma = triângulo	Então P	(9,10)
R4	Se Tamanho = médio	& Forma = elipse	Então S	(4)
R5	Se Tamanho = médio	& Forma = círculo	Então S	(6)

As regras geradas pelo IDRT nos mostram que o atributo Tamanho foi selecionado porque na matriz de relevância indica que ele é mais relevante para as conclusões do que o atributo Material, selecionado pelo ID3. Dessa forma, nesse caso, as regras que apresentavam o problema semântico desapareceram.

3.8 FRPRISM

O FRPRISM[9] tem a mesma estrutura do PRISM, apresentada no Código 3.3. A diferença entre os dois reside na definição da função de avaliação. Enquanto o PRISM utiliza a probabilidade de ocorrência, o FRPRISM utiliza a probabilidade condicional relevante modificada para receber as informações contidas na matriz de relevância nebulosa [Mongiovi 93a][Mongiovi 93b]. A probabilidade condicional relevante é definida de tal forma que as condições irrelevantes fiquem com os piores valores possíveis, resolvendo assim o problema semântico. Em casos extremos, tipo uma matriz de relevância vazia ou todas as condições serem relevantes, a escolha da condição se dará pela probabilidade condicional de ocorrência, mantendo, assim, uma coerência com o PRISM.

9 *Fuzzy Relevant PRISM*

Analisando o conjunto de treinamento do domínio "brinquedo seguro" e com base na matriz de relevância mostrada em TAB. 2.4, o FRPRISM gerou as seguintes regras:

R1	Se Material = couro	Então S		(1,2,3)
R2	Se Tamanho = grande	Então S		(1,2,3,5)
R3	Se Material = plástico	& Forma = elipse	Então S	(4,5,6)
R4	Se Forma = círculo	& Tamanho = médio	Então S	(6)
R5	Se Tamanho = pequeno	Então P		(7,8,11,12)
R6	Se Cor = amarelo	Então P		(10)
R7	Se Material = madeira	Então P		(8,9)

Podemos observar que o FRPRISM reformulou várias regras geradas pelo PRISM e criou uma nova regra (R1(FRPRISM)), com o propósito de torna-las mais úteis, em função das informações da matriz de relevância. Contudo, ele gerou as regras R6 e R7 que são idênticas às geradas pelo PRISM e que apresentavam o problema semântico. Mesmo assim, esta nova base de conhecimento gerada apresenta uma qualidade superior àquela gerada pelo PRISM, já que busca regras mais significativas e precisas.

3.9 EG2

O EG2[10] [Nuñez 91] utiliza duas formas de conhecimento preliminar, o custo e a generalização. Ele gera uma árvore de decisão e utiliza como função de avaliação o ICF[11] [Nuñez 91] que define uma relação custo/benefício do atributo. A fórmula do ICF (equação 3.2) foi deduzida de tal forma que existe uma relação entre o custo e o ganho de informação de um atributo, sendo que esta relação é inversamente proporcional a grandeza de seus valores. Assim sendo, o algoritmo busca sempre o atributo de menor ICF que será aquele com menor custo e o máximo de informação.

$$ICF = \frac{\left(\mathcal{FC}(a_i)+1\right)^{fe}}{2^{\Delta I}-1} \tag{3.2}$$

onde:

$\mathcal{FC}(a_i)$ é a função que fornece o custo do atributo a_i;

fe representa um fator de economia;

ΔI é o ganho de informação do atributo a_i.

Da mesma forma que os algoritmos TDIDT o EG2 inicia com um árvore vazia e coloca na raiz o atributo com menor ICF. Cria uma lista com as generalizações possíveis e os valores não generalizados do atributo selecionado e divide o conjunto de treinamento em subconjuntos, um para cada elemento pertencente a lista. O processo é aplicado

[10] *Economic Generalizer 2*

[11] *Information Cost Function*

recursivamente a cada subconjunto, provocando o aprofundamento dos ramos da árvore, até que exista somente um elemento de classificação no subconjunto ou que todos os exemplos já tenham sido mapeados. A folha de cada ramo da árvore é preenchida com o elemento de classificação que restou em cada subconjunto. Quando uma generalização é selecionada e a classificação não ocorre, esta generalização é decomposta em seus valores normais e a lista é classificada colocando sempre as generalizações na cabeça por ordem de quantidade de valores generalizados, os valores normais são classificados por ganho de informação com base na entropia [Nuñez 91].

A utilização das informações do conhecimento preliminar pode ser direcionada e para isso o algoritmo possui dois parâmetros definidos como:

- fe (ϖ no original) que representa o fator de economia, no intervalo [0,1], onde fe=0 significa desprezar a informação de custo e fe=1 solicita a busca da economia máxima;

- g (ct[12] 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 fe=0 e g=0 fazem com que o EG2 gere a mesma árvore gerada pelo ID3.

O Código 3.5 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.

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. 3.3. O algoritmo foi executado com os parâmetros fe=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.

---

[12] *Completness 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 de 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
fimEG2
```

CÓDIGO 3.5 - O algoritmo EG2

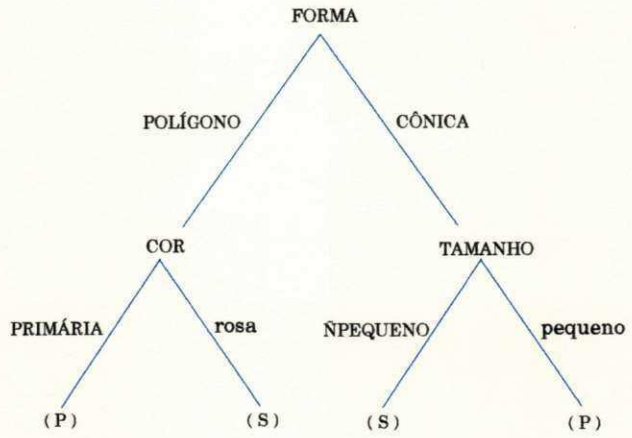

FIGURA 3.3 - Árvore de decisão gerada pelo EG2 para o domínio "brinquedo seguro".

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 = ÑPEQUENO	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 próximo capítulo faremos uma comparação entre o custo das bases de conhecimento geradas pelos algoritmos analisados.

3.10 Análise comparativa

Considerando os problemas sintático e semântico e a utilização de conhecimento preliminar, nas formas apresentadas, cada algoritmo estudado apresenta vantagens e deficiências que procuraremos salientar.

Devido à sua condição de pioneiro, e à comprovação prática dos seus resultados, o ID3 tem servido de referência para os estudos realizados. Obviamente, ele não contempla nenhum dos fatores em análise, tendo em vista que foi através da observação dos seus resultados que a quase totalidade das inovações apresentadas surgiram. Isso não significa que sempre iremos encontrar pontos negativos em uma base de conhecimento gerada pelo algoritmo. É perfeitamente possível que o ID3 gere uma árvore de decisão que não apresente os problemas sintático e/ou semântico, ou uma árvore de baixo custo. A observação correta a se fazer é que este algoritmo é mais suscetível aos problemas apresentados.

O PRISM, procura solucionar o problema sintático mas não elimina a possibilidade do problema semântico, nem reconhece outras formas de conhecimento preliminar. Sua grande virtude está na geração de regras modulares, eliminando a necessidade do uso da árvore de decisão.

O algoritmo IDRT, através da utilização do conhecimento preliminar definido como relevância semântica, procura solucionar o problema semântico, mas, por construir uma árvore de decisão, não está isento da possibilidade de apresentar o problema sintático. Também não reconhece outras formas de conhecimento preliminar.

A definição do FRPRISM foi uma tentativa de globalizar uma solução para os problemas sintático e semântico em um único algoritmo. No próprio exemplo apresentado, ficou claro que essa busca sem dúvida melhora bastante a qualidade das regras geradas, mas não garante a inexistência do problema semântico. Isto porque o peso da informação sintática presente no conjunto de treinamento pode ser forte o

suficiente para neutralizar a informação semântica contida na matriz de relevância. O não reconhecimento de outras formas de conhecimento preliminar, pode fazer com que a base de conhecimento gerada tenha, por exemplo, um alto custo envolvido.

As formas de conhecimento preliminar apresentadas, custo e generalização, são contempladas pelo algoritmo EG2 que por gerar uma árvore de decisão e não reconhecer a matriz de relevância pode apresentar os problemas sintático e semântico. Acreditamos, contudo, que a possibilidade desse algoritmo apresentar os referidos problemas seja reduzida pelo fato dele construir uma árvore de tamanho mínimo (altura e largura) e agrupar conclusões devido a utilização da generalização. Porém, a generalização não é um conhecimento que possa ser aplicado em qualquer domínio, o que pode tornar o algoritmo vulnerável. Mas, quando é possível utilizar a generalização, ela mostra toda sua capacidade de redução do tamanho da árvore e, principalmente, permite generalizar valores que não estão presentes no conjunto de treinamento. Como foi o caso das regras R1(EG2) e R2(EG2) que validam o valor pentágono nas suas conclusões sem que ele jamais tenha sido observado. Obviamente, como a generalização nada mais é do que um *OR* de condições, o limiar de generalização pode alterar significativamente os resultados apresentados pelo algoritmo. Por exemplo, se solicitarmos ao EG2 uma outra análise do conjunto de treinamento, agora utilizando os parâmetros fe=1 e g=1, que significam o custo máximo e generalização somente quando todos os valores ocorrerem, a árvore gerada dobra de tamanho se comparada com aquela apresentada na FIG. 3.3, uma vez que o algoritmo deixa de utilizar a generalização do atributo Forma.

Os algoritmos têm que ser analisados dentro do seu contexto, não sendo possível determinar com precisão quem é o melhor. Cada um atende a uma necessidade do usuário: se a intenção é uma base bem pequena, talvez a melhor solução seja o EG2 com generalização; se o interesse são regras semanticamente corretas, uma solução pode ser o FRPRISM; se a necessidade é de baixo custo, o EG2 tende a dar a melhor solução.

3.11 Conclusão

A praticidade e eficiência do processo de aquisição automática de conhecimento via indução a partir de exemplos têm sido um dos elementos motivadores da busca do aprimoramento deste processo. Esta pode ser uma explicação possível para o fato de corriqueiramente encontrarmos proposições de melhoramentos em algoritmos existentes ou propostas de novos algoritmos. Cada nova proposição, no entanto, procura solucionar um problema específico ou incorpora um novo tipo de conhecimento preliminar. Comportamento coerente com o processo de descoberta científica, mas sem um compromisso de incorporar as descobertas passadas.

Neste capítulo apresentamos os resultados encontrados por uma linha de pesquisa que busca sempre a melhoria da base de conhecimento gerada pelos algoritmos indutivos e que apresentaram ótimos resultados em cada um dos problemas abordados.

Contudo, nenhuma solução consegue contemplar todos os elementos analisados salientando, desta forma, a existência de uma lacuna entre os métodos indutivos a partir de exemplos. Lacuna esta que poderia ser preenchida com um algoritmo que buscasse minimizar a ocorrência dos problemas levantados e contemplasse os principais tipos de conhecimento preliminar, tendo um compromisso com a qualidade.

4 Qualidade de uma base de conhecimento

Dotar o processo automático de aquisição de conhecimento de um mecanismo, também automático, que possibilite a realização de uma análise qualitativa de uma base de conhecimento gerada por um algoritmo indutivo é o objetivo principal deste capítulo. Para isso, analisamos algumas formas de avaliação de uma base de conhecimento e propomos uma avaliação qualitativa com base na relevância semântica contida no conhecimento preliminar, denominado matriz de relevância. Mostraremos os valores obtidos pelos tipos de avaliação estudados e pelo proposto quando aplicados ao domínio definido no capítulo anterior.

4.1 Introdução

"A *American Society for Quality Control*, que existe há quase 50 anos, e se reúne anualmente, em conferências, não tem chegado a um acordo sobre a definição de qualidade" [Belchior 92]. Devemos então estudar a qualidade dentro das várias abordagens em que ela pode ser vista, algumas dessas abordagens foram descritas por [Paladini 90] e apresentadas em [Belchior 92], dentre as quais ressaltamos duas:

- **Uma que é centrada no produto**, na qual a qualidade é vista como passível de medição, através dos atributos do produto. Um produto de qualidade teria uma maior quantidade de atributos de melhores características, dando-se um caráter preciso à qualidade.

- **Outra centrada no usuário**, onde a qualidade é alcançada, quando atende, prontamente, as necessidades e as conveniências do usuário. O usuário é a fonte de toda a avaliação sobre a qualidade do produto.

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 está diretamente relacionado com a precisão e

a clareza da base gerada, uma vez que dela depende a capacidade do sistema em apresentar explicações convincentes e satisfatórias [Mongiovi 93a].

Os métodos automáticos indutivos não apresentam uma forma, também automática, de avaliação da qualidade da base de conhecimento por eles gerada. 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.

Nos interessa, então, utilizar na análise de uma base de conhecimento as duas abordagens destacadas sobre qualidade. Precisamos medir a base para saber sua qualidade antes de utilizá-la em um sistema baseado em conhecimento, dessa forma teremos a opção de utilizar a melhor base visando um aumento da satisfação do usuário. Além do mais, para manter o espírito automático da aquisição, todo esse procedimento de análise deve ser feito de forma automática com o mínimo possível de participação do especialista.

4.2 Formas de avaliação de uma base de conhecimento

Uma base de conhecimento pode ser analisada sob vários aspectos, atendendo à área de interesse 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. Outro fator importante na construção de um sistema baseado em conhecimento é a relação custo/benefício envolvida na utilização desse sistema. Assim sendo, podemos dividir a avaliação de uma base de conhecimento sob os aspectos: quantitativo, custo e semântico [Alexandre 93].

4.2.1 Aspecto quantitativo

O problema sintático tem influência direta no tamanho da regra e conseqüentemente no tamanho da base de conhecimento gerada. Por outro lado, independentemente do tamanho, essa base pode ter um bom nível de acerto nas classificações que lhe são submetidas. Dessa forma, basicamente existem duas formas de avaliar uma base de conhecimento sob o aspecto quantitativo: pelo tamanho e pela acurácia.

Em uma análise pelo tamanho pode-se medir a quantidade de regras geradas e o tamanho médio destas regras, que é obtido pelo número médio de condições por regra.

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 freqüência de acertos na classificação do conjunto de teste mede a qualidade da base, quanto maior melhor.

4.2.2 Aspecto custo

Como foi apresentado no capítulo 3 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.

4.2.3 Aspecto semântico

Entendemos como semanticamente correta a base de conhecimento que apresentar todas as suas regras como úteis e relevantes para classificar o domínio.

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 as questões já levantadas de indisponibilidade de tempo e do custo da hora de trabalho do especialista, aliado a forte subjetividade presente neste tipo de avaliação, que pode levar a resultados diferentes entre especialistas. Contudo, esse aspecto é muito importante pois ele pode detectar a presença do problema semântico.

Um processo automatizado de avaliação deste aspecto é uma solução para contornar os problemas encontrados na avaliação não automatizada.

4.3 Utilização da relevância semântica na avaliação de uma base de conhecimento

Como já foi demonstrado no capítulo anterior, a relevância semântica desempenha um papel fundamental na tentativa de minimizar a ocorrência do problema semântico. A matriz de relevância como forma específica de eliciação de conhecimento é uma ferramenta objetiva e precisa. Apesar da participação do especialista, sua objetividade evita que se recaia nos inconvenientes da aquisição de conhecimento cognitiva [Donato Júnior 94].

Uma base de conhecimento é composta por 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 um grau de relevância para a regra.

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

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.

Com esse grau de relevância, passaríamos a ter uma medida da base de conhecimento sob o aspecto semântico, permitindo uma avaliação qualitativa que atende as duas abordagens sobre qualidade apresentadas anteriormente.

4.4 Definição do grau de relevância de uma base de conhecimento

O grau de relevância é um número que mostra o nível de qualidade semântica de uma base de conhecimento, levando em conta as informações contidas na matriz de relevância. Para definirmos a equação que deve ser utilizada no cálculo desse grau, é necessário, inicialmente, introduzirmos algumas definições formais dos elementos que compõem seu ambiente de atuação.

Assim sendo, sejam:

- $\mathcal{BC} = \{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \ldots, \mathcal{R}_{n_r}\}$ uma base de conhecimento formada por um conjunto de n_r regras;
- $\mathcal{R}_r = (\mathcal{P}_r, \mathcal{E}_j)$ uma regra em que $\mathcal{P}_r$ é a premissa dessa regra e $\mathcal{E}_j$ seu elemento de classificação;
- $\mathcal{E} = \{\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \ldots, \mathcal{E}_{n_\mathcal{E}}\}$ um conjunto de $n_\mathcal{E}$ elementos de classificação;
- $\mathcal{P}_r = \{C_1, C_2, C_3, \ldots, C_{n_c}\}$ um conjunto de n_c condições representando a premissa da regra $\mathcal{R}_r$;
- $C_m = (a_i = v_{ik})$ uma condição em que a_i é um atributo e v_{ik} seu valor associado;
- $C_m = (a_i = g_{ik})$ uma condição em que a_i é um atributo e g_{ik} uma generalização associada a esse atributo;
- $\mathcal{A} = \{a_1, a_2, a_3, \ldots, a_{n_a}\}$ um conjunto de n_a atributos;
- $\mathcal{V}(a_i) = \{v_{i1}, v_{i2}, v_{i3}, \ldots, v_{inv}\}$ um conjunto de nv_i valores associados ao atributo a_i;
- $\mathcal{G}(a_i) = \{g_{i1}, g_{i2}, g_{i3}, \ldots, g_{ing}\}$ um conjunto de ng_i generalizações definindo os valores associados ao atributo a_i, sendo $0 \leq ng_i < nv_i$
- $g_{ik} = \{v_{i1}, v_{i2}, v_{i3}, \ldots, v_{invg}\}$ 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 $1 \leq j \leq n_\mathcal{E}$, onde cada elemento dessa matriz indica a relevância ρ do par $(a_i = v_{ik})$ para a classificação do elemento de classificação $\mathcal{E}_j$ e pode ser definido por: $\mathcal{MR}_{ij} = \{(\rho / v_{ik}) : \rho \in \mathrm{R}, 0 \leq \rho \leq 1 \text{ e } v_{ik} \in \mathcal{V}(a_i)\}$ sendo **R** o conjunto dos números reais;
- $\mathcal{F}_{ijk}$ uma função que fornece a relevância do atributo a_i para o elemento de classificação $\mathcal{E}_j$ no valor v_{ik}.
- $\mathcal{FC}(C_m)$ uma função que fornece o custo do atributo 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:

$$\Re\,(C_m, \mathcal{E}_j) \;=\; \begin{cases} \dfrac{\dfrac{1}{nvg_{ik}} \displaystyle\sum_{k=1}^{nvg_{ik}} \mathcal{F}_{ijk}}{1+\sigma^2} & \text{se } C_m = (a_i = g_{ik}) \\ \mathcal{F}_{ijk} & \text{se } C_m = (a_i = v_{ik}) \end{cases} \qquad (4.1)$$

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 (4.2)

$$\mathcal{GRR}\,(\mathcal{R}_r) \;=\; \frac{\dfrac{1}{|\mathcal{P}_r|} \displaystyle\sum_{C_c \in \mathcal{P}_r} \Re(C_c, \mathcal{E}_j)}{1+\sigma^2} \qquad (4.2)$$

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:

$$\mathcal{GRBC}\,(\mathcal{BC}) \;=\; \frac{\dfrac{1}{n_r} \displaystyle\sum_{r=1}^{n_r} \mathcal{GRR}(\mathcal{R}_r)}{1+\sigma^2} \qquad (4.3)$$

com σ^2 representando a variância da relevância das regras que compõem a base de conhecimento $\mathcal{BC}$.

A variância utilizada em cada uma das equações apresentadas é obtida pela aplicação da equação (4.4):

$$\sigma^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n} \qquad (4.4)$$

sendo x_i cada elemento utilizado no cálculo da média $\bar{x}$. 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 [Alexandre 93].

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 as 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 $\mathcal{R}_2$ [$\mathcal{GRR}(\mathcal{R}_2) = 0,5$] se manteria, mas para a regra $\mathcal{R}_1$ o resultado seria outro [$\mathcal{GRR}(\mathcal{R}_1) = 0,4$], ou seja a regra é penalizada por possuir uma condição irrelevante.

A TAB. 4.1 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 4.1 - 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 (4.2) e (4.3), 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. 4.1 mostra a convergência dos valores calculados como grau de relevância pela média e pelas equações definidas, aqui chamados de valores ajustados.

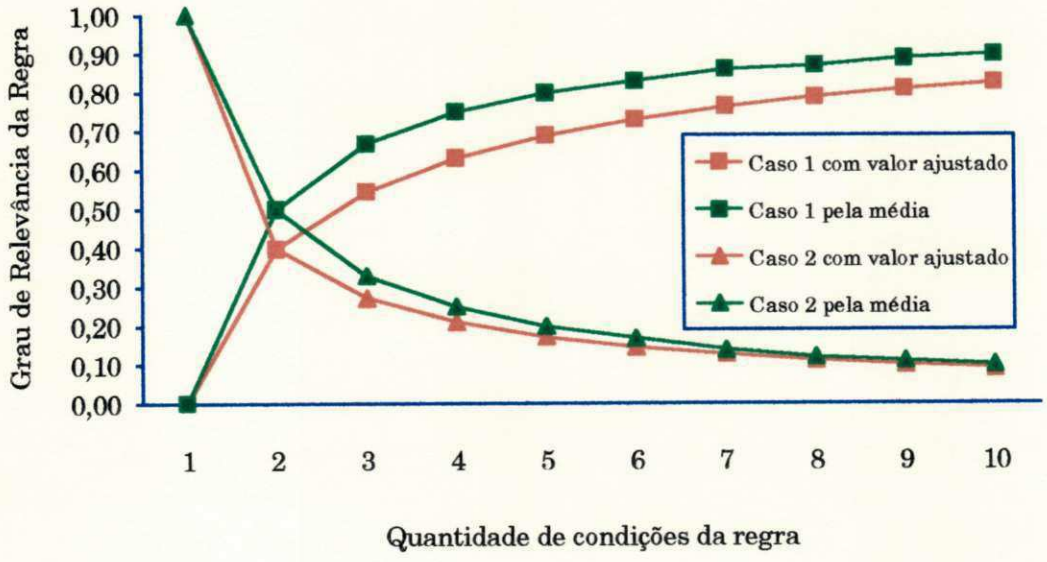

GRÁFICO 4.1 - Convergência da relevância de uma regra

4.5 Utilização das formas de avaliação de uma base de conhecimento

Anteriormente apresentamos, de uma forma empírica, algumas formas de avaliação quantitativa de uma base conhecimento. Utilizando o mesmo ambiente apresentado no item anterior, podemos definir as formas de cálculo dos tipos de avaliação quantitativa por:

- **quantidade de regras** como sendo 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 esta representada por n_r.

- **tamanho médio das regras** que 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:

$$\overline{TR}(\mathcal{BC}) = \frac{\sum_{n_r}\sum_{n_c} \mathcal{P}_r}{n_r} \tag{4.5}$$

- **custo médio de uma classificação** que vem a ser a média aritmética do custo de todas as classificações feitas para gerar a base de conhecimento. Entende-se por custo de uma classificação, como sendo a soma dos custos dos atributos que compõem a regra vezes o número de exemplos mapeados [Nuñez 91]. Dessa forma encontrar o custo médio de uma classificação é o mesmo que encontrar o custo médio de uma regra. Este custo é definido pela equação (4.6).

$$\overline{CR}(\mathcal{BC}) = \frac{\sum_{n_r}\sum_{n_c} f_i \cdot n_{ex_m}}{n_{ex}} \tag{4.6}$$

sendo:

f_i uma função que fornece o custo do atributo a_i, que está na condição C_m da premissa $\mathcal{P}_r$;

n_{ex} o número de exemplos contidos no conjunto de treinamento;

n_{ex_m} o número de exemplos mapeados pela regra que esta sendo analisada.

Utilizando os resultados apresentados pelos algoritmos estudados no capítulo anterior e com base nas formas de conhecimento preliminar bem como no conjunto de treinamento definidos no capítulo 2, apresentamos na TAB. 4.2 os valores encontrados para as formas de avaliação das bases de conhecimento geradas pelos algoritmos estudados. Apresentamos, também, como análise qualitativa o grau de relevância da base de conhecimento utilizando a equação (4.3) e o número de regras inúteis, ou seja, as regras que apresentaram um valor zero para a equação (4.2).

Podemos observar que os algoritmos que utilizam a relevância semântica (IDRT e FRPRISM) apresentam, numa análise geral, um bom desempenho. As duas formas de conhecimento preliminar, generalização e custo, utilizados pelo EG2, mostram toda sua utilidade, pois apesar do algoritmo gerar uma árvore de decisão, apresenta um bom grau de relevância além do menor custo dentre todos os algoritmos estudados. Evidentemente esse exemplo serve apenas como ilustração das questões levantadas, havendo a necessidade de um teste mais significativo com uma modelagem real utilizando um conjunto de treinamento com um bom número de exemplos.

TABELA 4.2 - Resultados da avaliação das bases de conhecimento geradas, pelos algoritmos estudados, para o domínio "brinquedo seguro".

Algoritmos	Aspecto Quantitativo		Custo Médio de Classificação	Aspecto Qualitativo	
	Quant. de Regras	Tam.Médio de Regra		Regras Inúteis[1]	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	4	2.00	95.00	-	0.48

4.6 Conclusão

A inexistência de um processo automatizado que permitisse uma avaliação qualitativa da base de conhecimento gerada, 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, representada na forma da matriz de relevância, foi motivada pela busca da qualidade nas bases de conhecimento geradas pelos métodos indutivos. Sua fácil eliciação minimizou a participação do perito humano e tornou disponível informações que podiam ser utilizadas na realização de uma avaliação qualitativa da base de conhecimento gerada para o seu domínio.

A definição do grau de relevância de uma base de conhecimento é uma contribuição visando dotar os métodos de aprendizagem automática de um procedimento que permita uma avaliação qualitativa da base de conhecimento por eles gerada.

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

5 Indução semântica de regras modulares, econômicas e generalizadas

Neste capítulo propomos um algoritmo que induz regras diretamente, sem gerar uma árvore de decisão, e que busca minimizar, ao mesmo tempo, a ocorrência dos problemas sintático e semântico, além de utilizar, de forma conjunta, as principais formas de conhecimento preliminar disponíveis. A dedução da função de avaliação utilizada pelo algoritmo é feita com base na teoria do ganho de informação e procura estabelecer uma relação custo/benefício para cada condição selecionada. Nos itens finais do capítulo são apresentados um procedimento geral para o algoritmo e as características do ambiente de implementação

5.1 Introdução

Conforme mostramos, e exemplificamos, no capítulo 3, 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 EG2, 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 pode apresentar 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 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 e semântico [Alexandre 94b].

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 e tamanho), a exemplo do IDRT.

A função de avaliação utilizada será deduzida de tal forma que permita fazer uma relação de custo/benefício, como na função ICF do EG2, utilizando a informação de relevância semântica, presente na matriz de relevância nebulosa, e permitindo a utilização de generalização de valores.

5.2 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 $\mathcal{GEF}$[1]. 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 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 $\mathcal{GEF}$ utilizaremos as definições apresentadas no capítulo 4 (item 4.4), havendo a necessidade de acrescentarmos apenas a definição de uma função que forneça o custo da condição C_m.

Por contemplar o conhecimento preliminar denominado generalização, a condição C_m pode ser formada pelo par atributo=valor ou atributo=generalização. Assim sendo, a função de custo pode ser definida como:

$$\mathcal{FC}(C_m) = \left\{ \begin{array}{l} \mathcal{FC}(a_i = v_{ik}) \\ \\ \mathcal{FC}(a_i = \mathcal{G}_{ik}) \end{array} \right\} \text{custo}(a_i) \qquad (5.1)$$

[1] *Global Evaluation Function*

• Ganho de informação

A quantidade de informação de um determinado elemento de classificação $\mathcal{E}$, contida num conjunto de treinamento, é dada por:

$$I(\mathcal{E}_j) = \log_2\left(\frac{1}{\mathcal{P}(\mathcal{E}_j)}\right) \tag{5.2}$$

onde: $\mathcal{P}(\mathcal{E}_j)$ é a probabilidade de um exemplo possuir a classe $\mathcal{E}_j$.

Agora se desejarmos saber a quantidade de informação para a classificação do mesmo elemento $\mathcal{E}_j$ mas agora devido a presença de uma condição C_m, este valor é obtido por:

$$I(\mathcal{E}_j|C_m) = \log_2\left(\frac{\mathcal{P}(\mathcal{E}_j|C_m)}{\mathcal{P}(\mathcal{E}_j)}\right) \tag{5.3}$$

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

No algoritmo PRISM, o objetivo da utilização do ganho de informação é obter as menores regras possíveis, para isso basta escolher o maior valor de $I(\mathcal{E}_j|C_m)$. Porém, como $\mathcal{P}(\mathcal{E}_j)$ é o mesmo para todo C_m, isso se reduz a obter o maior $\mathcal{P}(\mathcal{E}_j|C_m)$. Portanto, para efeito de escolha da melhor condição o ganho de informação será representado por:

$$\mathcal{P}(\mathcal{E}_j|C_m) \tag{5.4}$$

• Probabilidade condicional relevante

A probabilidade condicional relevante, que aqui será representada pela letra Q, é uma função que engloba o ganho de informação, dado pela equação 5.3, com a relevância da condição C_m para a classificação de $\mathcal{E}_j$. Essa função foi definida para o algoritmo FRPRISM [Mongiovi 93b] e é dada pela equação:

$$Q(\mathcal{E}_j|C_m) = gr \;.\; \mathcal{P}(\mathcal{E}_j|C_m) \;+\; (1 - gr) \;.\frac{\min(1, n_{ce})}{n_c} \tag{5.5}$$

onde: n_{ce} 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;
gr é o grau de relevância da condição C_m para a classificação de $\mathcal{E}_j$.

Como gr varia no intervalo [0,1], podemos observar na equação 5.4 que quando a condição C_m é totalmente relevante para concluir $\mathcal{E}_j$ (gr=1), o valor de Q fica idêntico ao do ganho de informação (equação 5.4). Quando ocorre da condição C_m ser

totalmente irrelevante para concluir $\mathcal{E}_j$ ($\mathcal{GI}$=0), o valor de Q é penalizado e, na maioria das vezes, é dado por $1/n_{ce}$. Nos demais casos o ganho de informação sofre uma redução em função da relevância da condição C_m.

- **Função** $\mathcal{GEF}$

Seguindo a idéia da relação custo/benefício do algoritmo EG2, a função de avaliação global foi definida como $\mathcal{GEF}$, que é utilizada através da equação:

$$\mathcal{GEF}(\mathcal{E}_j|C_m) = \frac{(\mathcal{FC}(C_m)+1)^{fe}}{fq \,.\, Q(\mathcal{E}_j|C_m) \;+\; (1 - fq) \,.\, \mathcal{P}(\mathcal{E}_j|C_m)} \qquad (5.6)$$

onde: fe é um fator de economia que pondera o custo e fica no intervalo [0,1];
fq é um fator de qualidade, atua no intervalo [0,1];
$\mathcal{FC}(C_m)$ é a função que fornece o custo da condição C_m.

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

A melhor condição será aquela que apresentar o menor valor de $\mathcal{GEF}$, significando a maior qualidade.

5.3 O algoritmo 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 ($\mathcal{GEF}$). 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.e.*, 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 5.1 apresenta um procedimento geral para o algoritmo ISREG.

