

Pedro Ribeiro de Andrade Neto

**Descoberta de Conhecimento, Utilizando a Teoria de Conjuntos
Aproximados, em SRCs com Informação Agregada**

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências da disciplina Projeto Orientado II para obtenção do título de Bacharel em Ciência da Computação.

Orientador
Prof. Joaquim Quinteiro Uchôa

Lavras
Minas Gerais - Brasil
Dezembro - 2002

Pedro Ribeiro de Andrade Neto

**Descoberta de Conhecimento, Utilizando a Teoria de Conjuntos
Aproximados, em SRCs com Informação Agregada**

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências da disciplina Projeto Orientado II para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 17 de dezembro de 2002

Prof. Bruno de Oliveira Schneider

Prof. Joaquim Quinteiro Uchôa
(Orientador)

Lavras
Minas Gerais - Brasil

*A meus pais,
Madaly e Gabriel.*

Agradecimentos

Agradeço a Deus por ter me ajudado nesta etapa da minha vida.

Agradeço ao Prof. MSc. Joaquim Quinteiro Uchôa, pelo incentivo e dedicação na orientação deste trabalho.

Agradeço aos meus familiares por terem me apoiado e incentivado.

Agradeço aos amigos José Augusto e Giullia Maria pelo apoio durante os estudos.

Agradeço aos meus companheiros de república, amigos RPGistas e colegas de classe pelo incentivo.

Resumo

A Teoria de Conjuntos Aproximados, TCA, tem sido utilizada em várias áreas de pesquisa, principalmente naquelas relacionadas com representação de conhecimento e aprendizado de máquina. Este trabalho investiga a possibilidade do uso de informação agregada aos atributos e valores associados a esses atributos, modificando o RS1+, um algoritmo baseado no índice discriminante de atributos.

Abstract

The Rough Sets Theory, RST, has been used in several areas of research, mainly in those related with knowledge representation and machine learning. This work explores the possibility of using aggregate information in attributes and associated values of this attributes, modifying the RS1+, an algorithm based in the attribute discriminating index.

Sumário

1	Introdução	1
2	Sistemas Baseados em Conhecimento	5
2.1	Introdução	5
2.2	Sistemas Baseados em Conhecimento	5
2.3	Aquisição de Conhecimento	8
2.4	Aprendizado de Máquina	9
2.4.1	Aprendizado Indutivo de Máquina	10
2.4.2	Tratamento de Incerteza em Aprendizado de Máquina	13
3	Elementos da Teoria de Conjuntos Aproximados	15
3.1	Introdução	15
3.2	Teoria de Conjuntos Aproximados – TCA	15
3.2.1	Espaços Aproximados	15
3.2.2	Classificação Aproximada de um Conjunto	16
3.2.3	Regiões do Espaço Aproximado	16
3.2.4	Medidas em um Espaço Aproximado	19
3.3	Representação de Conhecimento	20
3.3.1	Sistemas de Representação de Conhecimento (SRCs)	20
3.3.2	Tabelas de Decisão	22
3.3.3	Redução do Conjunto Inicial de Atributos	24
4	Algoritmo de Aprendizado Indutivo de Máquina Baseado em Índice Discriminante de Atributos	27
4.1	Introdução	27
4.2	Aprendizado usando RS1+	28
4.3	Comentários Finais	34

5	Descoberta de Conhecimento Utilizando Informação Agregada	37
5.1	Introdução	37
5.2	Alterações no Algoritmo RS1+	37
5.2.1	Normalização de Custos	37
5.2.2	Índice Discriminante de Atributos	38
5.2.3	Escolha do Melhor Atributo	39
5.2.4	Redutos	39
5.2.5	Valores associados aos atributos	40
5.3	Testes Realizados	41
5.3.1	Teste Inicial	41
5.3.2	Testes com Base de Dados Reais	42
5.3.3	Comentários	46
6	Conclusão	47

Lista de Figuras

2.1	Arquitetura básica de um SBC	6
2.2	Esquema geral de aprendizado indutivo de regras	11
3.1	Conjunto X no espaço aproximado $A = (U, R)$	17
3.2	Aproximações de X em A	18
3.3	Regiões de X em A	18

Lista de Tabelas

3.1	SRC onde $U = \{x_1, x_2, \dots, x_9, x_{10}\}$ e $Q = \{a, b, c, d\}$	21
4.1	SRC onde $U = \{x_1, x_2, \dots, x_{11}, x_{12}\}$ e $Q = \{a, b, c, d, e\}$	29
4.2	$Class_S(e)$	30
4.3	Cálculo de $\alpha_{B'}(X)$, com $X = \{x_1, x_2, x_3, x_7\}$ e $B = \emptyset$	30
4.4	Cálculo de $\alpha_{B'}(X)$, com $X = \{x_7\}$ e $B = \{b\}$	31
4.5	Cálculo de $\alpha_{B'}(X)$, com $X = \{x_4, x_5, x_6, x_8, x_{11}, x_{12}\}$ e $B = \emptyset$	32
4.6	Cálculo de $\alpha_{B'}(X)$, com $X = \{x_9, x_{10}\}$ e $B = \emptyset$	33
4.7	Cálculo de $\alpha_{B'}(X)$, com $X = \{x_9, x_{10}\}$ e $B = \{b\}$	34
5.1	SRC onde $U = \{x_1, x_2, x_3, x_4, x_5\}$ e $Q = \{a, b, c\}$	40
5.2	Pesos dos valores associados aos atributos do SRC da Tabela 5.1	40
5.3	SRC da Tabela 5.1 com custos agregados aos valores associados aos atributos	41
5.4	Síglas das tabelas de resultados	42
5.5	Resultados, utilizando custos, com o SRC da Tabela 4.1	42
5.6	Resultados com o exemplo <i>cars</i>	44
5.7	Resultados com o exemplo <i>hepatitis</i>	44
5.8	Custos e porcentagem das regras geradas do exemplo <i>hepatitis</i>	45
5.9	Resultados com o exemplo <i>heart-disease</i>	45
5.10	Custos e porcentagem das regras geradas do exemplo <i>heart-disease</i>	46

Capítulo 1

Introdução

Uma das mais proeminentes áreas em Inteligência Artificial é a de Sistemas Baseados em Conhecimento (SBCs), ou uma de suas principais formas, os Sistemas Especialistas (SEs). Estes sistemas são definidos formalmente como programas de computador que resolvem problemas utilizando conhecimento representado explicitamente e que, não fosse essa representação, exigiriam um especialista humano no domínio do problema para a sua solução.

O conhecimento que um sistema possui pode ser fornecido por especialistas, ou adquirido utilizando técnicas de extração de conhecimento. Entre essas técnicas encontra-se o Aprendizado de Máquina (AM). Inúmeras técnicas de AM têm sido desenvolvidas nos últimos anos, a maioria baseada em representação simbólica.

O aprendizado simbólico conhecido como aprendizado baseado em exemplos é o que mais tem contribuído efetivamente para a implementação de sistemas de aprendizado de máquina. A partir de um determinado conjunto de exemplos, expressões para tarefas classificatórias podem ser aprendidas (induzidas) como, por exemplo, diagnóstico de doenças, previsão meteorológica, predição do comportamento de novos compostos químicos, predição de propriedades mecânicas de metais com base em algumas propriedades químicas, etc. Alguns sistemas, conhecidos como Sistemas Baseados em Regras (SBRs), utilizam-se de regras da forma ($a \Rightarrow b$), – lê-se se a é verdade, então b também é verdade, ou resumidamente se a , então b – para codificação do conhecimento.

Via literatura é possível evidenciar que a pesquisa sendo conduzida na área de aprendizado indutivo proposicional se polariza, principalmente, na investigação de novos algoritmos que, via de regra, são refinamentos, adaptações e/ou melhoramentos dos algoritmos básicos (ID3 [Quinlan et alii(1986)], AQ [Michalski (1983)]

e CN2 [Clark & Niblett (1989)], ou então na proposta de novos modelos. Todos esses modelos têm como base de informações um conjunto de exemplos e decisões (ambos contidos em tabelas), e a partir daí gera-se as regras de decisão, baseando-se em padrões. Nota-se que eles utilizam apenas informações contidas nas tabelas (valores associados aos atributos) para prover conclusões. Mas é visto que no mundo real existem mais informações, tais como informações agregadas aos próprios atributos e até aos valores associados a esses atributos, o que geralmente não é tratado por estes modelos da literatura.

Por exemplo: existem dois tipos de exames para determinar se uma pessoa tem uma suposta doença. Um dos exames tem custo baixo, e fornece resultados negativos com 100% de certeza, já os resultados positivos não são totalmente confiáveis; o segundo exame é mais caro, mas fornece o resultado com um maior grau de confiança. O recomendado é fazer o primeiro exame, e, caso o resultado seja positivo, faz-se então o segundo, para remover as dúvidas. Um SBC normal recomendaria fazer o segundo exame, pelo fato de todas as regras terem o mesmo valor quantitativo, e a regra associada ao segundo exame ter um maior valor qualitativo.

Uma outra questão pertinente quando da utilização e representação de conhecimento é a imperfeição da informação. Muitas vezes, o que se convencionou chamar de informações imperfeitas abrange informações imprecisas, inconsistentes, parcialmente ignoradas e mesmo incompletas. Esta presença da incerteza em SBCs pode se originar de várias fontes:

- Confiabilidade parcial que se tem na informação;
- Imprecisão inerente da linguagem de representação na qual a informação é expressa;
- Não completude da informação;
- Agregação/sumarização da informação que provêm de múltiplas fontes.

A Teoria de Conjuntos Aproximados (TCA), surgida a partir de estudos de Pawlak em Sistemas de Informação, pode ser utilizada com sucesso para a implementação de métodos de representação do conhecimento incerto, bem como um formalismo, subsidiando aprendizado de máquina. Essa abordagem tem sido utilizada em Inteligência Artificial e Ciência Cognitiva, especialmente nas áreas de:

- aprendizado de máquina;
- aquisição de conhecimento;

- raciocínio indutivo;
- descoberta de conhecimento em base de dados;
- sistemas de suporte à decisão;

Pelo fato de a Teoria de Conjuntos Aproximados ser um formalismo muito expressivo em aprendizado de máquina, e por ela ser uma teoria que pode ser utilizada com sucesso para implementação de métodos de representação de conhecimento incerto, ela foi escolhida como base para nossos estudos, surgindo o objetivo deste trabalho: propor um algoritmo para descoberta de conhecimento que, utilizando a TCA e informações relacionadas aos atributos e aos valores associados a esses atributos, possa fornecer regras com diferentes graus de importância.

Esta monografia está organizada da seguinte forma: no Capítulo 2 são apresentados Sistemas Baseados em Conhecimento (SBCs), evidenciando a área de Aprendizado de Máquina (AM), como uma das maneiras de realizar Aquisição de Conhecimento para a construção da Base de Conhecimento em SBCs. No Capítulo 3 são apresentados os principais conceitos da TCA e descritas as medidas mais relevantes propostas na teoria. O Capítulo 4 descreve o algoritmo RS1+, o qual serão propostas modificações para utilizar informação agregada, descritas no Capítulo 5. Finalmente no Capítulo 6 são apresentadas as conclusões e trabalhos futuros do projeto.

Capítulo 2

Sistemas Baseados em Conhecimento

2.1 Introdução

Uma das mais proeminentes áreas em Inteligência Artificial é a de Sistemas Baseados em Conhecimento (SBCs), ou uma de suas principais formas, os Sistemas Especialistas (SEs).

Os objetivos deste capítulo (baseado no capítulo 2 de [Uchôa (1998)]) são: apresentar os principais componentes necessários à construção de SBCs e introduzir o conceito de Aprendizado de Máquina como um método de Aquisição de Conhecimento para a construção da Base de Conhecimento do SBC.

2.2 Sistemas Baseados em Conhecimento

SBCs são definidos formalmente como programas de computador que resolvem problemas utilizando conhecimento representado explicitamente e que, não fosse essa representação, seria exigido um especialista humano no domínio do problema para a sua solução. Conhecimento e processo de resolução de problemas são pontos críticos e essenciais durante o desenvolvimento de um SBC. A arquitetura básica de um SBC pode ser visualizada na Figura 2.1.

