O Uso de Semântica na Construção e Expansão de Árvores de Decisão Indutivas

Giuseppe Mongiovi
Walfredo da Costa Cirne Filho

Universidade Federal da Paraíba
Centro de Ciências e Tecnologia
Departamento de Ciência da Computação
Av. Aprígio Veloso, s/n
Campina Grande - PB - CEP 58.100 - BRASIL
E-mail: wfredo@brufpb2.bitnet

Resumo: Neste trabalho é apresentado o problema das regras inúteis, geradas por algoritmos da família TDIDT, e uma solução baseada em semântica, que expande a árvore de decisão gerada por estes algoritmos, é analisada. É também apresentada uma nova solução para o problema, que consiste na construção direta da árvore de decisão. Esta construção é guiada por uma função de avaliação pragmática, em que os aspectos estrutural e semântico são observados. Uma análise comparativa entre as soluções é feita.

I. Introdução

Uma das principais áreas de pesquisa em Inteligência Artificial, do ponto de vista científico e comercial, constitui-se dos Sistemas Baseados em Conhecimento (em particular, dos Sistemas Especialistas). A grande dificuldade na construção desses sistemas reside no processo de aquisição do conhecimento necessário ao seu funcionamento. Para minimizar este problema foram desenvolvidos métodos automatizados para aquisição de conhecimento [Boy 87] [Boose 87]. Entre os processo automatizados, o da geração da Base de Conhecimento (BC) a partir de exemplos tem sido um dos mais explorados. Entre estes, destacam-se os da família TDIDT (*Top Down Induction of Decision Trees*) [Quinlan 86].

Esses métodos geram automaticamente uma Árvore de Decisão (AD), a partir de um conjunto de exemplos, imitando o processo humano de aprendizado por indução [Boman 89]. Isto é feito classificando os exemplos. Tal classificação é feita a partir dos pares atributo/valor que caracterizam cada exemplo. A geração é feita tentando minimizar a AD, usando uma função de avaliação (geralmente, cálculo de entropia) na escolha do melhor atributo a expandir [Mingers 89]. A AD gerada pode ser usada diretamente como BC ou transformada num conjunto equivalente de regras de produção. Entre os métodos que realizam este processo de aquisição destacam-se o ID3 [Quinlan 86], o C4 [Quinlan 87] e o CART [Breiman 84]. Estes métodos diferem entre si pela estrategia adotada para minimizar a AD.

Apesar do aparente sucesso alcançado por esses métodos, algumas limitações de cunho prático têm sido identificadas. A inabilidade de lidar com ruído nos dados de entrada, a exigência de valores discretos e disjuntos e a forma como os resultados da indução são apresentados têm sido alvo de pesquisas recentes na área. Vários trabalhos, solucionando parcialmente esses problemas, têm sido apresentados [Cendrowska 88] [Quinlan 83]. Basicamente estes trabalhos têm abordado apenas o aspecto estrutural da **AD**. Entretanto, um fato muito importante, do ponto de vista prático, não vem sendo lembrado. É o aspecto semântico da **AD** gerada. Isto é, será que todo caminho na **AD** corresponde a uma regra de utilidade prática? A resposta é não.

Este trabalho apresenta o problema das regras inúteis (regras que, embora estruturalmente corretas, não têm utilidade prática) e discute a solução apresentada pelos autores para o algoritmo **ID3** [Mongiovi 90] e posteriormente generalizada para toda a família TDIDT (algoritmo **ADEX**) [Cirne 90]. Esta solução é baseada em conhecimento e consiste na expansão semântica de uma **AD** gerada por um algoritmo TDIDT. Como, apesar da pouca validação, a expansão apresenta alguns problemas práticos, será apresentada uma nova solução para o problema das regras inúteis. Esta nova solução usa uma estratégia de indução baseada não só no aspecto estrutural mas também no aspecto semântico. Estratégia esta que representa uma grande mudança de enfoque em relação aos métodos tradicionais de indução em **AD**.

