

FERNANDO ANTONIO DE CARVALHO GOMES

APREND: UM SISTEMA DE AQUISIÇÃO AUTOMÁTICA DE CONHECIMENTO
A PARTIR DE EXEMPLOS

Dissertação apresentada ao Curso
de MESTRADO EM INFORMÁTICA da
Universidade Federal da Paraíba,
em cumprimento às exigências
para obtenção do grau de mestre.

GIUSEPPE MONGIOVI

Orientador

Campina Grande - PB

Abril - 1989

*1. Sistema Especializado em Informática
2. Aquisição Automática de Conhecimento (APREND)
3. APREND (Sistema)
4. Informática*



G633a Gomes, Fernando Antonio de Carvalho
Aprend : um sistema de aquisicao automatica de conhecimento a partir de exemplos / Fernando Antonio de Carvalho Gomes. - Campina Grande, 1989.
65 f. : il.

Dissertacao (Mestrado em Informatica) - Universidade Federal da Paraiba, Centro de Ciencias e Tecnologia.

1. Sistemas Especialistas - 2. Aquisicao Automatica de Conhecimentos (APREND) 3. APREND (Sistema) 4. Informatica 5. Dissertacao I. Mongiovi, Giuseppe, Prof. II. Universidade Federal da Paraiba - Campina Grande (PB) III. Título

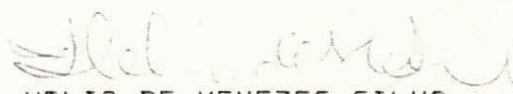
CDU 004.891(043)


APREND: UM SISTEMA DE AQUISIÇÃO AUTOMÁTICA DE CONHECIMENTO
A PARTIR DE EXEMPLOS

FERNANDO ANTONIO DE CARVALHO GOMES

Dissertação apresentada em 24/04/89


GIUSEPPE MONGIOVI
- Presidente -


HELIO DE MENEZES SILVA
- Examinador -


EMMANUEL PISECES LOPES PASSOS
- Examinador -

CAMPINA GRANDE

ABRIL - 1989

ABSTRACT

The major barrier to building expert systems is the knowledge acquisition problem. This work presents the development of the APREND system. The system is a tool for automatic knowledge acquisition. It uses examples as input data, and is based on inductive inference in order to provide two kinds of production rules: complete rules (with certainty factors 1,0, and one or more conditions), and atomic rules (with certainty factors in the interval [-1 1], and just one condition). Some criteria for good rules identification are presented. The aspects of the system design, implementation and using are showed. An application on nephrology is reported, in order to illustrate the knowledge acquisition process using APREND.

Aos queridos avós

José Valdivino e Maria Adair

(in memoriam)

AGRADECIMENTOS

Aos meus pais, pelo apoio dado ao longo de todos estes anos de estudo e, principalmente, durante a realização deste trabalho.

A minha esposa e ao meu filho, pela paciência e compreensão nos momentos em que troquei o convívio em família pela "tese".

Aos professores Giuseppe Mongiovi e Hélio de Menezes Silva, pelo incentivo e pelos esclarecimentos, indispensáveis nos momentos de dúvida.

Aos colegas e funcionários da UFPB, principalmente a Ana Lúcia Guimarães, cujo apoio facilitou a finalização deste trabalho.

Aos alunos do DSC que contribuíram para o sucesso deste trabalho, e especialmente a Túlio Ramalho Costa, pela ajuda na implementação e depuração dos programas.

A CAPES (PICD), pelo apoio financeiro.

SUMARIO

1 - Introdução	01
2 - Localização do APREND	08
2.1 - Métodos de Aquisição de Conhecimento	08
2.2 - Estratétias Fundamentais de Aprendizado	09
2.3 - O Sistema APREND	14
3 - O Algoritmo ID3	15
3.1 - Apresentação da Tabela de Exemplos	15
3.2 - Formação aleatória de uma AD	16
3.3 - Indução de AD's usando o ID3	19
3.3.1 - Obtenção do Conjunto Janela	19
3.3.2 - Cálculo da Entropia de um Atributo	20
3.4 - Vantagens do ID3	24
3.5 - Desvantagens do ID3	24
3.6 - Modificações efetuadas pelo APREND	25
4 - Tratamento da Incerteza nas Regras	27
4.1 - Revisão de Regras com tratamento da incerteza "a la MYCIN"	27
4.2 - Importância das RA's	28
4.3 - Critérios para a escolha das RA's	29
4.4 - Critério para a escolha das RC's	31
4.5 - Processo de Cálculo dos FC's	33
4.5.1 - Cálculo do MC	33
4.5.2 - Cálculo do MD	34

5 - Projeto, Implementação e Uso do APREND	36
5.1 - Etapas de Projeto	37
5.1.1 - Edição de Exemplos	37
5.1.2 - ID3: Construção da AD	37
5.1.3 - ATM: Geração de RA's	41
5.2 - Estruturas de Dados	41
5.2.1 - O Descritor de Classes, Atributos e Valores ...	41
5.2.2 - A Tabela de Exemplos	43
5.2.3 - A Arvore de Decisão	44
5.3 - Guia de Utilização	45
6 - Uma Aplicação em Nefrologia	50
6.1 - Apresentação dos Dados	50
6.2 - Análise dos Resultados	52
7 - Conclusões e Trabalhos Futuros	54
Referências Bibliográficas	57
Apêndice A	61
Apêndice B	62
Apêndice C	63
Resumo	65

1. INTRODUÇÃO

Definir inteligência é uma tarefa difícil, pois envolve conceitos que não são completamente compreendidos, tais como: criatividade, curiosidade, aprendizado, raciocínio, etc.

Definir com precisão Inteligência Artificial (IA), torna-se difícil por envolver o conceito de inteligência. Uma consequência disto, é que existe uma definição de IA para cada pessoa envolvida com a área. Porém, uma boa definição para IA, foi dada por Patrick H. Winston (Winston, 1987):

"Inteligência Artificial é a área do conhecimento, relacionada a sistemas de computação, que procura imitar as complexas ações humanas."

Do ponto de vista comercial, entre as muitas atividades de pesquisa que se enquadram atualmente na área de IA, os Sistemas Especialistas (SE's) estão presentes de forma marcante no mercado de software.

SE's, são sistemas de computação que usam o conhecimento de um ou mais peritos em um certo domínio, com o objetivo de encontrar resultados semelhantes aos encontrados por estes peritos.

Basicamente, um SE é composto de duas partes, a Base de Conhecimento (BC) e o motor de inferências. A BC contém todo o conhecimento extraído do(s) perito(s) no domínio. O motor de inferências utiliza os dados residentes na BC para simular o raciocínio do perito, com o objetivo de encontrar resultados coerentes, do ponto de vista prático. É usual e conveniente que

um SE tenha também um módulo de explanação e uma interface amigável com o usuário, através de linguagem natural, menus, gráficos, etc. Alguns SE's têm módulos de aquisição e verificação do conhecimento, podendo ser automatizados, semi-automatizados ou manuais.

O conhecimento utilizado deve ser estruturado de forma a facilitar a resolução do problema a que o SE se propõe. Alguns modelos de representação do conhecimento são mais adaptáveis a domínios fortemente dedutivos, outros são mais convenientes a domínios descritivos, enquanto que outros mapeiam muito bem os domínios que se apresentam estruturados hierarquicamente. Atualmente, a grande maioria dos SE's (mais de 90%) utiliza regras de produção como meio de representação do conhecimento (Voyer, 1987). Outras representações usuais são redes semânticas e quadros (frames). Uma rede semântica é um grafo onde os nós representam conceitos e as ligações representam relações entre os conceitos. Os frames representam características comuns a um objeto e são usados em domínios que se apresentam estruturados hierarquicamente.

O processo que vai desde a escolha do modelo mais adequado de representação do conhecimento até a construção da BC, é chamado de "engenharia de conhecimento".

O engenheiro de conhecimento interage com o(s) perito(s) no domínio estudado, com a finalidade de descobrir seus procedimentos, estratégias e regras, para assim poder construir a BC. Chama-se esta fase de "aquisição de conhecimento".

Muitos autores têm observado que a maior barreira ao se

construir SE's, é a tarefa de adquirir o conhecimento necessário ao seu funcionamento (Quinlan, 1987a). Edward Feigenbaum (Feigenbaum, 1981), escreveu:

"Existem muitos problemas importantes de representação, utilização e aquisição que devem ser resolvidos, porém, o problema da aquisição de conhecimento é o "gargalo" da construção de SE's."

A comunicação entre o(s) perito(s) e o engenheiro de conhecimento é dificultada porque o perito tem a tendência de explicar seu raciocínio e suas conclusões em termos gerais, i.e. as partes do conhecimento básico são combinadas de forma tão rápida que se torna difícil descrever todo o processo de raciocínio (Waterman, 1986). A comunicação entre dois peritos do mesmo domínio é feita através de uma linguagem própria e impossível de ser entendida por qualquer leigo.

As dificuldades encontradas pelo perito ao tentar enunciar o seu conhecimento reside em quatro pontos:

- (1) O conhecimento é compilado, i.e. o longo processo de aprendizagem por que passa o perito, faz com que o conhecimento se reestruture ao longo do tempo, dificultando a enunciação;
- (2) O conhecimento é subjetivo, i.e. cada perito possui uma visão muito especializada do domínio;
- (3) O conhecimento é volátil, i.e. em alguns domínios que estão em constante evolução, uma afirmação que hoje é aproveitada como conhecimento, amanhã pode deixar de ser aproveitada total ou parcialmente;
- (4) O conhecimento é formado de várias partes, i.e. em muitos

domínios, diversos peritos detêm, cada um, uma parte do conhecimento. Seria interessante se estas partes fossem agrupadas na BC (Boy et al., 1987).

Para amenizar os problemas na fase de aquisição de conhecimento, é necessário o desenvolvimento de um ambiente de suporte ao trabalho do engenheiro de conhecimento. O ambiente de suporte agrega ferramentas de aquisição e verificação de conhecimento, interfaces amigáveis, módulos de explicação do raciocínio (explanação) e de entrada de dados.

As primeiras tentativas de automatizar o processo de aquisição de conhecimento foram iniciadas em 1982, quando Davis desenvolveu o sistema TEIRESIAS (Davis et al., 1982), que auxilia o engenheiro do conhecimento a escrever e depurar as regras de produção. O TEIRESIAS faz a edição e o teste de consistência de regras, porém não faz a geração automática de conhecimento.

Em 1984, foi desenvolvido por J. BOOSE (Boose, 1984) o sistema ETS ("Expertise Transfer System"). O ETS é baseado em teorias psicológicas ("Personal Construct Theory"). Esta ferramenta descreve os elementos do conhecimento e suas características. As regras geradas pelo ETS possuem fatores de certeza. Porém, para gerar as regras, o ETS entrevista o perito de forma muitas vezes cansativa. O ETS foi desenvolvido na Boeing Computer Services como um sistema de pesquisa. Uma limitação do sistema é o fato de todas os atributos das regras geradas permitirem apenas dois valores (verdadeiro ou falso), quando o ideal seria o uso de qualquer número de valores discretos ou mesmo contínuos.

Paralelamente ao ETS surgiram os sistemas que constroem automaticamente árvores de decisão. Estes sistemas permitem a extração de regras a partir de exemplos (casos passados no cotidiano do perito), através de um processo indutivo. Atualmente as pesquisas em indução de árvores de decisão visam a geração rápida de uma BC consistente. Várias aplicações do processo indutivo foram implementadas recentemente: o sistema de diagnóstico de falhas para circuitos impressos da ITT Europe, o sistema de separação gás-óleo da British Petroleum e o sistema de avaliação da performance do motor do ônibus espacial da NASA (Carter e Catlett, 1987).

Este trabalho descreve o desenvolvimento do **APREND**, um sistema de aquisição automática de conhecimento a partir de exemplos, utilizando indução de árvores de decisão.

O APREND é uma ferramenta de auxílio do engenheiro do conhecimento. O seu objetivo é o de facilitar o processo de formação da BC, tornando a aquisição de regras mais rápido e eficiente, reduzindo a necessidade de intervenções do perito.

O sistema possui dois grandes módulos principais, o ID3 e o ATM. O engenheiro de conhecimento usará primeiro o ID3 e depois o ATM, obtendo respectivamente regras completas (com fator de certeza 1,0) e regras atômicas (com fator de certeza no intervalo [-1 +1]). Porém, em domínios acentuadamente não determinísticos, o engenheiro de conhecimento usará somente o ATM, pois o ID3 não se adapta bem a estes domínios.

O ID3 foi escolhido, porque possui três características desejáveis num sistema de aquisição automática de conhecimento:

(1) Utiliza exemplos como dados de entrada. Os exemplos são dados

disponíveis no dia-a-dia do perito, são fáceis de serem coletados e permitem um tratamento probabilístico;

(2) Tem complexidade baixa, da ordem do número de conclusões finais possíveis vezes o número de conclusões intermediárias possíveis; e

(3) Representa o conhecimento adquirido na forma de regras de produção, que é o formalismo mais natural, adequado e adotado para a maioria dos SE's.