```
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 )
FimISREG
```

CÓDIGO 5.1 - 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, ou se as duas regras devem permanecer.

Quando ocorre um empate nos valores encontradas pela função $\mathcal{GEF}$ 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.

5.4 O algoritmo ISREG no ambiente A4

O algoritmo ISREG foi implementado em C++ e pertence à classe algoritmos do ambiente A4. O A4 (Ambiente de Apoio a Aquisição Automática de Conhecimento) 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 92][Vasco 93]. 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 e FRPRISM, contudo, não haviam sido submetidos a testes mais rigorosos. Como o ISREG usa princípios de outros algoritmos, tivemos que implementar os algoritmos IDRT e EG2, além de rever e testar criteriosamente os algoritmos já implementados.

Sendo o A4 uma primeira versão de um ambiente, é natural, e desejável, que esteja em constante evolução. Assim sendo, várias alterações se fizeram necessárias à medida que o ambiente era utilizado, com o propósito de aprimorar suas ferramentas e uniformizar processos.

Dentre as alterações 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;

- viabilização de importação de tabelas de exemplos disponibilizadas por diversas universidades em diretórios públicos das redes de comunicação.

O algoritmo ISREG solicita a informação de três parâmetros (fe, fq 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.

Várias outras modificações de menor impacto, mas necessárias para facilitar o uso do ambiente, foram realizadas.

5.5 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,

UFPb/BIBLIOTECA/PRAI

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.

6 Análise dos resultados obtidos com o algoritmo ISREG

Analisar os resultados apresentados pelo algoritmo ISREG e verificar seu comportamento diante de domínios reais, que possuem um grande número de exemplos, é o principal objetivo deste capítulo. Inicialmente, mostraremos os resultados obtidos pelo algoritmo quando da análise do domínio "brinquedo seguro", que vem sendo utilizado ao longo deste trabalho, possibilitando uma comparação mais minuciosa desses resultados com aqueles apresentados pelos outros algoritmos estudados. Posteriormente, mostraremos os resultados alcançados na utilização de domínios maiores extraídos do mundo real.

6.1 Introdução

Segundo [Lucena 83] existem várias técnicas, provenientes da área da lógica matemática, que são aplicadas no processo de programação para provar, matematicamente, que o programa escrito está correto, *i.e.*, produz os resultados esperados para a especificação realizada e que o programa terminará eventualmente a sua execução. Essas provas, no entanto, centram-se na verificação do código do programa. Esse é um aspecto importante de ser observado, no entanto, no caso dos algoritmos indutivos, que geram uma base de conhecimento que posteriormente poderá ser utilizada por um sistema especialista, os fatores mais importantes a serem observados estão relacionados com a base gerada. Ou seja, o importante é a qualidade, sob diversos aspectos, dos resultados apresentados.

O usuário do ISREG têm a possibilidade de escolher, via parâmetros, os aspectos a serem priorizados ou desprezados. 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** - o número de regras geradas pelo algoritmo;

- **Tamanho médio das regras** $(\overline{\text{Tam.}})$ - 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** ($\mathcal{GRBC}$) - a média aritmética corrigida do grau de relevância semântica das regras que compõem a base de conhecimento, obtida através da aplicação da equação 4.3;

- **Custo médio de classificação** $(\overline{\text{Custo}})$ - a média aritmética do custo de todas as classificações feitas para gerar a base de conhecimento (equação 4.6);

- **Freqüência de acerto** - 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.

- **Tempo** - tempo de execução do algoritmo, em segundos, medido pela utilização do comando "gprof" disponível no ambiente UNIX da SUN microsystems.

O pequeno domínio "brinquedo seguro" foi utilizado ao longo deste trabalho 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. No entanto, ele não possibilita um avaliação mais consistente das vantagens e desvantagens do novo algoritmo. Em função disso, apresentaremos os resultados encontrados pelo algoritmo ISREG para outros domínios além do "brinquedo seguro".

Para cada algoritmo estudado no capítulo 3 existe uma combinação de parâmetros do ISREG que permite fazer uma comparação entre os resultados obtidos. Essa comparação é mostrada nas tabelas de resultados apresentadas nos itens seguintes. Os resultados do ISREG que sejam superiores ou iguais àqueles apresentados pelo algoritmo com quem 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 e de generalização e solicitam a busca da qualidade máxima. Os melhores resultados obtidos pelo ISREG, levando em consideração um balanceamento entre os fatores analisados (com exceção do tempo de execução), serão mostrados nas últimas linhas de cada tabela.

6.2 O domínio "brinquedo seguro"

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. 6.1 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 6.1 - 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 em FIG. 6.2.

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.HRQ
Matriz de Relevância	- BrinqSeguro.REL
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 6.2 - Relatório padrão de saída dos algoritmos.

TABELA 6.1 - 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. 6.1 foi mostrado um resumo de todos os resultados apresentados pelos algoritmos estudados e o resultado mais significativo do algoritmo ISREG.

Ao executar o ISREG desprezando as informações de custo, qualidade semântica e generalização, *i.e.*, utilizar o algoritmo na sua forma mais simples, mesmo assim obtemos uma base de conhecimento 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 fq=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.

6.3 Outros domínios

Com o objetivo de verificar o comportamento do algoritmo ISREG diante de domínios reais e com um número de exemplos significativo, 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.

A TAB. 6.2 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 6.2 - 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

• 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, que confirmou a total relevância do atributo perfil hormonal. Não existem generalizações. Os resultados encontrados estão mostrados em TAB. 6.3.

TABELA 6.3 - Resultados dos algoritmos para o domínio "amenorréia".

Algoritmos	Parâmetros	Regras			$\mathcal{GRBC}$	$\overline{\text{Custo}}$	Freq. Acerto	Tempo (Sec.)
		Qtde	$\overline{\text{Tam.}}$	Inut.				
ID3	-	5	1.00	0%	0.82	9842.56	100%	7.99
PRISM	-	5	1.00	0%	0.82	7614.04	100%	8.72
ISREG	fe=0.0 fq=0.0	5	1.00	0%	0.82	7614.04	100%	8,72
IDRT	p=0.5	5	1.00	0%	0.82	9842.56	100%	7.86
ISREG	fe=0.0 fq=0.5	5	1.00	0%	0.82	7614.04	100%	8.14
FRPRISM	-	5	1.00	0%	0.82	7614.04	100%	8.50
ISREG	fe=0.0 fq=1.0	5	1.00	0%	0.82	7614.04	100%	8.32
EG2	fe=1.0	41	4.90	0%	0.53	5087.72	62%	8.44
ISREG	fe=1.0 fq=0.0	41	5.05	0%	0.53	5403.51	62%	9.26
ISREG	fe=0.2 fq=0.4	25	3.64	0%	0.60	4666.67	100%	8.74

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 freqüência 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 freqüência de acerto em 100%, conforme mostra a última linha da TAB. 6.3.

• 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. 6.4 contém os resultados apresentados pelos algoritmos.

TABELA 6.4 - Resultados dos algoritmos para o domínio "zoo".

Algoritmos	Parâmetros	Regras			$GRBC$	$\overline{\text{Custo}}$	Freq. Acerto	Tempo (Sec.)