Um SBC possui, então, três módulos principais, a saber:

1. **Base de Conhecimentos (BC):** contém o conhecimento específico do domínio da aplicação. É composto de fatos sobre o domínio, regras que des-

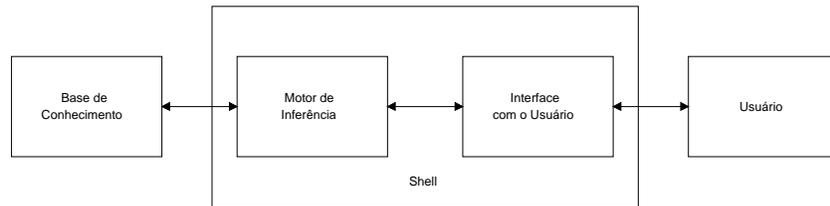


Figura 2.1: Arquitetura básica de um SBC

crevem relações no domínio e métodos e heurísticas para resolução de problemas no domínio.

2. **Motor de Inferência (MI):** mecanismo responsável pelo processamento do conhecimento da BC, utilizando-se de alguma linha de raciocínio. Implementa as estratégias de inferência e controle do SBC. Quando o conhecimento do SBC está expresso como regras, as estratégias de controle empregadas pelo MI normalmente são encadeamento para trás (*backward chaining*) ou encadeamento para frente (*forward chaining*). Quando o MI usa a estratégia de encadeamento para trás, uma lista de hipóteses é pesquisada, procurando reunir evidências para viabilizar a conclusão da validade de alguma(s) dela(s). Essa estratégia corresponde à pergunta: “É possível provar as hipóteses a partir dos dados disponíveis?”. Se a estratégia de controle for encadeamento para frente, o MI parte dos dados e, com base nas regras de conhecimento, deduz outras asserções, procurando chegar à solução do problema. Essa estratégia corresponde à pergunta: “O que é possível concluir a partir dos dados disponíveis?”.
3. **Interface com o Usuário (IU):** módulo responsável pela comunicação entre o usuário e o sistema. Deve fornecer, também, justificativas e explicações referentes às conclusões obtidas da BC, bem como ao raciocínio utilizado, comprovando a sua veracidade.

Sistemas Especialistas (SEs) constituem uma importante subclasse dos SBCs. Um SE pode ser definido como um SBC que resolve problemas bem específicos do mundo real, problemas estes que requerem considerável conhecimento, habilidade e heurísticas para sua resolução. Na literatura, os termos Sistemas Baseados em Conhecimento e Sistemas Especialistas são, muitas vezes, utilizados quase sem distinção. Adota-se essa abordagem neste texto.

A grande maioria dos SBCs apresenta-se sob a forma de Sistemas Baseados em Regras (SBRs), que utilizam-se de regras para codificação do conhecimento. Uma regra é uma estrutura da forma $(a \Rightarrow b)$, e é lida se a é verdade, então b também é verdade, ou resumidamente se a , então b . O termo a é denominado antecedente da regra, e expressa as condições de execução da mesma. O termo b , por sua vez, é denominado conseqüente da regra e, caso satisfeitas as condições expressas no antecedente, este torna-se um fato. Um fato é uma regra da forma $(\Rightarrow b)$, ou, resumidamente, (b) . O fato de o antecedente ser vazio implica na verdade incondicional de b .

Observe que, em um SBR, o Motor de Inferência é um interpretador de regras. Os SBCs também podem apresentar-se sob a forma de Sistemas Baseados em Árvores de Decisão, que utilizam-se de uma árvore de decisão para codificação do conhecimento. Numa árvore de decisão, os nós não-folha são considerados como os antecedentes da regra, e sua análise permite que se percorra a árvore em busca de um nó folha que é considerado o conseqüente da regra. Note que, em um SBC baseado em árvore de decisão, o Motor de Inferência é um algoritmo que percorre uma árvore em busca de um nó folha. Os nós são analisados, indicando o caminho que deve ser seguido na árvore, da raiz até um nó folha. Os nós não-folha possuem um ou mais valores (atributos) que compõem o antecedente (suas condições) de uma regra, enquanto que o nó folha, que também possui um ou mais valores, irá compor o conseqüente da regra (sua conclusão). O número de regras existentes numa árvore de decisão é equivalente ao número de nós-folha da mesma.

A capacidade de resolver problemas que demandam informações referentes a um determinado domínio de conhecimento e explicar seu comportamento tornaram-se os principais aspectos de um SBC. Um SBC deve ser capaz de explicar seu comportamento e suas decisões ao usuário, respondendo *por que* e *como* chegou a determinada solução. De uma maneira geral, as perguntas *por que* referem-se a qual conhecimento respalda a conclusão; as perguntas *como*, por sua vez, referem-se aos passos de raciocínio seguidos para determinar a solução do problema. Esta característica é especialmente necessária quando o SBC lida com domínios incertos (diagnóstico médico, por exemplo); a explicação pode, de certa forma, aumentar o grau de confiança que o usuário deposita no sistema, ou então, ajudá-lo a encontrar alguma falha no seu raciocínio.

Uma outra característica frequentemente necessária é a habilidade de lidar com incertezas e informações incompletas: a informação a respeito do problema a ser resolvido pode estar incompleta ou ser parcialmente confiável, bem como as relações no seu domínio podem ser aproximadas. Espera-se, também, que um SBC

seja flexível o suficiente para permitir, facilmente, a acomodação de novo conhecimento.

2.3 Aquisição de Conhecimento

Uma das principais atividades relacionadas ao desenvolvimento de Sistemas Baseados em Conhecimento consiste na transferência do conhecimento – informações e/ou formas de condução do raciocínio – do especialista humano (ou de qualquer outra fonte) à Base de Conhecimento do SBC. Esse processo é conhecido como Aquisição de Conhecimento (AC) e é, reconhecidamente, o processo mais difícil durante o desenvolvimento de SBCs, exigindo um grande investimento em tempo e esforço.

Além da extração do conhecimento necessário, tal conhecimento deve ser traduzido para o esquema formal de representação usado pelo SBC, e deve ser repetidamente refinado, até que o sistema atinja um grau de desempenho próximo ao de um especialista humano na resolução do problema. Sobre esse processo, pode ser observado em [Nicoletti (1994)] que:

...o processo de extração do conhecimento do especialista envolve um intenso questionamento que, em certas situações, pode interferir na sua própria percepção de como elabora o raciocínio. Geralmente um especialista acha difícil detalhar descrições de seu conhecimento e de como o usa; por outro lado, muitas vezes é difícil para o engenheiro do conhecimento traduzir com exatidão aquele conhecimento, geralmente amplo e multifacetado, em uma linguagem de representação restrita. Devido a essas dificuldades, muitas vezes o conhecimento extraído pode ser inconsistente (como consequência de diferenças individuais entre especialistas), incompleto e impreciso.

Existe um vasto conjunto de métodos e ferramentas que facilitam a tarefa da AC. A disponibilidade dessa variedade reflete o fato da AC ser um processo multidimensional, podendo ocorrer em diferentes estágios do desenvolvimento de um SBC, e podendo envolver vários tipos de conhecimento. Uma classificação proposta em [Boose (1989)] agrupa os métodos e técnicas para AC em:

1. **manuais:** consistem tipicamente de entrevistas e análise de protocolos;
2. **baseadas em computador:**

- (a) *interativas*: englobam, principalmente, ferramentas que entrevistam o especialista, fazem análise textual, extraem e analisam conhecimento de múltiplas fontes separadamente e as combinam para uso;
- (b) *baseadas em aprendizado*: são as que, via regras, generalizam situações específicas em conceitos.

2.4 Aprendizado de Máquina

Como observado em [Michalski & Tecuci (1993)], o aprendizado pode ser caracterizado como um processo multidimensional que, via de regra, ocorre através da aquisição de conhecimento declarativo, do desenvolvimento de habilidades motoras e cognitivas através de instrução e prática, da organização do conhecimento existente em representações mais efetivas, da descoberta de novos fatos e/ou teorias através da observação e experimentação ou, então, através da combinação e/ou composição dessas dimensões.

Um sistema inteligente deve ser capaz de formar conceitos, i.e., classes de entidades unidas por algum princípio. Esse pode ser o fato das entidades terem algum uso ou objetivo comum ou, simplesmente, por terem características perceptuais similares. Para a utilização do conceito, o sistema deve também desenvolver métodos eficientes para o reconhecimento de pertinência de uma dada entidade ao conceito. O estudo e a modelagem de processos pelos quais o sistema adquire, refina e diferencia conceitos são objetivos da área de pesquisa chamada de *Aprendizado de Conceito*, uma subárea de AM.

Em Aprendizado de Conceito, o termo *conceito* usualmente indica uma classe de equivalência de entidades, que pode ser descrita via um conjunto relativamente limitado de expressões e que deve ser suficiente para a diferenciação entre conceitos. Entidades individuais na classe são chamadas de *instâncias* do conceito. O fato do conceito ser definido como uma classe de equivalência torna cada instância de uma classe igualmente representativa do conceito em questão; tal definição estabelece, também, os limites da descrição daqueles conceitos – uma entidade satisfaz ou não satisfaz o conceito. Mais detalhes sobre aprendizado de conceito podem ser encontrados em [Mitchell (1997)] e [Langley (1996)].

Observe que, dada a complexidade existente no processo de aprendizado, não há, em relação a ele, uma definição única e tampouco um entendimento suficiente. Podem ser encontradas na literatura definições que se diferenciam por causa da ênfase dada a diferentes aspectos do processo de aprendizado.

Uma abordagem mais limitada de AM, geralmente adotada por pessoas que

trabalham especificamente com SBCs, é a de que aprendizado é a aquisição de conhecimento explícito. Muitos SBCs representam conhecimento como um conjunto de regras que necessitam ser adquiridas, organizadas e estendidas. Isso enfatiza a importância de tornar explícito o conhecimento adquirido, de maneira que ele possa facilmente ser verificado, modificado e explicado.

Na literatura há diversas taxinomias de sistemas de AM. Algumas das várias razões que dão origem a esta diversidade de classificações são: diferentes estratégias de aprendizado adotadas, existência de diferentes representações do conhecimento, domínios de aplicação variados, etc. Uma possível taxinomia de métodos de Aprendizado de Máquina classifica os paradigmas em:

Aprendizado Simbólico: aquisição de conceitos expressos em símbolos, através de conjuntos de exemplos;

Aprendizado Baseado em Instância: métodos que simplesmente armazenam as instâncias de treinamento;

Aprendizado Baseado em Algoritmos Genéticos: que inclui ambos: algoritmos genéticos, que induzem hipóteses descritas usando cadeias de bits, e programação genética, que induz hipóteses descritas como programas;

Aprendizado Conexionista: buscam modelar o processo de funcionamento dos neurônios e/ou áreas do cérebro humano;

Aprendizado Analítico: aprendizado baseado em explicações e certas formas de métodos de aprendizado analógicos e baseados em casos.

Note que, dentre todos esses modelos, apenas o Aprendizado Analítico é de natureza dedutiva; todos os demais são indutivos.

2.4.1 Aprendizado Indutivo de Máquina

Entre os vários modelos existentes para aprendizado apontados anteriormente, o aprendizado simbólico conhecido como aprendizado indutivo baseado em exemplos é o que mais tem sido pesquisado e o que mais tem contribuído efetivamente para a implementação de sistemas de aprendizado de máquina. A partir de um conjunto de exemplos, expressões para tarefas classificatórias podem ser aprendidas (induzidas) como, por exemplo, diagnóstico de doenças, previsão meteorológica, previsão do comportamento de novos compostos químicos, previsão de propriedades mecânicas de metais com base em algumas de suas propriedades químicas, etc. A Figura 2.2, extraída de [Shaw & Gentry (1990)], ilustra esse processo.