O ID3 apresenta algumas deficiências, principalmente por que não apresenta resultados garantidos para os domínios acentuadamente não determinísticos.

O módulo ATM, que pode ser usado independentemente do ID3, se adequa bem aos domínios acentuadamente não determinísticos e foi desenvolvido para o APREND com o objetivo de superar a maior parte das deficiências do ID3. O ATM se baseia em estatística indutiva.

Organização da Dissertação :

O capítulo 2 localiza o APREND dentro do universo dos sistemas de aquisição de conhecimento.

O capítulo 3 apresenta o algoritmo ID3, abrangendo as suas características, vantagens e desvantagens.

No capítulo 4 apresenta-se o ATM, método estatístico concebido para lidar com a incerteza, e os critérios para escolher as boas regras que farão parte da BC.

No capítulo 5 serão mostrados os aspectos de projeto, implementação e uso do sistema, tais como, gráfico de estruturas, estruturas de dados e um guia de uso.

O capítulo 6 apresenta uma ilustração do processo de aquisição de conhecimento através do APREND, mostrando os dados e os resultados obtidos em uma aplicação em nefrologia.

Finalmente, no capítulo 7 apresentam-se as conclusões e alguns trabalhos a serem desenvolvidos futuramente.

Os programas fonte, bem como a coleção completa dos exemplos, regras completas e resultados, referentes ao capítulo 6, estão à disposição dos interessados mediante solicitação ao Grupo de Inteligência Artificial/DSC/UFPB - Campina Grande.

2. LOCALIZAÇÃO DO APREND

A habilidade de aprender é uma característica central da inteligência humana. Os humanos são capazes de aprender a partir de fontes de informação que se apresentam de diversas formas, e com diferentes níveis de precisão e relevância.

Um dos principais objetivos da Inteligência Artificial (IA), na tentativa de imitar a inteligência humana, é implantar na máquina características de aprendizado. Esta é uma tarefa complexa e as pesquisas na área de aprendizado da máquina estão apenas começando. Porém, com o desenvolvimento de SE's, tornou-se urgente implementar algum modo de aprendizado na máquina, mesmo que seja de uma forma limitada.

Desta forma, os primeiros sistemas comerciais de aprendizado já começam a ser apresentados, com o objetivo de diminuir o tempo que se leva para construir a base de conhecimento (BC) de um SE, no processo de aquisição de conhecimento.

2.1 Métodos de Aquisição de Conhecimento

Os métodos de aquisição de conhecimento podem ser classificados em: cognitivos e automatizados (Boy et al., 1987).

Métodos Cognitivos

Os métodos cognitivos são caracterizados pela presença de um mediador entre o perito e o SE, chamado de engenheiro de

conhecimento.

A maior dificuldade dos métodos cognitivos, reside no fato de que o engenheiro de conhecimento deve ser um bom analista de sistemas e deve ter ou ser capaz de alcançar um bom nível de conhecimento do domínio estudado. Evidentemente, pessoas com essas características são raras e caras. Ademais o processo é lento e dispendioso.

Atualmente, as pesquisas são realizadas por psicólogos interessados no processo de aprendizado cognitivo.

Entre os métodos cognitivos, os mais usados são: entrevistas, análise de protocolos e observação direta.

Métodos Automatizados

Os métodos automatizados são aqueles onde a máquina consegue extrair do ambiente que a envolve, o conhecimento necessário ao funcionamento de um SE.

Na aprendizagem da máquina, assim como na aprendizagem humana, a fonte de informação pode se apresentar de diversas formas. O volume de inferência envolvida na aquisição de conhecimento depende da estratégia de aprendizado que se está utilizando.

2.2 Estratégias Fundamentais de Aprendizado

São seis as estratégias fundamentais de aprendizado, seja ele cognitivo ou automatizado.

Aprendizado por Repetição (rote learning)

Repetição ou imitação não requer qualquer forma de

inferência ou transformação de conhecimento por parte do aprendiz. Esta é a estratégia onde o aprendiz extrai o conhecimento a partir de imitação ou memorização.

Um programa algorítmico convencional, está enquadrado nesta estratégia, pois todo o conhecimento pertence ao programador, que simplesmente o transfere para o programa.

Aprendizado a Partir de Instrução (learning by being told)

Esta forma de aprendizado requer a transformação da linguagem de entrada numa linguagem usada internamente. É a forma mais usual de aprendizado humano e é feita a partir de professores, livros, publicações e fontes similares. Quando um aluno está tentando aprender em sala de aula, ele está transformando a linguagem do professor numa linguagem interna.

Nos métodos cognitivos de aquisição de conhecimento usa-se esta forma de aprendizado nas entrevistas com o perito.

Um sistema com esta abordagem seria capaz de aceitar instrução, transformar esta instrução numa representação interna e aplicar o conhecimento aprendido.

Podem-se classificar os sistemas que usam entrevistas automáticas (por exemplo, o ETS - Expertise Transfer System) como sistemas que utilizam esta forma de aprendizado.

Aprendizado por Dedução

No aprendizado por dedução, o conhecimento é adquirido a partir da reestruturação de um dado domínio numa forma útil e efetiva. Portanto requer, por parte do aprendiz, um raciocínio dedutivo.

O ser humano está constantemente reestruturando conhecimentos adquiridos anteriormente. Neste caso, o aprendizado por dedução requer conhecimentos acumulados que começaram a ser adquiridos a partir do nascimento.

Um sistema que usa esta forma de aprendizado, tem que acessar uma grande quantidade de dados no domínio, que dizem respeito ao conceito que está sendo aprendido e a outros conceitos correlatos. Estes sistemas chamam-se EBS ("Explanation-Based Systems").

Aprendizado por Analogia

Esta estratégia envolve a extensão de um conhecimento já existente, com o objetivo de aplicar o conhecimento resultante a um domínio similar ao anteriormente aprendido.

Por exemplo: um experiente esquiador da neve, usaria toda a sua habilidade, já adquirida, para aprender a esquiar na água, pois as duas atividades são similares.

Aprendizado por Indução

O processo indutivo é feito a partir de observações específicas de um certo conceito. Através da indução, consegue-se extrair uma descrição geral do conceito observado.

O aprendizado indutivo pode ser subdividido em dois tipos: aprendizado a partir de exemplos (learning from examples) e aprendizado por observação e descoberta (learning from observation and discovery).

Dentro da categoria de aprendizado a partir de exemplos distinguem-se dois tipos de generalização: instância-classe e parte-global.

Instância-classe, caracteriza um sistema de aprendizado que induz uma descrição geral de uma classe quando entra-se com uma coleção de descrições específicas daquela classe. As descrições específicas (instâncias ou exemplos) podem ser representações de objetos físicos, sons, imagens, ações, sintomas, etc.

A generalização instância-classe permite duas formas de se chegar às descrições gerais de um objeto: dirigido pelos dados ("data-driven") e dirigido para o modelo ("model-driven") (El-Khomi, 1988). No método "data-driven" o conjunto de dados usado como entrada não possui, a priori, nenhuma descrição geral das classes envolvidas. No método "model-driven", o conjunto de exemplos usados como entrada, já passou por um processo anterior de generalização. Este processo anterior pode ser o fornecimento de regras de uma forma cognitiva, que serão posteriormente generalizadas através do método "model-driven". O programa INDUCE (Dietterich e Michalski, 1981) se utiliza do modo "model-driven" com o objetivo de generalizar as regras fornecidas pelo perito.

Na generalização parte-global, o objetivo é conceituar um objeto, a partir de exemplos que representam partes do objeto. O sistema SPARC (Dietterich e Michalski, 1986) se utiliza de indução parte-global, para gerar regras.

Aprendizado por Observação

A estratégia de aprendizado por observação consiste em desenvolver uma descrição geral (uma teoria) a partir de fatos observados. Um sistema que utiliza esta abordagem, não trabalha com uma coleção de exemplos associados a classes, mas usa como

entrada uma coleção de fatos dispersos que, a priori, não possuem classes.

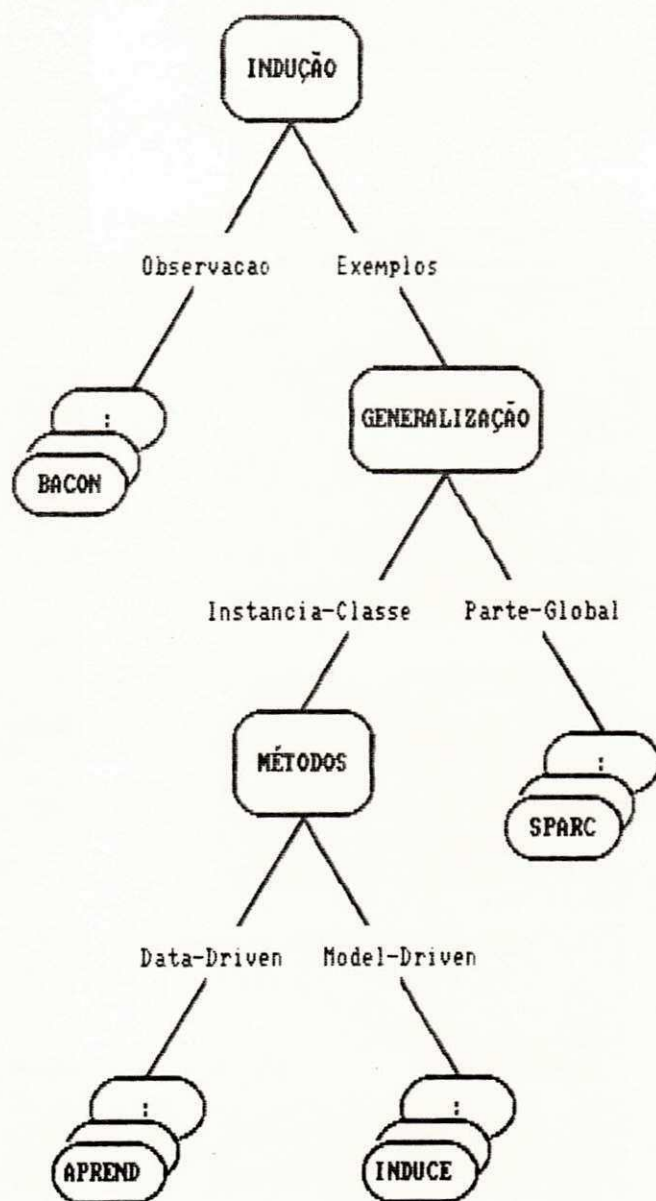


Figura 2.1 - Classificação de sistemas que usam estratégias indutivas de aprendizado.

O primeiro passo no desenvolvimento de uma teoria sobre uma coleção de fatos é a criação de uma taxonomia, que classifica os fatos observados e é considerada uma descrição geral destes fatos.

O sistema BACON (Langley et al., 1983) emprega um pequeno conjunto de heurísticas que detectam características semelhantes em dados numéricos e simbólicos.

A figura 2.1 apresenta a classificação de sistemas que se utilizam de estratégias indutivas de aprendizado.

2.3 O Sistema APREND

O APREND se utiliza de métodos de aprendizagem indutiva. O tipo de generalização usada é o instância-classe e o método é o "data-driven" (veja a figura 2.1).

O sistema gera, através do algoritmo ID3, o conhecimento na forma de regras completas (RC's), com o Fator de Certeza (FC) 1,0 (Vide Capítulo 4). Através do ATM, sistema fornece regras que possuem apenas uma condição, com o FC, "a la MYCIN" (Buchanan e Shortliffe, 1984), variando no intervalo [-1 1], o que se chamou de "Regras Atômicas (RA's)".

3. O ALGORITMO ID3

O algoritmo ID3 foi criado por John Ross Quinlan (Quinlan, 1986a). O algoritmo faz parte de uma família de algoritmos de classificação, chamada de TIDT ("Top-Down Induction of Decision Trees"). O ID3 é um descendente do CLS ("Concept Learning System"), que foi apresentado por Hunt, Stone e Marin em 1966 (Hunt et al., 1966).

A estratégia usada é o aprendizado não incremental a partir de exemplos. Ao sistema é apresentado um conjunto de casos relevantes para a tarefa de classificação e desenvolvimento de uma Árvore de Decisão (AD), da raiz para as folhas. O sistema é guiado pela frequência da informação e não pela ordem em que ela é apresentada.

3.1 Apresentação da tabela de exemplos

O conjunto de exemplos é formado de pares atributo-valor. Cada par atributo-valor registra uma característica importante de um objeto. Os valores possíveis a cada atributo devem ser discretos e mutuamente exclusivos.

A cada objeto associa-se uma classe. Essa classe pode ser um diagnóstico ou um conceito a ser classificado. As classes devem ser mutuamente exclusivas.