		Qtde	$\overline{\text{Tam.}}$	Inut.				
ID3	-	13	2.31	0%	0.61	2.12	96%	8.19
PRISM	-	14	2.07	7%	0.57	2.03	96%	8.95
ISREG	fe=0.0 fq=0.0	14	2.07	7%	0.57	2.03	96%	9.01
IDRT	p=0.5	15	4.53	0%	0.81	7.03	96%	9.73
ISREG	fe=0.0 fq=0.5	15	2.07	6%	0.59	2.12	96%	9.50
FRPRISM	-	16	2.38	0%	0.69	3.39	100%	9.06
ISREG	fe=0.0 fq=1.0	16	2.38	0%	0.69	3.39	100%	9.62
EG2	fe=1.0	15	2.67	7%	0.58	0.00	96%	9.05
ISREG	fe=1.0 fq=0.0	13	2.23	0%	0.62	0.00	92%	9.44
ISREG	fe=0.3 fq=1.0	17	3.18	0%	0.71	1.52	100%	9.75
ISREG	fe=1.0 fq=0.7	15	3.00	0%	0.70	0.00	96%	9.74

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, no entanto essa base possui o maior custo 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, o EG2 gerou uma base de conhecimento com um nível de qualidade semântica menor e um maior tamanho médio de regras. As duas últimas linhas da tabela mostram resultados onde houve uma tentativa de equilibrar todos os índices envolvidos, eles apresentam um nível de qualidade semântica bem próximo do máximo obtido, baixos custos, além de oferecer uma freqüência de acerto dentro dos níveis apresentados pelos demais algoritmos.

- **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 aprendizado automático. Os diagnósticos apontam, ou não, a presença de doenças do coração, com base em um exame denominado

7 Conclusão

Neste capítulo apresentaremos nossas conclusões com base na avaliação global dos resultados obtidos e com as observações feitas durante a realização deste trabalho. Relacionaremos, também, algumas sugestões para trabalhos futuros.

7.1 Considerações finais

A prática de desenvolvimento de sistemas baseados em conhecimento salientou um ponto de estrangulamento nesse processo. Esse "gargalo" é a aquisição de conhecimento, tarefa desempenhada pelo engenheiro do conhecimento. A indisponibilidade de tempo, problemas de comunicação com o especialista devido ao pouco conhecimento sobre o domínio por parte do engenheiro do conhecimento, adequação de vocabulário, etc., são algumas barreiras que o processo de aquisição de conhecimento precisa transpor a cada modelagem. O aprendizado automático procura minimizar esses problemas, procurando realizar a aquisição de conhecimento através de métodos que diminuam, e por vezes eliminem, a necessidade da presença física do especialista durante o processo.

Dentre os métodos de aquisição automática de conhecimento, o aprendizado indutivo a partir de exemplos tem sido o mais explorado por pesquisadores e por isso tem recebido propostas de aperfeiçoamento. Esses métodos utilizam-se de algoritmos que a partir de um conjunto de exemplos (casos) procuram generalizar conceitos sobre o domínio em questão. Dois problemas foram identificados nesses algoritmos indutivos, um de natureza estrutural (problema sintático) e outro de natureza semântica (problema semântico). O problema sintático resulta da forma utilizada para representar o conhecimento adquirido, os métodos que apresentam esse problema sempre utilizam árvores de decisão para a representação do conhecimento. O problema semântico advém do fato dos algoritmos não levarem em consideração uma possível relação de relevância entre os elementos de classificação e os atributos do

domínio que está sendo trabalhado. E essa relação pode ser obtida através da utilização de um conhecimento adicional, denominado conhecimento preliminar.

Alguns algoritmos propostos por pesquisadores nesta área procuram minimizar a ocorrência dos problemas citados de forma isolada, *i.é.*, uns mudam a forma de representação do conhecimento adquirido, outros utilizam formas variadas de conhecimento preliminar. No entanto, nenhum algoritmo procura abordar os dois problemas de forma conjunta.

Neste trabalho apresentamos um algoritmo, o ISREG, que além de procurar minimizar a ocorrência dos problemas sintático e semântico também possibilita a utilização de forma conjunta das principais formas de conhecimento preliminar disponíveis. Definimos também, com base nos princípios do conhecimento preliminar denominado matriz de relevância, uma grandeza que permite mensurar a qualidade semântica de uma base de conhecimento.

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.

Com a utilização do grau de relevância de uma base de conhecimento foi possível verificar a importância da matriz de relevância na geração dessas bases, à medida que era solicitado ao algoritmo mais qualidade para a base gerada, o grau de relevância, como era esperado, ia aumentando de valor. Com relação a esse grau de relevância, o experimento realizado neste trabalho (APÊNDICE A) serviu, além de validar a fórmula deduzida, para comprovar a facilidade de eliciação da matriz de relevância junto aos especialistas, minimizando o problema da falta de disponibilidade de tempo desse especialista. Essa eliciação, inclusive, pode ser feita à distância, como aconteceu no caso da matriz de relevância obtida, via FAX, para o domínio "hd-disease-Cleveland" utilizado nos testes do algoritmo.

Este trabalho mostrou, também, a necessidade de dar continuidade nas pesquisas que envolvem a modelagem de um domínio, iniciada pelo projeto do ambiente A4, principalmente no que diz respeito a domínios que apresentam atributos com valores contínuos, que leva à necessidade de uma discretização desses valores.

7.2 Sugestões de trabalhos futuros

Ao atingirmos um ponto de validação de um trabalho, invariavelmente, nos deparamos com novas idéias que visam aprimorar seu estágio atual mas que esbarram em duas situações: fogem do escopo previamente definido, podendo levar a um desvio do objetivo central do trabalho, ou então, devido ao fator tempo torna-se inviável sua execução.

Em virtude da implementação deste trabalho ter ocorrido no âmbito de um ambiente de aquisição de conhecimento, os pontos apontados como sugestões de trabalhos futuros não restringem-se apenas ao escopo dos algoritmos indutivos, procuram, também, indicar pontos do ambiente utilizado que merecem um trabalho mais apurado. Dessa forma, relacionamos as seguintes tarefas:

- 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;

- Estudar a possibilidade de dotar de alguma inteligência o relatório de análise dos parâmetros, com o propósito de auxiliar o usuário na escolha da melhor combinação de fatores de avaliação da base de conhecimento gerada;

- 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;

- Incorporar ao ambiente A4 um mecanismo que permita selecionar, previamente, da tabela de exemplos os exemplos que devem compor o conjunto de teste. Atualmente essa seleção é feita de forma aleatória, ou então permitir que o conjunto de teste seja criado e informado separadamente para os algoritmos;

- Definir um procedimento para determinar, via 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 6;

- Implementar, no ambiente, mecanismos que permitam eliminar, logicamente, os atributos de uma tabela de exemplos para efeito de execução de um algoritmo;

- Disponibilizar o relatório de análise da variação dos parâmetros do algoritmo, implementado no algoritmo ISREG, para todos os algoritmos que possuam passagem de parâmetros;

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 [Mongiovi 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:

$$\Re\ (C_m, \mathcal{E}_j) \;=\; \frac{\dfrac{1}{nvg_{ik}} \sum_{k=1}^{nvg_{ik}} \mathcal{F}_{ijk}}{1+\sigma^2} \qquad \text{(A.1)}$$

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 $G(a_i) = \{g_{i1}, g_{i2}, g_{i3}, \ldots, g_{ing}\}$, sendo $0 \leq ng_i < nv_i$;

$\mathcal{E}_j$ é um elemento de classificação pertencente ao conjunto de elementos de classificação $\mathcal{E} = \{\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \ldots, \mathcal{E}_{n_{\mathcal{E}}}\}$;