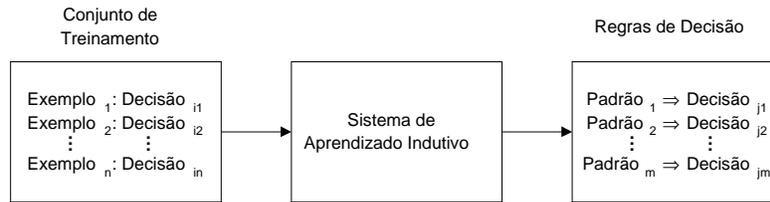


Figura 2.2: Esquema geral de aprendizado indutivo de regras

No aprendizado indutivo baseado em exemplos, também referenciado como aprendizado indutivo, o conjunto de exemplos, também denominado *conjunto de treinamento*, é fornecido ao sistema por um instrutor ou pelo ambiente (base de dados, sensores, etc.). Esse conjunto de treinamento é geralmente composto de *exemplos positivos* (exemplos do conceito) e *exemplos negativos* (contra-exemplos do conceito). A indução do conceito corresponde a uma busca no espaço de hipóteses, de forma a encontrar aquelas que *melhor* classificam os exemplos. Neste contexto, *melhor* pode ser definido em termos de precisão e compreensibilidade.

De uma maneira geral, um sistema que aprende a partir de exemplos recebe como dados informações na forma de situações específicas, cada uma delas devidamente classificadas (geralmente por especialista humano no domínio), caracterizando o que se convencionou chamar de *aprendizado supervisionado*, e produz como resultado uma(s) hipótese(s) que generaliza(m) aquelas situações inicialmente fornecidas. Entre as principais características de sistemas de aprendizado indutivo de máquina, encontram-se:

Incrementabilidade: num sistema incremental, a expressão do conceito vai sendo construída exemplo a exemplo, e implica em constante revisão do conceito (ou conceitos) até então formulado(s). Um novo exemplo pode, eventualmente, causar um rearranjo da expressão do conceito. A expressão do conceito vai se modificando à medida que os exemplos vão se tornando disponíveis. No caso não-incremental, o conjunto de treinamento deve estar totalmente disponível no início do processo de aprendizado, uma vez que o conceito é induzido considerando-se todos os exemplos de uma vez.

Teoria de Domínio: no caso em que o sistema não possui informação a respeito do problema de aprendizado sendo tratado, o sistema induz a expressão do conceito apenas a partir dos exemplos disponíveis. Entretanto, para que a solução de problemas difíceis de aprendizado sejam encontradas, é fundamen-

tal que um volume substancial de conhecimento sobre o problema esteja disponível, de maneira a subsidiar a indução do conceito. Esse conhecimento prévio existente é conhecido como Teoria de Domínio ou Conhecimento de Fundo.

Linguagem de Descrição: em inferência indutiva, os exemplos, a Teoria de Domínio e as hipóteses formuladas necessitam ser expressos em alguma linguagem, e geralmente são utilizadas linguagens formais.

Uma vez desenvolvido o sistema de AM, é interessante avaliar o resultado deste sistema. De uma forma geral, entre os critérios mais usuais para se medir o sucesso de um sistema de AM encontram-se:

Precisão de classificação: é geralmente definida como o percentual de exemplos corretamente classificados pela hipótese induzida.

Transparência da descrição induzida do conceito: em muitos domínios de aplicação (por exemplo, diagnóstico médico), é fundamental que a descrição do conceito, gerada por um sistema de AM, seja compreensível pelo ser humano. O entendimento do conceito, por parte do ser humano, não apenas aumenta a credibilidade do sistema de AM, mas também permite que o conceito possa ser assimilado e utilizado pelo especialista humano. Em muitas situações, a transparência da descrição é medida pelo número de descritores e operadores usados na descrição do conceito.

Complexidade computacional: está relacionada com os recursos computacionais necessários (tempo e espaço) para realizar o aprendizado.

Como já comentado, para a expressão de um modelo de aprendizado indutivo são necessárias linguagens que descrevam os exemplos de treinamento, assim como linguagens que descrevam os conceitos aprendidos. Vários formalismos lógicos têm sido usados em sistemas de aprendizado indutivo para a representação de exemplos e conceitos. Em geral, distinguem-se dois tipos de descrição: *descrição baseada em atributos* (ou proposicional) e *descrição relacional*. Em uma descrição baseada em atributos, exemplos são descritos como vetores de pares atributo-valor e uma classe associada. Em uma descrição relacional, exemplos do conceito a ser aprendido são fornecidos como fatos. O aprendizado relacional se caracteriza também por viabilizar o uso da Teoria de Domínio; a incorporação desse tipo de conhecimento ao processo de aprendizado quase sempre contribui para uma expressão mais concisa e mais natural do conceito a ser aprendido.

2.4.2 Tratamento de Incerteza em Aprendizado de Máquina

Um dos principais problemas em Aprendizado de Máquina é o de representação de incerteza. É fato que todo SBC possui problemas relativos à incerteza. Esta pode se manifestar de diversas formas: imprecisão, incompletudeza, inconsistência, etc. Torna-se, portanto, desejável que a estratégia de AM implementada, bem como o próprio SBC, possua mecanismos de tratamento da incerteza. Esses mecanismos freqüentemente surgem sob a forma de medidas de incerteza.

A Teoria de Conjuntos Aproximados, formalismo tratado neste trabalho, possui mecanismos para expressão de, entre outros, um tipo fundamental de incerteza: a indiscernibilidade. A indiscernibilidade surge quando não é possível distinguir objetos de um mesmo conjunto; representa a situação em que esses objetos parecem todos ser um único objeto.

Capítulo 3

Elementos da Teoria de Conjuntos Aproximados

3.1 Introdução

Um conjunto aproximado é um modelo matemático usado para tratar um tipo de incerteza – a *indiscernibilidade*. De uma forma simplista, conjuntos aproximados podem ser considerados conjuntos com fronteiras nebulosas, ou seja, conjuntos que não podem ser caracterizados precisamente utilizando-se o conjunto de atributos disponíveis. Uma das principais vantagens da Teoria de Conjuntos Aproximados (TCA) é a de não necessitar de qualquer informação adicional ou preliminar a respeito de dados, tais como: distribuição de probabilidade, atribuição de crenças, grau de pertinência ou possibilidade.

Este capítulo tem dois objetivos, a saber: o primeiro é apresentar os principais conceitos e medidas da Teoria de Conjuntos Aproximados (TCA), como subsídio para os estudos realizados e, conseqüentemente, para compreensão do próximo capítulo; o segundo é apresentar os principais conceitos de Sistemas de Representação de Conhecimento (SRCs) sob a ótica da TCA. Este capítulo foi baseado em [Pawlak (1991)] e em [Uchôa (1998)].

3.2 Teoria de Conjuntos Aproximados – TCA

3.2.1 Espaços Aproximados

Um *espaço aproximado* é um par ordenado $A = (U, R)$, onde:

- U é um conjunto não vazio, denominado *conjunto universo*;
- R é uma relação de equivalência sobre U , denominada *relação de indiscernibilidade*. Dados $x, y \in U$, se xRy então x e y são *indiscerníveis* em A , ou seja, a classe de equivalência definida por x é a mesma que a definida por y , i.e., $[x]_R = [y]_R$.

As classes de equivalência induzidas por R em U são denominadas *conjuntos elementares*. Se E é um conjunto elementar, $des(E)$ denota a descrição dessa classe de equivalência. Essa descrição é função do conjunto de atributos que define R . Note que, dados $x, y \in E$, onde E é um conjunto elementar em A , x e y são indiscerníveis, i.e., no espaço $A = (U, R)$ não se consegue distinguir x de y , pois $des(x) = des(y) = des(E)$.

3.2.2 Classificação Aproximada de um Conjunto

Dado um espaço aproximado $A = (U, R)$ e um conjunto $X \subseteq U$, com o objetivo de verificar o quão bem X é representado pelos conjuntos elementares de A , são definidas:

- a *aproximação inferior* de X em A , $A_{A-inf}(X)$, como a união de todos os conjuntos elementares que estão contidos em X :

$$A_{A-inf}(X) = \{x \in U \mid [x]_R \subseteq X\};$$

- a *aproximação superior* de X em A , $A_{A-sup}(X)$, como a união de todos os conjuntos que possuem intersecção não vazia com X :

$$A_{A-sup}(X) = \{x \in U \mid [x]_R \cap X \neq \emptyset\}.$$

Nas notações utilizadas, quando o espaço aproximado for conhecido e não houver risco de confusão, a referência ao espaço será abolida. Assim, $A_{sup}(X)$ será usado em substituição a $A_{A-sup}(X)$.

3.2.3 Regiões do Espaço Aproximado

Dado um espaço aproximado $A = (U, R)$ e $X \subseteq U$, as aproximações inferior e superior permitem a classificação do espaço aproximado em regiões:

1. *região positiva* de X em A , formada por todas as classes de equivalência de U contidas inteiramente no conjunto X , dada por

$$pos_A(X) = A_{A-inf}(X)$$

2. *região negativa* de X em A , formada pelos conjuntos elementares de A que não estão contidos na aproximação superior de X , dada por

$$neg_A(X) = U - A_{A-sup}(X)$$

3. *região duvidosa* de X em A , formada pelos elementos que pertencem à aproximação superior, mas não pertencem à aproximação inferior, dada por

$$dub_A(X) = A_{A-sup}(X) - A_{A-inf}(X)$$

Exemplo 1 Seja U um conjunto universo e R uma relação de equivalência em U , definindo o espaço aproximado $A = (U, R)$. Seja também X , como ilustra a Figura 3.1. A aproximação inferior e a aproximação superior de X em $A = (U, R)$ são mostradas na Figura 3.2(a) e 3.2(b), respectivamente. A Figura 3.3, por sua vez, apresenta as regiões de X .

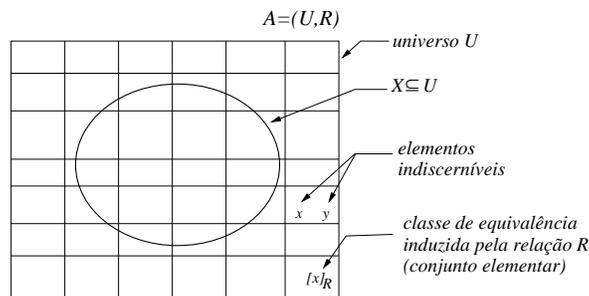


Figura 3.1: Conjunto X no espaço aproximado $A = (U, R)$.

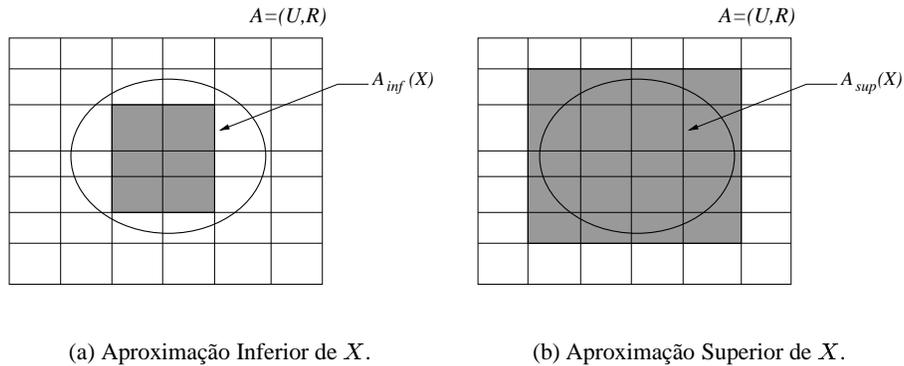


Figura 3.2: Aproximações de X em A .

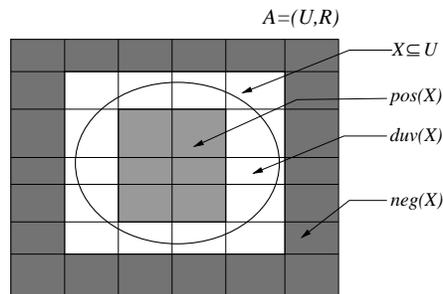


Figura 3.3: Regiões de X em A .

Seja $A = (U, R)$ um espaço aproximado e seja $X \subseteq U$. O conjunto X pode ou não ter suas fronteiras claramente definidas em função das descrições dos conjuntos elementares de A . Isso leva ao conceito de conjuntos aproximados: um *conjunto aproximado* em A é a família de todos os subconjuntos de U que possuem a mesma aproximação inferior e a mesma aproximação superior em A . Ou seja, possuem a mesma região positiva, negativa e duvidosa.