Na seção II levanta-se o problema das regras inúteis. A seção III apresenta o conceito de Matriz de Relevância (**MR**) que formaliza o conhecimento necessário para os métodos aqui descritos. A seção IV apresenta e discute o algoritmo **ADEX**. Na seção V encontram-se a descrição da nova solução proposta, denominada de algoritmo IDRT, e uma análise comparativa entre esta solução e os algoritmos **ID3** e **ADEX**. Finalmente, as conclusões estão na seção VI.

II. O problema das regras inúteis

A árvore de decisão gerada pelos algoritmos da família TDIDT é uma estrutura despida de semântica. Ela apenas mapeia, de forma reduzida, o conjunto de exemplos. Devido a este fato, estes algoritmos podem gerar, em certas circunstâncias, regras sem significado que na prática são inúteis. Uma circunstância em que isto frequentemente acontece é quando o conjunto de exemplos contém "casos raros".

Numa regra inútil, nota-se que a premissa é irrelevante para implicar a conclusão. Por exemplo, na construção de um Sistema Especialista em ginecologia, num universo de

pacientes adultas, com apenas alguns casos de crianças, o sistema de aquisição automática de conhecimento APREND [Gomes 88] [Gomes 89] forneceu a seguinte regra:

SE IDADE = criança
ENTÃO DIAGNÓSTICO = vaginite

Isto aconteceu porque no conjunto de exemplos, por coincidência, todas as crianças estavam com vaginite (já que os dados foram obtidos de fichario médico) e, portanto, o atributo **IDADE** era o de menor entropia. Além disso, o sistema APREND encerrou neste ponto a construção do ramo correspondente ao caso, já que todos os exemplos ficam classificados apenas com **IDADE = criança**. O exemplo (caso raro) ficou, portanto, mapeado numa regra inútil, uma vez que a premissa da regra acima ("a paciente é criança") é irrelevante para concluir se a paciente está ou não com vaginite. Note que, embora mais fácil de ocorrer em casos raros (pois as chances de erro numa pequena amostragem são maiores que em uma amostragem mais significativa), esse problema pode aparecer também em outros casos.

Este tipo de problema pode ocorrer até mesmo em sistemas "de brinquedo". Por exemplo, seja a tabela da figura 2.1.

	COME	DIABÉTICO	IDADE	VEGETARIANO	DIAGNÓSTICO
ex1	pouco	sim	velho	sim	MAGRO
ex2	médio	não	velho	sim	MAGRO
ex3	muito	sim	velho	não	GORDO
ex4	pouco	não	velho	não	MAGRO
ex5	médio	sim	jovem	não	GORDO
ex6	pouco	não	jovem	sim	MAGRO
ex7	muito	não	velho	não	GORDO
ex8	médio	não	jovem	não	GORDO

Figura 2.1 - Tabela de exemplos

A árvore gerada pelo sistema APREND, correspondente a tabela acima, está na figura abaixo:

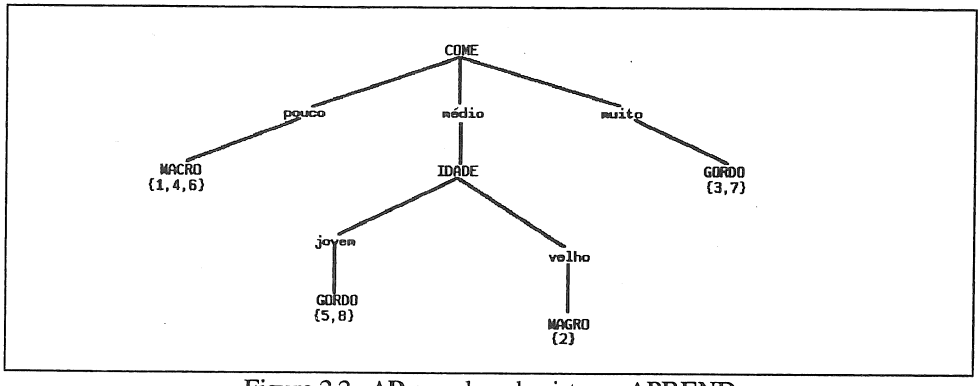

Figura 2.2 - AD gerada pelo sistema APREND

Cada folha da árvore acima é composta por um elemento de classificação e por um conjunto que representa os exemplos por ela mapeados. A informação dos nós está em maiúsculo e a dos ramos em minúsculo. As regras correspondentes à árvore são:

r_1) SE COME = pouco ENTÃO DIAGNÓSTICO = MAGRO
r_2) SE COME = médio E IDADE = jovem ENTÃO DIAGNÓSTICO = GORDO
r_3) SE COME = médio E IDADE = velho ENTÃO DIAGNÓSTICO = MAGRO
r_4) SE COME = muito ENTÃO DIAGNÓSTICO = GORDO

As regras r_2 e r_3, embora não mapeando casos raros, têm, como se pode ver, pouca utilidade prática e são ditas inúteis. Isto porque, na prática, **DIABÉTICO** e **VEGETARIANO** são fatores muito mais indicativos do que **IDADE**, para se concluir se alguém é **MAGRO** ou **GORDO**. Além do mais, o atributo **COME** não é importante para a conclusão (**MAGRO/GORDO**) com o valor **médio**.

III. Matriz de Relevância

A solução encontrada para contornar o problema das regras inúteis consiste na expansão da árvore de decisão fornecida por qualquer um dos métodos da família TDIDT. A expansão é feita a partir de uma Matriz de Relevância (**MR**), construída pelo especialista do domínio em questão.

Uma **MR** para um dado domínio é o relacionamento de relevância entre os elementos de classificação e os atributos deste domínio. Em uma **MR** a linha **i** representa o atributo A_i e a coluna **j** o elemento de classificação D_j. Um elemento R_{ij} de uma **MR** indica o grau de relevância do atributo A_i para a classificação de D_j. Este grau é dado pelo conjunto de valores

do atributo A_i que são relevantes para D_j. O elemento R_{ij} é, na verdade, um subconjunto do conjunto de valores do atributo A_i. Se R_{ij} for vazio, significa que o atributo A_i é irrelevante na classificação de D_j. Ao passo que, R_{ij} igual ao conjunto de valores de A_i significa que A_i é totalmente relevante na classificação de D_j. Uma **MR** é dita completa se todos os atributos forem totalmente relevantes para todos os elementos de classificação. Uma **MR** é vazia se todos os seus elementos forem iguais ao conjunto vazio.

Observa-se que aparentemente uma **MR** é semelhante à uma rede de repertório (*rating grid*) dos sistemas tipo ETS [Boose 84]. Entretanto, existe uma grande diferença entre elas. Enquanto numa rede de repertório o relacionamento é total (todos os elementos de classificação se relacionam com todas as características), numa **MR** apenas os relacionamentos tidos relevantes são considerados. Isto minimiza a participação do especialista no processo de aquisição de conhecimento.

Para o exemplo em questão (figura 2.1), uma **MR** poderia ser a da figura abaixo.

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

Figura 3.1 - Exemplo de uma **MR**

A figura 3.1 mostra que:

i) O atributo **COME** é relevante para classificar **MAGRO** no valor **pouco**, e é relevante, no valor **muito**, para classificar **GORDO**.

ii) O atributo **DIABÉTICO** é relevante para classificar **GORDO** no valor **sim** e **MAGRO** no valor **não**.

iii) O atributo **IDADE** é irrelevante para classificar **MAGRO** e **GORDO** (casos deste tipo alertam ao **EC** que possivelmente o atributo **IDADE** deveria ser eliminado da tabela de exemplos).

iv) O atributo **VEGETARIANO** é relevante para classificar **GORDO** no valor **não** e **MAGRO** no valor **sim**.

IV. Algoritmo ADEX

A árvore de decisão pragmática é uma expansão da árvore gerada por um algoritmos TDIDT. A finalidade dessa expansão é colocar a semântica obtida do especialista (contida na MR) na árvore. Pretende-se, desta forma, evitar que se tenha uma regra onde as premissas são irrelevantes para a conclusão, isto é, evitar regras inúteis.