nv_i é o número de valores do atributo a_i;

$\mathcal{F}_{ijk}$ é uma função que fornece a relevância do atributo a, para o elemento de classificação $\mathcal{E}_j$ no valor k;

σ^2 é a variância da média das relevâncias dos valores que compõem a generalização, calculada conforme a equação 4.4.

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, Joguem Minha Mãe do trem, Corra que Policia 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 Policia 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 que fornece o mesmo grau de relevância (A.1) 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 dos gráficos e das tabelas, utilizadas 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) Infanto-juvenil (07-11)	G1f	G1e	Infantil (01-11)
	Sênior (40-59) Velho (60-79)	G2f	G2e	Maduro (40-79)
Nível de instrução	Primário Ginásio Científico	G3f	G3e	Médio
	Graduado Pós-graduado	G4f	G4e	Superior

A codificação da generalização segue um formato G*n*X 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 estão mostrados em TAB. A.3 e GRAF. A.1.

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

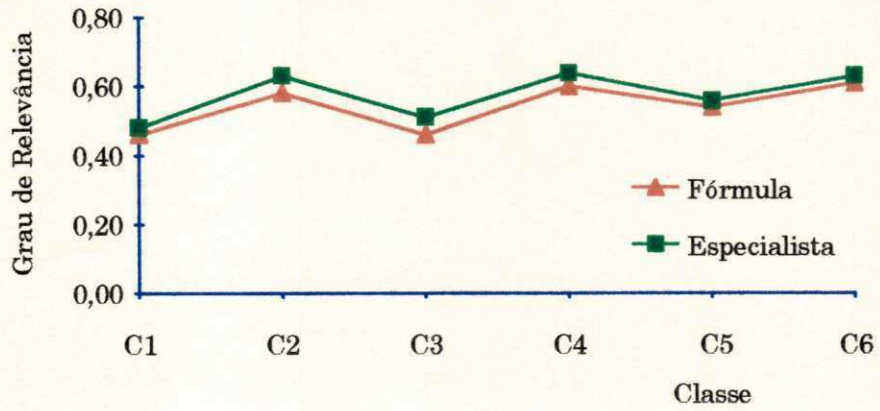

GRÁFICO A.1 - Resultado final do experimento

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:

- a diferença máxima encontrada entre as médias finais foi de 0,05 o que representa uma margem de erro de 5%, valor que consideramos aceitável;

- 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.

Na seqüência apresentaremos os dados de cada especialista mostrando sempre os valores em uma tabela e um gráfico para cada generalização analisada.

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

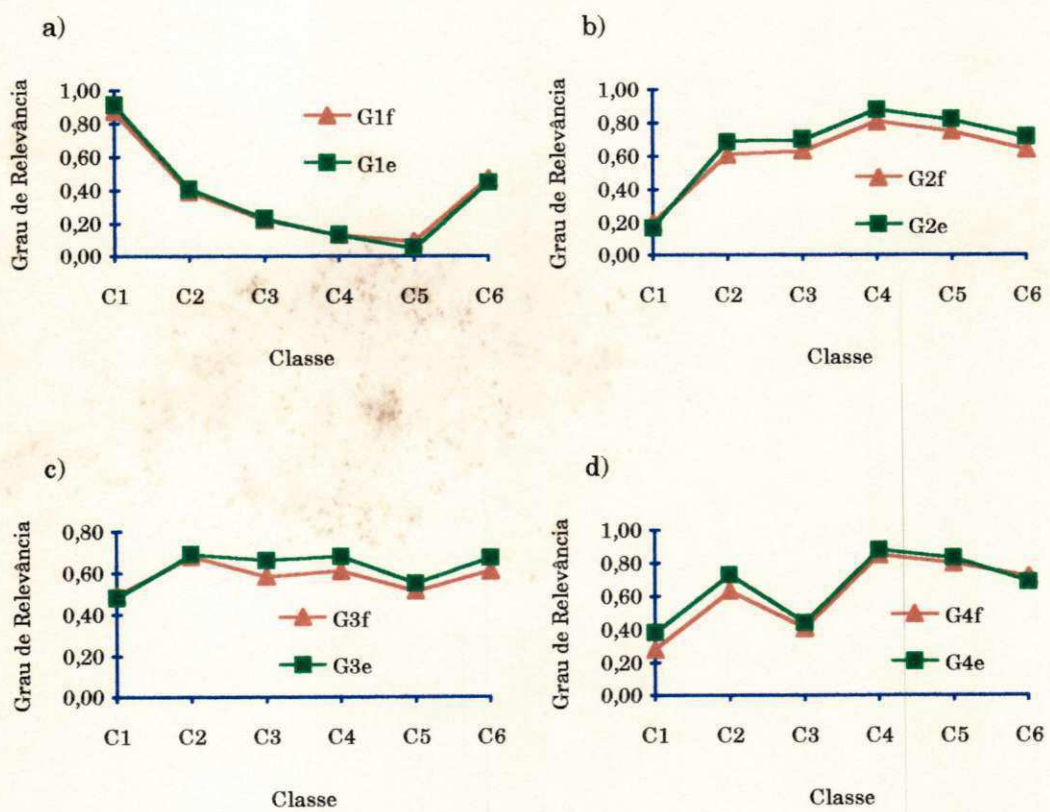

GRÁFICO A.2 - Resultados por generalização.

TABELA A.5 - Resultados do especialista 1.

Generalização	Elemento de Classificação (Classe)					
	C1	C2	C3	C4	C5	C6
G1f	1.00	0.29	0.00	0.00	0.00	0.34
G1e	1.00	0.10	0.00	0.00	0.00	0.10
G2f	0.00	0.37	0.80	0.83	0.79	0.75
G2e	0.00	0.70	0.90	1.00	1.00	0.70
G3f	0.32	0.60	0.77	0.63	0.63	0.63
G3e	0.50	0.60	0.75	0.50	0.50	0.50
G4f	0.00	0.70	0.65	0.85	0.85	0.85
G4e	0.00	0.70	0.60	1.00	1.00	0.70

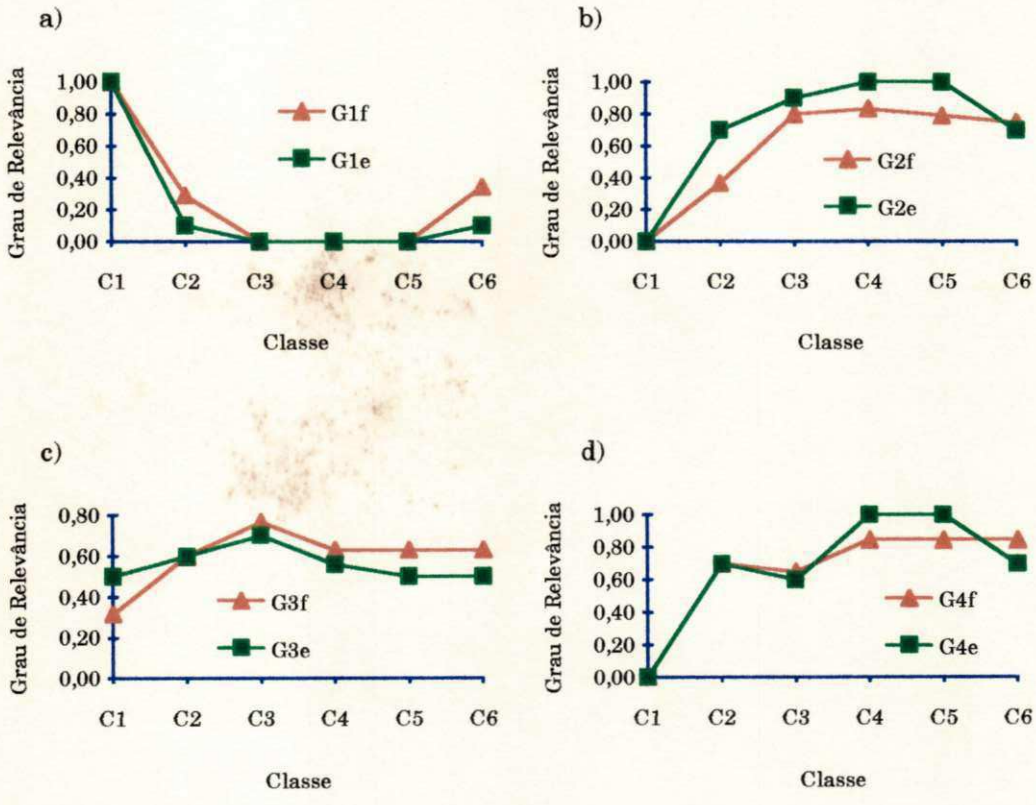

GRÁFICO A.3 - Resultados do especialista 1.

TABELA A.6 - Resultados do especialista 2.

Generalização	Elemento de Classificação (Classe)					
	C1	C2	C3	C4	C5	C6
G1f	0.85	0.83	0.24	0.15	0.15	0.65
G1e	0.80	1.00	0.00	0.20	0.20	0.50
G2f	0.00	0.90	0.60	0.90	0.85	0.75
G2e	0.00	1.00	0.70	1.00	0.90	0.90
G3f	0.20	0.62	0.40	0.89	0.62	0.69
G3e	0.00	1.00	0.60	1.00	0.70	0.90
G4f	0.00	0.90	0.65	1.00	0.95	0.95
G4e	0.00	1.00	0.70	1.00	0.90	0.90

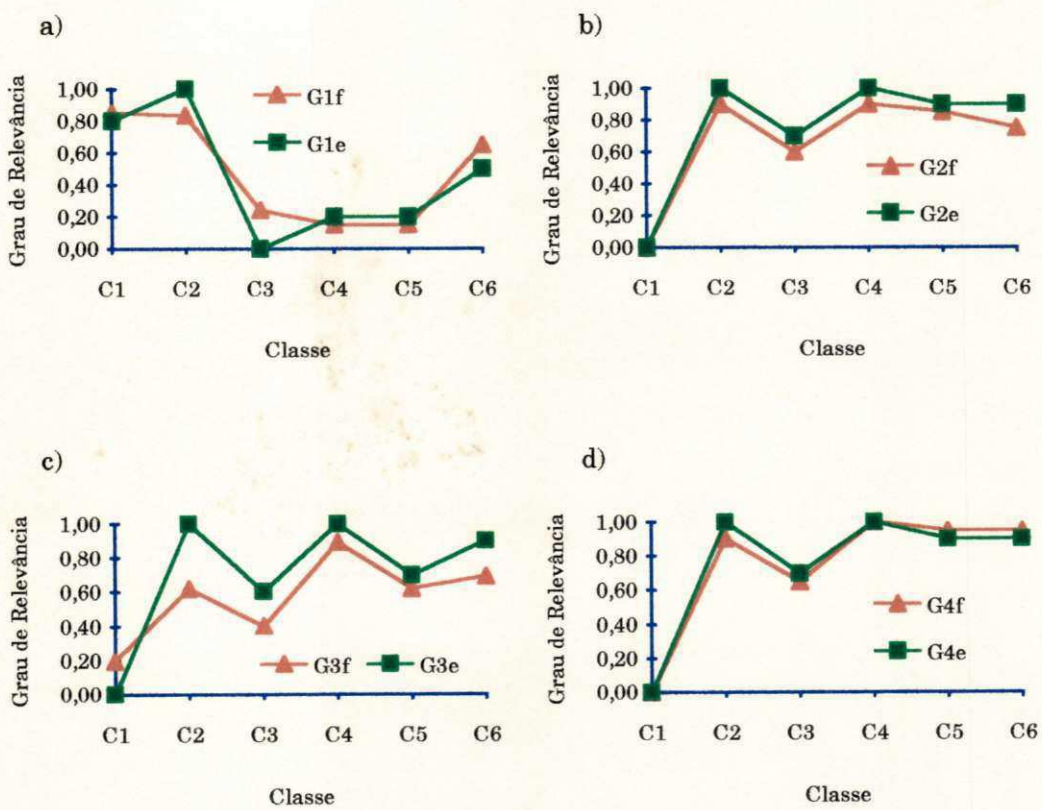

GRÁFICO A.4 - Resultados do especialista 2.

TABELA A.7 - Resultados do especialista 3.

Generalização	Elemento de Classificação (Classe)					
	C1	C2	C3	C4	C5	C6
G1f	0.75	0.48	0.19	0.15	0.10	0.49
G1e	0.90	0.60	0.50	0.10	0.10	0.80
G2f	0.50	0.69	0.60	0.80	0.80	0.75
G2e	0.20	0.60	0.70	0.80	0.80	0.80
G3f	0.60	0.80	0.73	0.60	0.50	0.56
G3e	0.60	0.90	0.80	0.60	0.50	0.60
G4f	0.60	0.80	0.45	0.80	0.90	0.90
G4e	0.70	0.90	0.60	0.80	0.90	0.70

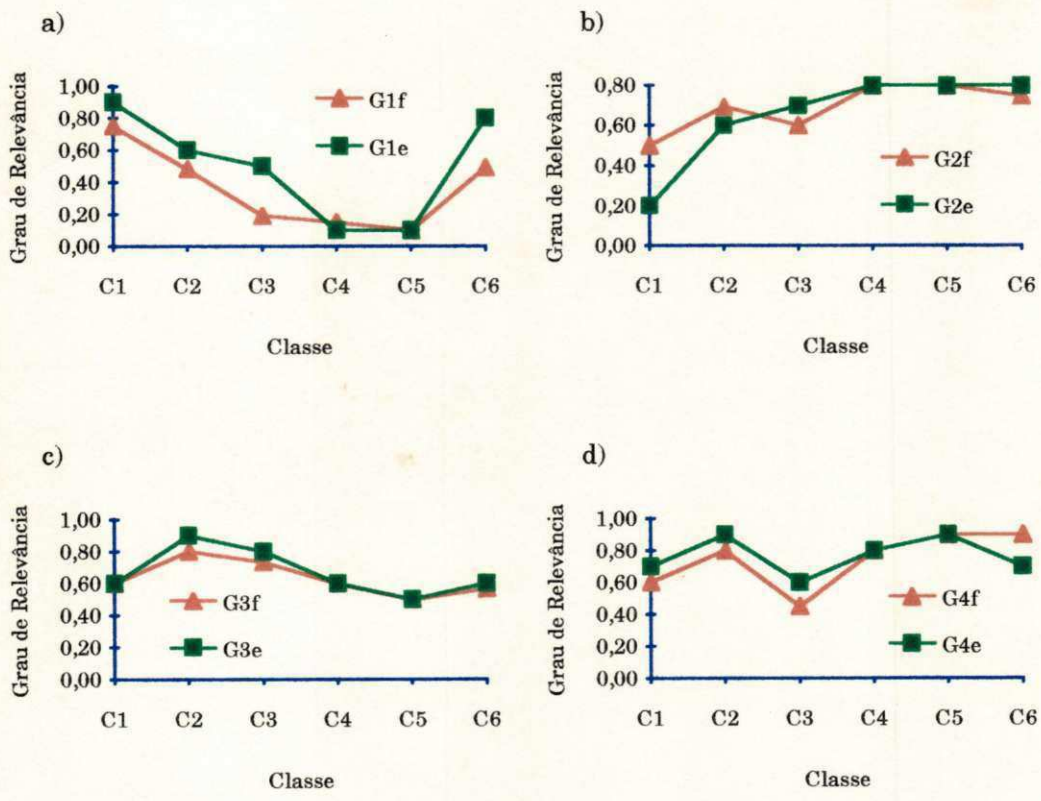

GRÁFICO A.5 - Resultados do especialista 3.

TABELA A.8 - Resultados do especialista 4.

Generalização	Elemento de Classificação (Classe)					
	C1	C2	C3	C4	C5	C6
G1f	0.79	0.20	0.10	0.10	0.15	0.71
G1e	0.90	0.10	0.20	0.10	0.10	0.80
G2f	0.20	0.44	0.55	0.64	0.64	0.55
G2e	0.20	0.60	0.70	0.90	0.70	0.90
G3f	0.40	0.53	0.53	0.48	0.56	0.63
G3e	0.10	0.10	0.80	0.40	0.40	0.70
G4f	0.15	0.35	0.25	0.80	0.75	0.55
G4e	0.10	0.10	0.40	0.80	0.58	0.80

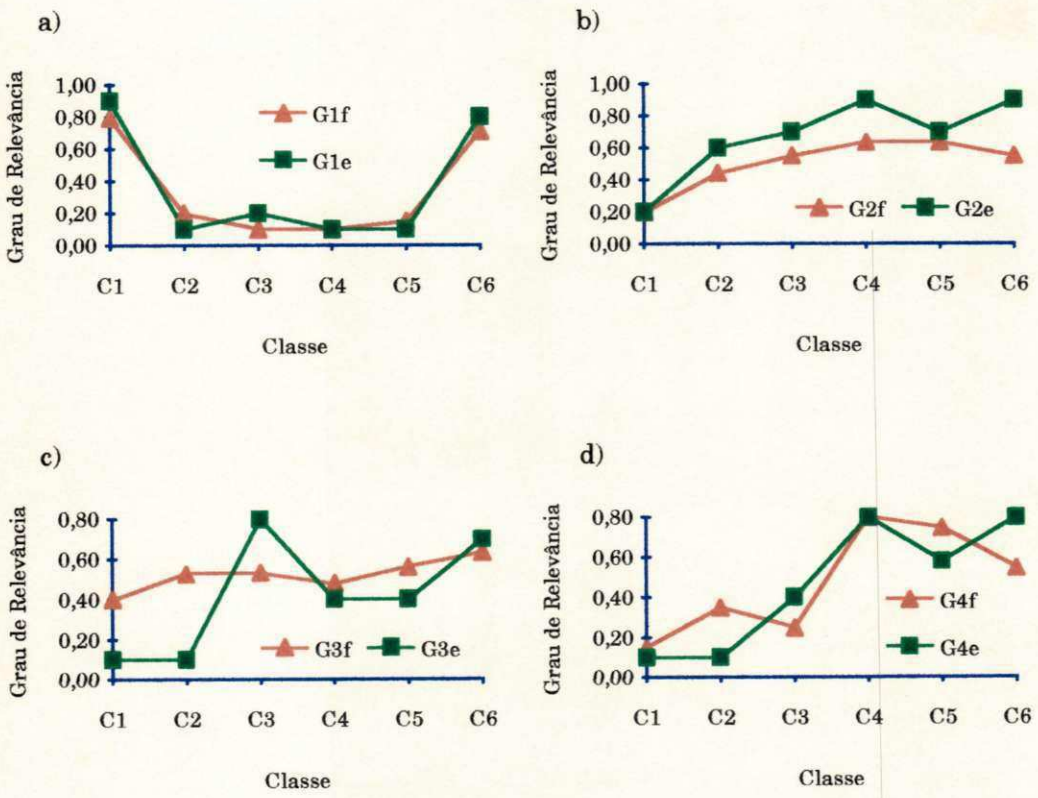

GRÁFICO A.6 - Resultados do especialista 4.

TABELA A.9 - Resultados do especialista 5.

Generalização	Elemento de Classificação (Classe)					
	C1	C2	C3	C4	C5	C6
G1f	0.89	0.37	0.39	0.15	0.05	0.20
G1e	1.00	0.30	0.30	0.20	0.00	0.20
G2f	0.25	0.40	0.79	0.69	0.35	0.50
G2e	0.30	0.50	0.80	0.60	0.40	0.50
G3f	0.53	0.67	0.67	0.50	0.23	0.33
G3e	0.40	0.50	0.60	0.50	0.20	0.40
G4f	0.25	0.49	0.55	0.85	0.45	0.50
G4e	0.20	0.50	0.50	0.90	0.70	0.50

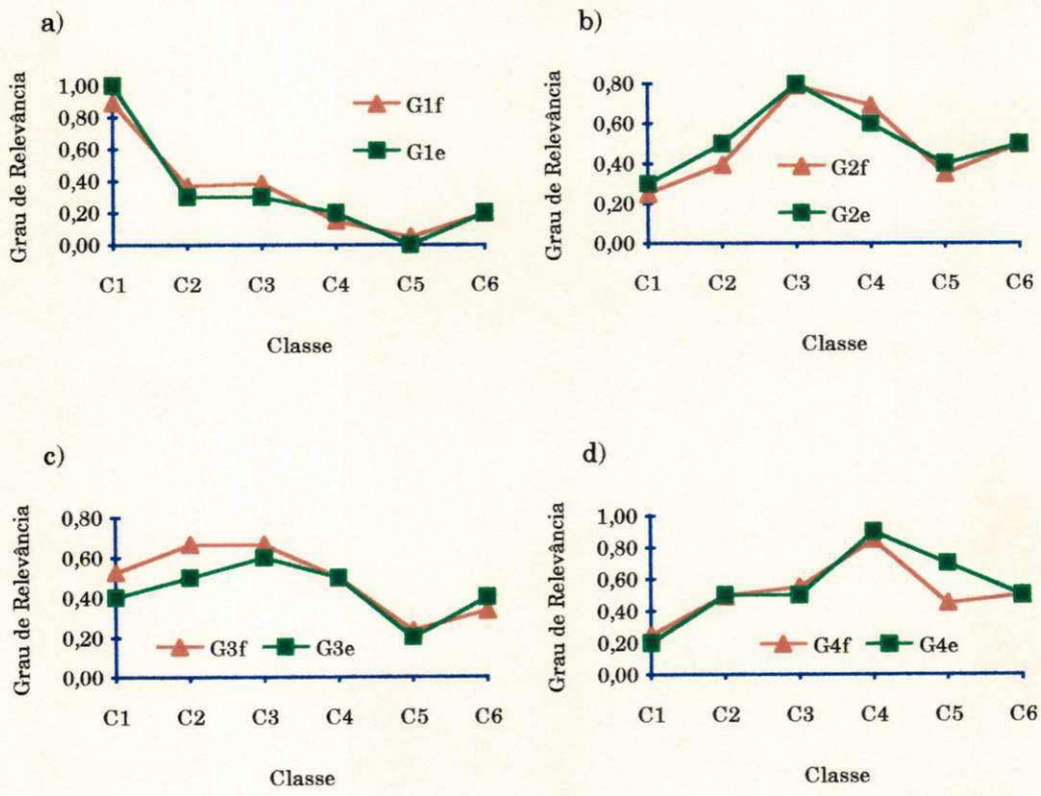

GRÁFICO A.7 - Resultados do especialista 5.

TABELA A.10 - Resultados do especialista 6.

Generalização	Elemento de Classificação (Classe)					
	C1	C2	C3	C4	C5	C6
G1f	0.64	0.59	0.05	0.20	0.20	0.50
G1e	0.70	0.70	0.00	0.30	0.00	0.50
G2f	0.05	0.50	0.10	0.90	0.90	0.50
G2e	0.00	0.50	0.10	0.90	0.90	0.50
G3f	0.17	0.63	0.10	0.53	0.53	0.50
G3e	0.20	0.70	0.20	0.90	0.50	0.50
G4f	0.00	0.40	0.10	0.90	0.90	0.40
G4e	0.00	0.50	0.10	0.90	0.90	0.50

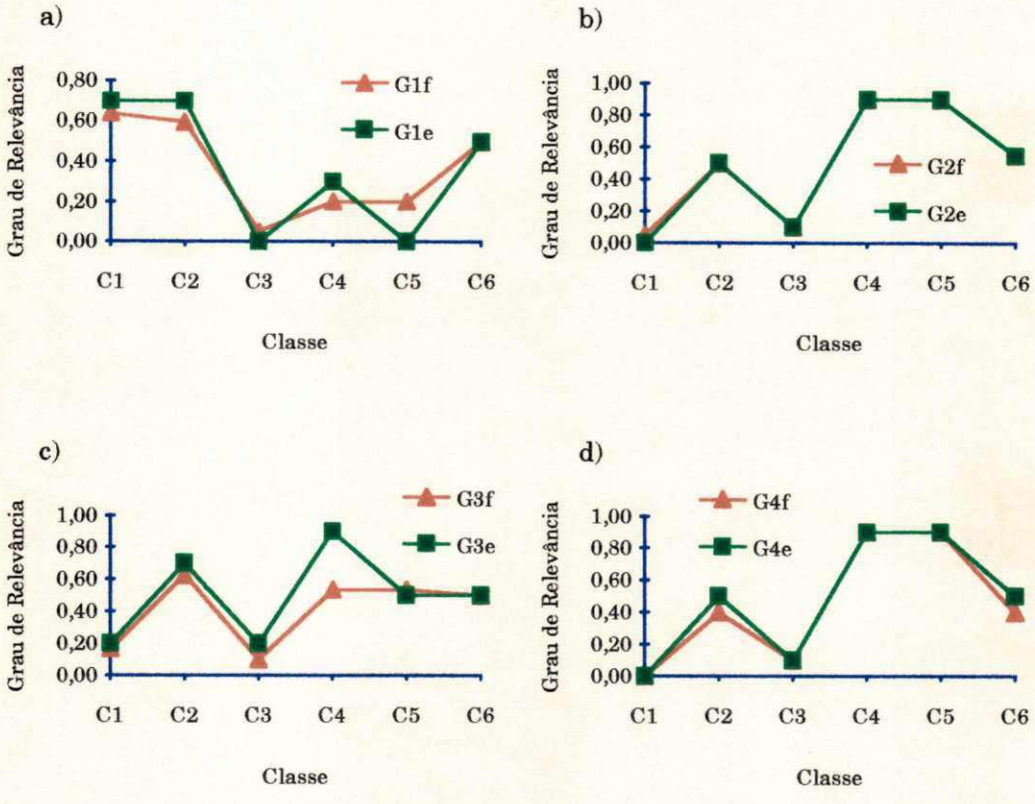

GRÁFICO A.8 - Resultados do especialista 6.

TABELA A.11 - Resultados do especialista 7.

Generalização	Elemento de Classificação (Classe)					
	C1	C2	C3	C4	C5	C6
G1f	1.00	0.15	0.15	0.00	0.00	0.23
G1e	1.00	0.30	0.00	0.00	0.00	0.50
G2f	0.20	0.90	0.85	0.75	0.75	0.80
G2e	0.20	0.80	0.80	0.90	0.80	0.80
G3f	0.00	0.20	0.10	0.33	0.10	0.20
G3e	0.70	0.90	0.70	0.50	0.80	0.70
G4f	0.00	0.20	0.10	0.69	0.10	0.20
G4e	0.70	0.90	0.70	0.90	0.80	0.70

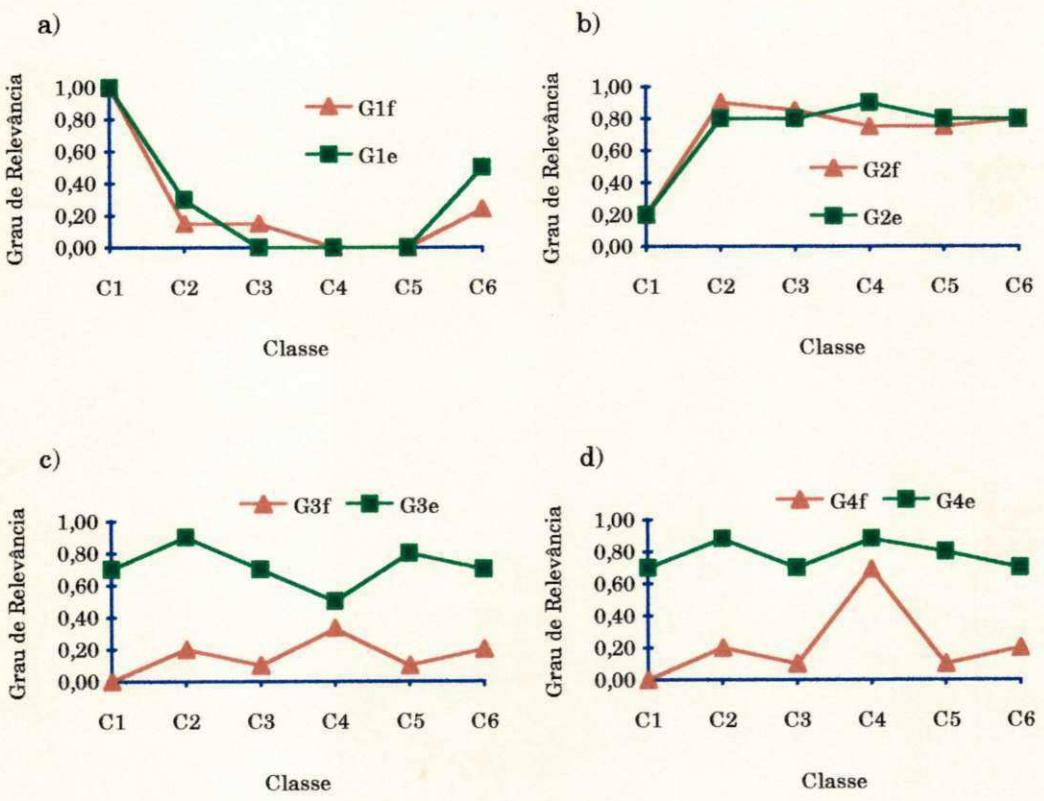

GRÁFICO A.9 - Resultados do especialista 7.

TABELA A.12 - Resultados do especialista 8.

Generalização	Elemento de Classificação (Classe)					
	C1	C2	C3	C4	C5	C6
G1f	1.00	0.19	0.52	0.19	0.00	0.71
G1e	1.00	0.20	0.60	0.20	0.00	0.50
G2f	0.75	0.89	0.75	0.89	0.83	0.95
G2e	0.70	1.00	0.70	1.00	1.00	1.00
G3f	0.93	0.93	0.96	0.84	0.80	0.73
G3e	1.00	1.00	0.90	1.00	1.00	0.80
G4f	0.90	0.80	0.70	0.69	1.00	0.90
G4e	1.00	1.00	0.50	0.60	1.00	0.60

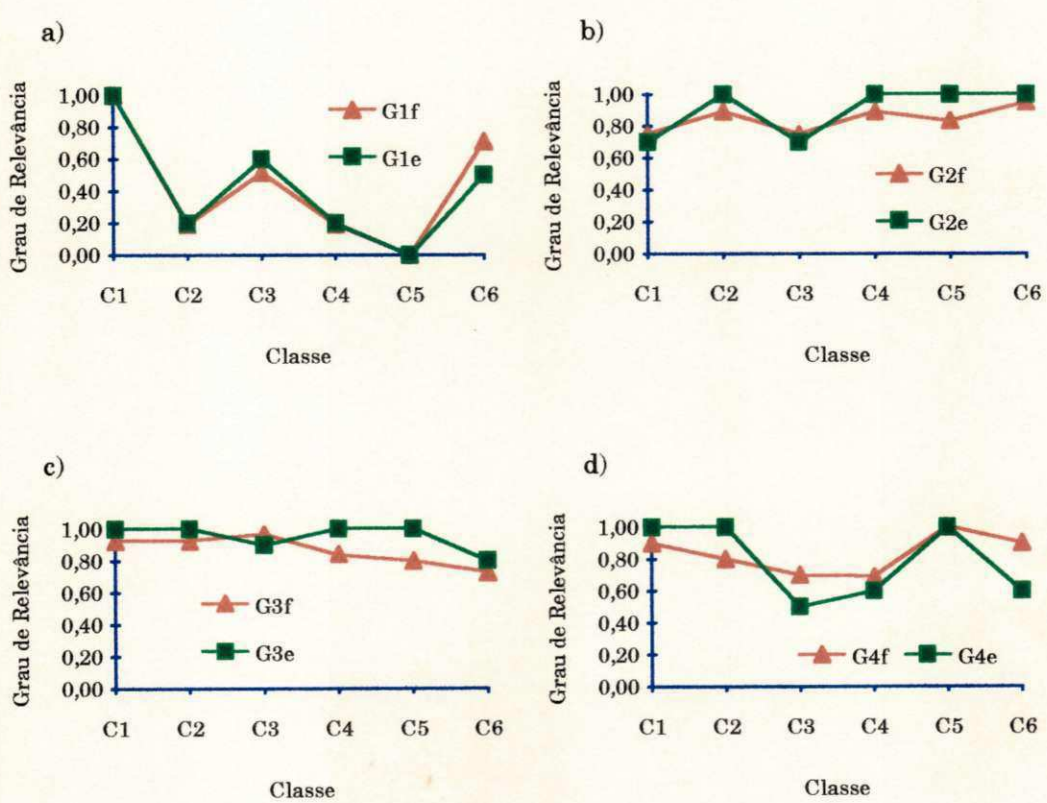

GRÁFICO A.10 - Resultados do especialista 8.

TABELA A.13 - Resultados do especialista 9.

Generalização	Elemento de Classificação (Classe)					
	C1	C2	C3	C4	C5	C6
G1f	1.00	0.58	0.42	0.30	0.30	0.77
G1e	1.00	0.60	0.50	0.20	0.20	0.90
G2f	0.20	0.80	0.90	1.00	1.00	0.64
G2e	0.20	0.90	1.00	1.00	1.00	0.80
G3f	0.71	1.00	0.83	0.83	0.83	0.93
G3e	0.80	0.80	1.00	0.90	0.70	0.90
G4f	0.25	0.80	0.35	1.00	1.00	0.90
G4e	0.40	0.90	0.50	1.00	0.80	0.80

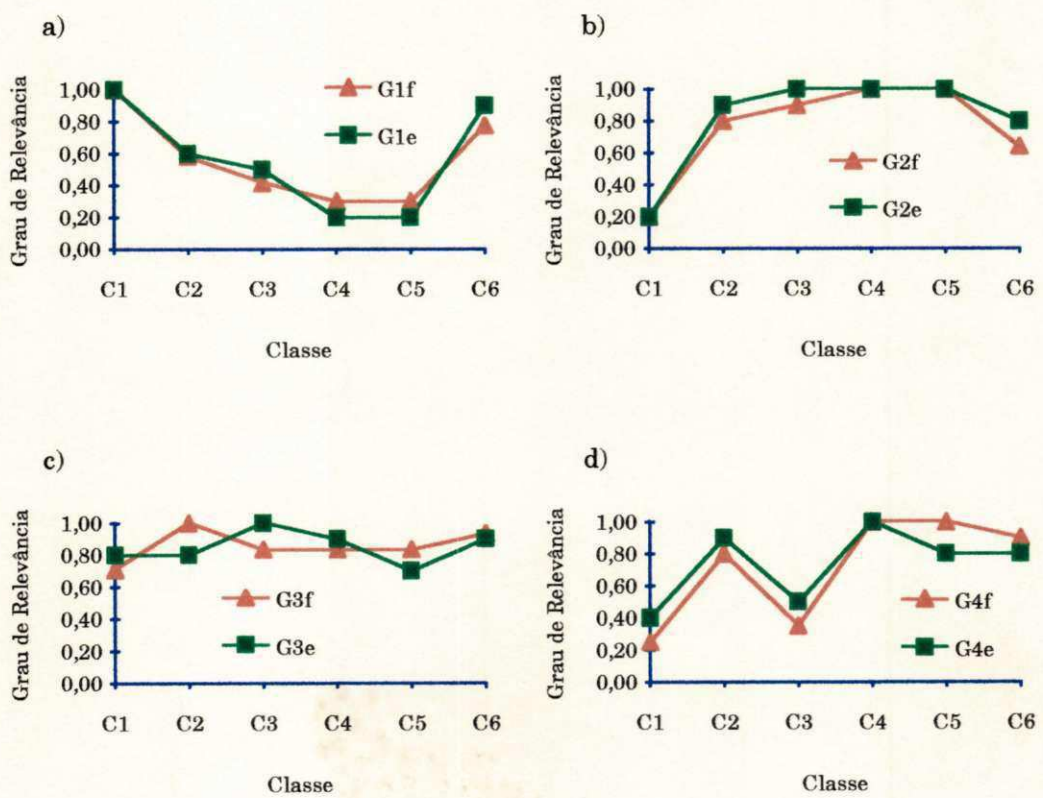

GRÁFICO A.11 - Resultados do especialista 9.

TABELA A.14 - Resultados do especialista 10.

Generalização	Elemento de Classificação (Classe)					
	C1	C2	C3	C4	C5	C6
G1f	0.69	0.29	0.09	0.05	0.00	0.15
G1e	0.80	0.30	0.00	0.00	0.00	0.00
G2f	0.10	0.40	0.65	0.90	0.70	0.64
G2e	0.00	0.60	0.80	0.90	0.90	0.70
G3f	0.77	0.80	0.53	0.46	0.43	0.66
G3e	0.50	0.70	0.30	0.60	0.40	0.60
G4f	0.20	0.50	0.20	0.85	0.95	0.80
G4e	0.50	0.80	0.10	0.90	0.90	0.70

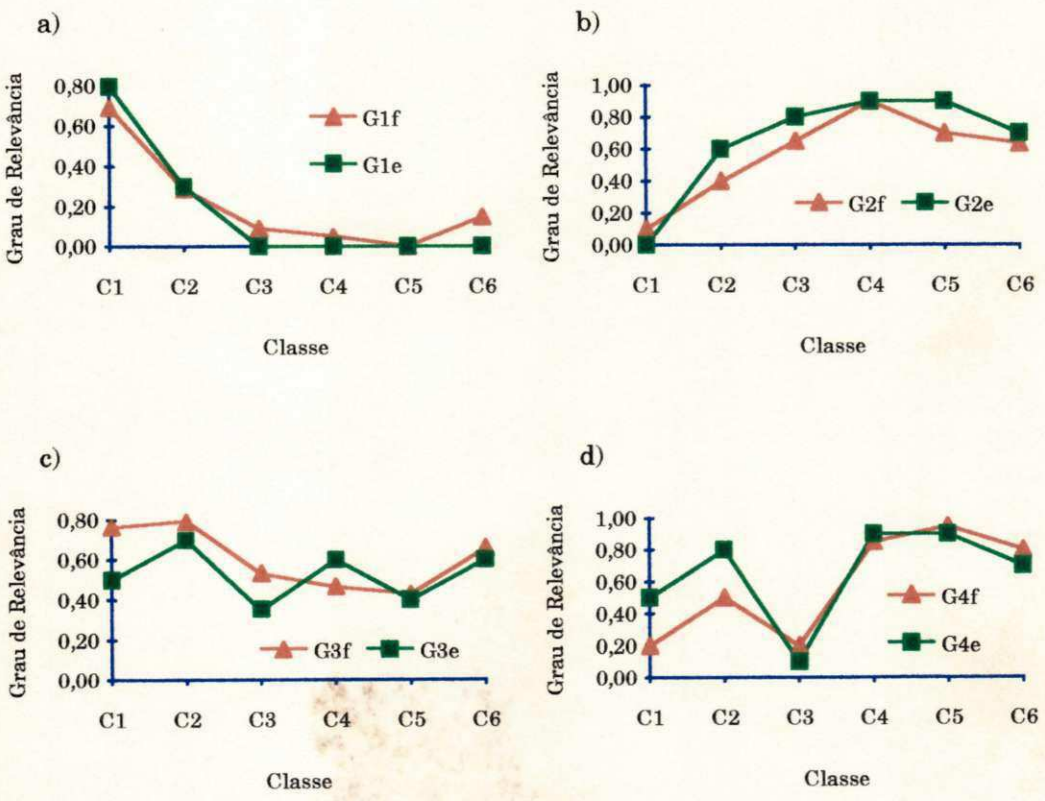

GRÁFICO A.12 - Resultados do especialista 10.

TABELA A.15 - Resultados do especialista 11.

Generalização	Elemento de Classificação (Classe)					
	C1	C2	C3	C4	C5	C6
G1f	0.85	0.24	0.30	0.20	0.05	0.54
G1e	0.80	0.20	0.10	0.00	0.00	0.50
G2f	0.05	0.49	0.64	0.75	0.69	0.59
G2e	0.10	0.40	0.70	0.80	0.60	0.50
G3f	0.61	0.77	0.73	0.67	0.48	0.77
G3e	0.40	0.50	0.60	0.70	0.40	0.80
G4f	0.60	0.75	0.45	0.90	0.85	0.80
G4e	0.20	0.60	0.20	0.90	0.70	0.50

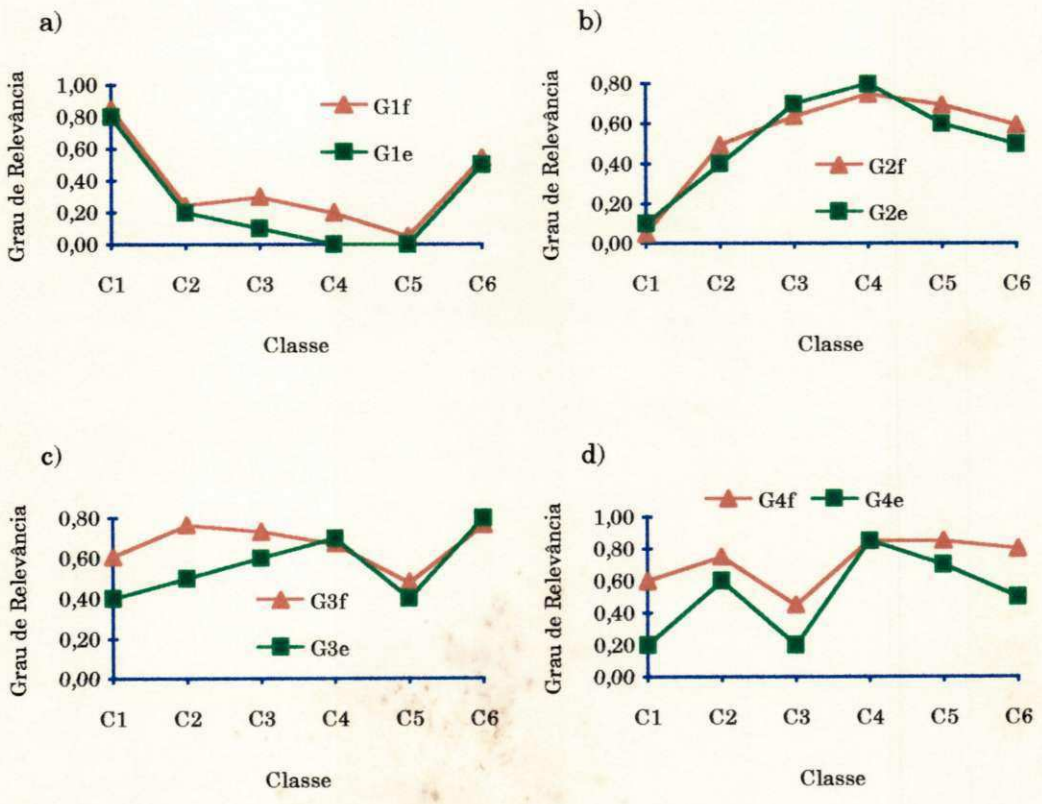

GRÁFICO A.13 - Resultados do especialista 11.

TABELA A.16 - Resultados do especialista 12.

Generalização	Elemento de Classificação (Classe)					
	C1	C2	C3	C4	C5	C6
G1f	1.00	0.49	0.24	0.05	0.05	0.33
G1e	1.00	0.50	0.50	0.20	0.00	0.10
G2f	0.10	0.59	0.29	0.69	0.69	0.28
G2e	0.10	0.70	0.50	0.80	0.80	0.50
G3f	0.70	0.74	0.63	0.53	0.46	0.68
G3e	0.50	0.70	0.70	0.60	0.50	0.60
G4f	0.45	0.90	0.44	0.90	0.90	0.90
G4e	0.70	0.90	0.40	0.90	0.90	0.90

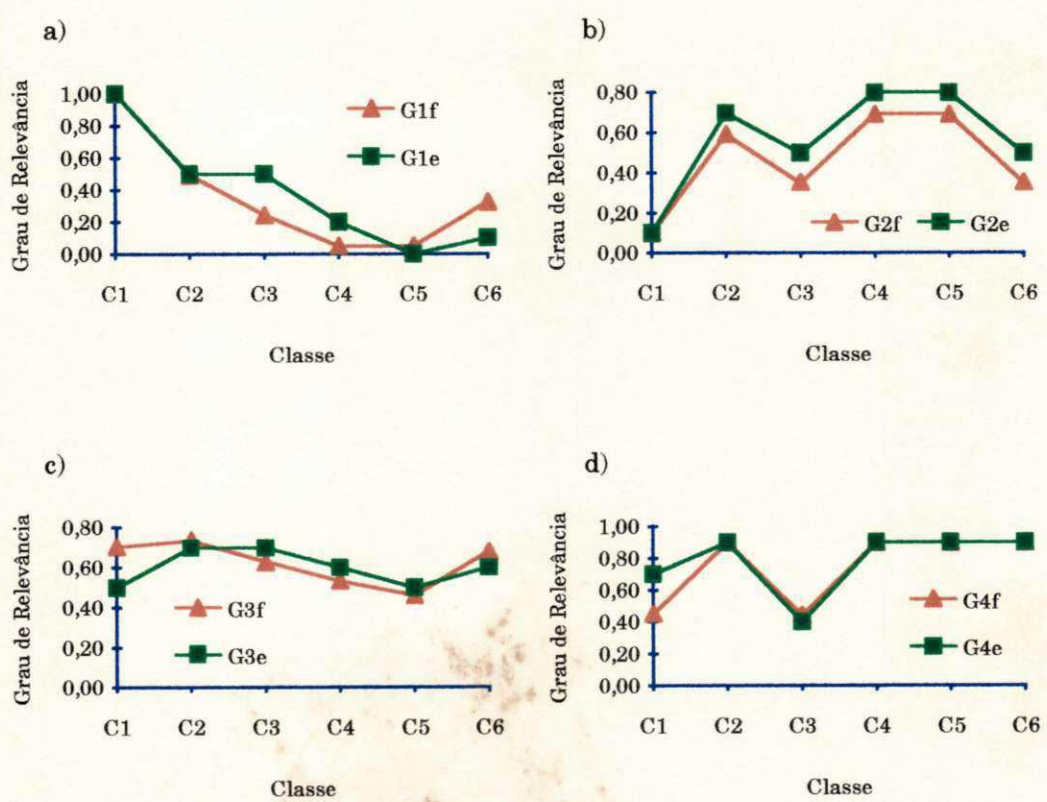

GRÁFICO A.14 - Resultados do especialista 12.

APÊNDICE B - O algoritmo ISREG

Apresentaremos a seguir o código fonte do algoritmo ISREG, este código é uma sub-classe da classe algoritmos do ambiente A4.

========== Início do A4ISREG.H