3.2.4 Medidas em um Espaço Aproximado

Dado um espaço aproximado $A = (U, R)$ e $X \subseteq U$, pode-se definir medidas que indiquem o quão bem $X \subseteq U$ pode ser representado pelo espaço aproximado $A = (U, R)$.

Acuracidade de um Aproximação

Seja um espaço aproximado $A = (U, R)$. Com a finalidade de verificar o quão bem o conjunto $X \subseteq U$ pode ser representado por A , são definidas as seguintes medidas:

- *medida interna* de X em A , $\omega_{A-inf}(X) = |A_{A-inf}(X)|$;
- *medida externa* de X em A , $\omega_{A-sup}(X) = |A_{A-sup}(X)|$;
- *qualidade da aproximação inferior* de X em A , $\gamma_{A-inf}(X) = \frac{\omega_{A-inf}(X)}{|U|}$;
- *qualidade da aproximação superior* de X em A , $\gamma_{A-sup}(X) = \frac{\omega_{A-sup}(X)}{|U|}$.

A medida de *acuracidade* de X é definida pela relação entre a cardinalidade da aproximação inferior de X pela cardinalidade da aproximação superior de X , em símbolos:

$$\omega_A(X) = \frac{\omega_{A-inf}(X)}{\omega_{A-sup}(X)} = \frac{|A_{inf}(X)|}{|A_{sup}(X)|}.$$

Quando A é conhecido e não há risco de confusão, escreve-se ω_{inf} , ω_{sup} , γ_{inf} , γ_{sup} e ω em substituição a ω_{A-inf} , ω_{A-sup} , γ_{A-inf} , γ_{A-sup} e ω_A .

Índice Discriminante de Atributos

Seja o espaço aproximado $A = (U, R)$ e $X \subseteq U$. O número total de objetos em U que podem, com certeza, ser classificados em dois subconjuntos disjuntos, X e $U - X$, é igual ao número de objetos que não pertencem à região $duv(X)$. Esse número é, pois:

$$|U - duv(X)| = |U - (A_{sup}(X) - A_{inf}(X))| = |U| - |A_{sup}(X) - A_{inf}(X)|.$$

Dessa forma, a razão:

$$\alpha_R(X) = \frac{|U| - |A_{sup}(X) - A_{inf}(X)|}{|U|},$$

definida com *índice discriminante de R com relação a X*, fornece uma medida do grau de certeza na determinação da pertinência (ou não) de um elemento de U no conjunto X .

Função de Pertinência Aproximada

Em [Pawlak (1994)] é proposta um função de pertinência aproximada, que permite calcular a pertinência de um elemento do universo U a qualquer das regiões definidas por um conjunto $X \subseteq U$. Dado um espaço aproximado $A = (U, R)$, $X \subseteq U$ e $x \in U$, a pertinência de x a X no espaço A é dada por:

$$\mu_X^A(x) = \frac{|[x]_R \cap X|}{|[x]_R|},$$

onde $[x]_R$ denota a classe de equivalência induzida pela relação de equivalência R , que contém o elemento x (e todos os seus equivalentes, por R). Quando A é conhecido e não há risco de confusão, escreve-se μ_X em substituição a μ_X^A .

3.3 Representação de Conhecimento

3.3.1 Sistemas de Representação de Conhecimento (SRCs)

Os conceitos da TCA são utilizados principalmente no contexto de Sistemas de Representação de Conhecimento. Um *Sistema de Representação de Conhecimento* (SRC) é uma quádrupla $S = (U, Q, V, \rho)$, onde U é o universo finito de S .

Os elementos de U são chamados objetos, que são caracterizados por um conjunto de atributos Q e seus respectivos valores. O conjunto de valores de atributos é dado por $V = \bigcup_{q \in Q} V_q$, onde V_q é o conjunto de valores do atributo q . Por sua vez, $\rho: U \times Q \rightarrow V$ é uma *função de descrição* tal que $\rho(x, q) \in V_q$, para $x \in U$ e $q \in Q$.

Dado um SRC $S = (U, Q, V, \rho)$, é importante observar que cada subconjunto de atributos $P \subseteq Q$ define um único espaço aproximado $A = (U, \tilde{P})$ onde \tilde{P} é a relação de indiscernibilidade (equivalência) induzida por P .

Exemplo 2 *Seja o SRC representado pela Tabela 3.1, onde $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$, $Q = \{a, b, c, d\}$ e $V = \{1, 2, 3, 4, 5, 6\}$. Neste sistema de representação de conhecimento tem-se:*

- $\rho(x_1, a) = 4$;

- $\rho(x_2, b) = 1$;
- $\rho(x_5, d) = 1$;
- $\rho(x_3, c) = 5$.

Tabela 3.1: SRC onde $U = \{x_1, x_2, \dots, x_9, x_{10}\}$ e $Q = \{a, b, c, d\}$

U	a	b	c	d
x_1	4	1	3	1
x_2	6	1	4	1
x_3	6	2	5	2
x_4	6	1	5	1
x_5	4	2	4	1
x_6	4	2	4	2
x_7	4	2	3	1
x_8	4	1	4	1
x_9	6	1	4	2
x_{10}	4	1	4	2

Exemplo 3 Seja S o SRC do Exemplo 2 e $P = \{d\}$. Neste caso tem-se que os elementos x_3, x_6, x_9 e x_{10} são indiscerníveis com relação a P , pois possuem o mesmo valor. Além disto, $A = (U, \tilde{P})$ é um espaço aproximado e seus conjuntos elementares são $E_1 = \{x_1, x_2, x_4, x_5, x_7, x_8\}$, $E_2 = \{x_3, x_6, x_9, x_{10}\}$.

O Exemplo 4 demonstra o cálculo dos principais conceitos e medidas da TCA.

Exemplo 4 Seja S o SRC definido no Exemplo 2 e seja $P = \{a, b\}$, um subconjunto de atributos de Q . Nesse caso, os conjuntos elementares do espaço aproximado $A = (U, \tilde{P})$ são: $E_1 = \{x_1, x_8, x_{10}\}$, $E_2 = \{x_2, x_4, x_9\}$, $E_3 = \{x_3\}$ e

$E_4 = \{x_5, x_6, x_7\}$. Seja $X = \{x_2, x_3\}$. Tem-se que:

$$\begin{aligned} A_{A-inf}(X) &= \{x_3\} \\ A_{A-sup}(X) &= \{x_2, x_3, x_4, x_9\} \end{aligned}$$

$$\alpha_P(X) = \frac{10 - |4 - 1|}{10} = \frac{7}{10} = 0.7$$

$$\begin{aligned} \omega_{A-inf}(X) &= 1 \\ \omega_{A-sup}(X) &= 4 \\ \omega_A(X) &= 0.25 \end{aligned}$$

$$\begin{aligned} \gamma_{A-inf}(X) &= 0.1 \\ \gamma_{A-sup}(X) &= 0.4 \end{aligned}$$

$$\mu_X^A(x_3) = 1$$

3.3.2 Tabelas de Decisão

No contexto da TCA o interesse recai, principalmente, sobre tabelas de decisão, um tipo particular de SRC. Uma *tabela de decisão* é um SRC onde os atributos de Q são divididos em condições e decisões. Tem-se então $Q = C \cup D$, onde C é o conjunto das condições e D o conjunto das decisões. Como geralmente o conjunto D é unitário, uma tabela de decisão é descrita por $S = (U, C \cup \{\delta\}, V, \rho)$, onde U , V e ρ são tais como num SRC, C é o conjunto de condições e δ é o atributo de decisão. Por $Class_S(\delta)$ entende-se a classificação de S , i.e., a família de conjuntos elementares do espaço aproximado induzido por $\{\delta\}$.

Exemplo 5 *Seja S o SRC do Exemplo 2. Seja d o atributo de decisão em S , i.e., $\delta = d$. S é uma tabela de decisão, onde $C = \{a, b, c\}$. Os conjuntos elementares do espaço aproximado induzido por C são $\{x_1\}$, $\{x_2, x_9\}$, $\{x_3\}$, $\{x_4\}$, $\{x_5, x_6\}$, $\{x_7\}$ e $\{x_8, x_{10}\}$. Por sua vez, a classificação de S é dada por $Class_S(\delta) = \{\{x_1, x_2, x_4, x_5, x_7, x_8\}, \{x_3, x_6, x_9, x_{10}\}\}$.*

Dada uma tabela de decisão $S = (U, C \cup \{\delta\}, V, \rho)$, é importante verificar o quão bem a família de conjuntos elementares induzidos pelas condições $P \subseteq C$ espelha a família de conjuntos elementares induzidos por $\{\delta\}$. Para isso, considerando o espaço aproximado induzido por P , são definidas:

- região positiva de δ induzida por P ,

$$pos(P, \delta) = \bigcup_{X \in Class_S(\delta)} A_{P-inf}(X);$$

- grau de dependência de δ com relação a P ,

$$\kappa(P, \delta) = \frac{|pos(P, \delta)|}{|U|};$$

- fator de significância de um atributo $a \in P$, com relação à dependência existente entre δ e P ,

$$FS(a, P, \delta) = \frac{(\kappa(P, \delta) - (\kappa(P - \{a\}, \delta)))}{\kappa(P, \delta)},$$

se $\kappa(P, \delta) > 0$.

Exemplo 6 Seja S o SRC definido no Exemplo 2 e seja $P = \{a, b, c\}$. Os conjuntos elementares do espaço aproximado induzido por P são: $E_1 = \{x_1\}$, $E_2 = \{x_2, x_9\}$, $E_3 = \{x_3\}$, $E_4 = \{x_4\}$, $E_5 = \{x_5, x_6\}$, $E_6 = \{x_7\}$ e $E_7 = \{x_6, x_{10}\}$. Por sua vez, $Class_S(d) = \{\{x_1, x_2, x_4, x_5, x_7, x_8\}, \{x_3, x_6, x_9, x_{10}\}\}$. Com isso, é possível verificar os seguintes resultados:

$$\begin{aligned} pos(P, d) &= \bigcup_{X \in Class_S(d)} A_{P-inf}(X) = \\ &= A_{P-inf}(\{x_1, x_2, x_4, x_5, x_7, x_8\}) \cup A_{P-inf}(\{x_3, x_6, x_9, x_{10}\}) = \\ &= \{x_1, x_4, x_7\} \cup \{x_3\} = \{x_1, x_3, x_4, x_7\} \\ \kappa(P, d) &= \frac{|pos(P, d)|}{|U|} = \frac{4}{10} = 0.4. \end{aligned}$$

Para o cálculo de:

$$FS(a, P, d) = \frac{(\kappa(P, d) - (\kappa(P - \{a\}, d))}{\kappa(P, d)},$$

e de :

$$FS(b, P, d) = \frac{(\kappa(P, d) - (\kappa(P - \{b\}, d))}{\kappa(P, d)},$$

bem como o de :

$$FS(c, P, d) = \frac{(\kappa(P, d) - (\kappa(P - \{c\}, d))}{\kappa(P, d)},$$

é necessário calcular antes o valor de $\kappa(\{b, c\}, d)$, $\kappa(\{a, c\}, d)$ e $\kappa(\{a, b\}, d)$, respectivamente, para este exemplo, considerando-se que $P_1 = \{b, c\}$, $P_2 = \{a, c\}$ e $P_3 = \{a, b\}$, tem-se que as famílias dos conjuntos elementares dos espaços aproximados induzidos por P_1 , P_2 e P_3 são dadas, respectivamente, por $\{\{x_1\}, \{x_2, x_8, x_9, x_{10}\}, \{x_5, x_6\}, \{x_7\}, \{x_3\}, \{x_4\}\}$, $\{\{x_1, x_7\}, \{x_5, x_6, x_8, x_{10}\}, \{x_3, x_4\}, \{x_2, x_9\}\}$ e $\{\{x_1, x_8, x_{10}\}, \{x_2, x_4, x_9\}, \{x_3\}, \{x_5, x_6, x_7\}\}$. Donde:

$$\begin{aligned} \text{pos}(P_1, d) &= \{x_1, x_3, x_4, x_7\} \text{ e} \\ \kappa(P_1, d) &= \frac{4}{10} = 0.4, \end{aligned}$$

$$\begin{aligned} \text{pos}(P_2, d) &= \{x_1, x_7\} \text{ e} \\ \kappa(P_2, d) &= \frac{2}{10} = 0.2, \end{aligned}$$

$$\begin{aligned} \text{pos}(P_3, d) &= \{x_3\} \text{ e} \\ \kappa(P_3, d) &= \frac{1}{10} = 0.1. \end{aligned}$$

