

KD – O conhecimento: formas de representação e explicitação

Adelmo L. Cechin*

Fernando S. Osório**

Resumo

A descoberta de conhecimentos (KD - Knowledge Discovery) a partir de bases de dados é um processo que está intimamente ligado as técnicas de aquisição, representação e explicitação dos conhecimentos contidos nestas bases de dados. Conhecimento, segundo [FAY 96, capítulo 1] é um padrão E pertencente a uma linguagem L caso este conhecimento ultrapasse um limite definido pelo usuário em termos de validade para novos casos, atualidade, utilidade e compreensibilidade. O conhecimento assim é uma expressão em uma linguagem e como tal recebe uma forte influência da forma ou sintaxe da linguagem usada.

Nosso trabalho visa abordar algumas das principais técnicas usadas em KD, abordando suas principais propriedades em relação à aquisição, representação e explicitação dos conhecimentos. O processo de aquisição de conhecimentos é usualmente o responsável pela escolha de um tipo particular de representação vinculado à validade, atualidade, utilidade e compreensibilidade.

As técnicas empregadas na aquisição automática de conhecimentos são os chamados métodos de aprendizado de máquina (*Machine Learning*) onde podemos destacar o uso de: algoritmos de indução de árvores de decisão (IDT – Induction of Decision Trees, e.g. C4.5) [NIK 97, QUI 92], técnicas de agrupamento de dados (*Clustering*, e.g. K-Nearest Neighbors) [MIT 97] e aprendizado baseado em redes neurais artificiais (ANN – Artificial Neural Networks, e.g. Back-Propagation MLP)[RUM 86, BRA 00]. A primeira forma de representação no contexto da *Machine Learning* é feita diretamente sob a forma de matriz de dados. O CBR (*Case-Based Reasoning*) é um exemplo desta forma de representação e manipulação de dados [KOL 93].

Uma questão na representação do conhecimento relaciona-se ao critério de Occam para encontrar o modelo subjacente nos dados, como sendo o mais “simples”, que ao mesmo tempo confunde-se com critérios utilizando a quantidade de informação em uma certa representação (e portanto em um certo formato). A questão “será o modelo mais simples o mais correto” não é aceito por todos autores. Um outro aspecto intimamente relacionado à este é o da compressão de dados, onde o conhecimento representado em uma

* cechin@exatas.unisinos.br

** osorio@exatas.unisinos.br

forma flui para outra forma mais “simples”. Critérios de simplicidade são porém relativos, o que torna critérios de compreensibilidade usados para definir conhecimento também relativos e dependentes de conhecimentos previamente existentes.

A utilização do critério *utilidade* parece ser o mais adequado para resolver estas questões, porém vincula a representação ao processamento desta e à sua utilização em um contexto, e assim torna a análise do problema mais complexa (por exemplo, pela quantidade de opções).

Nas árvores de decisões [QUI 92], selecionamos os atributos de entrada um após o outro, usando-os como elementos discriminantes em relação aos dados, ou seja, usando-os para dividir os dados em subconjuntos. Para a escolha dos atributos faz-se uso de conceitos como a entropia dos dados para obter a maior discriminação entre os grupos e a maior semelhança dentro dos dados em um grupo. Este tipo de critério tem como consequência a geração de árvores de pequena profundidade, no sentido também de simplificar o modelo obtido e assim atingir o critério da compreensibilidade do modelo (critério do modelo mais simples de Occam). A Figura 1(a) apresenta um esquema da representação dos conhecimentos usando-se árvores de decisão.

Em algoritmos de clusterização [MIT 97, OSO 99], os dados são agrupados de acordo com uma métrica que permite “medir” as suas semelhanças e diferenças. Há basicamente três abordagens para o cálculo de distâncias entre *clusters*: medidas baseadas em proximidade (distância definida pelos casos mais próximos entre dois *clusters*), em interconectividade (distância definida pela soma de distâncias individuais entre casos de *clusters* diferentes) e em casos representativos (distância entre casos representativos dos *clusters*) [KAR 99].

A Fig. 1(b) mostra um exemplo simples de clusterização de dados usando elipses como delimitadores da área de influência de um *cluster*, onde podemos notar definido o protótipo ideal, indicado pelo centro da elipse, e onde os raios indicam a região de influência (agrupamento) ao redor deste protótipo. Uma versão alternativa deste método de agrupamento de dados também pode ser obtida usando-se protótipos baseados em regiões retangulares. Algoritmos de clusterização mais complexos usam modelos livres onde o conhecimento ou os *clusters* podem ser representados através de qualquer forma. Tais sistemas representam o conhecimento não apenas por um conjunto de atributos representando um *cluster*, mas através de uma lista de conjunto de atributos. A representação das métricas pode ser realizada não somente pela definição de uma função de distância, mas também de forma independente, através de matrizes de distância (espaços não-métricos, não-lineares ou distorcidos).