Como ilustração, apresenta-se a caracterização de um exemplo. Para classificar o conceito de um objeto, usam-se os pares atributo-valor:

Preço	Constituição	Num. de pernas	Classe
Barato	Madeira	Quatro	Mesa

No exemplo acima foi caracterizado o objeto "Mesa", para isso foram usados três pares atributo-valor.

A figura 3.1 apresenta um conjunto de exemplos positivos e negativos (as classes podem ser "Mesa" e "Não-Mesa"), com relação ao conceito de "mesa".

PREÇO	CONSTITUIÇÃO	NUM. DE PERNAS	CLASSI
1. Barato	Metal	Quatro	Mesa
2. Barato	Metal	Duas	Não-Mesa
3. Barato	Animal	Quatro	Não-Mesa
4. Caro	Metal	Duas	Não-Mesa
5. Caro	Madeira	Quatro	Mesa
6. Caro	Animal	Quatro	Não-Mesa
7. Caro	Metal	Quatro	Mesa
8. Caro	Animal	Duas	Não-Mesa

Figura 3.1 - Conjunto de Exemplos

3.2 Formação aleatória de uma AD

A partir da tabela de exemplos da figura 3.1, pode-se construir aleatoriamente uma AD. Para isto, basta que se escolha

qualquer um dos atributos para compor a raiz da árvore. Cada ramo da AD representa um valor do atributo (nó) correspondente.

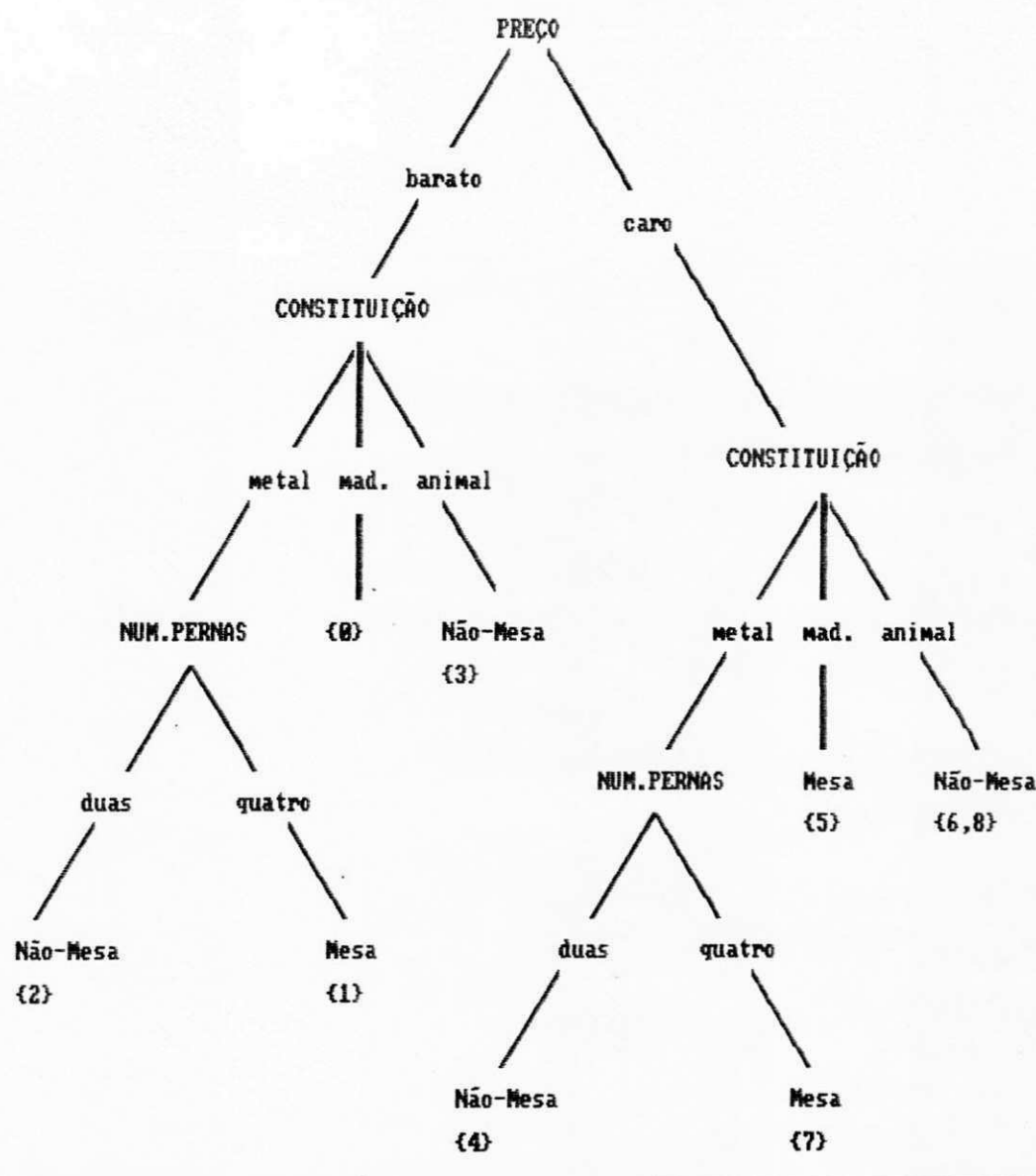


Figura 3.2 - Árvore de decisão construída aleatoriamente a partir da tabela de exemplos da figura 3.1

Para se formar as sub-árvores, é necessário que se escolha aleatoriamente um novo atributo, excluindo-se os atributos já usados.

Quando se verifica que existe uma ou nenhuma classe que rotula corretamente uma folha, para-se de ramificar a AD.

A AD estará totalmente construída, quando todos os exemplos estiverem classificados.

Quando dois exemplos possuem os mesmos valores para todos os atributos e as classes diferem, diz-se que estes exemplos são inadequados. Para que se possa executar o ID3, os exemplos inadequados devem ser retirados da tabela. Os exemplos inadequados são originários de exemplos atípicos ou contra-exemplos.

A figura 3.2 representa uma AD construída de forma aleatória, que classifica todos os exemplos da tabela da figura 3.1. Os números dos exemplos classificados por cada caminho da AD, estão entre chaves.

Na árvore da figura 3.2 escolhe-se, ao acaso, o atributo "Preço" como raiz. O atributo "Preço" possui dois valores: "barato" e "caro". Na tabela da figura 3.1 tem-se três exemplos (1, 2 e 3) onde "Preço"="barato". O exemplo 1 possui a classe "Mesa", porém os exemplos 2 e 3 possuem a classe Não-Mesa. Desta forma, escolhe-se outro atributo aleatório, até que se tenha apenas uma classe para rotular a folha. Pode ocorrer que um caminho da AD não classifique nenhum exemplo. Neste caso, rotula-se a folha com a classe $\{\emptyset\}$ ou "Nulo".

A escolha aleatória da ordem dos atributos, geralmente produz uma AD de uso ineficiente.

3.3 Indução de AD's usando ID3

A tarefa do ID3 é produzir uma AD de tamanho reduzido e uso eficiente, que classifique todos os exemplos apresentados (outra forma de se obter tal AD poderia ser a geração de todas as AD's possíveis e, em seguida, escolher-se a melhor, entretanto, isto seria impraticável por ocupar muito tempo e espaço).

Chama-se o conjunto de todos os exemplos apresentados ao ID3, de "conjunto de trabalho".

O ID3 consegue produzir AD's simples usando apenas um subconjunto do conjunto de trabalho e aplicando o cálculo da entropia (usado na Teoria da Informação) para escolher os atributos que serão as raízes das AD's e sub-AD's.

O subconjunto do conjunto de trabalho usado para construir a AD é chamado de "conjunto janela".

A figura 3.3 mostra o algoritmo ID3, que será descrito a seguir.

3.3.1 Obtenção do conjunto janela

O primeiro passo para se construir a AD é a obtenção do conjunto janela, escolhendo-se aleatoriamente 10% dos elementos do conjunto de trabalho (Quinlan, 1987a).

A AD é gerada a partir dos elementos do conjunto janela e, como consequência, classifica todos os elementos deste conjunto.

Utilizam-se subconjuntos (sub-janelas) do conjunto janela para se construir as sub-AD's.

Algoritmo ID3

Procedimento Construir_arvore

Inicio

Repita

Obter_o_conjunto_janela

Obter_o_atributo_de_menor_entropia

Colocar_na_raiz_o_atributo_de_menor_entropia

Se ainda_ramifica Entao Construir_Arvore

Ate nao_ramifica

Rotular_folha

Fim

Inicio

Repita

Enquanto nao_fim_de_arvore faca

Construir_arvore

Se nao_classifica_conjunto_trabalho Entao

Incluir_exemplos_nao_classificados_no_conjunto_janela

Ate arvore_classificar_o_conjunto_de_trabalho

Fim

Figura 3.3 - O algoritmo ID3

3.3.2 Cálculo da entropia de um atributo

A entropia mede a quantidade de informação contida em um atributo (Klir e Folger, 1988).

Quanto menor for a entropia de um atributo, menor será o seu grau de incerteza. Portanto, o atributo de menor entropia

é o que possui maior quantidade de informação e sempre será escolhido para ser a raiz de uma AD.

A entropia (H) de um atributo (A) é a média ponderada das entropias dos valores de A . A entropia de um valor v_i de A é dada por:

$$H(A=v_i) = \sum_{j=1}^M -P(C_j|v_i) \cdot \log_2 P(C_j|v_i)$$

onde M é o número de classes da tabela de exemplos e $P(C_j|v_i)$ é a probabilidade de se ter, no conjunto janela, um exemplo com o valor v_i , se a classe é C_j . A entropia de A é dada por:

$$H(A) = \sum_{i=1}^N \frac{nv_i}{n} \cdot H(A=v_i)$$

onde N é o número de valores do atributo A , nv_i é o número de exemplos que possuem o valor v_i e n é o número de elementos do conjunto janela.

Como ilustração, apresenta-se o cálculo da entropia do atributo "Constituição", considerando-se a figura 3.1 como sendo o nosso conjunto janela. O primeiro passo é calcular a entropia de cada valor desse atributo:

$$H(\text{Constituição}=\text{Metal}) = -\frac{2}{4} \log_2 \frac{2}{4} - \frac{2}{4} \log_2 \frac{2}{4} = 1,0$$

$$H(\text{Constituição}=\text{Madeira}) = -\frac{1}{1} \log_2 \frac{1}{1} = 0,0$$

$$H(\text{Constituição=Animal}) = - \frac{3}{3} \log_2 \frac{3}{3} = 0,0$$

a entropia de "Constituição" será a média ponderada dos seus valores:

$$H(\text{Constituição}) = \frac{4}{8} \cdot 1,0 + \frac{1}{8} \cdot 0,0 + \frac{3}{8} \cdot 0,0 = 0,50$$

calculando-se as entropias de todos os atributos, teremos:

$$H(\text{Preço}) = 0,10 \text{ e}$$

$$H(\text{Num. de Pernas}) = 0,05$$

Desta forma, "Num. de Pernas" é o atributo de menor entropia e, conforme a figura 3.4, será a raiz da AD gerada pelo ID3.

Apenas 3 exemplos no conjunto janela possuem o valor "Duas" (2,4,8). Todos eles estão associados à classe "Não-Mesa". Assim, não será necessário ramificar a sub-árvore e a folha será rotulada com a classe "Não-Mesa".

Existem 5 exemplos onde "Num. de Pernas"="Quatro" (1,3,5,6,7). Porém eles estão associados a classes diferentes. Portanto, deve-se ramificar recursivamente a sub-árvore. A raiz da sub-árvore será o atributo de menor entropia. Neste caso, a sub-janela será o conjunto dos exemplos que possuem o valor "Quatro".

Comparando-se a AD obtida aleatoriamente (figura 3.2), com a AD fornecida pelo ID3 (figura 3.4), nota-se que a segunda é uma árvore menor e de uso mais eficiente. Na AD fornecida pelo ID3 não foi usado o atributo "Preço". Isto mostra que, de acordo com a tabela de exemplos apresentada, o atributo "Preço" não é

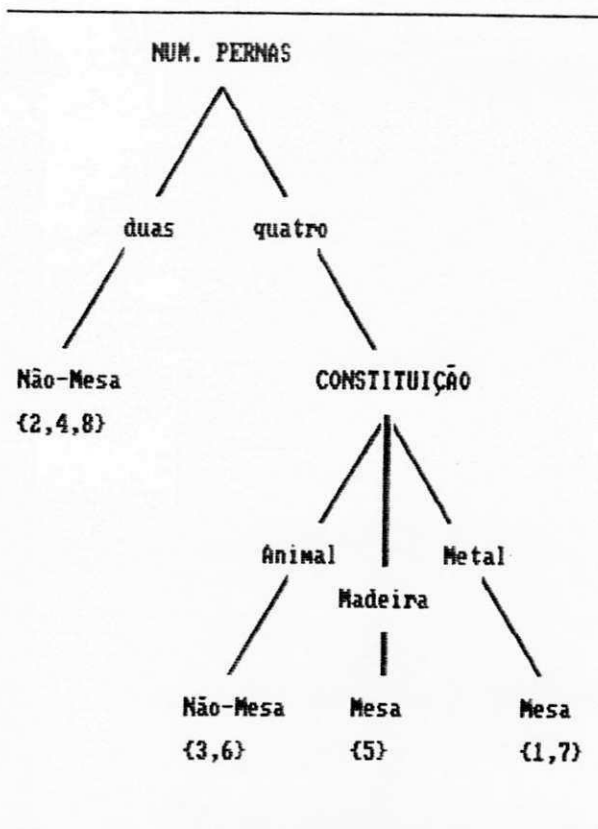


Figura 3.4 - Árvore de Decisão gerada pelo ID3 a partir dos exemplos da figura 3.1

relevante para se conceituar "mesa".