Portanto,

$$FS(a, P, d) = \frac{(\kappa(P, d) - (\kappa(\{b, c\}, d)))}{\kappa(P, d)} = \frac{\frac{4}{10} - \frac{4}{10}}{\frac{4}{10}} = 0,$$

$$FS(b, P, d) = \frac{(\kappa(P, d) - (\kappa(\{a, c\}, d)))}{\kappa(P, d)} = \frac{\frac{4}{10} - \frac{2}{10}}{\frac{4}{10}} = 0.5,$$

$$FS(c, P, d) = \frac{(\kappa(P, d) - (\kappa(\{a, b\}, d)))}{\kappa(P, d)} = \frac{\frac{4}{10} - \frac{1}{10}}{\frac{4}{10}} = \frac{3}{4} = 0.75.$$

Diz-se, ainda com respeito a uma tabela de decisão $S = (U, C \cup \{\delta\}, V, \rho)$ e $P \subseteq C$ que P é independente com relação à dependência existente entre δ e P se, para todo subconjunto próprio $R \subset P$, for verdade que $\text{pos}(P, \delta) \neq \text{pos}(R, \delta)$, i.e., $\kappa(P, \delta) \neq \kappa(R, \delta)$. Caso haja algum $R \subset P$ tal que $\text{pos}(P, \delta) = \text{pos}(R, \delta)$, i.e., $\kappa(P, \delta) = \kappa(R, \delta)$, então P é dito ser dependente com relação à dependência existente entre δ e P .

3.3.3 Redução do Conjunto Inicial de Atributos

Um problema crucial no contexto de Sistemas de Representação de Conhecimento é o de encontrar subconjuntos do conjunto original de atributos com o mesmo poder discriminatório deste. A obtenção desses subconjuntos pode auxiliar tanto na redução do custo computacional de tarefas que utilizem SRCs, como até mesmo

custo temporal ou financeiro (caso atributos de difícil obtenção ou alto custo possam ser eliminados). No contexto da TCA, a obtenção de redutos está intimamente ligada à análise de dependência entre atributos.

Um conjunto $R \subset P$ é dito ser um *reduto* de P com relação à dependência existente entre δ e P se for independente com relação à dependência existente entre δ e R , e $pos(P, \delta) = pos(R, \delta)$, i.e., $\kappa(P, \delta) = \kappa(R, \delta)$.

Exemplo 7 Seja S o SRC do Exemplo 2, onde d é o atributo de decisão em S . Como já visto, S é uma tabela de decisão, onde a classe de condições é dada por $C = \{a, b, c\}$ e a classe de decisão é dada por $\delta = d$. A família $Class_S(d)$ foi determinada no Exemplo 5. Tem-se os seguintes resultados em S :

$$\kappa(C, d) = \frac{|pos(C, d)|}{|U|} = \frac{|4|}{|10|} = 0.4$$

$$\kappa(\{a, b\}, d) = \frac{|1|}{|10|} = 0.1$$

$$\kappa(\{a, c\}, d) = \frac{|2|}{|10|} = 0.2$$

$$\kappa(\{b, c\}, d) = \frac{|4|}{|10|} = 0.4$$

Neste caso C é dependente, e $\{b, c\}$ é o único subconjunto de C a possuir um reduto de C , pois $\{b, c\}$ é o único subconjunto com o mesmo grau de dependência de C . Tem-se ainda:

$$\kappa(\{b\}, d) = \frac{|0|}{|10|} = 0$$

$$\kappa(\{c\}, d) = \frac{|2|}{|10|} = 0.2$$

Esses valores são menores que $\kappa(C, d)$, portanto o conjunto $R = \{b, c\}$ é o único reduto do conjunto de condições com relação a dependência entre δ e C .

Capítulo 4

Algoritmo de Aprendizado Indutivo de Máquina Baseado em Índice Discriminante de Atributos

4.1 Introdução

No capítulo anterior, foram apresentados conceitos da Teoria de Conjuntos Aproximados que podem ser utilizados para desenvolvimento de algoritmos de aprendizado de máquina. Esses algoritmos tem como objetivo deduzir regras de uma dada base de dados. As regras de decisão geradas a partir destes algoritmos são da forma:

$$(a = v) \Rightarrow (b = w)$$

lida como: "se o atributo a possui o valor v , então o atributo b possui valor w ". Formalmente, uma regra de decisão gerada por esses algoritmos tem a forma:

$$(a_i = v_{a_i}) \& \dots \& (a_n = v_{a_n}) \Rightarrow (b = v_b),$$

onde o símbolo "&" corresponde ao termo "e" lógico.

O objetivo deste capítulo é apresentar o algoritmo de aprendizado RS1+, apresentado em [Uchôa (1998)], utilizado como base dos nossos estudos.

4.2 Aprendizado usando RS1+

Esse algoritmo é baseado no índice discriminante de atributos, descrito no capítulo anterior. Ele utiliza um conjunto B , onde são inseridos os atributos de maior índice discriminante. Regras consistentes são geradas quando houver alguma aproximação inferior nesse conjunto, e são geradas regras inconsistentes quando o índice discriminante não aumentar com a inserção de qualquer atributo de condição ou não existirem mais atributos de condição para serem inseridos. O algoritmo utiliza uma variável de controle, *oldDiscrim*, para checar se o índice discriminante aumentou.

Dada uma tabela de decisão $S = (U, C \cup \{\delta\}, V, \rho)$, o algoritmo RS1+ executa os seguintes passos:

1. Calcular $Class_S(\delta) = \{X_1, \dots, X_n\}$, a família de conjuntos elementares do espaço aproximado induzido por $\{\delta\}$.
2. Fazer $j = 1$
3. Fazer $U' = U, C' = C, B = \emptyset, X = X_j, oldDiscrim = -1$ e $S' = \{U', C \cup \{\delta\}, V, \rho\}$.
4. Calcular o conjunto de índices discriminantes $\{\alpha_{B'}(X) | B' = B \cup \{c\}, \forall c \in C'\}$ em S' .
5. Se o maior valor de $\alpha_{B'}(X)$ for menor que *oldDiscrim*, ir para o passo 15.
6. Selecionar o conjunto de atributos $B' = B \cup \{c\}$ com maior valor $\alpha_{B'}(X)$ e fazer *oldDiscrim* = maior valor de $\alpha_{B'}(X)$.
7. Fazer $B = B'$.
8. Se $A_{B-inf}(X) = \emptyset$, ir para o passo 11.
9. Identificar os conjuntos elementares $E = \{E_1, \dots, E_r\}$ do espaço aproximado induzido por B que estão contidos em $A_{B-inf}(X)$.
10. Para cada elemento $E_k \in E$, gerar uma regra de decisão determinística (consistente). Considerando que B possui m atributos, então cada regra tem a forma $(a_i = v_{a_i}) \& \dots \& (a_m = v_{a_m}) \Rightarrow (b = v_b)$, onde $a_1, \dots, a_m \in B$. Cada elemento $(a_k = v_{a_k})$ do antecedente é construído da seguinte forma: a_k recebe o nome do k -ésimo atributo de B e v_{a_k} recebe o valor atribuído a E_k por esse atributo.
11. Fazer $U' = U' - ((U' - A_{B-sup}(X)) \cup A_{B-inf}(X))$, $X = X - A_{B-inf}(X)$ e $S' = \{U', C \cup \{\delta\}, V, \rho\}$.

12. Se $U' = \emptyset$ ir para o passo 17.
13. Fazer $C' = C' - B$.
14. Se $C' \neq \emptyset$, voltar ao passo 4.
15. Identificar todos conjuntos elementares $E = \{E_1, \dots, E_r\}$ do espaço aproximado induzido por B .
16. Para cada elemento $E_k \in E$, gerar uma regra de decisão não-determinística (inconsistente) de forma semelhante à descrita no passo 9.
17. Fazer $j = j + 1$.
18. Se $j \leq n$ voltar ao passo 3.
19. Devolver todas as regras encontradas nos passos 10 e 16.

Exemplo 8 Considere a tabela de decisão definida na Tabela 4.1. Dado $\delta = e$, aplicando-se os passos do algoritmo, onde cada numeração indica o passo sendo executado, tem-se:

Tabela 4.1: SRC onde $U = \{x_1, x_2, \dots, x_{11}, x_{12}\}$ e $Q = \{a, b, c, d, e\}$

U	a	b	c	d	e
x_1	1	2	1	1	1
x_2	1	2	2	3	1
x_3	1	2	3	2	1
x_4	1	1	2	2	2
x_5	2	1	2	2	2
x_6	2	1	2	2	2
x_7	2	3	1	2	1
x_8	2	1	2	5	2
x_9	3	3	1	2	3
x_{10}	3	3	3	1	3
x_{11}	3	1	2	5	2
x_{12}	3	1	2	5	2

1. A Tabela 4.2 mostra o valor de $Class_S(e)$.
2. $j = 1$.

Tabela 4.2: $Class_S(e)$

X	Conjunto
X_1	$\{x_1, x_2, x_3, x_7\}$
X_2	$\{x_4, x_5, x_6, x_8, x_{11}, x_{12}\}$
X_3	$\{x_9, x_{10}\}$

3. $U' = \{x_1, \dots, x_{12}\}$
 $C' = \{a, b, c, d\}$
 $B = \emptyset$
 $X = X_1 = \{x_1, x_2, x_3, x_7\}$
 $oldDiscrim = -1$
 $S' = (U', C \cup \{e\}, V, \rho)$
4. A Tabela 4.3 mostra os valores dos índices discriminantes de $\{\alpha_{B'}(X) | B' = B \cup \{q\}, \forall q \in C'\}$ em S' .

Tabela 4.3: Cálculo de $\alpha_{B'}(X)$, com $X = \{x_1, x_2, x_3, x_7\}$ e $B = \emptyset$

$q \in C'$	$B \cup \{q\}$	$\alpha_{B'}\{X\}$
a	{a}	$\frac{12 - 8 - 0 }{12} = 0,33$
b	{b}	$\frac{12 - 6 - 3 }{12} = 0,75$
c	{c}	$\frac{12 - 12 - 0 }{12} = 0$
d	{d}	$\frac{12 - 12 - 0 }{12} = 0$

5. $oldDiscrim (-1) < 0,75$
6. Como $B' = B \cup \{b\}$ é o conjunto de atributos com maior índice discriminante, $B' = \emptyset \cup \{b\} = \{b\}$. $oldDiscrim = 0,75$
7. $B = \{b\}$.
8. $A_{B-inf}(X) = \{x_1, x_2, x_3\} \neq \emptyset$.
9. A família formada pelos conjuntos elementares contidos em $A_{B-inf}(X)$ é $\{E_1\}$, onde $E_1 = \{x_1, x_2, x_3\}$.
10. Gerar regra ($\mathbf{b} = 2$) \Rightarrow ($\mathbf{e} = 1$).
11. $U' = U' - ((U' - A_{B-sup}(X)) \cup A_{B-inf}(X))$
 $= \{x_1, \dots, x_{12}\} - ((\{x_1, \dots, x_{12}\} - \{x_1, x_2, x_3, x_7, x_9, x_{10}\}) \cup \{x_1, x_2, x_3\})$
 $= \{x_7, x_9, x_{10}\},$

$$X = X - \{x_1, x_2, x_3\} = \{x_7\} \text{ e}$$

$$S' = (U', C \cup \{e\}, V, \rho)$$

12. $U \neq \emptyset$.
13. $C' = C' - B = \{a, b, c, d\} - \{b\} = \{a, c, d\}$
14. $C' \neq \emptyset$, voltar ao passo 4.
4. A Tabela 4.4 mostra os valores dos índices discriminantes de $\{\alpha_{B'}(X) | B' = B \cup \{q\}, \forall q \in C'\}$ em S' .