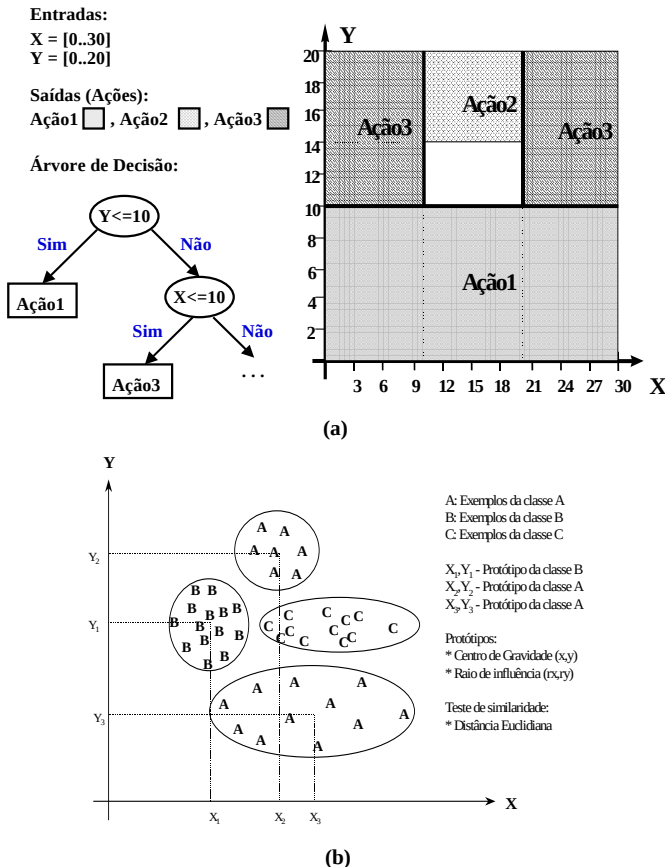


Figura 1. Representação de Conhecimentos – 2 atributos quantitativos:
 (a) Árvores de decisão (b) Clusters

Sistemas baseados em casos (*Case Based Reasoning* [KOL 93]) realizam uma representação do conhecimento através de protótipos ou casos representativos, podendo-se armazenar todos os casos ou apenas os casos mais representativos. O processamento desta informação também requer a definição de métricas.

As redes neurais artificiais do tipo MLP (*Multi-Layer Perceptron*) [RUM 86, NIK 97, MIT 97] permitem a discriminação dos dados através de uma combinação da contribuição dos seus atributos de entrada, o que equivale a determinar retas (ou planos n-dimensionais) de separação dos dados. Estas retas podem ser posicionadas e combinadas livremente de forma a dividir o espaço de entradas, conforme apresentamos na Figura 2(a). Portanto, usando as redes neurais, podemos delimitar áreas bem específicas junto ao espaço de entradas, como mostra a Figura 2(b).

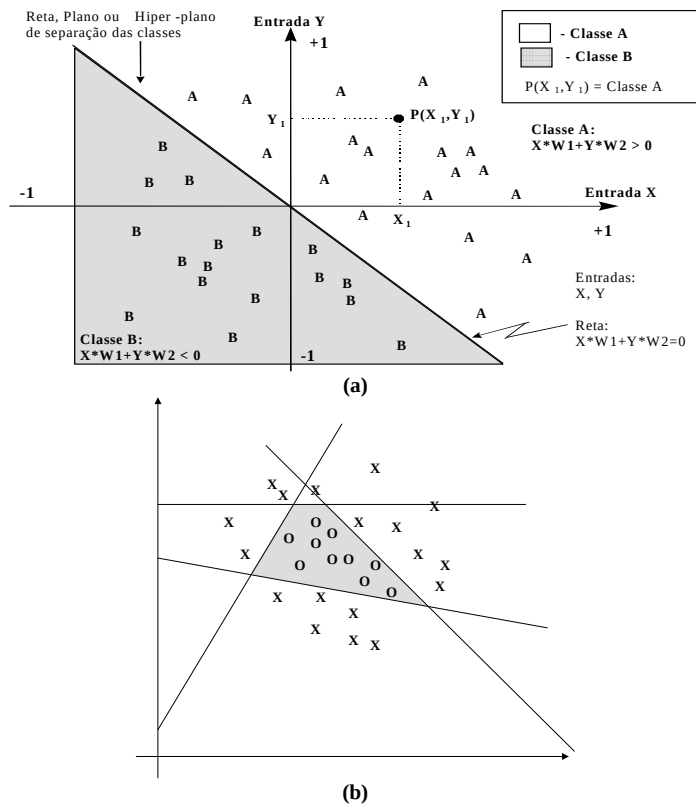


Figura 2. Representação de Conhecimentos nas Redes Neurais:
(a) Um neurônio (b) Combinação de neurônios