Para obter as regras, a partir da AD, basta percorrer todos os caminhos no sentido raiz-folha. Na AD da figura 3.4, obtêm-se 4 regras:

1. Se Num. de Pernas=duas Então Não-Mesa;
2. Se Num. de Pernas=quatro & Constituição=Metal Então Mesa;
3. Se Num. de Pernas=quatro & Constituição=Animal Então Não-Mesa;
4. Se Num. de Pernas=quatro & Constituição=Madeira Então Mesa.

A complexidade computacional do algoritmo é $O(C \cdot |A|)$, onde $|A|$ é o número de atributos e C é o número de elementos no conjunto de trabalho (Quinlan, 1986a).

3.4 Vantagens do ID3

(1) O ID3 obtém regras que são geradas a partir da experiência de um perito no domínio, devido a isto, as regras obtidas são confiáveis.

(2) As regras obtidas pelo ID3 estão livres de inconsistências (Gascuel, 1988), pois o processo de formação da AD não permite a ocorrência de redundância, conflito, circularidade de regras e regras desnecessárias. Isto elimina o processo de verificação e ratificação no que diz respeito a redundância e conflito de regras, regras desnecessárias e regras circulares (Nguyen et al., 1987).

(3) Outra grande vantagem do algoritmo é a sua facilidade de implementação e execução, principalmente em ambientes que permitam recursividade.

3.5 Desvantagens do ID3

(1) O ID3 é aplicado apenas a domínios onde tanto as classes como os valores dos atributos são mutuamente exclusivos. Porém, sabe-se que diversos domínios do conhecimento possuem diagnósticos não mutuamente exclusivos. Tomando como ilustração os domínios médicos, frequentemente ocorre de um paciente ter, ao mesmo tempo, a confirmação de dois diagnósticos diferentes. Por exemplo: gripe e pneumonia, ao mesmo tempo.

(2) O ID3 falha na sua tentativa de construir a AD quando, entre os exemplos, existe uma grande quantidade de contra-exemplos,

caracterizando um domínio não determinístico. Desta forma, quando o domínio é acentuadamente não determinístico, não se recomenda o uso do ID3.

(3) Durante a descrição dos exemplos, pode-se cometer erros de digitação, indefinição de valores (Quando uma pessoa é alta?) e ausência de valores. Estes erros constituem-se "ruídos" na tabela de exemplos. Originalmente, o ID3 não lida com domínios cujos exemplos contenham ruído, porém o APREND incorpora algumas facilidades, descritas em 3.6, planejadas para contornar alguns tipos de ruído.

(4) Para o ID3, a repetição de exemplos exatamente iguais, não é levada em conta, i.e. será construída a mesma AD caso um exemplo apareça apenas uma vez ou centenas de vezes na tabela. Isto acarreta a geração de regras opacas (Quinlan, 1987b), que regras que não constituem conhecimento utilizável.

(5) As regras geradas possuem o $FC=1,0$, ou seja, regras ideais e completas, quando sabemos que o mundo real possui características de aproximação e incerteza.

3.6 Modificações efetuadas pelo APREND no ID3 original

Para minimizar as desvantagens apresentadas pelo ID3, o APREND incorporou algumas modificações ao referido algoritmo. As principais modificações são:

(1) Quando o domínio aplicado é acentuadamente não determinístico, o usuário do APREND pode e deve usar só o módulo ATM, descrito no próximo capítulo, gerando regras atômicas e com alguma iteração manual, regras compostas, com FC no mesmo

intervalo das regras atômicas $([-1 \ 1])$.

(2) O cálculo da quantidade de elementos do conjunto janela foi alterado, como apresentado na subseção 5.1.2, visando uma maior rapidez no processo de construção da AD.

(3) Para eliminar as regras opacas, o APREND associou às regras completas, a influência de cada condição, com relação à conclusão da regra. Ao mesmo tempo, foram apresentados critérios para se identificar o que chamamos de "boas regras completas". Vide a seção 4.3.

4. TRATAMENTO DA INCERTEZA NAS REGRAS

SE's devem ser capazes de inferir a partir de conhecimento aproximado, a fim de solucionar um grande número de problemas do mundo real. O FC representa a característica de incerteza inerente a um determinado domínio do conhecimento.

As RA's nos revelam a influência que cada par atributo/valor exerce sobre uma classe. Esta influência é dada pelo FC da regra atômica.

No módulo ID3 as RC's são geradas sem incerteza, i.e. $FC=1,0$. O módulo ATM gera RA's com FC's "a la MYCIN (Buchanan e Shortliffe, 1984)".

4.1 Revisão de regras com tratamento da incerteza "a la MYCIN"

Uma regra de produção é uma expressão da forma:

SE Condições ENTÃO Conclusões

Cada condição ou premissa é a descrição de uma situação representada dentro do formalismo da lógica proposicional ou da lógica de predicados. As conclusões são ações que deverão ser tomadas, caso as premissas sejam verdadeiras.

O sistema MYCIN (Buchanan & Shortliffe, 1984), associa a cada regra residente na sua BC, um Fator de Certeza (FC). Este FC é um número dentro do intervalo $[-1 +1]$, que indica o grau com que as condições da regra confirmam ($FC > 0$), ou desconfirmam ($FC < 0$) a conclusão.

Se o FC é -1, diz-se que a(s) evidência(s) (condições

da regra) desconfirma(m) totalmente a hipótese (conclusão da regra). Se o cálculo nos revela $FC = 0$, a(s) evidência(s) não confirma(m) nem desconfirma(m) a hipótese. Quando o FC é $+1$, a(s) evidência(s) confirma(m) totalmente a hipótese. Se o FC é um número não inteiro, significa que a hipótese é confirmada ou desconfirmada, com aproximação, i.e. em proporção ao valor do FC .

A aproximação implantada pelo MYCIN é muito útil em SE's porque grande parte dos domínios do conhecimento, no mundo real, é aproximada.

Alguns SE's comerciais disponíveis, se utilizam apenas de incerteza positiva, i.e. o FC varia no intervalo $[0 \ 1]$, onde $FC = 0,5$ indica que a hipótese não é confirmada nem desconfirmada. Estes sistemas não conseguem exprimir muito bem a desconfirmação de uma hipótese, já que todos os FC 's são positivos. A incerteza "a la MYCIN" exprime a desconfirmação com FC 's negativos, isto revela uma boa associação com o mundo real e acarreta uma maior percentagem de acertos com um menor número de perguntas (veja o capítulo 6).

4.2 Importância das RA's

Suponha que existe na base de conhecimento duas regras, uma completa (R1) e uma atômica (R2):

R1 - Se a & b & c Então d ($FC = 1,0$) e

R2 - Se c Então d ($FC = 0,4$)

se o fato "a" fosse negado ($FC = -1,0$), a regra R1 não poderia

ser usada. Porém, a regra R2 garante o FC 0,4 para d, mesmo que a regra R1 não seja aplicada. Isto significa que as RA's contribuem com a propagação do seu FC, mesmo que as RC's não sejam aplicadas.

Com a introdução do maior número possível de RA's, na BC, temos uma grande flexibilidade na propagação dos FC's.

Para cada classe, o número possível de RA's é igual ao número de pares atributo-valor. Entretanto, deve-se excluir algumas RA's da base de conhecimento. Esta exclusão, deve seguir os critérios que serão apresentados na seção 4.2.

Outra função importante das RA's, é a indicação das RC's (geradas pelo ID3), que serão consideradas boas e farão parte da base de conhecimento. O critério de escolha das RC's, será mostrado na seção 4.3.

Os critérios de escolha e rejeição de uma regra ainda não estão implementados com o APREND, portanto trata-se de critérios a serem aplicados manualmente pelo engenheiro de conhecimento e pelo(s) perito(s).

4.3 Critérios para a escolha de RA's

Cada domínio do conhecimento apresenta características particulares, no que diz respeito à incerteza. Devido a isto, os critérios de escolha das boas regras são subjetivos e dependem da orientação dada pelo perito. Portanto, a presença do perito no domínio estudado é necessária e obrigatória.

Existem dois critérios para se excluir uma RA da base de conhecimento.

O primeiro critério diz respeito a uma RA isoladamente. Para cada domínio do conhecimento e, às vezes, para cada classe desse domínio, deve ser estabelecido um limiar de aceitação do FC. Isto ocorre porque as regras que possuem FC muito próximo de 0,0, não confirmam, nem desconfirmam com significância um diagnóstico, podendo ser omitidas da BC.

O segundo critério de exclusão de RA's diz respeito à importância de uma RA com relação a todos os valores de um mesmo atributo, considerando-se todas as classes. Se, para todos os valores de um mesmo atributo, a diferença entre o maior e o menor FC for inferior ao limiar de aceitação, todas as RA's correspondentes ao atributo focalizado, deverão ser excluídas. Qualquer que seja o valor do atributo, o FC será praticamente o mesmo. Isto significa que o atributo como um todo não influencia as classes na tabela de exemplos.

Como ilustração, considere o atributo idade, com os valores (novo, velho). O atributo idade produzirá duas RA's para cada classe, pois possui dois valores possíveis. Para um conjunto fictício de exemplos tem-se como classe os diagnósticos gripe e pneumonia. O atributo idade produziu, as seguintes RA's:

R3 - Se Idade = novo Então Gripe (FC = 0,40),

R4 - Se Idade = velho Então Gripe (FC = 0,41),

R5 - Se Idade = novo Então Pneumonia (FC = 0,39) e

R6 - Se Idade = velho Então Pneumonia (FC = 0,40).

Nota-se que o atributo idade, para qualquer valor, terá os FC's muito próximos. Portanto, é irrelevante saber se o paciente é novo ou velho. Desta forma, as quatro regras (R3 a R6) deverão ser excluídas.

4.4 Critério de escolha das RC's

Para cada condição da RC, será mostrada a sua influência com relação à conclusão da regra. Esta influência é igual ao FC da RA que possui o mesmo par atributo-valor da condição.

Uma RC não deverá ser incluída na BC se a combinação das influências das suas condições resultar no valor 1,0 ou próximo de 1,0, de acordo com o perito. Isto é: a combinação das RA's que possuem as condições correspondentes às condições desta RC (mesmo par atributo-valor) já fornece um valor igual, ou próximo de 1,0 (FC da RC). Mesmo se todas as condições desta RC forem verdadeiras, ela propagará um FC próximo ou igual ao FC relativo à combinação das RA's correspondentes. Portanto torna-se desnecessária a presença desta RC na BC.

De forma análoga, uma RC é considerada boa quando a combinação das influências (Inf) das suas condições resulta num valor muito abaixo de 1,0 (de acordo com o perito). Isto ocorre porque a combinação dos FC's das RA's correspondentes é muito abaixo de 1,0. Portanto esta RC, se todas as suas condições forem verdadeiras, propagará um FC = 1,0. Enquanto que a combinação das RA's correspondentes propagará um FC bem menor.

O processo de combinação dos FC's no MYCIN é um caso particular da Teoria das Evidências de Dempster-Shafer (Buchanan e Shortliffe, 1984). Quando duas regras (RA's) independentes confirmam uma mesma hipótese a combinação dos FC's é dada pela expressão:

$$FC = FC1 + (1 - FC1) \cdot FC2 \quad (4.1)$$

A combinação das influências de uma RC é dada pela expressão 4.1.

Considere a RC:

R7 - Se $a = x$ & (Inf = 0,20)

$b = y$ & (Inf = 0,19)

$c = z$ (Inf = 0,25)

Então d (FC = 1,0)

A combinação das influências das condições resulta em 0,51, que é menor que o FC da regra (1,0). Portanto a regra R7 deverá ser incluída na BC.

A regra R8, mostrada a seguir, é o exemplo de uma regra que não deverá ser incluída na BC.

R8 - Se $e = k$ & (Inf = 0,80)

$f = l$ & (Inf = 0,60)

$g = w$ (Inf = 0,55)

Então h (FC = 1,0)

A combinação das influências da regra R8 resulta em 0,96. Portanto não será necessária a presença da R8 na BC, pois as RA's, correspondentes às condições da R8, irão propagar um valor muito próximo de 1,0.