```
#ifndef A4_HEADER_ISREG
#define A4_HEADER_ISREG

class ISREG : public AlgoritmoRegular
{
      public:
       void executa();

//        VETOR DE GENERALIZAÇÕES DISCARTADAS
       static VetGenDisc[QTDEMAXGEN];

//        ESTRUTURA QUE COMUNICA OS VALORES CALCULADOS PELA FUNÇÃO DE
//        CUSTO RELEVANTE DA INFORMAÇÃO (FCRI)
       typedef struct fcri {
                     float result;
                     int   at;
                     int   gen;
                     int   vlr;
                     float freq;
                     float nvgen;
                     float relev;
       };
       fcri FCRI[1];

//        ESTRUTURA QUE GUARDA O MENOR FCRI
       typedef struct menor_fcri {
                     float result;
                     int   at;
                     int   gen;
                     int   vlr;
                     float freq;
                     float nvgen;
                     float relev;
       };
       menor_fcri MenorFCRI[1];
```

```
//      LISTA DE GENERALIZACOES E VALORES OBSERVADOS
      typedef struct lista_gv {
                  int gen;
                  int vlr;
      };

//      MACRO PARA CALCULO DO MÁXIMO ENTRE TRES NÚMEROS
#define      _MAX3(x,y,z)  ((x) > (y) ? 1 : ((x) > (z) ? 1 :((y) > (z) ? 2 : ((z) >
(y) ? 3 : 0))))

//      MÉTODOS
      void      SetaParametros ( );
      void      InicializaAmbiente ( );
      void      GeraRegras ( );
      void      SelecionaMelhorCondicao ( condicao *, int );
      void      FCRIGeneralizacao ( fcri *, int, int );
      void      FCRIValor ( fcri *, int, int, int );
      void      GeraSubConjunto ( condicao * );
      int       GenCompleta ( condicao *, int );
      void      ReinicializaVetores ( );
      void      RetornaSubConjunto ( condicao * );
      void      TrataConflito ( condicao *, int );
      int       RetiraExemplosMapeados ( condicao * );
      void      RetornaSubConjuntosRegra ( );
      void      GravaTabRegras ( );
      void      GravaDadosAnalise ( int, float, float );
      void      GravaResumo ( );
};
#endif
```

========== **Início do A4ISREG.C**

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <xview/xview.h>
#include <xview/panel.h>
#include <xview/xv_xrect.h>
#include <gdd.h>
#include "A4_ui.H"
#include "A4define.H"
#include "A4exem.H"
#include "A4hiera.H"
#include "A4rel.H"
#include "A4conf.H"
#include "A4alg.H"
#include "A4algreg.H"
#include "A4ISREG.H"

extern A4_pop_inclui_algoritmo_objects  A4_pop_inclui_algoritmo;
extern A4_pop_param_algoritmos_objects  A4_pop_param_algoritmos;
extern A4_window1_objects               A4_window1;
extern Xv_opaque vet_valores_param[];
extern nome_alg;
extern Configuracao *conf;
extern Exemplo *ex;
extern Hierarquia *hrq;
extern Relevancia *rel;
```

```
void ISREG::executa ( )                    // método inicial obrigatório
{
     if ( StatusParametros == 1 ) {        // controle da solicitação
         SetaParametros ( );               // dos parâmetros
         return;
     }

     SelecionaExemplosTeste ( );

     // A forma de execução do algoritmo é controlada pelo parâmetro de solici-
     // tação da análise de cobinação dos valores dos parâmetros, ou seja:
     // 0 = não gera o relatório, gerando a base de conhecimento conforme os
     //     os parâmetros informados;
     // 1 = gere o relatório variando o parâmetro fe conforme o valor informado
     // 2 = gere o relatório variando o parâmetro fq conforme o valor informado
     // 3 = gere o relatório variando os parâmetros fe e fq conforme os valoreo
     //     informado;
     // ---------------------------------------------------------------------
     int exec = 0; int tot_exec = 0; int nexe_fe; int nexe_fq; int status = 0;
     float incr_fe = 0; float incr_fq = 0; float inic_fe = 0; float inic_fq =
     char msg [30];
     float opcao = lista_parametros[3].valor;
     switch ( (int) lista_parametros[3].valor ) {
         case 1:
              inic_fe = 0;
              incr_fe = lista_parametros[0].valor;
              incr_fq = 0;
              inic_fq = lista_parametros[1].valor;
              break;
         case 2:
               incr_fe = 0;
               inic_fe = lista_parametros[0].valor;
               inic_fq = 0;
               incr_fq = lista_parametros[1].valor;
               break;
         case 3:
               inic_fe =0;
               incr_fe = lista_parametros[0].valor;
               inic_fq =0;
               incr_fq = lista_parametros[1].valor;
      }
      if ( opcao > 0 ) {
         // Calcula quantidade de execuções para informar seu andamento
         // A soma do valor 0.001 é feita para forçar um arredondamento
         // em duas casas decimais no formato float
         if ( incr_fe > 0 )
            nexe_fe = (int) ((1/incr_fe+0.001) + 1);
         else nexe_fe = 1;
         if ( incr_fq > 0 )
            nexe_fq = (int) ((1/incr_fq+0.001) + 1);
         else nexe_fq = 1;
         if ( rel->arq_carregado )
            tot_exec = nexe_fe * nexe_fq;
         else tot_exec = nexe_fe;
         for ( float e=0; e < nexe_fe; e++ ) {       // busca economia
             lista_parametros[0].valor = e*incr_fe+inic_fe;
             for ( float p=0; p < nexe_fq; p++ ) {  // busca qualidade
                 lista_parametros[1].valor = p*incr_fq+inic_fq;
                 sprintf (msg, "Aguarde.  Execução %d de %d", ++exec,
                                                              tot_exec)
```

```
                xv_set(A4_pop_inclui_algoritmo.pop_inclui_algoritmo,
                        FRAME_LEFT_FOOTER, msg, NULL);
                InicializaAmbiente ( );
                GeraRegras ( );
                GravaDadosAnalise (status, e*incr_fe+inic_fe,
                                           p*incr_fq+inic_fq)
                if ( !rel->arq_carregado ) // sem matriz de relevância não
                    break;                 // tem sentido buscar qualidade
                status = 2;
            }
            status = 1;
        }
    }
    else {
        // Gera regras c/base nos parâmetros informados
        InicializaAmbiente ( );
        GeraRegras ( );
        GravaRegras ( );
        GravaResumo ( );
    }
    return;
}// fim_executa


// Este método indica os valores sugeridos para os parâmetros solicitados pelo
// algoritmo
void ISREG::SetaParametros ( )
{
    lista_parametros[0].valor = 1.0;
    lista_parametros[0].limite_inferior = 0.0;
    lista_parametros[0].limite_superior = 1.0;
    sprintf(lista_parametros[0].descricao,
        "Fator de Economia (fe = [0,1] onde: 1=economia máxima)");
    xv_set(vet_valores_param[0], XV_HELP_DATA, "A4alg:ISREG_fe", NULL);
    lista_parametros[1].valor = 0.5;
    lista_parametros[1].limite_inferior = 0.0;
    lista_parametros[1].limite_superior = 1.0;
    sprintf(lista_parametros[1].descricao,
        "Fator de Qualidade (fq = [0,1] onde: 1=qualidade máxima)");
    xv_set(vet_valores_param[1], XV_HELP_DATA, "A4alg:ISREG_fq", NULL);
    lista_parametros[2].valor = 1.0;
    lista_parametros[2].limite_inferior = 0.0;
    lista_parametros[2].limite_superior = 1.0;
    sprintf(lista_parametros[2].descricao,
        "Limiar de Generalização (lg = [0,1])");
    xv_set(vet_valores_param[2], XV_HELP_DATA, "A4alg:ISREG_lg", NULL);
    lista_parametros[3].valor = 0;
    lista_parametros[3].limite_inferior = 0;
    lista_parametros[3].limite_superior = 1;
    sprintf(lista_parametros[3].descricao,
        "Opcão do relatório de análise dos parâmetros");
    xv_set(vet_valores_param[3], XV_HELP_DATA, "A4alg:ISREG_ra", NULL);
    QtdeParametros = 4;
    TrataParametros ( );
    return;

}// fim_SetaParametros
```

```
void ISREG::InicializaAmbiente ( )
{
     // Inicializa vetores de trabalho
     for ( int i=0; i < ex->QualQtdeAtributos ( ); i++ )
         VetAtSel[i] = 0;
     for ( i=0; i < hrq->QtdeGeneralizacoes; i++ )
         VetGenDisc[i] = 0;
     for ( i=0; i < ex->QualQtdeValorTot ( ); i++ )
         VetVlrDisc[i] = 0;

     InicializaTabRegras ( );
     InicializaTabRegrasConflito ( );
     HouveConflito = 0;        // flag de controle da existência de conflito na base

     return;

}// fim_InicializaAmbiente



void ISREG::GeraRegras ( )
{
     for ( int cl=0; cl < ex->QualQtdeClasse (); cl++ ) {
         while ( FreqClasse (cl) > 0 ) {
             do {
                 SelecionaMelhorCondicao (Cond, cl);
                 GeraSubConjunto (Cond);
                 if ( GenCompleta (Cond, cl) ) {
                     TabRegras[r][p].at    = Cond->at;
                     TabRegras[r][p].gen   = Cond->gen;
                     TabRegras[r][p++].vlr = Cond->vlr;
                 }
                 else RetornaSubConjunto (Cond);
             } while ( AtualQtdeClasse ( ) > 1   &&   !HouveConflito );
             TrataConflito (Cond, cl);
             TabRegras[r][--p].classe = cl;
             TabRegras[r][p].nexmap = RetiraExemplosMapeados (Cond);
             RetornaSubConjuntosRegra ( );
             r++; p = 0;              // Prepara ponteiros p/nova regra
         }
         RestauraConjTreinamento ( );
      }
      return;

}// fim_GeraRegras



// A melhor condição é formada pelo par atributo/valor ou atributo/generalização
// e é escolhida por apresentar a menor Função de Custo Relevante da Informação.
// Em caso de empate entre duas condições, foram adotados os seguintes critérios
// de desempate, por ordem de prioridade:
// 1. uma generalização é sempre melhor que um valor;
// 2. a generalização com mais valores presentes no conjunto de treinamento;
// 3. segue uma prioridade variavel estabelecida pelos valores informados nos
//    parâmetros (custo, qualidade, tamanho);
// 4. a primeira condição.
```

```
void ISREG::SelecionaMelhorCondicao ( condicao *Cond, int classe )
{
     MenorFCRI->result = MAXFLOAT;      // guarda a menor FCRI ocorrido
     MenorFCRI->at     = -1;            // guarda o atributo do menor FCRI ocorrido
     MenorFCRI->gen    = -1;    // guarda a generalização do menor FCRI ocorrido
     MenorFCRI->vlr    = -1;            // guarda o valor do menor FCRI ocorrido
     int selecionado = 0;               // flag de controle dos desempates

     Cond->at = Cond->gen = Cond->vlr = -1;

     // Testa a ocorrência de conflito (contra-exemplos)
     int qtde_at = ex->QualQtdeAtributos ( );
     for ( int at=0; at < qtde_at  &&  VetAtSel[at] == 1; at++ );
     if ( at == qtde_at ) {
         HouveConflito = 1;
         return;
     }

     for ( at=0; at < ex->QualQtdeAtributos ( ); at++ ) {
         if ( VetAtSel[at] == 0 ) {
            for ( int vlr=0; vlr < ex->QtdeValorAtrib (at); vlr++ ) {
                int valor = ex->lista_atrib[at].valores[vlr];
                int gen = hrq->QualGenAtribVal (at, valor);
                if ( gen != -1  &&  VetGenDisc[gen] == 0 )
                   FCRIGeneralizacao (FCRI, gen, classe);
                else if ( VetVlrDisc[valor] == 0 )
                        FCRIValor (FCRI, at, valor, classe);
                if ( FCRI->result < MenorFCRI->result )
                   selecionado = 1;
                else if ( FCRI->result == MenorFCRI->result  &&
                          FCRI->result != MAXFLOAT ) {
                        if ( MenorFCRI->vlr != -1 && FCRI->gen != -1 )
                            selecionado = 1;
                        else {
                            float fe = lista_parametros[0].valor;
                            float fq = lista_parametros[1].valor;
                            float ft = 1 - fq;
                            float menor = 0;
                            float atual = 0;
                            while ( menor == atual ) {
                              switch (_MAX3(fe, fq, ft)) {
                                 case 1: // economia
                                      menor = ex->CustoAtributo (MenorFCRI->at);
                                      atual = ex->CustoAtributo (at);
                                      fe = 0;
                                      if ( atual < menor )
                                         selecionado = 1;
                                      break;
                                 case 2: // qualidade (relevância)
                                      menor = MenorFCRI->relev;
                                      atual = FCRI->relev;
                                      fq = 0;
                                      if ( fe = 0 ) ft = 1;
                                      if ( atual < menor )
                                         selecionado = 1;
                                      break;
                                 case 3: // tamanho
                                      atual = MenorFCRI->freq;
                                      menor = FCRI->freq;
                                      ft = 0;
```

```
                                        if ( atual < menor )
                                           selecionado = 1;
                                        break;
                                    default:
                                        menor = -1;
                                }
                            }
                        }
                    }
                    if ( selecionado ) {
                        MenorFCRI->result = FCRI->result;
                        MenorFCRI->at     = FCRI->at;
                        MenorFCRI->gen    = FCRI->gen;
                        MenorFCRI->vlr    = FCRI->vlr;
                        MenorFCRI->freq   = FCRI->freq;
                        MenorFCRI->nvgen  = FCRI->nvgen;
                        MenorFCRI->relev  = FCRI->relev;
                        selecionado = 0;
                    }
                }
            }
        }

        if ( MenorFCRI->result == MAXFLOAT ) {
            HouveConflito = 1;
            return;
        }

        // Monta condição com valores selecionados
        Cond->at  = MenorFCRI->at;
        Cond->gen = MenorFCRI->gen;
        Cond->vlr = MenorFCRI->vlr;
        VetAtSel[MenorFCRI->at] = 1;
        return;

}// fim_SelecionaMelhorCondicao



// A Função de Custo Relevante da Informação(FCRI) é calculada por :
// ((Custo(at) + 1) elevado a 'e') / p * Q_E_C + (1-p) * P_E_C
// onde: e = fator de economia;
//       p = fator de ponderação entre menor tamanho (0) e maior qualidade (1)
//   P_E_C = é a probablidade de um exemplo, dado que contém a condição C,
//           pertencer a classe E;
//   Q_E_C = é a probabilidade condicional relevante que é calculada como:
//           gr * P_E_C + (1 - gr) * ( MIN(1, Nce) / Nc )
//           onde: gr = é o grau de relevância da condição C p/a classe E;
//                 Nce = o número de exemplos que tem a condição C e classe E;
//                 Nc = o número de exemplos que tem a condição C;
// Para uma generalização o grau de relevância(gr) é calculado como sendo:
// a média das relevâncias da generalização dividido pela variância desta média.
void ISREG::FCRIGeneralizacao ( fcri *FCRI, int h, int classe )
{
     float fe = lista_parametros[0].valor;   // Fator de Economia
     float fq = lista_parametros[1].valor;   // Fator de Qualidade
     float lista_rel[QTDEMAXVLRGEN];        // lista das relevâncias da generalização
     float freq_cond = 0;                    // freqüência da condição
     float freq_classe = 0;                  // freqüência da classe
     int qtde_vlr_ocorridos = 0;             // valores da generalização presentes
```

```
    int qtde_valores_gen = hrq->lista_hiera[h].qtde_valores;
    int at = hrq->lista_hiera[h].atrib;
    FCRI->result = MAXFLOAT;              // Inicializa estrutura com valores
    FCRI->at     = at;                    // padrões
    FCRI->gen    = hrq->lista_hiera[h].gen;
    FCRI->vlr    = -1;
    FCRI->freq   = FCRI->nvgen  = 0;

    // Percorre os valores da generalização informada
    for ( int i=0; i < qtde_valores_gen; i++ ) {
        int vlr = hrq->lista_hiera[h].val[i];
        lista_rel[i] = rel->CoeficienteRelevancia (at, vlr, classe);