A expansão é feita de modo a colocar pelo menos uma condição relevante em todo caminho que vai da raiz a uma folha qualquer. Ou seja, toda regra terá um componente relevante em sua premissa. Abaixo, segue o algoritmo **ADEX**, que expande uma **AD** gerada por um algoritmo TDIDT em uma árvore pragmática.

```
adex(AD)
        PARA TODA folha f (composta pelo elemento de classificação D
        e pelo conjunto de exemplos E) da AD
                tentar_expandir(f);
tentar_expandir(f)
        SE nenhuma das condições pertencentes ao caminho(raiz,f) é
        relevante ENTÃO
                SEJA ai um atributo não pertencete ao caminho(raiz,f);
                SEJA Vi = {Vi1,Vi2,...,Vin} o conjunto de valores que
                são relevantes para ai concluir D;
                SEJA A = {a1,a2,...,ak} o conjunto de todos os ai's
                tais que existe um exemplo pertencente a E com o valor
                de ai pertencente a Vi.
                SE A não é vazio ENTÃO
                        SEJA am o atributo de "melhor" função de avaliação
                        pertencente a A;
                        substituir f pelo nó não-terminal am ligado pelos
                        ramos Vm1, Vm2, ..., Vmn e * às folhas f1, f2,
                        ..., fn e f*, respectivamente;
                        associar a cada uma destas folhas o elemento D e
                        redistribuir o conjunto de exemplos E entre elas;
                        tenta_expandir(f*);
                SENÃO marca f como inútil;
```

No algoritmo **ADEX**, o símbolo *, filho do atributo a_m, representa todos os valores não relevantes de a_m. Por exemplo, aplicando o algoritmo à árvore da figura 2.2, tem-se a árvore da figura 3.2.

As regras correspondentes à árvore **ADEX** são:

e_1) **SE** COME = pouco **ENTÃO** DIAGNÓSTICO = MAGRO
e_2) **SE** COME = médio **E** IDADE = jovem **E** DIABÉTCO = sim
ENTÃO DIAGNÓSTICO = GORDO
e_3) **SE** COME = médio **E** IDADE = jovem **E** VEGETARIANO = não
ENTÃO DIAGNÓSTICO = GORDO
e_4) **SE** COME = médio **E** IDADE = velho **E** DIABÉTCO = não
ENTÃO DIAGNÓSTICO = MAGRO
e_5) **SE** COME = muito **ENTÃO** DIAGNÓSTICO = GORDO