Mesmo que a inclusão de uma RC seja considerada desnecessária pelo critério apresentado, a palavra final deve ser do perito. Por exemplo: uma regra que, pelos critérios supra citados, é considerada má, poderá ser mantida na BC, se o perito considerar esta regra importante.

4.5 Processo de cálculo dos FC's

No MYCIN o FC de uma evidência com relação a uma hipótese, é encontrado a partir da expressão:

$$FC = \frac{MC - MD}{1 - \text{Min}(MC, MD)} \quad (4.2)$$

onde MC é a medida de crença, que indica o aumento da crença em uma hipótese, devido a uma certa evidência. E o MD é a medida de descrença, que indica o aumento da descrença em uma hipótese, dada a mesma evidência.

O cálculo do MC e MD é feito através da "comparação de duas proporções" (Costa Neto, 1977).

4.5.1 Cálculo do MC

Seja a regra: Se A=a Então C, onde A=a é o par atributo(A)-valor(a) que representa a condição e C é a classe que representa a conclusão da regra. E seja n_1 o número total de elementos do conjunto de trabalho, f_1 a frequência com que ocorre a classe C, n_2 o número elementos onde se verifica o par A=a e f_2 a frequência com que ocorre a classe C quando se verifica o par A=a. Assim, MC é dado pela expressão :

$$MC = \frac{f_1}{n_1} \cdot (1 - \frac{f_2}{n_2})$$

onde $\frac{f_2}{n_2}$ é a proporção com que ocorre C quando se verifica o par

A=a, ou seja:

$$P = \frac{f_2}{n_2}$$

e P é encontrado na tabela de probabilidades da distribuição normal de Gauss. O parâmetro z usado na tabela de Gauss é dado por:

$$z = \frac{P_1 - P_2}{\sqrt{B \cdot (1 - B) \cdot \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

onde P₁ é a proporção com que ocorre C no conjunto de trabalho, ou seja:

$$P_1 = \frac{f_1}{n_1}$$

e B é dado por:

$$B = \frac{f_1 + f_2}{n_1 + n_2}$$

4.5.2 Cálculo do MD

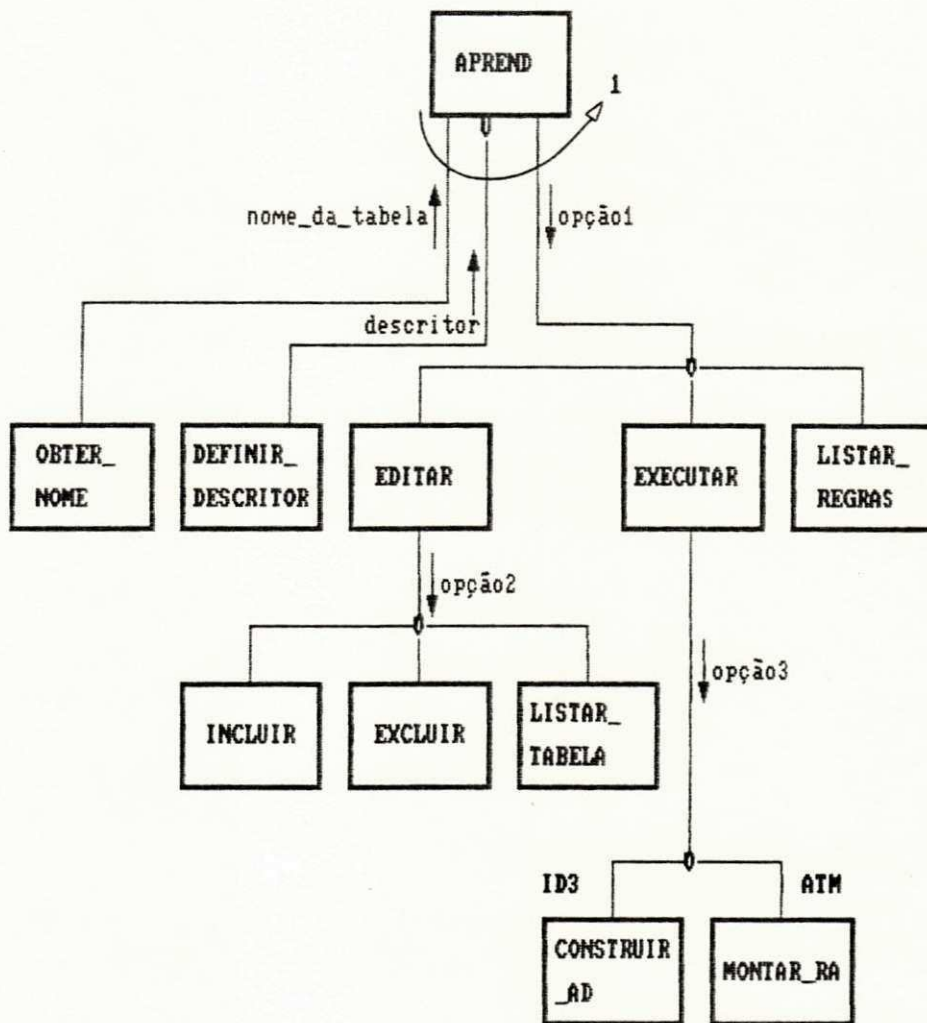
Para se calcular o MD procede-se da mesma forma que o MC, porém modificando-se apenas as definições de f₁ e f₂. Deste modo, f₁ é a frequência com que ocorre a classe ~C (qualquer

classe que não seja C) e f é a frequência com que ocorre a classe $\sim C$ quando se verifica $R=a$.

De posse dos valores de MC e MD, aplica-se a expressão 4.2 e encontra-se o FC.

5. PROJETO, IMPLEMENTAÇÃO E USO DO APREND

O projeto do sistema APREND foi dividido em três etapas: formação da tabela de exemplos, construção da AD e montagem das RA's.



1. Enquanto opção1 ≠ ESC faça

Figura 5.1 - Gráfico de Estrutura do APREND

5.1 Etapas de projeto

A figura 5.1 apresenta o Gráfico de Estrutura (GE) do APREND. O módulo OBTER_NOME recebe o nome de uma tabela e verifica se ela está no diretório. Se a tabela for nova, o módulo DEFINIR_DESCRITOR é acionado. O módulo DEFINIR_DESCRITOR solicita a descrição dos pares atributo-valor e das classes envolvidas. Os módulos EDITAR, CONSTRUIR_AD e MONTAR_RA, correspondem às três fases citadas acima.

5.1.1 Edição de exemplos

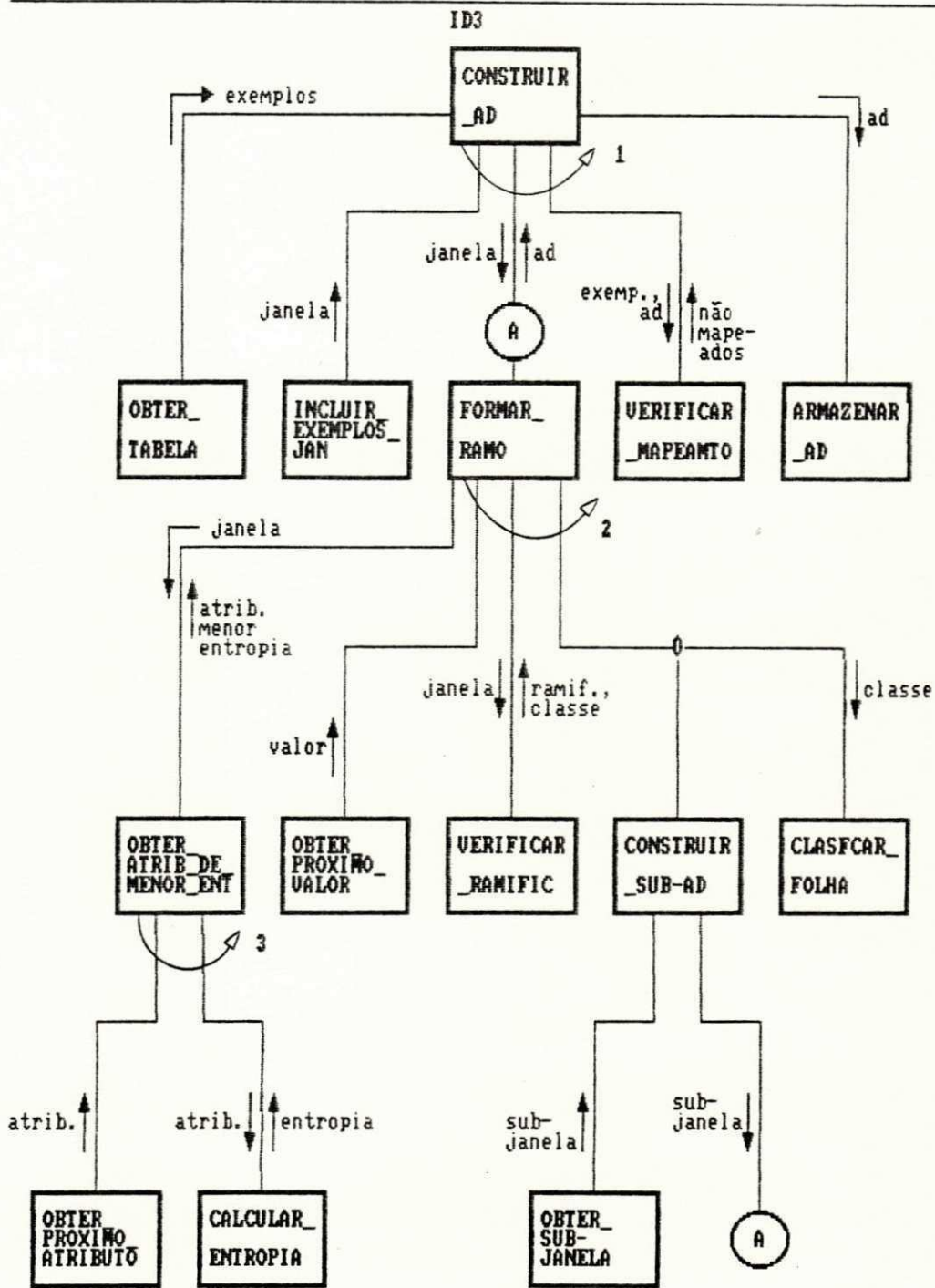
Durante a fase de edição, são permitidas as seguintes operações: inclusão de um novo exemplo, exclusão de um exemplo ou um grupo de exemplos, listar a tabela de exemplos no vídeo ou impressora, e gravá-la num arquivo externo (veja a figura 5.1).

5.1.2 ID3: Construção da AD

O GE que mostra o processo recursivo de construção da AD é apresentado na figura 5.2.

Para que a construção da AD seja iniciada, exige-se que na tabela de exemplos exista pelo menos um elemento de cada classe.

A inclusão de exemplos no conjunto janela é efetuada pelo módulo INCLUIR_EXEMPLOS_JAN.



1. Enquanto existir algum exemplo não mapeado Faça
2. Para todos os valores Faça
3. Para todos os atributos Faça

Figura 5.2 - Gráfico de estrutura da construção da árvore de decisão

O APREND modificou o cálculo do número de elementos do conjunto janela proposto pelo ID3 (10% do conjunto de trabalho). Esta modificação visa adaptar o algoritmo para conjuntos de trabalho que possuem poucos elementos (< 150). O cálculo modificado é feito da seguinte forma: seja J o número de elementos do conjunto janela e T o número de elementos da tabela de exemplos, então:

$$J = \max((T/10), \min(T, 15)) \quad (5.1)$$

desta forma, para $T < 15$, tomar $J = T$, para $15 \leq T \leq 150$, tomar $J = 15$ e para $T > 150$, tomar $J = T/10$.

O objetivo do conjunto janela é construir, com poucas iterações, a AD que mapeia todos os exemplos do conjunto de trabalho. Portanto, o conjunto janela não será escolhido de forma aleatória, mas considerando a mesma proporção de classes do conjunto de trabalho. I.e., se no conjunto de trabalho a proporção de elementos com a classe C é P , então a proporção de elementos com a classe C , no conjunto janela, também será P . Esta é uma heurística que não está descrita no ID3 original, mas foi implementada pelo APREND.

No módulo VERIFICAR_RAMIFICAÇÃO, toma-se a decisão de ramificar a sub-AD ou rotular a folha correspondente. Se todos os exemplos da sub-janela corrente possuírem a mesma classe, então rotula-se a folha. Se as classes dos exemplos da sub-janela forem diferentes, então ramifica-se uma nova sub-árvore, com o objetivo de classificar todos os exemplos da sub-janela.

O módulo FORMAR_RAMO é chamado recursivamente pelo CONSTRUIR_SUB_AD, até que todas as folhas estejam classificadas.