        if ( FreqCond (at+1, vlr)  !=  0 ) {
           freq_cond += FreqCond (at+1, vlr);
           freq_classe += FreqCondClasse (at+1, vlr, classe);
           qtde_vlr_ocorridos += 1;
        }
    }

    if ( qtde_vlr_ocorridos == 0 )     // generalização não presente no
        return;                        // conjunto de treinamento

    if ( AtualQtdeExemplos (at) == freq_cond )// a seleção desta condição manteria
     return;                                  // o mesmo conjunto de treinamento

    // gr = grau de relevância da condição com generalização
    float gr    = Media (lista_rel, qtde_valores_gen) /
                      (1 + Variancia (lista_rel, qtde_valores_gen));
    // P_E_C = probabilidade de um exemplo
    float P_E_C = freq_classe/_MAX(1, freq_cond);

    // Q_E_C = probabilidade condicional relevante
    float Q_E_C = gr * P_E_C + (1 - gr) * ((float) _MIN(1, freq_classe) /
                                                  _MAX(1, freq_cond));

    // Monta a estrutura da FCRI
    FCRI->result = pow((ex->CustoAtributo(at) + 1), fe) /
                     (fq * Q_E_C + (1 - fq) * P_E_C);
    FCRI->nvgen  = qtde_vlr_ocorridos / qtde_valores_gen;
    FCRI->relev  = gr;
    return;

}// fim_FCRIGeneralização


// Para um valor gr é simplesmente o valor presente na matriz de relevância.
void ISREG::FCRIValor ( fcri *FCRI, int at, int vlr, int classe )
{
    float fe = lista_parametros[0].valor;      // Fator de Economia
    float fq = lista_parametros[1].valor;      // Fator de Qualidade
    FCRI->result = MAXFLOAT;                   // Inicializa estrutura com valores
    FCRI->at     = at;                         // padrões
    FCRI->gen    = -1;
    FCRI->vlr    = vlr;
    FCRI->freq   = 0;
    FCRI->nvgen  = 0;

    float freq_classe = (float) FreqCondClasse (at+1, vlr, classe);
    float freq_cond = (float) FreqCond (at+1, vlr);
```

```
        if ( freq_cond == 0  ||  freq_classe == 0 )// valor não presente no CT
            return;                                  // ou nao ocorre p/a classe

        // gr = grau de relevância da condição com valor
        float gr = rel->CoeficienteRelevancia (at, vlr, classe);

        // P_E_C = probabilidade de um exemplo, dado que contém a condição C,
        //         pertencer à classe E
        float P_E_C = freq_classe/_MAX(1, freq_cond);

        // Q_E_C = probabilidade condicional relevante
        float Q_E_C = gr * P_E_C + (1 - gr) * ((float) _MIN(1, freq_classe) /
                                                    _MAX(1, freq_cond));

        // Monta a estrutura da FCRI
        FCRI->result = pow((ex->CustoAtributo(at) + 1), fe) /
                           (fq * Q_E_C + (1 - fq) * P_E_C);
        FCRI->freq   = freq_classe;
        FCRI->relev  = gr;
        return;

}// fim_FCRIValor



// Um subconjunto é uma janela do conjunto de treinamento selecionado com base na
// condição fornecida
void ISREG::GeraSubConjunto ( condicao *Cond )
{
        int qtde_valores_gen = 0;
        int vlr_at_exemplo = 0;

        if ( Cond->at == -1 )       // Condição não fornecida
         return;

        if ( Cond->gen != -1 ) {
            qtde_valores_gen = hrq->QualQtdeValporGen (Cond->gen, Cond->at);
            vlr_at_exemplo = -9;
        }

         // Percorre a tabela de exemplos marcando todos os exemplos que possuam
         // um par atributo/valor diferente daquele fornecido pela condição Cond
        for ( int e=0; e < ex->QualQtdeExemplos (); e++ ) {
            int qtde_valores_diferentes = 0;
            int h = hrq->ExisteHierarquia (Cond->gen, Cond->at);
            for ( int k=0; k < qtde_valores_gen; k++ ) {
                if ( ex->ValAtribExemplo (e, Cond->at) != hrq->lista_hiera[h].val[k] )
                qtde_valores_diferentes += 1;
            }
            if ( Cond->vlr != -1 )
                vlr_at_exemplo = ex->ValAtribExemplo (e, Cond->at);
            if ( qtde_valores_diferentes == qtde_valores_gen  &&
                Cond->vlr != vlr_at_exemplo ) {
                if ( VetDelEx[e] == 0 )
                    for ( int j=0; j < ex->QualQtdeAtributos ( ); j++ )
                      ex->matriz_classe_par[ex->existe_par_atrib_val (j+1, ex->Val
Atrib Exemplo(e, j))][ex->ClassedoExemplo (e)]  -=  ex->freq_exemplo[e];
                if ( VetDelEx[e] >= 0 )
                    VetDelEx[e] += 1;
            }
```

```
    }
    return;

}// fim_GeraSubConjunto


// A generalização será completa quando no sub-conjunto atual a ocorrência dos
// valores de todas as generalizações da regra atendem ao limite estabelecido
// pelo parametro 'lg'.  Quando isso não ocorre a generalização atualmente se-
// selecionada em Cond será marcada como discartada.
int ISREG::GenCompleta ( condicao *Cond, int classe )
{
    float lg = lista_parametros[2].valor;     // Limiar de Generalização

    if ( HouveConflito )
       return 0;

    if ( AtualQtdeExemplos (0) == 0 )   // inconsistêcia detectada
       return 0;

    if ( Cond->at == -1 ) {                    // não conseguiu selecionar
       ReinicializaVetores ( );                // nenhuma condição
       if ( TabRegras[r][p-1].gen != -1 )
          VetGenDisc[TabRegras[r][p-1].gen] = 1;      // despreza a última condição
       else VetVlrDisc[TabRegras[r][p-1].vlr] = 1;    // selecionada
       Cond->at = TabRegras[r][p-1].at;
       Cond->gen = TabRegras[r][p-1].gen;
       Cond->vlr = TabRegras[r][p-1].vlr;
       p -= 1;
       return 0;
     }

      // Percorre toda a regra gerada até a condição p (atual)
      for ( int c=0; c < p; c++ ) {
         if ( TabRegras[r][c].gen != -1 ) {
            int h = hrq->ExisteHierarquia (TabRegras[r][c].gen, TabRegras[r][c].at);
            int qtde_valores_gen = hrq->QualQtdeValporGen (TabRegras[r][c].gen,
                                                           TabRegras[r][c].at);
            int cont = 0;
            for ( int v=0; v < qtde_valores_gen; v++ ) {
                 if ( FreqCondClasse (TabRegras[r][c].at+1,
                                      hrq->lista_hiera[h].val[v], classe)  >  0 )
                      cont += 1;
            }
            if ( ((float) cont / qtde_valores_gen) < lg ) {
               if ( Cond->gen != -1 )
                  VetGenDisc[Cond->gen] = 1;
               else VetVlrDisc[Cond->vlr] = 1;
               VetAtSel[Cond->at] = 0;
               return 0;
            }
         }
      }

      // Caso a regra esteja OK, verifica a integralidade da condição Cond
      if ( Cond->gen != -1 ) {
         int h = hrq->ExisteHierarquia (Cond->gen, Cond->at);
         int qtde_valores_gen = hrq->QualQtdeValporGen (Cond->gen, Cond->at);
         int cont = 0;
```

```
        for ( int v=0; v < qtde_valores_gen; v++ ) {
          if ( FreqCondClasse (Cond->at+1,
                               hrq->lista_hiera[h].val[v], classe)  >  0 )
              cont += 1;
        }
        if ( ((float) cont / qtde_valores_gen) < lg ) {
             VetGenDisc[Cond->gen] = 1;
             VetAtSel[Cond->at] = 0;
             return 0;
        }
    }
    return 1;

}// fim_GenCompleta



void ISREG::ReinicializaVetores  ( )
{
     for ( int i=0; i < ex->QualQtdeAtributos ( ); i++ )
          VetAtSel[i] = 0;
     for ( i=0; i < hrq->QtdeGeneralizacoes; i++ )
          VetGenDisc[i] = 0;
     for ( i=0; i < ex->QualQtdeValorTot ( ); i++ )
          VetVlrDisc[i] = 0;
     return;
}// fim_ReinicializaVetores
// Este método devolve ao conjunto de treinamento o subconjunto retirado pelo mé-
// todo GeraSubConjunto com base na condição informada
void ISREG::RetornaSubConjunto ( condicao *Cond )
{
     int qtde_valores_gen = 0;
     int vlr_at_exemplo = 0;

     if ( Cond->at == -1 )        // condição não fornecida
         return;

     int at = Cond->at;
     if ( Cond->gen != -1 ) {
         qtde_valores_gen = hrq->QualQtdeValporGen (Cond->gen, Cond->at);
         vlr_at_exemplo = -9;
     }

     for ( int e=0; e < ex->QualQtdeExemplos (); e++ ) {
         int qtde_valores_diferentes = 0;
         int h = hrq->ExisteHierarquia (Cond->gen, Cond->at);
         for ( int k=0; k < qtde_valores_gen; k++ ) {
             if ( ex->ValAtribExemplo (e, at) != hrq->lista_hiera[h].val[k] )
                qtde_valores_diferentes += 1;
         }
         if ( Cond->vlr != -1 )
             vlr_at_exemplo = ex->ValAtribExemplo (e, at);
         if ( qtde_valores_diferentes == qtde_valores_gen   &&
             Cond->vlr != vlr_at_exemplo ) {
             if ( VetDelEx[e] == 1 )
                 for ( int j=0; j < ex->QualQtdeAtributos (); j++ )
                    ex->matriz_classe_par[    ex->existe_par_atrib_val    (j+1,    ex-
>ValAtrib Exemplo (e, j))][ex->ClassedoExemplo (e)]  +=  ex->freq_exemplo[e];
                 if ( VetDelEx[e] > 0 )
                    VetDelEx[e] -= 1;
```

```
            }
        }
        VetAtSel[at] = 0;
        return;

}// fim_RetornaSubConjunto



// Definimos como conflito a existência de contra-exemplos no conjunto de trei-
// namento. Não adotamos nenhuma estratégia especial para tratar este evento,
// simplesmente marcamos a regra para uma posterior análise pelo especialista.
// Essa marca consiste em copiar a referida regra p/uma tabela de regras em con-
// flito para, no final do algoritmo, oferecermos uma referência cruzada de todas
// as regras em conflito (esse procedimento é realizado no método GravaRegras())
void ISREG::TrataConflito ( condicao *Cond, int classe )
{
     if ( !HouveConflito )
         return;

     // Ativa última condição válida
     Cond->at  = TabRegras[r][p-1].at;
     Cond->gen = TabRegras[r][p-1].gen;
     Cond->vlr = TabRegras[r][p-1].vlr;

     // Copia todas as condições da regra atual p/a tabela de regras em conflito
     if ( p > 0 ) {
         for ( int c=0; c < p; c++ ) {
            TabRegConflito[rc][c].at     = TabRegras[r][c].at;
            TabRegConflito[rc][c].gen    = TabRegras[r][c].gen;
            TabRegConflito[rc][c].vlr    = TabRegras[r][c].vlr;
            TabRegConflito[rc][c].classe = TabRegras[r][c].classe;
            TabRegConflito[rc][c].nexmap = r+1;// nexmap passa a receber o número
                                               // da regra em conflito
         }
         TabRegConflito[rc++][p-1].classe = classe;
     }
     HouveConflito = 0;          // limpa flag de controle da existência de conflito
     return;

}// fim_TrataConflito



// Os exemplos que mapearam a classe serão marcados no vetor de exemplos
// deletados com -1
int ISREG::RetiraExemplosMapeados ( condicao *Cond )
{
    int qtde_valores_gen = 0;
    int vlr_at_exemplo = 0;
    int cont_ex_mapeados = 0;

    if ( Cond->at == -1 )       // condição não fornecida
       return 0;

    if ( Cond->gen != -1 ) {
       qtde_valores_gen = hrq->QualQtdeValporGen (Cond->gen, Cond->at);
       vlr_at_exemplo = -9;
    }
```

```
    // Percorre a tabela de exemplos retirando retiramdo aqueles que tiverem
    // o par atributo/valor diferente do informado na condição Cond
    for ( int e=0; e < ex->QualQtdeExemplos ( ); e++ ) {
         int vlr_pertence_gen = 0;
         int h = hrq->ExisteHierarquia (Cond->gen, Cond->at);
         for ( int k=0; k < qtde_valores_gen; k++ ) {
            if ( ex->ValAtribExemplo (e, Cond->at) == hrq->lista_hiera[h].val[k] ) {
               vlr_pertence_gen = 1;
               break;
            }
         }
         if ( Cond->vlr != -1 )
            vlr_at_exemplo = ex->ValAtribExemplo (e, Cond->at);
         if ( vlr_pertence_gen   ||   Cond->vlr == vlr_at_exemplo ) {
            if ( VetDelEx[e] == 0 ) {
               for ( int j=0; j < ex->QualQtdeAtributos ( ); j++ )
                     ex->matriz_classe_par[ex->existe_par_atrib_val     (j+1,     ex-
>ValAtribExemplo(e, j))][ex->ClassedoExemplo (e)]  -=  ex->freq_exemplo[e];
               cont_ex_mapeados += ex->freq_exemplo[e];
               VetDelEx[e] = -1;
            }
         }
    }
    QtdeExemplosMapeados += cont_ex_mapeados;
    return cont_ex_mapeados;
}// fim_RetiraExemplosMapeados



// Este método devolve ao conjunto de treinamento todos os subconjuntos retirados
// pelo método GeraSubConjunto para compor a regra atual.
// Esta devolução é feita com base nas condições da regra atual (r)
void ISREG::RetornaSubConjuntosRegra ( )
{
     for ( int c=0; TabRegras[r][c].at != -1; c++ ) {
         int qtde_valores_gen = 0;
         int vlr_at_exemplo = 0;
         int at = TabRegras[r][c].at;
         if ( TabRegras[r][c].gen != -1 ) {
            qtde_valores_gen = hrq->QualQtdeValporGen (TabRegras[r][c].gen,
                                                       TabRegras[r][c].at);
            vlr_at_exemplo = -9;
            VetGenDisc[TabRegras[r][c].gen] = 0;
         }
         else VetVlrDisc[TabRegras[r][c].vlr] = 0;
         for ( int e=0; e < ex->QualQtdeExemplos (); e++ ) {
            int qtde_valores_diferentes = 0;
            int h = hrq->ExisteHierarquia (TabRegras[r][c].gen, TabRegras[r][c].at);
            for ( int k=0; k < qtde_valores_gen; k++ ) {
            if ( ex->ValAtribExemplo (e, at) != hrq->lista_hiera[h].val[k] )
               qtde_valores_diferentes += 1;
            }
            if ( TabRegras[r][c].vlr != -1 )
               vlr_at_exemplo = ex->ValAtribExemplo (e, at);
            if ( qtde_valores_diferentes == qtde_valores_gen   &&
               TabRegras[r][c].vlr != vlr_at_exemplo ) {
               if ( VetDelEx[e] > 0 ) {
                  for ( int j=0; j < ex->QualQtdeAtributos (); j++ )
                     ex->matriz_classe_par[   ex->existe_par_atrib_val   (j+1,   ex-
>ValAtribExemplo (e, j))][ex->ClassedoExemplo (e)]  +=  ex->freq_exemplo[e];
```

```
                    VetDelEx[e] = 0;
                }
            }
        }
    }
    ReinicializaVetores ( );
    return;

}// fim_RetornaSubConjuntosRegra



// Este método grava, no arquivo de saída, um resumo das informações sobre as
// bases de dados geradas nas execuções provocadas pela combinação dos parâme-
// tros 'fe' e 'fp'.
void ISREG::GravaDadosAnalise ( int float e, float p )
{
    FILE *fpr;
    if ( e == 0  &&  p == 0 ) {
        fpr = fopen ( conf->nome_saida, "w" );
        fprintf (fpr, "   Arquivo Saída: %s\n\n",conf->nome_saida);
        fprintf (fpr, "Domínio                          - %s\n", conf->nome_dominio);
        if ( rel->arq_carregado )
           fprintf (fpr, "Matriz de Relevância          - %s\n", rel->nome_matriz);
        if ( hrq->arq_carregado ) {
           fprintf (fpr, "Tabela de Generalização       - %s", hrq->nome_hiera);
           fprintf (fpr, "                    Limiar de Generalização - %3.2f\n",
lista_parametros[2].valor);
        }
        fprintf (fpr, "N.Exemp.do Conj.Treinamento  - %d\n",
                      ex->QualQtdeExemplosReal ( ));
        if ( QtdeExTeste != 0 )
           fprintf (fpr, "N.Exemp.do Conj.Teste        - %d\n", QtdeExTeste);
        fprintf (fpr, "\nfe => Fator de Economia   = [0,1] onde: 0=perdulário ...
                      1=economia máxima");
        fprintf (fpr, "\nfp => Fator de Qualidade  = [0,1] onde: 1=qualidade máxima
                      \n\n\n");
        fprintf (fpr, "                     ----------- Regras -------------
                      Nível     Freq.    Custo\n");
        fprintf (fpr, "                      Qtde  Tam.Med  Conflito  Inúteis
                      Qualidade   Acerto   Médio\n");
    }
    else fpr = fopen ( conf->nome_saida, "a" );
    char nivel_qualidade [5];
    char regras_inuteis [5];
    float nq = NivelQualidadeDaBase ( );
    if ( nq != -1 ) {
        sprintf (nivel_qualidade, "%3.2f", nq);
        sprintf (regras_inuteis, "%4d", RegrasInuteis);
    }
    else {
        sprintf (regras_inuteis,  " -- ");
        sprintf (nivel_qualidade, " -- ");
    }
    char freq_acerto [5];
    float fa = FreqAcertoConjTeste ( );
    if ( QtdeExTeste > 1 )
        sprintf (freq_acerto, "%6.2f%%", fa);
    else sprintf (freq_acerto,  "  --- ");
    if ( p == 0 )
```