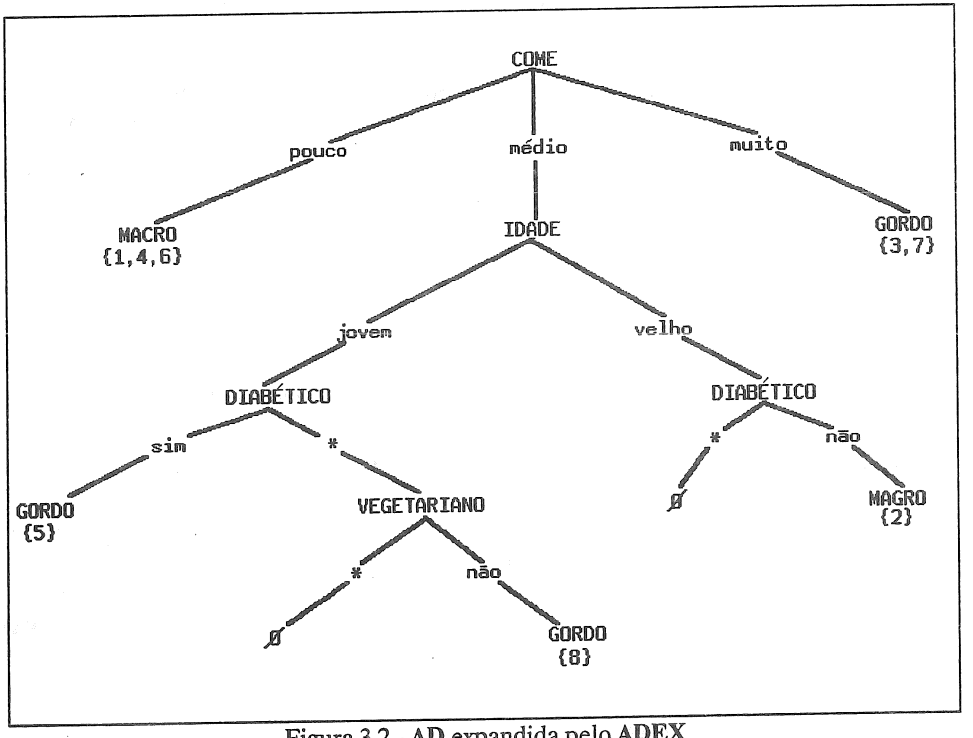

Figura 3.2 - **AD** expandida pelo **ADEX**

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

Da comparação das regras originais com as regras pragmáticas, nota-se que a regra r_2 inicialmente tida como inútil foi substituída, com o uso do algoritmo **ADEX**, pelas regras úteis e_2 e e_3. Além disso, a outra regra inútil r_3 foi transformada na regra útil e_4. A conclusão de que as regras e_2, e_3 e e_4 são úteis deve-se ao fato de que, na prática, os atributos **DIABÉTICO** e **VEGETARIANO**, contantes destas regras, são bastantes significativos para distinguir entre **MAGRO** e **GORDO**.

Embora no exemplo citado todas as regras inúteis foram substituidas por regras úteis, na prática isto nem sempre acontece [Cirne 90] [Mongiovi 90]. Entretanto, isso não é devido à uma limitação do **ADEX** e sim à pouca informação contida na **MR**. Mesmo quando o **ADEX** não retira regras inúteis seu uso é recompensador, pois as regras inúteis ficam marcadas de forma que quando o especialista for revisar o banco de regras obtido, pode concentrar sua atenção sobre elas.

Mesmo para os casos extremos, a **MR** completa ou vazia, o **ADEX** ainda produz resultados coerentes. Em ambos os casos ele não muda a árvore original. Se a **MR** for completa, conclui-se que todas as regras são úteis (já que tudo é relevante). Ao passo que, se a **MR** for vazia, obtem-se que todas as regras são inúteis (como nada é relevante, tudo é inútil).

V. Construção Direta de uma Árvore de Decisão Pragmática

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

Uma forma, sugerida pelos autores, de contornar os problemas acima mencionados consiste na construção direta da **AD** levando em consideração não só o aspecto estrutural como também o aspecto semântico. Esta construção será guiada por uma Função de Avaliação Pragmática (**FAP**) que engloba ambos os aspectos. Este novo algoritmo (que utiliza a **FAP** como função de avaliação) será chamado de **IDRT** (*Induction of Decision Relevants Trees*).

V.1. Função de Avaliação Pragmática

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

$$m(A_i) = p \bullet N(A_i) + (1 - p) \bullet IR(A_i)$$

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

$N(A_i)$ é a função de avaliação estrutural normalizada

$IR(A_i)$ é a intensidade de relevância do atributo A_i

Na ausência de uma melhor indicação para compor uma **FAP** equilibrada, um valor razoável para **p** seria $p = 1/2$.

V.1.1. Função de Avaliação Estrutural Normalizada

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

$$N(A_i) = \exp(-\mathbf{ent}(A_i))$$

V.1.2. Intensidade de Relevância

Para definir $IR(A_i)$, algumas definições se fazem necessárias. Considere, então:

n_v como sendo o número de valores do atributo A_i
n_d como sendo o núemro de elementos de classificação
$IR_v(v_k|A_i)$ como sendo a intensidade de relevância do valor v_k do atributo A_i

Pode-se agora definir a intensidade de relevância do atributo A_i como segue:

$$IR(A_i) = (1/n_v) \bullet \sum_{k=1}^{n_v} IR_v(v_k|A_i)$$

$$\text{Onde: } IR_v(v_k|A_i) = (1/n_d) \bullet \sum_{j=1}^{n_d} f(v_k|Dj)$$

$$\text{Sendo: } f(v_k|Dj) = \begin{cases} 1, \text{ se } v_k \text{ pertence a } Dj \\ 0, \text{ caso contrário} \end{cases}$$

V.2. Exemplo de uma AD IDRT

Considerando os mesmos dados de entrada (conjunto de exemplos da fig. 2.1 e **MR** da fig. 3.1) utilizados no exemplo com o **ADEX**, o **IDRT** gera a **AD** mostrada na figura abaixo.