Tabela 4.4: Cálculo de $\alpha_{B'}(X)$, com $X = \{x_7\}$ e $B = \{b\}$

$q \in C'$	$B \cup \{q\}$	$\alpha_{B'}\{X\}$
a	{b, a}	$\frac{3- 1-1 }{3} = 1$
c	{b, c}	$\frac{3- 2-0 }{3} = 0,33$
d	{b, d}	$\frac{3- 2-0 }{3} = 0,33$

5. $oldDiscrim(0, 75) < 1$
6. $B' = B \cup \{a\} = \{a, b\}$. $oldDiscrim = 1$
7. $B = \{a, b\}$
8. $A_{B-inf}(X) = \{x_7\} \neq \emptyset$
9. $E = \{E_1\}$, onde $E_1 = \{x_7\}$
10. Gerar regra ($\mathbf{b} = 3$) & ($\mathbf{a} = 1$) \Rightarrow ($\mathbf{e} = 1$)
11. $U' = U' - ((U' - A_{B-sup}(X)) \cup A_{B-inf}(X))$
 $= \{x_7\} - ((\{x_7\} - \{x_7\}) \cup \{x_7\})$
 $= \emptyset,$
 $X = X - \{x_7\}$
 $= \emptyset \text{ e}$
 $S' = (U', C \cup \{e\}, V, \rho)$
12. $U = \emptyset$, ir para o passo 17.
17. $j = 2$.
18. $j \leq 3$. Voltar para o passo 3.

$$\begin{aligned}
3. \quad U' &= \{x_1, \dots, x_{12}\} \\
C' &= \{a, b, c, d\} \\
B &= \emptyset \\
X = X_2 &= \{x_4, x_5, x_6, x_8, x_{11}, x_{12}\} \\
oldDiscrim &= -1 \\
S' &= (U', C \cup \{e\}, V, \rho)
\end{aligned}$$

4. A Tabela 4.5 mostra os valores dos índices discriminantes de $\{\alpha_{B'}(X) | B' = B \cup \{q\}, \forall q \in C'\}$ em S' .

Tabela 4.5: Cálculo de $\alpha_{B'}(X)$, com $X = \{x_4, x_5, x_6, x_8, x_{11}, x_{12}\}$ e $B = \emptyset$

$q \in C'$	$B \cup \{q\}$	$\alpha_{B'}\{X\}$
a	{a}	$\frac{12- 12-0 }{12} = 0$
b	{b}	$\frac{12- 6-6 }{12} = 1$
c	{c}	$\frac{12- 7-0 }{12} = 0,417$
d	{d}	$\frac{12- 9-3 }{12} = 0,5$

5. $oldDiscrim(-1) < 1$

6. $B' = B \cup \{b\} = \{b\}$.

7. $B = \{b\}$.

8. $A_{B-inf}(X) = \{x_4, x_5, x_6, x_8, x_{11}, x_{12}\} \neq \emptyset$.

9. $E = \{E_1\}$, onde $E_1 = \{x_4, x_5, x_6, x_8, x_{11}, x_{12}\}$

10. Gerar regra (**b = 1**) \Rightarrow (**e = 2**)

$$\begin{aligned}
11. \quad U' &= U' - ((U' - A_{B-sup}(X)) \cup A_{B-inf}(X)) \\
&= \{x_1, \dots, x_{12}\} - ((\{x_1, \dots, x_{12}\} - \{x_4, x_5, x_6, x_8, x_{11}, x_{12}\}) \cup \{x_4, x_5, x_6, x_8, x_{11}, x_{12}\}) \\
&= \emptyset, \\
X &= X - \{x_4, x_5, x_6, x_8, x_{11}, x_{12}\} \\
&= \emptyset \text{ e} \\
S' &= (U', C \cup \{e\}, V, \rho)
\end{aligned}$$

12. $U = \emptyset$, ir para o passo 17.

17. $j = 3$

18. $j \leq 3$, voltar para o passo 3.

3. $U' = \{x_1, \dots, x_{12}\}$
 $C' = \{a, b, c, d\}$
 $B = \emptyset$
 $X = X_3 = \{x_9, x_{10}\}$
 $oldDiscrim = -1$
 $S' = (U', C \cup \{e\}, V, \rho)$
4. A Tabela 4.6 mostra os valores dos índices discriminantes de $\{\alpha_{B'}(X) | B' = B \cup \{q\}, \forall q \in C'\}$ em S' .

Tabela 4.6: Cálculo de $\alpha_{B'}(X)$, com $X = \{x_9, x_{10}\}$ e $B = \emptyset$

$q \in C'$	$B \cup \{q\}$	$\alpha_{B'}\{X\}$
a	{a}	$\frac{12- 4-0 }{12} = 0,66$
b	{b}	$\frac{12- 3-0 }{12} = 0,75$
c	{c}	$\frac{12- 5-0 }{12} = 0,417$
d	{d}	$\frac{12- 8-0 }{12} = 0,33$

5. $oldDiscrim(-1) < 0,75$
6. $B' = B \cup \{b\} = \{b\}$ e $oldDiscrim = 0,75$.
7. $B = \{b\}$.
8. $A_{B-inf}(X) = \emptyset$. Ir para o passo 11.
11. $U' = U' - ((U' - A_{B-sup}(X)) \cup A_{B-inf}(X))$
 $= \{x_1, \dots, x_{12}\} - ((\{x_1, \dots, x_{12}\} - \{x_7, x_9, x_{10}\}) \cup \emptyset)$
 $= \{x_7, x_9, x_{10}\},$
 $X = X - \emptyset$
 $= \{x_9, x_{10}\}$ e
 $S' = (U', C \cup \{e\}, V, \rho)$
12. $U' \neq \emptyset$.
13. $C' = C' - B = \{a, b, c, d\} - \{b\} = \{a, c, d\}$.
14. $C' \neq \emptyset$. Ir para o passo 4.
4. A Tabela 4.7 mostra os valores dos índices discriminantes de $\{\alpha_{B'}(X) | B' = B \cup \{q\}, \forall q \in C'\}$ em S' .
5. $oldDiscrim(0,75) < 1$.
6. $B' = B \cup \{a\} = \{a, b\}$ e $oldDiscrim = 1$.

Tabela 4.7: Cálculo de $\alpha_{B'}(X)$, com $X = \{x_9, x_{10}\}$ e $B = \{b\}$

$q \in C'$	$B \cup \{q\}$	$\alpha_{B'}\{X\}$
a	{b, a}	$\frac{3- 2-2 }{3} = 1$
c	{b, c}	$\frac{3- 3-0 }{3} = 0$
d	{b, d}	$\frac{3- 2-0 }{3} = 0,33$

7. $B = \{a, b\}$
8. $A_{B-inf}(X) \neq \emptyset$
9. $E = \{E_1\}$, onde $E_1 = \{x_9, x_{10}\}$
10. Gerar a regra ($\mathbf{b} = \mathbf{3}$)&($\mathbf{a} = \mathbf{3}$) \Rightarrow ($\mathbf{e} = \mathbf{3}$).
11. $U' = U' - ((U' - A_{B-sup}(X)) \cup A_{B-inf}(X))$
 $= \{x_7, x_9, x_{10}\} - ((\{x_7, x_9, x_{10}\} - \{x_9, x_{10}\}) \cup \{x_9, x_{10}\})$
 $= \emptyset,$
 $X = X - \{x_9, x_{10}\}$
 $= \emptyset$ e
 $S' = (U', C \cup \{e\}, V, \rho)$
12. $U' = \emptyset$. Ir para o passo 17.
17. $j = 4$.
18. $j > 3$.
19. As regras geradas são, portanto:
 - ($\mathbf{b} = \mathbf{2}$) \Rightarrow ($\mathbf{e} = \mathbf{1}$)
 - ($\mathbf{b} = \mathbf{3}$)&($\mathbf{a} = \mathbf{2}$) \Rightarrow ($\mathbf{e} = \mathbf{1}$)
 - ($\mathbf{b} = \mathbf{1}$) \Rightarrow ($\mathbf{e} = \mathbf{2}$)
 - ($\mathbf{b} = \mathbf{3}$)&($\mathbf{a} = \mathbf{3}$) \Rightarrow ($\mathbf{e} = \mathbf{3}$)

4.3 Comentários Finais

Percebe-se facilmente que o algoritmo RS1+ possui as seguintes características:

- Permite ordenar os elementos do antecedente de uma regra de decisão descoberta de acordo com o valor do índice discriminante. Isso significa que em uma regra do tipo ($b = 0$)&($a = 1$) \Rightarrow ($e = 1$), por exemplo, o valor de b será checado primeiro, o que é desejável, pois ele é mais significativo para a determinação de ($e = 1$).

- Possibilita uma avaliação das regras geradas, através do fator de credibilidade, utilizando a função de pertinência aproximada.
- Pode-se facilmente separar regras consistentes das inconsistentes (as consistentes têm fator de credibilidade 1).

Capítulo 5

Descoberta de Conhecimento Utilizando Informação Agregada

5.1 Introdução

O algoritmo RS1+ permite gerar regras supondo que todos os atributos e valores associados a esses atributos tenham o mesmo peso (ou custo). Mas pode ser que alguns atributos e(ou) valores associados não sejam fáceis de serem obtidos, tendo preferência por gerar regras com outros atributos, caso seja possível. O objetivo deste capítulo é propor modificações no algoritmo RS1+, de forma que ele manifeste preferência pelos atributos e valores de menor custo.

5.2 Alterações no Algoritmo RS1+

5.2.1 Normalização de Custos

Dado T um vetor de custos, a normalização dos seus valores é feita da seguinte forma:

$$T[i] = \frac{k}{T[i]},$$

onde k é o valor do menor custo.

Esta normalização garante que o menor custo tenha valor 1, e, quanto maior o custo de um atributo em relação aos outros, mais próximo de 0 será o seu valor. Para o uso deste cálculo estão proibidos custos de atributos com valor 0.

5.2.2 Índice Discriminante de Atributos

A escolha dos atributos para as regras está diretamente ligada aos valores dos seus índices discriminantes (α). Para restringir um atributo de ser usado, deve-se diminuir o seu poder discriminatório. Mas ele não deve ser totalmente eliminado pois, mesmo que seja mais caro, talvez ainda assim compense usá-lo. Daí surge uma nova fórmula para o cálculo do índice discriminante.

Seja o SRC $S = (U, C \cup \{\delta\}, V, \rho)$, um conjunto de custos T e uma função de custos σ . U , V e ρ são tais como num SRC, C é o conjunto de condições, δ é o atributo de decisão, T é um conjunto de custos normalizados e $\sigma: C \rightarrow T$ é uma função de custo, tal que $\sigma(x) \in T$, se $x \in C$. Seja também o espaço aproximado $A = (U, R)$ e $X \subseteq U$, o novo valor do índice discriminante será:

$$\alpha_R(X) = \frac{|U| - |A_{sup}(X) - A_{inf}(X)|}{|U|} \cdot \frac{P(R)}{|R|}, \text{ onde}$$

$$P(R) = \sum_{a \in R} \sigma(a).$$

No RS1+, apenas um atributo é inserido por vez no conjunto de decisão. Em cada iteração, a quantidade de elementos no conjunto R é constante, e o valor de $P(R)$ também. Portanto esses valores podem ser substituídos pelo custo do novo atributo. Então o índice discriminante ficará da seguinte forma:

$$\alpha_{R \cup \{c\}}(X) = \frac{|U| - |A_{sup}(X) - A_{inf}(X)|}{|U|} \cdot \sigma(c)$$

Utilizando essa nova fórmula, pode-se concluir que:

- pelo fato de os custos estarem normalizados, α continua pertencendo ao intervalo $[0, 1]$. Será 0 quando o α anterior (RS1+) for 0, e será 1 quando o anterior for 1 e $\sigma(c)$ tiver valor 1.
- caso os custos sejam todos iguais, os resultados do cálculo do $\alpha_R(X)$ serão os mesmos do RS1+, pois todos os custos normalizados terão valor 1, e, portanto, esses valores não vão interferir nas multiplicações.
- um atributo com custo n vezes maior que um outro terá que ter índice discriminante n vezes maior para ser escolhido.