Em VERIFICAR_MAPEAMENTO percorre-se o conjunto de trabalho, para marcar os exemplos que não foram classificados pela AD gerada. Parte destes exemplos serão incluídos no conjunto janela. O número de exemplos a serem incluídos no conjunto janela é calculado pela expressão 5.1, onde T é o número de exemplos não classificados. A proporção de classes é obtida com relação aos exemplos não classificados.

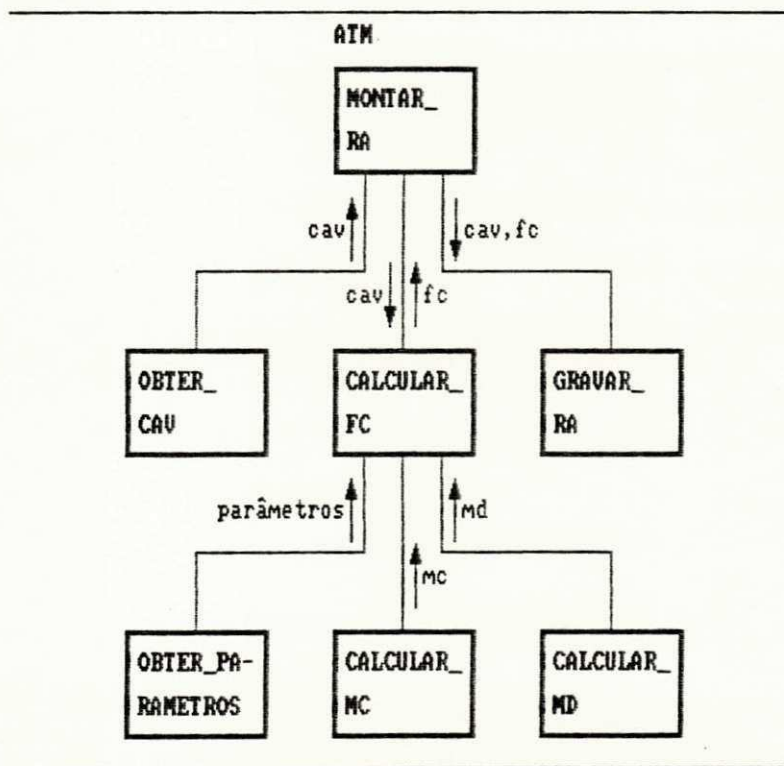


Figura 5.3 - Gráfico de Estrutura da montagem das RI's

5.1.3 ATM: Geração de RA's

A obtenção de RA's é iniciada pelo módulo OBTER_CAV (CAV = Classe-Atributo-Valor), conforme a figura 5.3. Este módulo informa qual a classe e o par atributo-valor que formarão a RA.

O módulo OBTER_PARAMETROS encontra todos os parâmetros necessários ao cálculo do MC e MD.

Os módulos CALCULAR_MC e CALCULAR_MD efetuam, respectivamente, os cálculos da crença e da descrença em uma hipótese, baseados em estatística indutiva. O detalhe de como os cálculos são feitos podem ser vistos nas subseções 4.5.1 e 4.5.2.

5.2 Estruturas de Dados

As estruturas de dados do descritor de classes, atributos e valores, está representada na figura 5.4. A tabela de exemplos e a AD, estão representadas nas figuras 5.5 e 5.6, respectivamente.

A tabela de exemplos da figura 3.1 e a AD da figura 3.4 servem de modelo ilustrativo para as estruturas de dados apresentadas.

5.2.1 O descritor de classes, atributos e valores

O descritor de classes, atributos e valores associa o nome de cada classe, atributo ou valor, a um índice na matriz que representa a estrutura de dados. Este procedimento faz com que a tabela de exemplos seja composta apenas de números inteiros,

economizando-se espaço.

O descritor de atributos aponta para o descritor de valores e classes. O índice 0 do descritor de atributos é usado para indicar a quantidade de atributos e o índice da última classe do descritor de classes, conforme a figura 5.4.

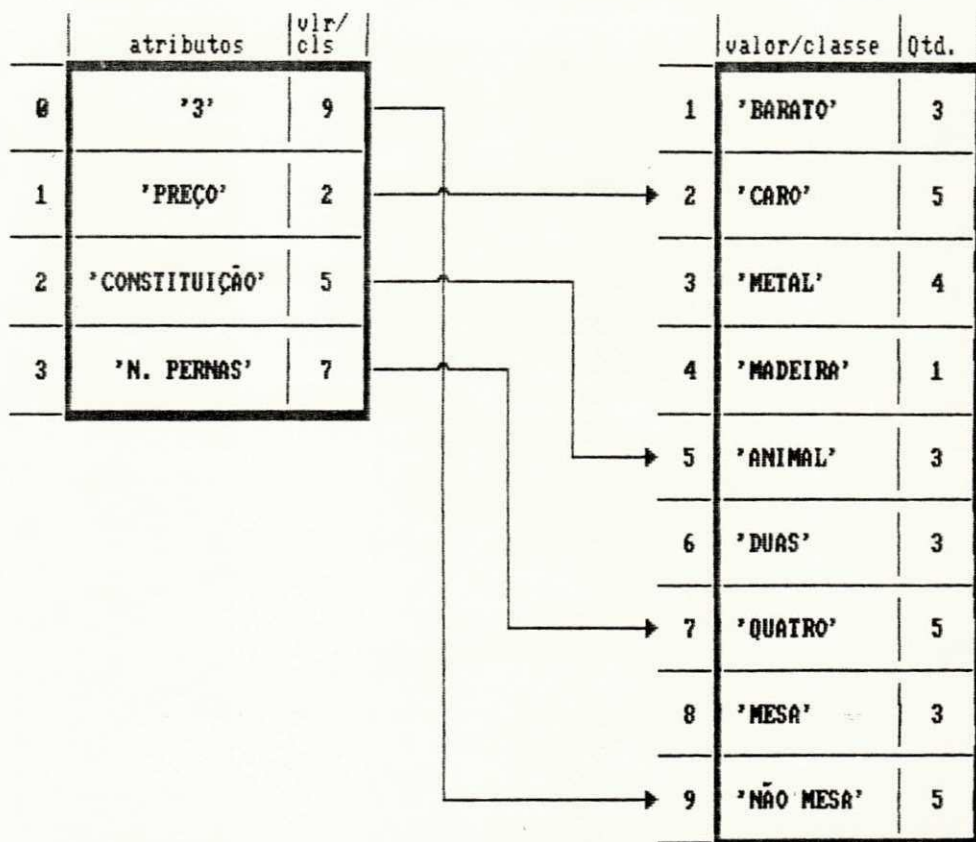


Figura 5.4 - Estrutura de dados que representa o descritor de classes, atributos e valores da tabela de exemplos da figura 3.1

O campo "Qt.", indica o número de exemplos que possuem o valor ou classe correspondente, na tabela de exemplos. Esta informação é usada no cálculo do FC.

5.2.2 A tabela de exemplos

A estrutura da tabela de exemplos é referenciada por linhas e colunas. Nas linhas são apresentados os exemplos e nas colunas os atributos.

	classe	atributo/valor			mesma classe	contador
	0	1	2	3	4	5
1	8	1	3	7	5	1
2	9	1	3	6	3	1
3	9	1	5	7	4	1
4	9	2	3	6	6	1
5	8	2	4	7	7	1
6	9	2	5	7	8	1
7	8	2	3	7	8	1
8	9	2	5	6	8	1

Figura 5.5 - Estrutura de dados que representa a tabela de exemplos da figura 3.1

A última e a penúltima colunas, representam respectivamente, a lista de exemplos com a mesma classe e o contador de exemplos. A lista de exemplos com a mesma classe é usada no módulo VERIFICAR_MAPEAMENTO (Figura 5.2), para evitar que se percorra toda a tabela sempre que se queira verificar o

mapeamento. Desta forma, apenas a lista de exemplos de mesma classe é percorrida.

O contador de exemplos representa o número de repetições de cada exemplo na tabela. Economiza espaço e facilita a obtenção de parâmetros para o cálculo do FC.

5.2.3 A árvore de decisão

A estrutura de dados da AD representa cada nó da árvore usando cinco características (campos): o valor incidente, o atributo ou classe rotulada no nó, o primeiro filho, o próximo irmão e o pai. A estrutura é preenchida à medida em que a AD vai sendo construída.

	valor incidente	atributo /classe	primeiro filho	próximo irmão	pai
1	0	3	2	0	0
2	6	9	0	3	1
3	7	2	4	0	1
4	3	8	0	5	3
5	5	9	0	6	3
6	4	8	0	0	3

Figura 5.6 - Estrutura de dados que representa a árvore de decisão da figura 3.4

O "valor incidente" é o índice na matriz do descritor, que representa o valor do atributo que incide no nó. Por exemplo: verificando-se a figura 5.4, nota-se que 1 representa o valor "Barato", 2 representa o valor "Caro", e assim sucessivamente. A raiz da AD não possui um valor incidente, portanto o valor incidente da raiz é "0".

O "atributo/classe" é o número do atributo que representa o nó, ou o número da classe que rotula uma folha.

O "primeiro filho" é o índice, na estrutura da AD, onde está representado o primeiro filho do nó. Uma folha da AD não possui filhos, portanto o primeiro filho de uma folha é "0".

O "próximo irmão" é o índice, na estrutura da AD, onde está representado o próximo irmão. Um nó é considerado irmão de outro nó, quando está no mesmo nível e o valor incidente pertence ao mesmo atributo. A raiz da AD não possui irmãos, portanto o próximo irmão da raiz é "0".

O "pai" representa o índice, na estrutura da AD, onde está representado o pai de um nó ou de uma folha. A raiz não possui pai, portanto o pai da raiz é 0.

A AD terá que ser percorrida diversas vezes durante o seu processo de formação e na transformação da AD em regras. A estrutura de dados da AD permite que se percorra os caminhos da árvore com rapidez e economia de espaço.

5.3 Guia de Utilização

O APREND foi implementado em micro-computador compatível com o IBM PC-At, utilizando-se a linguagem PASCAL.

Esta linguagem foi escolhida devido ao ambiente de programação oferecido, apresentando facilidades só encontradas em ambientes profissionais de programação, dos quais não dispúnhamos na época.

O código fonte do sistema como um todo, foi escrito com um total de 7.400 linhas. O código executável ocupa 88 kbytes. A configuração mínima aconselhável é de 512 kbytes de memória, isto deve-se à tabela de exemplos, que estará residente na memória, durante a execução do programa.

Ao se executar o sistema, a identificação de uma tabela de exemplos será requisitada. Neste momento deve-se entrar com um

<ESC> Encerra a Operacao									
Qual o nome da tabela ? NEFRO									
<table border="1"><thead><tr><th colspan="3">TABELAS</th></tr></thead><tbody><tr><td>DERMAT</td><td>528</td><td>15/09/88</td></tr><tr><td>PNEUMO</td><td>1.100</td><td>01/02/89</td></tr></tbody></table>	TABELAS			DERMAT	528	15/09/88	PNEUMO	1.100	01/02/89
TABELAS									
DERMAT	528	15/09/88							
PNEUMO	1.100	01/02/89							

Figura 5.7 - Identificação da tabela

nome alfanumérico com até 8 caracteres. Se o nome digitado não existir no diretório de tabelas, passa-se para a fase de descrição dos diagnósticos possíveis na nova tabela. Todos os nomes de diagnósticos e atributos serão digitados com até 10 caracteres (veja a figura 5.7).

A seguir define-se o nome de cada atributo acompanhado dos nomes dos seus respectivos valores. Sempre que necessita-se retornar à fase anterior, ou encerrar uma operação dentro do sistema, pressiona-se a tecla <ESC>.

Após a descrição de todos os diagnósticos e pares

<ESC> Encerra a Operacao										
Qual o diagnostico?										
<p>0 CLARA</p> <p>1 NEGRA</p>										
NUM.	DIAGNOSTICO	IDOSO	COR	AUC	ICC	ANG	RIM	ALCOOL	SOPROABD	COLES
1	H.LEVE	F								

Figura 5.8 - Inclusão de um novo exemplo, durante a escolha do valor do atributo COR.

atributo-valor, entra-se na fase de inclusão de novos exemplos. Nesta fase, o sistema apresentará menus com opções de diagnósticos e valores, conforme a figura 5.8. Ao encerrar a entrada de um exemplo, o sistema requisitará a quantidade de exemplos iguais àquele, 1(um) é o "default".

Ao definir-se todo o conjunto de exemplos, será solicitada a gravação da tabela de exemplos.

Feita a gravação, escolhe-se no menu, uma das opções: editar, executar, listar tabela ou listar regras. A figura 5.9 mostra a listagem de uma regra gerada pelo ID3.