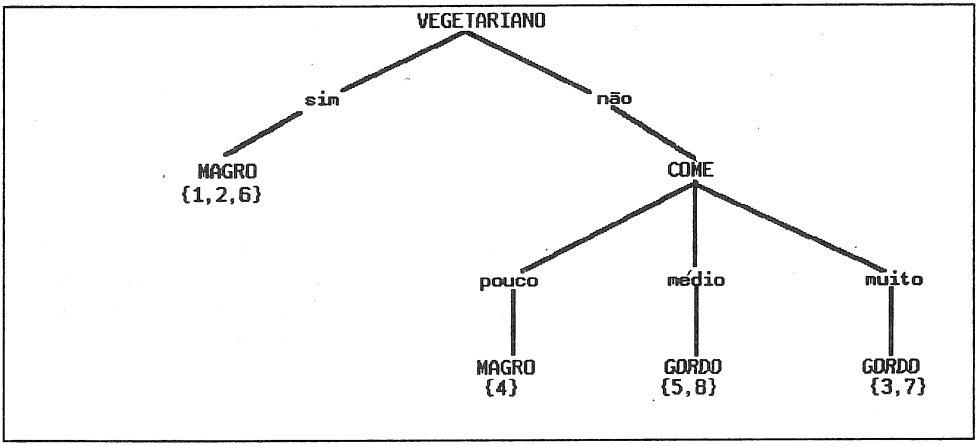

Figura 5.1 - **AD** gerada pelo algoritmo **IDRT**

A **AD** da figura 5.1, por sua vez, pode ser mapeada no seguinte conjunto de regras:

p_1) **SE** VEGETARIANO = sim **ENTÃO** DIAGNÓSTICO = MAGRO
p_2) **SE** VEGETARIANO = não **E** COME = pouco **ENTÃO** DIAGNÓSTICO = MAGRO
p_3) **SE** VEGETARIANO = não **E** COME = médio **ENTÃO** DIAGNÓSTICO = GORDO
p_4) **SE** VEGETARIANO = não **E** COME = muito **ENTÃO** DIAGNÓSTICO = GORDO

Analisando as regras fornecidas pelo **IDRT** e comparando-as com as do **ID3** e do **ADEX**, conclui-se que as primeiras são melhores que as duas últimas, visto que:

i) As regras do **IDRT**, apesar de ligeiramente maiores que as do **ID3**, em sua totalidade, são úteis do ponto de vista semântico.

ii) Embora as regras do **ADEX** sejam todas úteis, semanticamente falando, elas são bem maiores que as do **IDRT**.

Apesar dos bons resultados obtidos neste exemplo, não se pode afirmar *a priori* que isto ocorerá em todos os casos. No entanto, pelas idéias qui apresentadas, espera-se que este resultado se repita para a maioria dos casos.

VI. Conclusão

Sendo a aquisição de conhecimento o ponto de estrangulação no desenvolvimento de sistemas baseados em conhecimento, todo esforço no sentido de se obter novos métodos de aquisição ou melhorar os métodos existentes é de suma importância. O processo de aquisição automática de conhecimento a partir de exemplos é um dos mais promissores e, por isto, vem sendo alvo de frequentes investigações.

Neste trabalho são apresentadas, analisadas e comparadas duas soluções para o problema das regras inúteis. A primeira (**ADEX**), originalmente encontrada para o **ID3**, e posteriormente generalizada para todos os algoritmos da família TDIDT, consiste em expandir semanticamente as **AD**'s geradas com uso de conhecimento (**MR**) fornecido pelo especialista. A segunda (**IDRT**), uma nova solução proposta pelos autores, consiste na construção direta da árvore de decisão, onde esta construção é guiada por uma função de avaliação pragmática que engloba não só o aspecto estrutural como também o aspecto semântico.