UFPb/BIBLIOTECA/PRAI

```
        fprintf (fpr, "\nfe=%3.2f  fp=%3.2f    %4d    %3.2f     %4d      %4s
                %4s     %5s  %.3f\n", e, p, r, TamMedioRegra ( ), rc,
                regras_inuteis, nivel_qualidade, freq_acerto, CustoMedio ( ));
    else fprintf (fpr, "          fp=%3.2f    %4d    %3.2f     %4d      %4s
                %4s     %5s  %.3f\n", p, r, TamMedioRegra ( ), rc, regras_inuteis,
                nivel_qualidade, freq_acerto, CustoMedio ( ));
    fclose (fpr);
    return;

}// fim_GravaDadosAnalise



// Este método grava, no arquivo de saída, um resumo com informações sobre
// uma execução do algoritmo com base nos parâmetros: fe, fp e lg.
void ISREG::GravaResumo ( )
{
    FILE *fpr;
    fpr = fopen (conf->nome_saida, "a");
    fprintf (fpr, "\n\nResumo (%s):\n\n",nome_alg);
    fprintf (fpr, "Domínio                          - %s\n", conf->nome_dominio);
    if ( hrq->arq_carregado )
        fprintf (fpr, "Tabela Hierarquia              - %s\n", hrq->nome_hiera);
    if ( rel->arq_carregado )
        fprintf (fpr, "Matriz de Relevância           - %s\n", rel->nome_matriz);
    fprintf (fpr, "N.Exemp.do Conj.Treinamento  - %d\n", ex->QualQtdeExemplosReal (
            ));
    if ( QtdeExTeste != 0 )
        fprintf (fpr, "N.Exemp.do Conj.Teste          - %d\n", QtdeExTeste);
    fprintf (fpr, "Fator de Economia (fe)       - %3.2f\n",
                lista_parametros[0].valor);
    fprintf (fpr, "Fator de Qualidade (fq)      - %3.2f\n",
                lista_parametros[1].valor);
    fprintf (fpr, "Limiar da Generalização (lg) - %3.2f\n",
                lista_parametros[2].valor);
    fprintf (fpr, "No. Total de Regras          - %d\n", r);
    fprintf (fpr, "Tamanho Médio de Regra       - %3.2f\n", TamMedioRegra ( ));
    fprintf (fpr, "Custo Médio de Classificação - %.2f\n", CustoMedio ( ));
    float nq = NivelQualidadeDaBase ( );
    if ( nq != -1 )
        fprintf (fpr, "Nível de Qualidade da Base   - %3.2f\n", nq);
    if ( QtdeExTeste != 0 )
        fprintf (fpr, "Freq.Acerto do Conj.Teste    - %.2f%%\n",
                FreqAcertoConjTeste ( ));

    GravaRegrasInuteis ( fpr );

    GravaRegrasEmConflito ( fpr );

    fclose (fpr);

}// fim_GravaResumo
```