O método LDA (*Linear Discriminant Analysis*) [STA 00], correspondente à discriminação entre casos realizada por redes MLP, encontra funções discriminantes (normalmente equações lineares). O resultado da análise é uma equação que determina o grau de pertinência de um caso a uma dada classe.

Redes Neurais do tipo MLP e redes polinomiais podem ser usadas não apenas para representar conhecimento de classes, mas também de funções ou mapeamentos não lineares (função de aproximação). A representação do conhecimento neste caso não pode ser simplificada ou entendida como limites entre classes, porém é a própria função não-linear que a rede calcula. A representação do conhecimento exige então a definição da estrutura e o ajuste dos pesos da rede neural, tornando-a de difícil compreensibilidade. MLP's para aproximação podem ser representadas através de modelos gráficos [FAY 96, capítulo 3], que incluem cadeias de Markov, redes de Bayes e redes semânticas. Apesar do processamento e do significado dos grafos usados para representar estas estruturas serem

diferentes, a representação usada pode ser a mesma. Uma forma de representar grafos é através de matrizes, onde os elementos da matriz representam os pesos das redes neurais, ou a probabilidade de transição em uma cadeia de Markov ou probabilidades condicionais em redes de Bayes. A representação do conhecimento através de uma equação linear ou de um polinômio através de regressão é equivalente ao treinamento de uma rede MLP e o conhecimento armazenado nos pesos de uma rede neural correspondem aos coeficientes de um polinômio.

Redes Neurais não-supervisionadas como Mapas de Kohonen representam o conhecimento através de conjuntos de representativos [KOH 88], ou seja, as redes de Kohonen criam *clusters* através de um processo de auto-organização.

A explicitação de conhecimentos usualmente é feita através da obtenção de regras simbólicas em uma linguagem próxima à linguagem natural (descritas de forma compreensível para os seres humano) que representam os conhecimentos adquiridos e representados internamente no sistema. Esta tarefa de explicitação de conhecimentos pode ser simples e imediata, como é o caso das árvores de decisão [QUI 92], ou pode necessitar da aplicação de complexos algoritmos de extração de regras, como é o caso das redes neurais [AND 95, OSO 99, CEC 97]. Quando falamos em *regras simbólicas* é importante salientar que existem *diferentes capacidades de representação de conhecimentos* associadas aos diferentes tipos de regras usados.

Exemplos de regras tipo causa/feito ou condição/ação:

Se temp_paciente = normal ou temp_paciente <> febre	
Então tomar_remédio = Não	[Regras booleanas]
Se temp_paciente = média ou temp_paciente = alta	
Então tomar_remédio = Sim	[Entradas discretizadas]
Se temp_paciente > 37.0	
Então tomar_remédio = Sim	[Entradas quantitativas]
Se temp_paciente > temp_médico	
Então tomar_remédio = Sim	[Contexto: relação entre 2 entradas]
Se pertence_ao_intervalo (temp_paciente, 37.0, 39.0)	
Então tomar_remédio = Sim	[Intervalos do tipo fuzzy]
Se grau_de_pertinência (temp_paciente, 37.0, 39.0)	
Então tomar_remédio = quantidade (temp_paciente)	[Saída numérica]
Se temp_paciente > 37.0	
Então tomar_remédio = grau_de_certeza(diagnóstico)	[Saída com probabilidade]
Se aumentou(temp_paciente)	
Então aumenta(medicção)	[Análise de comportamento]

Exemplo de regras de associação:

Se pneus **Então** serviços automotivos (confiança=98.8%, suporte=5.79%)

[Regra de associação]

“98.8% dos carros que apresentavam problemas nos pneus necessitaram também serviços automotivos.”

“5.79% dos carros possuem problemas nos pneus ou necessitaram serviços automotivos.”