Numa análise comparativa, um mesmo exemplo foi utilizado pelos três algoritmos (**ID3**, **ADEX** e **IDRT**). Observando-se as regras geradas pelo **IDRT** nota-se que:

i) Do ponto de vista semântico elas são equivalentes às regras obtidas pelo **ADEX** e bem superiores às do **ID3**.

ii) Do ponto de vista estrutural (quantidade de regras mais comprimento das premissas) as regras **IDRT** são menores que as do **ADEX** e praticamente iguais as do **ID3**.

Esse resultado favorável do **IDRT** é explicado pelo fato de que na construção da **AD**, o algoritmo considera não só o aspecto estrutural mas também o aspecto semântico. Embora o resultado apresentado pelo **IDRT** seja encorajador, os autores estão cientes de que isto poderá nem sempre ocorrer. Uma análise da estabilidade do **IDRT** e ajustes na função de avaliação pragmática se fazem necessários.

Referências

[Boman 89] Boman, M.; *The Problem of Induction*; SYSLAB working paper nº 152; University of Stockholm; 1989.

[Breiman 84] Breiman, L., Friedman, J., Olshen, R. e Stone, C.; *Classification and Regression Trees*; Wadsworth; Belmont; 1984.

[Boose 84] Boose, J. H.; *Personal Construct Theory and the Transfer of Human Expertise*; Anais do AAAI; 1984.

[Boose 87] Boose, J. H.; *Expertise Transfer and Complex Problems: Usign AQUINAS as a Knowledge*; Aquisition Workbench for Knowledge-Based Systems; II Man-Machine Studies; Academis Press; London; 1987.

[Boy 87] Boy, G., Faller, B. e Sallantin, J.; *Acquisition et Ratification de Connaissances*; Relatório Técnico do Centre de Recherche en Informatique de Montpelier; CRIM; 1987.

[Cendrowska 88] Cendrowska, J.; *PRISM: An Algorithm for Inducing Modular Rules*; in Knowledge-Based Systems, vol. 1; Academic Press; 1988.

[Cirne 90] Cirne Filho, W. C. e Mongiovi, G.; *Expansão Semântica para os Métodos de Aquisição de Conhecimento da Família TDIDT*; Anais do 1º Simposio de Inteligencia Artificial y Robotica; Luján - Buenos Aires - Argentina; 1990

[Gomes 88] Gomes, F. A. C.; Mongiovi, G.; Silva, H. M.; *APREND - Um Sistema de Aquisição Automática de Conhecimento a Partir de Exemplos*; 14ª Conferência Latinoamericana de Informática; Buenos Aires; Outubro/1989.

[Gomes 89] Gomes, F. A. C.; *APREND: Um Sistema de Aquisição Automática de Conhecimento a Partir de Exemplos*; Dissertação de Mestrado; Mestrado de Informática da UFPB; Campina Grande; 1989.

[Mingers 89] Mingers, J.; *An Empirical Comparison of Selection Measures for Decision-Tree Induction*; Machine Learning 3; pg. 319-342; 1989.

[Mongiovi 90] Mongiovi, G. e Cirne Filho, W. C.; *Um Algoritmo Baseado em Conhecimento para Ampliar a Potencialidade do ID3*; Anais do 7º Simpósio Brasileiro de Inteligência Artificial; Campina Grande; 1990.

[Quinlan 83] Quinlan, J. R.; *Learning from Noisy Data*; Proceedings of the International Machine Learning Workshop; University of Illinois; 1983.

[Quinlan 86] Quinlan, J. R.; *Induction of Decision Tree*; in Machine Learninig Journal I; Kluver Academic Publisher; 1986.

[Quinlan 87] Quinlan, J. R.; *Inductive Knowledge Aquisition: A Case Study*; in Application of Expert Systems; Addison-Wesley; 1987.