========== **Fim do A4ISREG.C**

APÊNDICE C - 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 - BrinqSeguro

2 classes
4 atributos
12 exemplos

---------- Classes ----------

Cod	Nome	#Exs.
000	S	6
001	P	6

--------- Atributos ----------			---------- Valores ----------	
Cod	Nome	Custo	Cod	Nome	Cod.Generalização
000	Forma	10	000	quadrado	000
			001	triangulo	000
			002	elipse	001
			003	circulo	001
			004	pentagono	000
001	Cor	30	005	vermelho	002
			006	azul	002
			007	amarelo	002
			015	rosa	
002	Tamanho	140	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	SAUDAVEL	16

--------- Atributos ---------- ---------- Valores ----------

Cod	Nome	Custo	Cod	Nome	Cod.Generalização
000	Idade	0	000	jovem	
			001	madura	
			002	senil	
001	Vomitos_Enjoos	0	003	sim	
			004	nao	
002	Sexo_Periodo	0	003	sim	
			004	nao	
003	BHCG	2000	005	positivo	
			006	negativo	
004	Perfil_Hormonal	8000	007	e	
			008	pe	
			009	p	
			010	n	
			011	x	
005	Atraso_Menstrual	0	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 aclive
                                 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.REL

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	

Matriz de Relevância = Zoo.REL

Atributo	Valor	Classes 0	1	2	3	4	5	6
hair	no	0.00	1.00	1.00	1.00	1.00	0.00	1.00
	yes	0.50	0.00	0.00	0.00	0.00	0.50	0.00
feathers	no	1.00	0.00	1.00	1.00	1.00	1.00	1.00
	yes	0.00	1.00	0.00	0.00	0.00	0.00	0.00
eggs	no	1.00	0.00	0.00	0.00	0.00	0.00	0.00
	yes	0.00	1.00	0.00	0.00	0.00	0.00	0.75
milk	no	0.00	1.00	1.00	1.00	1.00	1.00	1.00
	yes	1.00	0.00	0.00	0.00	0.00	0.00	0.00
airborne	no	1.00	0.25	1.00	1.00	1.00	0.25	1.00
	yes	0.00	0.75	0.00	0.00	0.00	0.75	0.00
aquatic	no	0.50	0.00	0.50	0.00	0.00	1.00	0.00
	yes	0.00	0.00	0.00	1.00	1.00	0.00	1.00
predator	no	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	yes	0.00	0.00	0.00	0.00	0.00	0.00	0.00
toothed	no	0.00	1.00	0.75	0.00	0.00	0.00	0.75
	yes	1.00	0.00	0.00	0.00	0.00	0.00	0.00
backbone	no	0.00	0.00	0.00	0.00	0.00	1.00	1.00
	yes	1.00	1.00	0.50	1.00	0.00	0.00	0.00
breathes	no	0.00	0.00	0.75	1.00	0.75	0.00	1.00
	yes	1.00	0.75	0.00	0.00	0.00	0.00	0.00
venomous	no	0.75	1.00	0.00	0.75	0.00	1.00	0.75
	yes	0.00	0.00	0.25	0.00	0.50	0.00	0.00
fins	no	0.75	1.00	1.00	0.00	0.50	1.00	0.00
	yes	0.00	0.00	0.00	1.00	0.00	0.00	0.75
legs	0	0.00	0.00	0.75	1.00	0.00	0.00	0.00
	2	0.75	1.00	0.00	0.00	0.00	0.00	0.00
	4	0.75	0.00	0.25	0.00	0.50	0.00	0.00
	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	6	0.00	0.00	0.00	0.00	0.00	0.00	1.00
	8	0.00	0.00	0.00	0.00	0.00	0.00	1.00
tail	no	0.00	0.00	0.75	0.00	0.00	0.75	0.50
	yes	0.50	1.00	0.00	1.00	0.75	0.00	0.00
domestic	no	0.00	0.00	0.00	0.00	1.00	1.00	0.75
	yes	0.00	0.00	0.00	0.00	0.00	0.00	0.00
catsize	no	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	yes	1.00	1.00	1.00	1.00	1.00	1.00	1.00

ABSTRACT

Cognitive and semi-automated are plaged by problems such as experts having not enough time available, knowledge engineers and experts not communicating sufficiently well, etc. The field of automated knowledge acquisition arose with the objective of solving these problems. However, evaluating the knowledge bases generated by automated methods is a process still undertaken just by experts. Inductive methods for acquiring knowledge from a set of examples, though being perhaps the most used ways for achieving automated knowledge acquisition, face two serious problems, the first one ("the syntactic problem") having a structural nature, and the second one ("the semantic problem") having a content nature. Inductive methods are also named empirical, emphasizing the fact that they demand little, if any, background knowledge about the application domain. The choice of an adequate formalism for knowledge representation averts the syntactic problem and adoption of a relevance-matrix (a certain kind of background knowledge) makes the occurrence of the semantic problem rare. Other kinds (cost and generalization) of background knowledge may help the inductive methods to yield knowledge bases of better quality. In this work we propose an algorithm, ISREG, that: a) provides a combined use (not offered by any of the current inductive methods) of the main kinds of background knowledge; b) solves the syntactic problem; and c) minimizes the occurrence of the semantic problem. Furthermore, we propose an automated process for evaluating the semantic quality of knowledge bases generated by inductive methods.

Referências Bibliográficas

[Alexandre 93] Cláudio R. Alexandre, Giuseppe Mongiovi. *Qualidade semântica de bases de conhecimento geradas por métodos indutivos.* In: SIAR'93 - SIMPOSIO DE INTELIGENCIA ARTIFICIAL Y ROBOTICA, 1993. Buenos Aires. **Proceedings...** [s.n.t.].

[Alexandre 94a] Cláudio R. Alexandre, Giuseppe Mongiovi, Haroldo Cesar F. Bezerra. *Relevância semântica de uma generalização: Definição e um estudo de caso.* In: II CONGRESO y EXPOSICIÓN INTERNACIONAL DE INFORMÁTICA, 1994. Mendoza. e In: XX CONFERENCIA LATINOAMERICANA DE INFORMÁTICA, 1994. México. **Proceedings...** México: Centro Latinoamericano de Estudos en Informática, 1994. p.803-814.

[Alexandre 94b] Cláudio R. Alexandre, Giuseppe Mongiovi, José Antão B. Moura. *Semantic induction of generalized, cost-effective and modular rules.* In: Brazilian Symposium on Artificial Intelligence, 11, 1994. Fortaleza. **Proceedings...** Fortaleza: Universidade Federal do Ceará, 1994. p.187-201.

[Belchior 92] Arnaldo Dias Belchior. *Controle da qualidade de "software" financeiro.* Rio de Janeiro, COPPE/UFRJ, 1992, 189p. **Dissertação** (Mestrado em Informática) - Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro, COPPE, 1992.

[Carbonell 83] Jaime G. Carbonell, Ryszard S. Michalski, Tom M. Mitchell. *An overview of machine learning.* In:—, (Ed.). **Machine Learning.** Los Altos, CA:Morgan Kaufmann Publisher Inc., 1983. v.1, p.3-37.

[Carbonell 90] Jaime G. Carbonell. *Introduction: Paradigms for machine learning.* In:—, (Ed.). **Machine Learning: paradigms and methods.** Cambridge, MA:MIT Press, 1990. p.1-10.

[Cendrowska 88] Jadzia Cendrowska. *PRISM: An algorithm for inducing modular rules.* In: Brian R. Gaines, John H. Boose, (Ed.). **Knowledge-Based System.** London: Academic Press, 1987. v.1, p.255-276.

[Cirne Filho 91] Walfredo C. Cirne Filho, Giuseppe Mongiovi. *Indução semântica de regras modulares.* In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICAL, 8, 1991, Brasília. **Anais...** Brasília: Sociedade Brasileira de Computação, 1991. p.143-148.

[Cirne Filho 92] Walfredo C. Cirne Filho. *O uso de semântica na melhoria dos métodos indutivos de aquisição automática de conhecimento.* Campina Grande, UFPB, 1992, 64p. **Dissertação** (Mestrado em Informática) - Departamento de Sistemas e Computação, Universidade Federal da Paraíba, 1992.

[Donato Júnior 94] Edmundo T. Donato Júnior. *Uso de conhecimento preliminar na melhoria do aprendizado em um modelo simbólico-conexionista.* Campina Grande, UFPB, 1994, 105p. **Dissertação** (Mestrado em Informática) - Departamento de Sistemas e Computação, Universidade Federal da Paraíba, 1994.

[Feigenbaum 81] Edward A. Feigenbaum. *Expert Systems in 1980's.* In: A. Bond (Ed.). **The state of the art report on machine intelligence**. Oxford:Pergamon-Infotech, 1981.

[Firebaugh 89] Morris W. Firebaugh. *Artificial Intelligence - A knowledge-based approach.* Boston:PWS-Kent Publishing Company, 1989. 740p.

[Genaro 87] Sergio Genaro. *Sistemas Especialista - O conhecimento artificial.* São Paulo:LTC-Livros Técnicos e Científicos Editora S.A., 1986. 192p.

[Gaines 93] Brian R. Gaines, Mildred L. G. Shaw. *Eliciting knowledge and transfering it effectively to a knowledge-based system.* **IEEE Transactions on Knowledge and Data Engineering**, v.5, n.1, p.4-13, fev.1993.

[Gomes 89] Fernando A. Gomes. *APREND: Um sistema de aquisição automática de conhecimento a partir de exemplos.* Campina Grande, UFPB, 1989, 65p. **Dissertação** (Mestrado em Informática) - Departamento de Sistemas e Computação, Universidade Federal da Paraíba, 1989.

[Lucena 83] Carlos J. P. de Lucena. *Análise de síntese de programas de computador.* Brasília: Editora Universidade de Brasília, 1983. 185p.

[Hart 87] Anna Hart. *Role of induction in knowledge elicitation.* In: Alison L. Kidd (ed.). **Knowledge acquisition for expert systems: A pratical handbood.** New York: Plenum Press, 1987. p.165-189.

[Kidd 87] Alison L. Kidd. *Knowledge acquisition - An introdutory framework.* In:—, (Ed.). **Knowledge acquisition for expert systems: A pratical handbood.** New York: Plenum Press, 1987. 194p.

[Klir 88] J. G. Klir, T. A. Floger. *Fuzzy sets, uncertainty, and information.* New Gersey: Prentece-Hall, 1988.

[Michaelsen 85] Robert H. Michaelsen, Donald Michie, Albert Boulanger. *The techonology of expert systems.* **BYTE**, v.10, n.4, p.310, abr.1985.

[Michalski 83] Ryszard S. Michalski. *Pattern recognition as knowledge-guided computer iduction.* **IEEE Transactions on Pattern Analysis and Machine Intelligence**. v.PAMI-2, n.4, p.349-361, july 1983.

[Michalski 86] Ryszard S. Michalski. *Understanding the nature of learning: Issues and research directions.* In:—, Jaime G. Carbonell, Tom M. Mitchell (Ed.). **Machine Learning - A Artificial Intelligence Approach**. Los Altos, CA: Morgan Kaufmann Publisher Inc., 1986. v.2, p.3-25.

[Michalski 90] Ryszard S. Michalski, Y. Kodratoff. *Research in machine learning; Recent progress, classification of methods and future directions.* In:—, Y. Kodratoff (Ed.). **Machine Learning - A Artificial Intelligence Approach**. Los Altos, CA: Morgan Kaufmann Publisher Inc., 1990. p.3-30.

[Michie 85] Donald Michie. *Expert systems interview.* **Expert Systems**, v.2, n.1, p.21, jan.1985.

[Mongiovi 90] Giuseppe Mongiovi, Walfredo C. Cirne Filho. *Um algoritmo baseado em conhecimento para ampliar a potencialidade do ID3.* In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICAL, 7, 1990, Campina Grande. **Anais...** Campina Grande: [s.n.], 1990. p.8-16.

[Mongiovi 91] Giuseppe Mongiovi, Walfredo C. Cirne Filho. *O uso de semântica na construção e expansão de árvores de decisão indutivas.* In: XVII CONFERENCIA LATINOAMERICANA DE INFORMÁTICA, 1991. Caracas. **Proceedings...** [s.n.t.].

[Mongiovi 93a] Giuseppe Mongiovi. *Aquisição automática de conhecimento a partir de exemplos: uma abordagem pragmática.* Campina Grande, UFPB, 1993, 108p. **Tese** (concurso público para prof. titular) - Departamento de Sistemas e Computação, Universidade Federal da Paraíba, 1993.

[Mongiovi 93b] Giuseppe Mongiovi, João J.P.F. Vasco. *Relevância semântica nebulosa para os métodos indutivos de aquisição de conhecimento.* In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICAL, 10, 1993, Porto Alegre. **Anais...** Porto Alegre: Instituto de Informática da UFRGS, 1993. p.141-153.

[Nuñez 88] Marlon Nuñez. *El metodo de aprendizaje EG2: Una aplicacion de conocimiento de base a ejemplos estructurados*. Madrid, UPM, 1988, 74p. **Tese** (Master en Ingenieria del Conocimiento) - Facultad de Informatica, Universidad Politecnica de Madrid, 1988.

[Nuñez 91] Marlon Nuñez. *The use of background knowledge in decision tree induction*. **Machine Learning**, Boston, v.6, n.3, p.231-250, maio 1991.

[Paladini 90] Edson P. Paladini. *Controle da qualidade uma abordagem abrangente*. São Paulo:Editora Atlas S.A., 1990.

[Peirce 65] Charles S. Peirce. *Elements of logic*. Cambridge, MA: The Belknap Press Harvard University Press, 1965.

[Pires 93] Fernando J. G. Moura-Pires. *Aprendizagem por indução empírica*. Lisboa, UNL, 1993, 238p. **Tese** (Doutorado em Informática) - Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 1993.

[Quinlan 83] J. Ross Quinlan. *Learning efficient classification procedures and their application to chess end games*. In:—, Jaime G. Carbonell, Tom M. Mitchell (Ed.). **Machine Learning - A Artificial Intelligence Approach**. Los Altos, CA: Morgan Kaufmann Publisher Inc., 1983. p.468-482.

[Quinlan 86] J. Ross Quinlan. *Induction of decision trees*. **Machine Learning**, Boston, v.1, n.1, p.81-106, 1986.

[Rich 91] Elaine Rich, Kevin Knight. *Artificial Intelligence*. 2.ed. New York: McGraw-Hill Inc., 1991. 621p.

[Uthurusamy 93] Ramasamy Uthurusamy, Linda G. Means, Kurt S. Godden. *Extracting knowledge from diagnostic databases*. **IEEE Expert**, Los Alamitos, CA, v.8, n.6, p.27-38, dez. 1993.

[Vasco 92] João J.P.F. Vasco, Giuseppe Mongiovi, Walfredo C. Cirne Filho, Edmunto Tojal Donato Júnior. *A4 - Ambiente de apoio à aquisição automática de conhecimento*. In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICAL, 9, 1992, Rio de Janeiro. **Anais...** Rio de Janeiro: Sociedade Brasileira de Computação, 1992. p.186-199.

[Vasco 93] João J.P.F. Vasco. *A4 - Um ambiente de apoio à aquisição automática de conhecimento*. Campina Grande, UFPB, 1993, 105p. **Dissertação** (Mestrado em Informática) - Departamento de Sistemas e Computação, Universidade Federal da Paraíba, 1993.

Bibliografia

BOOSE, John H. *A knowledge acquisition program for expert systems based on personal construct psychology*. **International Journal of Man-Machine Studies**, London, v.23, p.495-525, abr. 1985.

CLEAL, D. M., HEATON, N. O. *Knowledge acquisition*. In: HORWOOD, Ellis (Ed.). **Knowledge-Based Systems: Implications for human-computer interfaces.** Chichester-England: Ellis Horwood Limeted, 1988. p.152-175.

CORMEN, Thomas H., LEISERSON, Charles E., RIVEST, Ronald L. *Introduction to Algorithms*. New York: McGraw-Hill Book Company, 1992. 1028p.

CUNHA, Horácio da., RIBEIRO, Sousa. *Introdução aos sistemas especialistas*. Rio de Janeiro: Livros Técnicos e Científicos Editora S.A., 1987. 142p.

EVANS, Bob, FISHER, Doug. *Overcoming Process delays with decision tree induction*. **IEEE Expert**, Los Alamitos, CA, v.9, n.1, p.60-66, fev. 1994.

FRANÇA, Júnia Lessa, et. al. *Manual para normalização de publicações técnico-científicas*. 2.ed. Belo Horizonte: Editora da UFMG, 1990. 196p.

GAINES, Brian R. *An overview of knowledge-acquisition and transfer*. **International Journal of Man-Machine Studies**, London, v.26, p.453-472, 1987.

GENARO, Sergio. *Sistemas especialistas - o conhecimento artificial*. Rio de Janeiro: Livros Técnicos e Científicos Editora S.A., 1986. 192p.

GOMES, Fernando A. de Carvalho. *Utilisation d'algorithmes stochastiques en apprentissage*. Montpellier, 1992, 156p. **Tese** (Doutorado em Informática) - Université des Sciences et Techniques du Languedoc, Université Montpellier II, 1992.

LAKATOS, Eva Maria, MARCONI, Marina de Andrade. *Metodologia do trabalho científico*. 4.ed. São Paulo: Editora Atlas S.A., 1992. 214p.

MICHALSKI, Ryszard S., CHILAUSKY, R. L. *Knowledge acquisition by encoding expert rules versus computer induction from examples: a case study involving soybean pathology*. **International Journal of Man-Machine Studies**, London, v.12, p.63-87, 1980.

MICHALSKI, Ryszard S. *A theory and methodology of inductive learning*. In:—, CARBONELL, Jaime G., MITCHELL, Tom M. (Ed.). **Machine Learning.** Los Altos, CA:Morgan Kaufmann Publisher Inc., 1983. v.1, p.83-125.

SALOMON, Délcio Vieira. *Como fazer uma monografia*. 2.ed. São Paulo: Livraria Martins Fontes Editora Ltda., 1991. 294p.

SCHILDT, Herbert. *Inteligência artificial utilizando linguagem C*. São Paulo: McGraw-Hill, 1989. 349p.

Índice remissivo