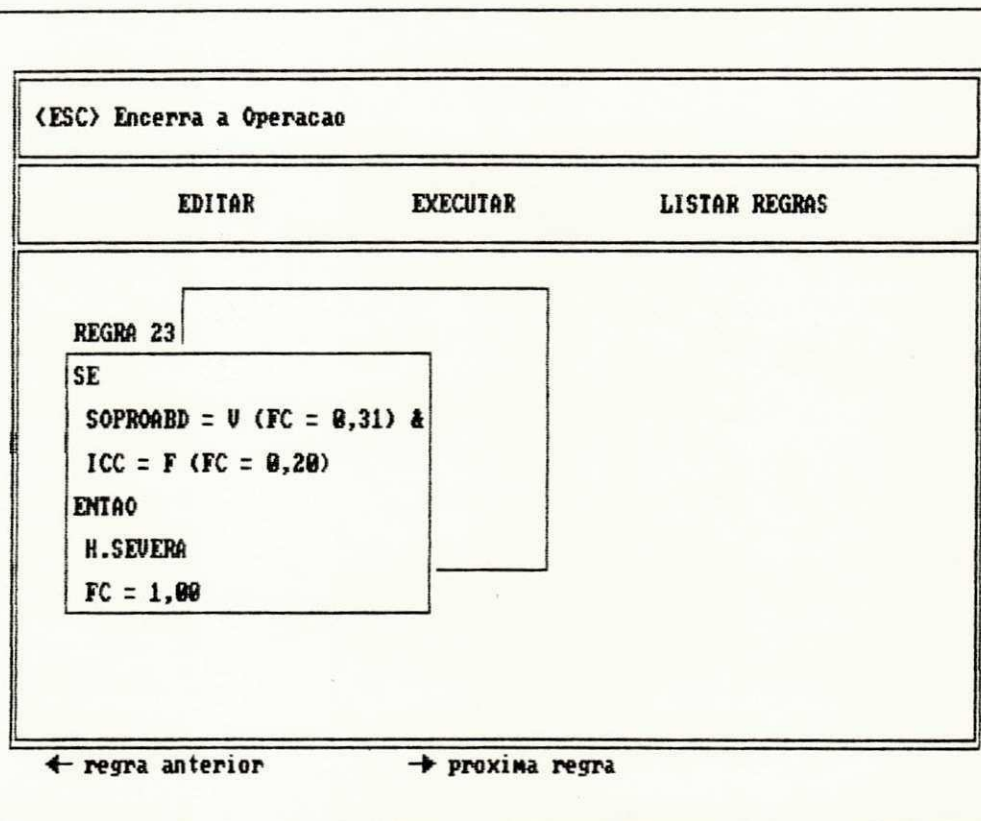


Figura 5.9 - Apresentação de uma regra gerada pelo ID3

A opção "listar regras" está condicionada à opção "executar", onde serão executados os módulos de geração de RC's (ID3) e de RA's (ATM). I.e., a listagem das regras só será feita se anteriormente foi executado o programa, caso contrário, não haverá regra a ser listada.

6. UMA APLICAÇÃO EM NEFROLOGIA

Será apresentada uma aplicação ilustrativa de como, utilizando-se o APREND, pode-se adquirir regras a partir de dados relativos a hipertensão arterial, no domínio de nefrologia.

Todos os dados utilizados foram gentilmente cedidos pela Escola Paulista de Medicina e representam casos clínicos passados no ambulatório de nefrologia daquela escola. Os resultados aqui apresentados são uma forma de ilustrar o uso do sistema. Para se chegar a um resultado final e definitivo, é necessário uma análise cuidadosa, por parte de um perito no domínio. Esta análise faz parte do processo de aquisição automática de conhecimento, e deve ser realizada tanto na interpretação dos dados de entrada, como na crítica das regras obtidas.

6.1 Apresentação dos dados

Os exemplos utilizados se compõem de 14 atributos. Cada atributo possui 2 valores possíveis. Os diagnósticos foram classificados em 3 tipos: hipertensão essencial leve, essencial moderada e essencial severa.

Os atributos associados coletivamente mais o diagnóstico atribuído, formam a caracterização de um exemplo, o que se pode notar na figura 6.1.

O conjunto total de casos disponíveis é composto de 336 exemplos. Usa-se 2/3 do conjunto total para gerar regras e 1/3

para testar as regras geradas, será o conjunto de teste. As

Idoso	<V,F>
Cor	<Clara, Negra>
AUC (Acidente Vascular Cerebral).....	<V,F>
ICC (Insuf. Cardio Coronariana).....	<V,F>
ANG (Angina).....	<V,F>
Rim (Problemas Renais).....	<V,F>
Álcool	<V,F>
SoproAbd (Sopro Abdominal).....	<V,F>
Colester (Colesterol Alterado).....	<V,F>
AntHiper (Antecedentes de Hipertensao)	<V,F>
AntDiab (Antecedentes de Diabetes)....	<V,F>
AntAUC (Antecedentes de AUC).....	<V,F>
AntAngor (Antecedentes de Angor).....	<V,F>
AntRenal (Antecedentes de Probs.Rim)..	<V,F>

U - Verdadeira
F - Falsa

Figura 6.1 - Atributos e valores

regras geradas serão avaliadas de acordo com a proporção de acertos, com relação ao conjunto de teste.

O conjunto usado para gerar regras é composto de 224 elementos, sendo 106 com o diagnóstico de hipertensão leve, 78 de hipertensão moderada e 40 de hipertensão severa. Desta forma, o conjunto de teste é composto de 112 elementos, sendo 53 com hipertensão leve, 39 com moderada e 20 com severa.

O Apêndice A apresenta uma parte do conjunto de exemplos usado para gerar as regras.

6.2 Análise dos resultados

Considerando-se o não-determinismo dos exemplos apresentados e levando-se em consideração o critério de escolha das boas RC's, somente 11 entre 45 regras geradas pelo ID3, podem ser consideradas boas regras. O ATM gerou 84 RA's. Como não dispúnhamos de um perito para auxiliar na eliminação das RA's, foram eliminadas apenas aquelas regras que tinham $FC = 0,00$, ou seja: eliminou-se 2 RA's.

Em termos globais, as regras geradas acertaram 78% dos casos apresentados no conjunto de teste. Com relação a cada diagnóstico, a proporção de acerto foi de 93% para hipertensão leve, 64% para moderada e 54% para severa. Ocorreu um maior índice de acerto para hipertensão leve porque este é o diagnóstico que mais aparece na tabela de exemplos. Desta forma, nota-se que quanto maior é a proporção de exemplos associado a

DIAGNÓSTICO	FREQUÊNCIA NO CONJ. DE TRAB.	FREQUÊNCIA NO CONJ. DE TESTE	PROPORÇÃO DE ACERTO
H. LEVE	106	53	93 %
H. MODERADA	78	39	64 %
H. SEVERA	40	20	54 %

Figura 6.2 - Frequência e Proporção de acertos dos Diagnósticos.
Regras geradas pelo APREND (ID3 + ATM).

certo diagnóstico, maior será a proporção de acertos deste diagnóstico (veja a figura 6.2). Isto deve-se ao cálculo dos FC's. Quanto maior é o número de elementos de um certo diagnóstico, maior é a precisão dos FC's relacionados com este diagnóstico.

Portanto aumentando-se a quantidade de exemplos, teremos regras com um maior nível de confiabilidade.

Para se verificar a importância das RA's que possuem o FC negativo, todas as RA's com esta característica foram retiradas da BC. Desta forma, a proporção geral de acertos caiu de 78% para 57%.

A acurácia das regras obtidas pode ser aumentada se houver o acompanhamento de um perito no domínio durante todo o processo.

Deve-se notar que o grau de determinismo dos exemplos também é fator importante para aumentar a proporção de acerto das regras geradas pelo ID3, pois quanto mais determinísticos são os exemplos, melhor é a performance do algoritmo.

O Apêndice B apresenta parte das RC's geradas pelo sistema. O Apêndice C apresenta as RA's geradas pelo ATM, excluindo-se as RA's que foram excluídas.

7. CONCLUSÕES E TRABALHOS FUTUROS

O APREND gera regras de produção, através da chamada, pelo usuário, de dois módulos independentes: o ID3 (baseado na Teoria da Informação), fornece regras completas com FC = 1,0, e o ATM (baseado em Estatística Indutiva), fornece regras atômicas com FC apresentando valores entre -1 e +1.

A implementação do algoritmo ID3 produz regras completas. Algumas dessas regras são consideradas boas e úteis pelos peritos, porém, outras são consideradas más, sem muita utilidade, devendo ser removidas da base de conhecimento. Para se distinguir as boas das más regras, usam-se os valores dos FC's das regras atômicas, correspondentes às condições das regras completas. Estes valores foram chamados de Influência de uma condição. Foram apresentados alguns critérios que, baseados na Influência, permitem distinguir as boas regras completas.

Além de revelar as boas e más regras completas, cada condição associada a um FC, forma uma regra atômica. As regras atômicas têm grande importância na performance de um SE, pois elas indicam, individualmente, o quanto cada condição influencia uma conclusão. O FC pode ser positivo, negativo ou zero, isto permite uma melhor representação das relações entre a evidência e a hipótese.

As regras obtidas através de sistemas de aquisição automática de conhecimento, similares ao APREND, podem acertar até 100% dos diagnósticos (Michalski & Chilauski, 1981), quando aplicados a domínios determinísticos. Em geral, o APREND como

ferramenta de auxílio ao engenheiro do conhecimento, não soluciona completamente o problema do "gargalo" da construção de SE's, mas contribui para a redução do tempo de aquisição de conhecimento.

O processo de escolha ou rejeição de uma regra deve ser orientado pelo perito no domínio do conhecimento. Ele apresentará, baseado nos FC's, os limites que indicarão se a regra será descartada ou usada. Assim, é proposto como trabalho futuro, o desenvolvimento de um módulo que, baseado nos critérios de escolha, apresente ao usuário apenas as boas regras.

Quando os exemplos pertencem a domínios acentuadamente não determinísticos, o ID3 não fornecerá boas regras completas. Contudo, o método de cálculo do FC, usado por ATM, poderia ser usado para encontrar regras compostas, com FC entre -1 e +1, substituindo o ID3, porém, com uma complexidade explosiva de ordem $(2^n \cdot C)$, onde n é o número de pares atributo-valor e C é o número de classes. Desta forma, o desenvolvimento de um módulo capaz de lidar com esta complexidade, ou mesmo reduzi-la, através de interações com o perito, seria um trabalho posterior que enriqueceria substancialmente o sistema APREND (veja na próxima página o número 9).

Serão propostos alguns trabalhos a serem desenvolvidos a partir do estado atual do APREND. São eles:

1. Permitir a inclusão de atributos com valores contínuos.
2. Desenvolver um módulo de tratamento do ruído, baseado no algoritmo C4 (Quinlan, 1986b);

3. Desenvolver um módulo de interface com sistemas gerenciadores de bancos de dados, permitindo a formação de exemplos a partir de SGBD's, ou aproveitar dados já editados.
4. Aplicar o APREND em SE's reais de médio porte, com a assistência de um perito no domínio.
5. É importante associar o APREND a uma ferramenta de construção de SE's que represente o conhecimento na forma de regras e propague a incerteza na forma do MYCIN. Esta ferramenta pode ser o BACKFORNEB (Aquino, 1987).
6. Adaptar o ID3 a domínios com diagnósticos e valores não mutuamente exclusivos.
7. Fazer um estudo visando a simplificação das RC's, através do cálculo do FC, considerando a influência do conjunto de condições de uma regra como um todo e não cada condição individualmente.
8. Desenvolver um módulo que, baseado nos critérios de escolha, apresente ao usuário apenas as regras que deverão ser implementadas na base de conhecimento.
9. Desenvolvimento de um módulo que associe os FC's de duas ou mais condições, formando regras compostas, a partir de sugestões dos peritos. Uma idéia inicial seria apresentar todas as RA's ao perito, e então pedir para que ele escolha intuitivamente as combinações que ele considerar importantes. A seguir o sistema forneceria o FC da combinação.

Referências Bibliográficas

- Aquino, M.S.; BACKFORNEB: Uma Ferramenta para Construção de Sistemas Especialistas. Diagnosticadores; Dissertação de Mestrado, apresentada ao DSC/UFPB, Campina Grande, 1987.
- Boy, G., Faller, B. e Sallantin, J.; Acquisition et Ratification de Connaissances; Publicado pelo Centre de Recherche en Informatique de Montpellier - CRIM, Montpellier, 1987.
- Boose, J.H.; Personal Construct Theory and The Transfer of Human Expertise. Anais do AAAI-84, 1984.
- Buchanan, G.B. e Shortliffe, E.H.; Rule Based Expert Systems: The MYCIN Experiments of The Stanford Heuristic Programming Project, Addison-Wesley, 1984.
- Carter, C. e Catlett, J.; Assessing Credit Card Applications Using Machine Learning; in IEEE Expert, fall, 1987.
- Costa Neto, P.L.O.; Estatística; Edgard Blucher, São Paulo, 1987.
- Davis, R., Shrobe, H., Hamscher, W., Wieckert, K., Shirley, M. e Polit, S.; Diagnosis Based on Description of Structure and Function; Anais do AAAI-82, 1982.
- Dietterich, T.G. e Michalski, R.S.; Inductive Learning of

Structural Descriptions: Evaluation Criteria and Comparative Review of Selected Methods, Artificial Intelligence, Vol. 16, 1981.