As regras podem ser divididas basicamente em dois grupos: regras baseadas em cálculo de predicados e regras baseadas em conjuntos fuzzy. As regras baseadas em cálculo de predicados possuem na premissa e na consequência uma proposição ou um predicado. Sistemas de inferência fuzzy necessitam de uma função de pertinência, que especifica o grau de validade da parte da consequência. Há dois tipos básicos de regras fuzzy: Mamdani e Sugeno [CEC 97]. A principal diferença está na parte da consequência. Nas regras Mamdani a consequência é representada através de uma variável lingüística definida por uma função de pertinência que calcula o grau de pertinência de uma variável de saída da regra ao correspondente conjunto fuzzy. Funções de pertinência típicas são triângulos e trapézios. Para regras Sugeno, a consequência é representada por uma equação que define diretamente o valor da variável de saída da regra. Um conjunto de regras define um mapeamento não-linear entre variáveis de entrada e de saída. Formas de representar a estrutura deste conjunto de regras é através de grafos (usada na implementação de sistemas especialistas) ou através de matrizes (usada na definição de controladores fuzzy). O conceito “fuzzy” na especificação de um sistema de inferência deve estar claro para o usuário: a lógica “fuzzy” pode especificar graus de probabilidade, possibilidade ou certeza, sendo diferente o significado do conhecimento especificado em cada um deles.

O conhecimento também pode ser representado na forma gráfica através de histogramas (uma dimensão) ou através de tabelas de contingência (duas dimensões) [FAY 96, capítulo 13].

Concluindo, nesse trabalho apresentamos alguns dos principais métodos de aprendizado de máquinas usados na descoberta de conhecimentos (KD), dando ênfase as diferentes técnicas de aquisição e representação de conhecimentos, as quais possuem particularidades que lhes atribuem vantagens e desvantagens próprias a cada método. A explicitação dos conhecimentos adquiridos é uma etapa muito importante neste processo de descoberta de conhecimentos, onde mostramos os diferentes tipos de regras que podem ser obtidos. Isto nos leva a uma questão final: qual será o melhor método de aquisição, representação e explicitação para uma dada aplicação?

Referências

- [AND 95] ANDREWS, R.; DIEDERICH, J.; TICKLE, A. **A Survey And Critique of Techniques For Extracting Rules From Trained ANN**. Technical Report - Neurocomputing Research Centre, Queensland University of Technology - Brisbane, Australia. January 1995 (Also published in: Knowledge-Based Systems Journal). [<http://www.fit.qut.edu.au/NRC/>]
- [BRA 00] BRAGA, A.; LUDERMIR, T.; CARVALHO, A. **Redes Neurais Artificiais: Teoria e Aplicações**. LTC Editora, Rio de Janeiro. 2000.
- [CEC 97] CECIN, A.L. **The Extraction of Fuzzy Rules from Neural Networks**. Shaker Verlag, Aachen, Alemanha. 1997.
- [FAY 96] FAYYAD, U.M.; PIATETSKY-SHAPIRO, G.; SMYTH, P.; UTHURUSAMY, R. (eds.). **Advances in Knowledge Discovery and Data Mining**. AIII Press/The MIT Press. Menlo Park, California. 1996.
- [KAR 99] KARYPIS, G.; HAN, E.-H. S.; KUMAR, V. **Chameleon: Hierarchical Clustering Using Dynamic Modeling**. Computer. Vol. 32, n.8. Agosto. 1999.
- [KOH 88] KOHONEN, T.. **Self-Organization and Associative Memory**. Springer Verlag. 1988.
- [KOL 93] KOLODNER, J.. **Case-Based Reasoning**. Morgan Kaufmann, San Mateo. 1993.
- [NIK 97] NIKOLOPOULOS, C. **Expert Systems: Introduction to first and second generation and Hybrid Knowledge Based Systems**. New York: Marcel Dekker Inc. Press, 1997.
- [MIT 97] MITCHELL, T, M. **Machine Learning**. WCB – McGraw-Hill, Boston, MA. 1997.
- [OSO 99] OSORIO, F. & VIEIRA, R. **Sistemas Híbridos Inteligentes**. Tutorial apresentado no ENIA'99 – Encontro Nacional de I.A., XIX Congresso da SBC. Rio de Janeiro, 1999. [<http://www.inf.unisinos.br/~osorio/enia99/>]
- [QUI 92] QUINLAN, J. **C4.5: Programs for Machine Learning**. Academic Press/Morgan Kaufmann, 1992.
- [RUM 86] RUMELHART, D.; HINTON, G. & WILLIAMS, R. Learning Internal Representations by Error Propagation. In: Parallel Distributed Processing - Explorations in the Microstructure of Cognition, Vol.1. Cambridge: MIT Press. 1986.
- [STA 96] STATSOFT, INC. **The Statistics Homepage**. <http://www.statsoft.com/textbook/streliab.html>. 1984-2000.