5.2.3 Escolha do Melhor Atributo

O fato de multiplicar o valor do atributo pelo seu custo diminui o índice discriminante dos atributos de maior custo. Há uma tendência de os melhores atributos possuírem custos mais elevados. Portanto, a multiplicação faz com que bons atributos sejam ignorados à medida que os seus custos aumentem.

Suponha um exame médico. Se ele for barato, poderá ser utilizado frequentemente. Caso o seu custo aumente, haverá uma rejeição natural por este exame, e as pessoas passarão a procurar por outros exames. Caso existam outros que possam substituí-lo totalmente, esse não será mais usado. Mas, se o exame não puder ser substituído, este ainda será usado, mas apenas em última instância, quando não houverem mais opções.

Essa é a estratégia proposta. Se um atributo aumentar de custo, outros serão usados no seu lugar, caso o índice discriminante aumente (ele será rejeitado caso outros atributos possam substituí-lo normalmente). Caso contrário, este atributo será usado.

Para fazer isto, inicialmente é escolhido o melhor atributo utilizando a nova fórmula para o índice discriminante. Caso seu valor aumente em relação ao α anterior, continuamos o algoritmo normalmente. Caso contrário, os índices são recalculados, mas sem utilizar os custos, ou seja, do mesmo modo que no RS1+, evitando assim regras inconsistentes. Dessa forma, os bons atributos de maior custo não são totalmente removidos, mas movidos para o lado mais à direita dos antecedentes das regras. Isso significa que eles ainda poderão ser usados, mas apenas em último caso.

5.2.4 Redutos

O cálculo de redutos, como citado no Capítulo 3, também pode ser utilizado para remover os atributos de custo elevado. Segundo [Uchôa (1998)], um reduto de um SRC diminui o tempo de execução do RS1 em relação ao mesmo SRC. Mas redutos não foram utilizados, pois o tempo de execução do algoritmo para gerar os redutos tem crescimento exponencial. Também o RS1 garante estar utilizando o melhor conjunto para definição dos valores de $Class(\delta)$, ao escolher o conjunto de atributos de maior índice discriminante. Também não se pode simplesmente remover um atributo de custo elevado, como explicado anteriormente.

5.2.5 Valores associados aos atributos

O novo algoritmo trata apenas custos associados aos atributos, mas não trata os custos dos valores associados aos atributos. Para esse novo caso, é proposto que a base de dados seja reformulada, criando-se novos atributos, antes de executar o algoritmo.

Para cada atributo a_i com n valores possíveis e custos associados a esses valores, deve-se eliminar esse atributo e gerar n novos atributos a_{ij} , $1 \leq j \leq n$, de forma que cada novo atributo seja booleano, demonstrando presença ou ausência do valor associado, ou seja,

$$\rho(x_k, a_i) = j \Rightarrow \rho(x_k, a_{ij}) = \mathbf{s}$$

$$\rho(x_k, a_i) \neq j \Rightarrow \rho(x_k, a_{ij}) = \mathbf{n}$$

O custo do novo atributo a_{ij} será então o custo do valor j associado a a_i .

O Exemplo 9 demonstra o funcionamento dessa modificação.

Exemplo 9 Dado o SRC da Tabela 5.1. Os atributos a , b e c têm pesos associados aos seus valores, descritos na Tabela 5.2. O novo SRC a ser utilizado como entrada do RS1+ encontra-se na Tabela 5.3, e os pesos dos seus atributos são os pesos dos custos dos valores na Tabela 5.2.

Tabela 5.1: SRC onde $U = \{x_1, x_2, x_3, x_4, x_5\}$ e $Q = \{a, b, c\}$

U	a	b	c
x_1	1	3	5
x_2	2	4	4
x_3	1	2	5
x_4	1	3	5
x_5	2	2	4

Tabela 5.2: Pesos dos valores associados aos atributos do SRC da Tabela 5.1

Atributo	a	a	b	b	b	c	c
Valor	1	2	2	3	4	4	5
Peso	1	3	2	3	4	3	3

Tabela 5.3: SRC da Tabela 5.1 com custos agregados aos valores associados aos atributos

U	a_1	a_2	b_2	b_3	b_4	c
x_1	s	n	n	s	n	5
x_2	n	s	n	n	s	4
x_3	s	n	s	n	n	5
x_4	s	n	n	s	n	5
x_5	n	s	s	n	n	4

Note que:

- quando os custos relacionados aos valores de um dado atributo forem os mesmos, não é necessário o uso dessa estratégia para aquele atributo;
- quando existirem dois valores com custos diferentes associados a um mesmo atributo, o atributo gerado do valor de maior custo poderá ser removido, como o atributo a_2 no Exemplo 9
- quando houverem 3 ou mais valores relacionados aos atributos, eles devem ser mantidos, pois os valores das colunas de cada novo atributo gerado serão diferentes (é o caso dos atributos b_1 , b_2 e b_3).

Devido à limitação de tempo para a entrega da monografia, esta modificação não foi implementada.

5.3 Testes Realizados

Nesta seção estão os resultados obtidos da implementação feita, modificando o algoritmo RS1+ do protótipo ILROS, uma implementação dos conceitos e algoritmos baseados na TCA. O ILROS está descrito em [Uchôa (1998)].

A Tabela 5.4 indica a nomenclatura utilizada na apresentação dos resultados.

5.3.1 Teste Inicial

Utilizando a tabela de decisão definida na Tabela 4.1 foram feitos vários testes, centrando-se no atributo b , que esteve presente em todas as regras obtidas pelo RS1+, no Exemplo 5. O seu custo foi aumentado gradativamente, e os resultados obtidos estão na Tabela 5.5.

Tabela 5.4: Siglas das tabelas de resultados

A_{MC}	Atributo de maior custo
A_{MP}	Atributo de maior presença nas regras
It	Iteração
M_{AR}	Média da quantidade de atributos por regra
M_{Cred}	Credibilidade média
M_P	Posição média do atributo nas regras em que ele aparece
M_{OP}	Moda da posição do atributo nas regras
Regras	Número de regras geradas
R_{MC}	Porcentagem de regras com o atributo de maior custo
R_{MP}	Porcentagem das regras com o atributo de maior presença
S_{AR}	Desvio padrão da quantidade de atributos por regra
S_{Cred}	Desvio padrão da credibilidade
S_P	Desvio padrão da posição média do atributo

Tabela 5.5: Resultados, utilizando custos, com o SRC da Tabela 4.1

Custo de b	Regras	Regras (b)	M_{Cred}	$M_P(b)$	$S_P(b)$	$M_{OP}(b)$
1	4	100,0%	100%	1,00	0,00	1
1,5	5	60,0%	100%	1,00	0,00	1
2,2	6	33,3%	100%	1,00	0,00	1
3	8	0,0%	100%	-	-	-

O novo algoritmo funciona da mesma forma que o RS1+ quando todos os atributos tem o mesmo custo. Percebe-se que, conforme o custo do atributo b aumenta, esse começa a ser rejeitado das regras, até ser totalmente eliminado. Isto ocorre porque existem atributos capazes de substituí-lo de forma a manter a credibilidade dos resultados. Percebe-se também que este atributo, mesmo aumentando o seu custo, sempre aparece na primeira posição das regras, o que significa que, individualmente, ele chega a ser até quase 3 vezes melhor que cada um dos outros.

5.3.2 Testes com Base de Dados Reais

Nesta seção são apresentados os resultados de testes com uma série de arquivos de dados de domínio público, disponíveis em [Merz & Murphy (1998)], um repositório de bases de dados, utilizado para análise de algoritmos de aprendizado de máquina. Foram utilizadas bases de dados com e sem custos.

Descrição dos Arquivos de Dados

1. **Car:** consiste de um conjunto de dados descrevendo a aceitação de vários modelos de carros. A descrição de cada instância é feita através de 6 atributos discretos, 4 com valores nominais e 2 com valores numéricos. Cada instância é classificada em uma das quatro possíveis classes: inaceitável, aceitável, bom e muito bom. O conjunto original, com 1728 elementos, não possui qualquer elemento com valor ausente, tendo sido usado integralmente.
2. **Hepatitis:** consiste de um conjunto de dados e exames de pessoas que tem (ou não) hepatite. São 19 atributos de decisão, sendo que dois deles foram removidos dos testes: o atributo *Class*, que indica se o paciente está vivo ou morto, por não ter um custo relacionado, e *Prottime*, por ter mais de 40% dos campos ausentes. Alguns atributos eram numéricos, e foram convertidos em valores discretos, conforme arredondamento proposto no próprio arquivo de descrição dos atributos. Essa base de dados possui originalmente 155 elementos, mas foram testados com 112 instâncias sem valores ausentes. Há uma pesquisa feita pelo Ontario Health Insurance Program com os custos dos atributos, e estes valores estão em Dólares Canadenses.
3. **Heart-Disease:** consiste de um conjunto de dados e exames de pessoas que sofrem (ou não) de doenças do coração. Esta é uma pesquisa feita por [Detrano (1988)]. São quatro valores de decisão, de 1 a 4. O valor 1 indica que o paciente não sofre de doença do coração, os outros indicam o "nível de sofrimento": quanto maior, pior. São 76 atributos, mas, segundo a descrição da base de dados, atualmente apenas 14 destes são utilizados em testes. São 303 instâncias, sem qualquer valor ausente. Alguns atributos eram contínuos, e foram arredondados para dificultar na geração das regras. Todos os atributos têm um custo em Dólares Canadenses.

Teste com *cars*

Não existem custos nesta base de dados, portanto foi utilizada a seguinte estratégia: primeiramente foram geradas regras sem custo. Então o custo do atributo de maior presença nas regras foi dobrado e foram geradas as regras novamente. A partir daí, a cada nova iteração, foram aumentados em 50% os custos dos atributos que têm custo diferente de um, e o custo do atributo de maior presença foi dobrado. Os

resultados obtidos estão na Tabela 5.6.

Tabela 5.6: Resultados com o exemplo *cars*

It	Regras	M _{AR}	S _{AR}	M _{Cred}	S _{Cred}	A _{MC}	R _{MC}	A _{MP}	R _{MP}
1	9	1,3	0,47	39,0%	0,25	-	-	<i>persons</i>	55,5%
2	9	1,3	0,47	39,0%	0,26	<i>persons</i>	0,0%	<i>safety</i>	66,6%
3	16	1,9	0,24	46,1%	0,28	<i>persons</i>	56,0%	<i>safety</i>	81,2%
4	10	1,5	0,49	38,5%	0,24	<i>safety</i>	20,0%	<i>buying</i> *	50,0%
5	15	1,8	0,40	45,3%	0,29	<i>safety</i>	80,0%	<i>safety</i> **	80,0%

* O atributo *buying* empatou com *persons*. *Buying* foi escolhido por ter custo 1.

** O atributo *safety* sempre foi escolhido como segundo atributo das regras, ou seja, esse atributo foi escolhido para não diminuir a credibilidade.

Como pode-se notar, mesmo aumentando o custo dos melhores atributos, algumas vezes a credibilidade chega até a aumentar. Isto ocorre pelo fato de combinações de atributos de menor índice gerar melhores índices que suas combinações individuais com os atributos de maior índice.

Percebe-se também que não é possível remover os atributos de melhor índice discriminante individual, e eles se revezam entre os mais usados. Isso significa que os outros atributos não são capazes de substituir à altura os atributos *persons*, *safety* e *buying*.

Teste com *hepatitis*

Foram feitos dois testes com esta base de dados, um sem utilizar os custos, e outro com eles. Os resultados obtidos estão na Tabela 5.7. Uma análise dos atributos nas regras geradas e os seus respectivos custos estão na Tabela 5.8.