Dietterich, T.G. e Michalski, R.S.; Learning to Predict Sequencies, in Machine Learning: An Artificial Intelligence Approach, Vol. II, R.S. Michalski, J.G. Carbonell e T.M. Mitchell (Eds.), Morgan Kaufmann, 1986.

El-Khomi, B.M.; Inductive Algorithms and Tools: A Survey; SYSLAB working paper no. 135, University of Stockholm, 1988.

Feigenbaum, E.A.; Expert Systems in the 1980's, in the State of The Art Report on Machine Intelligence, A. Bond (Ed.), Pergamon-Infotech, 1981.

Gascuel, D.; Critères pour Elaguer la Recherche Lorsque la Complétude et la Cohérence ne Sont pas Requises, publicado pelo CRIM, Montpellier, 1988.

Hunt, E.B., Marin, J. e Stone, P.J.; Experiments in Induction, Academic-Press, 1986.

Kguyen, T.A., Perkins, W.A., Laffey, T.J. e Pecora, D; Knowledge Base Verification, AI Magazine, Vol. 8, no. 2, 1987.

Klir, G.J. e Folger, T.A.; Fuzzy Sets, Uncertainty and Information; Prentice-Hall, 1988.

Langley, P.W., Zytkow, J.M., Simon, H.A., Bradshaw, G.L.; The Search for Regularity: Four Aspects of Scientific Discovery, in Machine Learning: An Artificial Intelligence Approach, R.S. Michalski, J.G. Carbonell e T.M. Mitchell (Eds.), Vol. II, Morgan Kaufmann, 1986.

Michalski, R.S. e Chialauski, R.L.; Knowledge Acquisition by Encoding Expert Rules versus Computer Induction from Examples: A Case Study Involving Soybean Pathology, in Fuzzy Reasoning and its Applications, E.H. Mamdani e B.R. Gaines (Eds.), Academic Press, 1981.

Quinlan, J.R.; Induction of Decision Tree, in Machine Learning Journal I, Kluver Academic Publisher, 1986a.

Quinlan, J.R.; The Effect of Noise on Concept Learning, in Machine Learning: An Artificial Intelligence Approach, Vol. II, R.S. Michalski, J.G. Carbonell e T.M. Mitchell (Eds.), Morgan Kaufmann, 1986b.

Quinlan, J.R.; Inductive Knowledge Acquisition: A Case Study, in Applications of Expert Systems, J.R. Quinlan (Ed.), Addison-Wesley, 1987a.

Quinlan, J.R.; Simplifying Decision Trees, New South Wales Institute of Technology Technical Report no. 87.4, 1987b.

Voyer, R.; Moteurs de Systemes Experts, Editions Eyrolles, Paris, 1987.

Waterman, D.A.; A Guide to Expert Systems, Addison-Wesley, 1986.

Winston, P.H.; The Commercial Debut of Artificial Intelligence, in Applications of Expert Systems, J.R. Quinlan (Ed.), Addison-Wesley, 1987.

APENDICE A

Alguns Exemplos Usados no Capitulo 6

(A listagem dos exemplos está à disposição dos interessados)

CLASSE	Idoso	Cor	AUC	ICC	ANG	Rim	Alc	SoproAbd	Colester	Anthiper	AntDiab	AntAUC	AntAngor	AntRenal
EL	U	C	F	F	F	F	F	F	U	F	F	F	F	F
EM	U	C	F	U	U	F	F	F	F	U	F	F	F	F
ES	U	C	F	F	F	U	F	F	F	U	U	F	F	F
ES	U	C	F	F	F	F	F	F	F	F	F	F	F	F
EL	U	C	F	U	F	U	F	F	F	F	F	F	F	F
EL	U	C	F	F	F	U	F	F	F	U	F	F	F	F
EM	U	C	F	F	F	F	U	F	F	U	U	U	U	F
ES	U	N	F	U	F	F	F	F	F	U	F	U	F	F
ES	U	C	F	F	F	F	F	F	U	F	F	F	F	F
EL	F	C	F	F	F	F	F	F	F	U	F	U	F	F
ES	U	C	F	F	F	F	U	F	F	F	F	F	U	F
EL	F	C	F	F	U	F	F	F	F	U	F	U	F	F
EL	U	N	F	F	F	F	F	F	F	U	F	F	U	F
EL	U	C	F	F	F	F	F	F	F	F	U	F	F	F
EM	F	C	F	F	F	U	U	F	F	F	F	F	F	U
EL	U	N	F	F	U	F	F	F	F	F	F	U	F	F
EL	U	C	F	F	F	F	F	F	F	F	F	F	F	F
EL	F	C	F	F	F	F	F	F	F	U	U	F	F	F
EL	F	N	F	F	F	U	U	F	F	U	F	U	F	U
EM	F	C	F	F	F	F	F	F	F	U	F	U	U	F
EL	U	N	F	U	F	F	F	F	F	U	U	U	F	F
EL	U	C	F	F	F	F	F	F	F	U	F	F	F	F
EM	U	C	F	U	F	F	F	F	U	U	F	F	F	F
EL	U	N	F	F	F	F	F	F	F	U	F	F	U	F

Fonte: Escola Paulista de Medicina

EL = Hipertensao Essencial Leve

EM = Hipertensao Essencial Moderada

ES = Hipertensao Essencial Severa

F = Falso

U = Verdadeiro

C = Clara

N = Negra

APENDICE B

Regras Completas Geradas Pelo APREND

SE	SOPROABD = V (Inf=0,98)	SE	SOPROABD = F & (Inf=-0,13)
ENTAO		SE	ANTAUC = F & (Inf=-0,09)
ES	FC=1,00	SE	ICC = F & (Inf=-0,22)
		SE	ANTRENAL = V & (Inf=-0,28)
		SE	ANTANGOR = F (Inf=-0,11)
SE	SOPROABD = F & (Inf=-0,35)	ENTAO	EM FC=1,00
SE	ANTAUC = V & (Inf=-0,59)		
SE	ANG = F & (Inf=-0,35)	SE	SOPROABD = F & (Inf=-0,13)
SE	ANTRENAL = F & (Inf=-0,29)	SE	ANTAUC = V & (Inf=-0,23)
SE	COR = N & (Inf=0,03)	SE	ANG = F & (Inf=-0,03)
SE	ALCOOL = F & (Inf=-0,30)	SE	ANTRENAL = F & (Inf=-0,13)
SE	ANTDIAB = F & (Inf=-0,36)	SE	COR = C & (Inf=-0,14)
SE	IDOSO = V (Inf=-0,37)	SE	ANTDIAB = F & (Inf=-0,19)
ENTAO	ES FC=1,00	SE	AUC = F & (Inf=-0,10)
		SE	ALCOOL = F & (Inf=-0,08)
		SE	RIM = V (Inf=0,00)
SE	SOPROABD = F & (Inf=0,00)	ENTAO	EM FC=1,00
SE	ANTAUC = F & (Inf=-0,31)		
SE	ICC = F & (Inf=0,02)	SE	SOPROABD = F & (Inf=0,00)
SE	ANTRENAL = F & (Inf=-0,08)	SE	ANTAUC = F & (Inf=-0,31)
SE	ALCOOL = F & (Inf=-0,12)	SE	ICC = V & (Inf=-0,24)
SE	COR = N & (Inf=-0,36)	SE	ANTHIPER = V & (Inf=0,02)
SE	ANG = F & (Inf=-0,12)	SE	ANTANGOR = V & (Inf=0,09)
SE	COLESTER = F & (Inf=0,04)	ENTAO	EL FC=1,00
SE	ANTHIPER = V & (Inf=0,02)		
SE	IDOSO = V & (Inf=-0,21)		
SE	ANTANGOR = V (Inf=0,09)		
ENTAO	EL FC=1,00		

APENDICE C

Regras Atômicas Geradas pelo APREND

IDOSO=F-----(-0,33)---> EL	ANG=F-----(-0,12)---> EL
IDOSO=F-----(-0,51)---> EM	ANG=F-----(-0,03)---> EM
IDOSO=F-----(-0,24)---> ES	ANG=F-----(-0,35)---> ES

IDOSO=U-----(-0,21)---> EL	ANG=U-----(-0,30)---> EL
IDOSO=U-----(-0,07)---> EM	ANG=U-----(-0,48)---> EM
IDOSO=U-----(-0,37)---> ES	ANG=U-----(-0,24)---> ES

COR=CLARA----(-0,15)---> EL	RIM=F-----(-0,02)---> EL
COR=CLARA----(-0,14)---> EM	RIM=F-----(-0,18)---> EM
COR=CLARA----(-0,48)---> ES	RIM=F-----(-0,29)---> ES

COR=NEGRA----(-0,36)---> EL	RIM=U-----(-0,05)---> EL
COR=NEGRA----(-0,16)---> EM	RIM=U-----(-0,43)---> ES
COR=NEGRA----(-0,03)---> ES	

AUC=F-----(-0,15)---> EL	ALCOOL=F----(-0,12)---> EL
AUC=F-----(-0,10)---> EM	ALCOOL=F----(-0,08)---> EM
AUC=F-----(-0,26)---> ES	ALCOOL=F----(-0,30)---> ES

AUC=U-----(-0,55)---> EL	ALCOOL=U----(-0,30)---> EL
AUC=U-----(-0,37)---> EM	ALCOOL=U----(-0,37)---> EM
AUC=U-----(-0,55)---> ES	ALCOOL=U----(-0,38)---> ES

ICC=F-----(-0,02)---> EL	SOPROABD=F--(-0,13)---> EM
ICC=F-----(-0,22)---> EM	SOPROABD=F--(-0,35)---> ES
ICC=F-----(-0,29)---> ES	

ICC=U-----(-0,24)---> EL	SOPROABD=U--(-1,00)---> EL
ICC=U-----(-0,21)---> EM	SOPROABD=U--(-1,00)---> EM
ICC=U-----(-0,43)---> ES	SOPROABD=U--(1,00)---> ES

APENDICE C

-continuação-

COLESTER=F---(0,04)---> EL	ANTAUC=F----(-0,31)---> EL
COLESTER=F--(-0,21)---> EM	ANTAUC=F----(-0,09)---> EM
COLESTER=F--(-0,33)---> ES	ANTAUC=F----(-0,10)---> ES

COLESTER=U--(-0,31)---> EL	ANTAUC=U-----(0,40)---> EL
COLESTER=U---(0,13)---> EM	ANTAUC=U----(-0,23)---> EM
COLESTER=U--(-0,29)---> ES	ANTAUC=U----(-0,59)---> ES

ANTHIPER=F--(-0,11)---> EL	ANTANGOR=F--(-0,07)---> EL
ANTHIPER=F--(-0,23)---> EM	ANTANGOR=F--(-0,11)---> EM
ANTHIPER=F--(-0,17)---> ES	ANTANGOR=F--(-0,32)---> ES

ANTHIPER=U---(0,02)---> EL	ANTANGOR=U---(0,09)---> EL
ANTHIPER=U--(-0,10)---> EM	ANTANGOR=U--(-0,25)---> EM
ANTHIPER=U--(-0,41)---> ES	ANTANGOR=U--(-0,33)---> ES

ANTDIAB=F----(0,06)---> EL	ANTRENAL=F--(-0,08)---> EL
ANTDIAB=F--(-0,19)---> EM	ANTRENAL=F--(-0,13)---> EM
ANTDIAB=F--(-0,36)---> ES	ANTRENAL=F--(-0,29)---> ES

ANTDIAB=U---(-0,17)---> EL	ANTRENAL=U---(0,33)---> EL
ANTDIAB=U---(-0,08)---> EM	ANTRENAL=U--(-0,28)---> EM
ANTDIAB=U---(-0,26)---> ES	ANTRENAL=U--(-0,48)---> ES

RESUMO

O "gargalo" da construção de um Sistema Especialista reside no processo de aquisição do conhecimento necessário ao seu funcionamento.

Este trabalho apresenta o desenvolvimento do APREND, uma ferramenta de aquisição automática de conhecimento para Sistemas Especialistas, utilizando exemplos como dados de entrada e gerando regras de produção.

São fornecidos dois tipos de regras: as completas e as atômicas. As primeiras têm uma ou mais condições, fator de certeza 1,0 e são geradas pelo algoritmo ID3 de Quinlan. As últimas, têm uma só condição, fator de certeza entre -1 e +1 e são geradas por algoritmo criado para o APREND, com base em estatística indutiva.

São apresentados detalhes do projeto, implementação e uso do sistema, inclusive critérios para identificar eventuais regras que não correspondam a conhecimento considerado útil e relevante pelos peritos.

Uma aplicação na área de nefrologia ilustra o processo de aquisição de conhecimento, utilizando o APREND.