Tabela 5.7: Resultados com o exemplo *hepatitis*

	Sem custo	Com custo
Regras	74	57
M _{AR}	3,65	3,96
S _{AR}	1,08	1,18
M _{Cred}	94,50%	87,70%
S _{Cred}	0,16%	0,23%

Tabela 5.8: Custos e porcentagem das regras geradas do exemplo *hepatitis*

Atributo	Custo	Sem Custo			Com Custo		
		Regras	M _P	S _P	Regras	M _P	S _P
age	\$1.00	86,48%	3,00	0,00	100,00%	1,00	0,00
malaise	\$1.00	52,70%	4,00	0,00	-	-	-
spiders	\$1.00	-	-	-	66,67%	4,00	0,00
ascites	\$1.00	-	-	-	96,49%	2,00	0,00
bilirubin	\$7.27	28,37%	5,00	0,00	85,96%	3,00	0,00
alk_phosphate	\$7.27	97,29%	2,00	0,00	-	-	-
sgot	\$7.27	-	-	-	47,36%	5,00	0,00
albumin	\$7.27	100,00%	1,00	0,00	-	-	-

Obs.: os atributos *sex*, *steroid*, *antviral*, *fatigue*, *anorexia*, *liver_big*, *liver_firm*, *spleen palpable* e *varices* (todos com custo \$1.0) não apareceram nas regras geradas.

Nota-se que os atributos *albumin* e *alk_phosphate*, originalmente os melhores atributos (1^a e 2^a posições, respectivamente), foram totalmente substituídos pelos atributos de menor custo *age* e *ascites*, e, para não haver uma maior perda na credibilidade, o algoritmo escolheu *bilirubin* na 3^a posição e *sgot*, na 5^a. Houve uma pequena queda na credibilidade (6,8%), mas é compensada pela melhor escolha dos atributos.

Teste com *heard-disease*

Foram feitos dois testes com esta base de dados, um sem utilizar os custos e outro com eles. Os resultados obtidos estão na Tabela 5.9, e a porcentagem de cada atributo nas regras geradas e os seus respectivos custos estão na Tabela 5.10.

Tabela 5.9: Resultados com o exemplo *heart-disease*

	Sem custo	Com custo
Regras	250	219
M _{AR}	2,71	4,00
S _{AR}	0,70	0,97
M _{Cred}	92,22%	89,95%
S _{Cred}	0,20%	0,21%

Neste exemplo, o atributo de maior presença inicial, *thal*, foi totalmente substituído, e o outro atributo de maior custo, *ca*, foi movido da 3^a para a 5^a posição.

Tabela 5.10: Custos e porcentagem das regras geradas do exemplo *heart-disease*

Atributo	Custo	Sem Custo				Com Custo			
		Regras	M _P	S _P	M _{OP}	Regras	M _P	S _P	M _{OP}
age	\$1.00	-	-	-	-	82,64%	2,23	0,70	2
cp	\$1.00	-	-	-	-	92,69%	2,58	0,98	3
trestbps	\$1.00	62,40%	1,70	0,71	2	100,00%	1,36	1,13	1
chol	\$7.27	98,80%	1,84	0,40	2	15,52%	1,00	0,00	1
oldpeak	\$87.30	8,40%	4,00	0,00	4	63,01%	4,13	0,74	4
thalach	\$102.90	-	-	-	-	2,73%	5,00	0,00	5
ca	\$102.90	5,20%	3,62	1,93	3	43,37%	4,63	1,01	5
thal	\$102.90	95,20%	1,50	1,17	1	-	-	-	-

Obs.: os atributos *sex* (\$1.00), *lbs* (\$5.20), *restecg* (\$15.50), *exang* (\$87.30) e *slope* (\$87.30) não apareceram nas regras geradas.

O atributo *oldpeak* teve um aumento de incidência nas regras, mas continuou nas últimas posições. Nota-se que os atributos de maior custo sempre foram usados nas últimas posições das regras, o que significa que eles foram utilizados apenas quando o índice discriminante não aumentou com os outros atributos (o que geraria regras inconsistentes), o que fez com que os custos fossem ignorados e, conseqüentemente, eles puderam ser escolhidos. Também houve uma pequena queda na credibilidade (2,27%), mas menor que no exemplo anterior.

5.3.3 Comentários

Pode-se perceber que, em todos os testes, o número de atributos por regra aumentou. Isso acontece devido ao fato que os melhores atributos não são escolhidos inicialmente, então o índice discriminante tem um crescimento mais lento que anteriormente. Isso significa que o custo total das regras pode aumentar, caso os atributos de maior custo sejam essenciais.

Um fator muito importante quanto ao uso das regras geradas é qual a ordem de teste dos atributos. Propõe-se que, de acordo com os resultados obtidos, essa ordem esteja diretamente ligada à posição média do atributo ou à sua moda, mas nunca ligada à quantidade de regras que o atributo aparece.

Capítulo 6

Conclusão

A Teoria de Conjuntos Aproximados é um formalismo bastante expressivo em aprendizado de máquina, e pode ser utilizada com sucesso para implementação de métodos de representação de conhecimento incerto.

O algoritmo RS1+, que utiliza os conceitos da TCA, baseia-se no índice discriminante dos atributos para geração de regras, e somente insere atributos no conjunto de decisão caso este índice aumente. Os testes realizados modificando o cálculo deste índice comprovaram as expectativas: com uma pequena perda de credibilidade, verificou-se que:

- os atributos de maior custo são removidos das regras, caso outros atributos possam substituí-los sem perda de credibilidade;
- esses atributos aparecerão mais a direita nos antecedentes das regras, quando o índice discriminante não aumentar sem os atributos. Isso faz com que um atributo de melhor custo será testado apenas quando nenhum outro atributo de menor custo puder substituí-lo.

Pretende-se, como trabalhos futuros:

- implementar a modificação proposta na seção 5.2.5, para utilizar informações agregadas aos valores associados aos atributos;
- implementar as modificações propostas utilizando um algoritmo que gere árvores de decisão, para facilitar a escolha dos atributos a serem testados, comprovando (ou não) a proposta de usar a ordem das modas ou da posição média do atributo no antecedente das regras;

- adicionar ao RS1+ o suporte a valores ausentes e contínuos, pois foi verificado que, em bases de dados testadas, alguns atributos continham muitos dados ausentes, e foram excluídos dos testes, e os valores contínuos foram arredondados.

Referências Bibliográficas

- [Bonissone (1991)] Bonissone, P. Plausible reasoning, In: Shapiro, S. C.; Eickroth, D. & Valassi, G. A. (Eds.) *Encyclopedia of Artificial Intelligence*. New York, John Wiley & Sons, 1991. p.854-863.
- [Boose (1989)] Boose, J. H. A knowledge acquisition: techniques and tools. *Knowledge Acquisition* (1): 3-37, 1989.
- [Clark & Niblett (1989)] Clark, P. & Niblett, T. The CN2 induction algorithm. *Machine Learning Journal*, 3(4):261-283, 1989.
- [Detrano (1988)] Detrano, R. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation.
- [Domingues & Uchôa (2001)] Domingues, M. A. & Uchôa, J. Q. *Implementação e Desenvolvimento de Técnicas de Descoberta de Conhecimento e Tratamento de Incerteza com Ênfase na Teoria de Conjuntos Aproximados*. Relatório de Iniciação Científica - Pibic/CNPq. Lavras, DCC-UFLA, 2001. 32p.
- [Fausett (1993)] Fausett, L. V. *Fundamental of newural networks: Architectures, Algorithms And Applications*. New York, Prentice Hall, 1993.
- [GNU (1999)] Free Software Foundation. *GNU's Not Unix! the GNU project and the Free Software Foundation (FSF)*. url: <http://www.gnu.org/>.
- [Klir & Yuan (1995)] Klir, J. G. & Yuan, B. *Fuzzi sets and fuzzy logic: theory and application*. New Jersey, Prentice Hall, 1995.
- [Langley (1996)] Langley, P. *Elements of machine learning*. San Francisco, Morgan Kaufmann, 1996.

- [McCulloch & Pitts (1943)] McCulloch, W. & Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematica Biophysics*, (5):115-137.
- [Merz & Murphy (1998)] Merz, C. J. & Murphy, P. M. *UCI repository of Machine Learning databases*. Irvine, University of California, Department of Information and Computer Science, 1998. url: <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
- [Michalski (1983)] Michalski, R. S. A theory and methodology of inductive learning. In: Michalski, R. S.; Carbonell, J. G. & Mitchell, T. M. (Eds.). *Machine learning: an Artificial Intelligence approach*. Palo Alto, Tioga, 1983. p.83-134.
- [Michalski & Tecuci (1993)] Michalski, R. S. & Tecuci, G. Multistrategy learning. Tutorial T15, IJCAI-1993, 1993.
- [Mitchell (1997)] Mitchell, T. M. Machine learning. New York, McGraw-Hill, 1997. [Ng & Abramsom (1990)] Ng, K.C. & Abramsom, B. Uncertainty management in expert systems. *IEEE Expert*, (April): 29-47, 1990.
- [Nicoletti (1994)] Nicoletti, M. C. *Ampliando os limites do aprendizado indutivo de máquina através das abordagens construtiva e relacional*. São Carlos, Universidade de São Paulo, 1994. (Tese de Doutorado).
- [Nicoletti & Uchôa (1997)] Nicoletti, M. C. & Uchôa, J. Q. *O uso de funcoes de pertinencia na caracterização dos principais conceitos da teoria de conjuntos aproximados*. Relatório Técnico do Departamento de Computação 005/97. São Carlos, DC-UFSCar, 1997. 26p.
- [Nicoletti; Uchôa & Baptistini (2001)] Nicoletti, M. C.; Uchôa, J. Q. & Baptistini M. T. Z. Rough relations properties. *Int. J. Appl. Math Comput. Sci.*, 11(3):621-635, 2001.
- [Pawlak (1982)] Pawlak, Z. Rough sets. *International Journal of Computer and Information Sciences*, 11(5):341-356, 1982.
- [Pawlak (1991)] Pawlak, Z. *Rough sets: theoretical aspects of reasoning about data*. London, Kluwer, 1991.

- [Pawlak (1994)] Pawlak, Z. Hard and soft sets. In: Ziarko, W. P. (Ed). *Rough sets, fuzzy sets and knowledge discovery*. London, Springer-Verlag, 1994. p.130-135.
- [Quinlan et alii(1986)] Quinlan, J. R; Compton, P. J; Horn, K. A. & Lazarus, L. *Inductive Knowledge Acquisition: a Case Study*. In: Proceedings of the 2nd Australian Conference on Applications of ES, Sydney, 1986, pp 183-203.
- [Shafer (1976)] Shafer, G. *A mathematical theory of evidence*. Princeton, Princeton University Press, 1976.
- [Uchôa et alii(1997)] Uchôa, J. Q.;Pantim, S. M. & Nicoletti, M. C. *Elementos da Teoria de Dempster-Shafer*. Relatório do Departamento de Computação 007/97. São Carlos, DC-UFSCar, 1997. 34p.
- [Uchôa (1998)] Uchôa, J. Q. *Representação e indução de conhecimento usando teoria de conjuntos aproximados*. São Carlos, UFSCar, 1998. 237p. (Dissertação de Mestrado).
- [Uchôa & Nicoletti (1998)] Uchôa, J. Q. & Nicoletti, M. C. *ILROS: um sistema de aprendizado indutivo de máquina baseado em conjuntos aproximados*.(Submetido para publicação como Relatório Técnico do Departamento de Computação da UFSCar).
- [Uchôa & Nicoletti (1999)] Uchôa, J. Q. & Nicoletti, M. C. ILROS: um sistema de aprendizado de máquina para domínios incompletos. In: *Anais do 4. SBAI - Simpósio Brasileiro de Automação Inteligente*. Escola Politécnica da USP, São Paulo (SP), 8 a 19 de Setembro de 1999. p. 314-319.
- [Shaw & Gentry (1990)] Shaw, M. J. & Gentry, J. A. Inductive learning for risk classification. *IEEE Expert*, (February):47-53, 1990.
- [Thrun et alii (1991)] Thrun, S. B.; Bala, J.; Bloedorn, E. et alii. *The monk's problems - a Performance comparison of different learning algorithms*. Technical Report CS-CMU-91-197, Carnegie Mellon University, December 1991.
- [Wong et alii (1986)] Wong, S. K. M.; Ziarko, W. & YE, R. L. Comparison of rough-set and statistical methods in inductive learning, *Internacional Journal of Man-Machine Studies*, (24):53-72, 1986.
- [Zadeh (1965)] Zadeh, L. A. Fuzzy sets. *Information and Control*, (8):338-353, 1965.