

Combinando Invenção de Predicados e Revisão de Teorias Probabilísticas de Primeira-ordem

Kate Revoredo , Aline Paes , Gerson Zaverucha , Vitor Santos Costa

¹Departamento de Engenharia de Sistemas e Computação - COPPE
Universidade Federal do Rio de Janeiro (UFRJ)

{kate, ampaes, gerson, vitor}@cos.ufrj.br

Resumo. Durante o processo de revisão, similarmente quando aprendendo em ILP, o algoritmo pode não ser capaz de propor uma modificação útil usando a linguagem existente. Neste caso, técnicas de invenção de predicados podem ser usadas para automaticamente estender a linguagem com a definição de novos predicados. Neste trabalho, estendemos o nosso sistema de revisão de teorias probabilísticas de primeira-ordem, chamado PFORTE, com dois novos operadores de revisão que propõem novos predicados, aprendem suas definições e CPDs. Experimentos mostram que existem situações onde somos capazes de representar aspectos não observados da base de dados e conseqüentemente aprender um modelo melhor, melhorando a classificação.

Abstract. Sometimes during the revision process, similarly when learning in standard ILP, the algorithm is not capable of proposing a useful modification using the existing language. In this case, techniques of predicate invention can be used to automatically extend a logical language with newly defined predicates. In this work, we extend our revision of probabilistic first-order theory system, called PFORTE, with two novel revision operators that propose new predicates, learning their definition and associated conditional probability distribution. Experimentation shows that there are indeed examples where we are able to represent unobserved aspects of the data and consequently learn a better model, which can improve classification accuracy.

1. Introdução

Tem sido grande o interesse em integrar formalismos baseados em lógica de primeira-ordem com mecanismos de raciocínio probabilístico, definindo então teorias probabilísticas de primeira-ordem. Alguns exemplos são PRM [Friedman et al. 1999], SLP [Muggleton 2002], BLP [Kersting and De Raedt 2002], CLP($\mathcal{BN}$) [Costa et al. 2003], MLN [Kok and Domingos 2005], entre outros. A tarefa de aprender tais teorias é denominada com frequência aprendizado estatístico relacional (Statistical Relational Learning (SRL)).

Muitos trabalhos em SRL assumem que queremos aprender uma teoria do zero. Este pode não ser o caso. Muitas vezes, uma teoria é fornecida. Se admitirmos que

esta teoria é aproximadamente correta, isto é, que apenas alguns pontos na sua estrutura impedem que esta modele corretamente a base de dados, é mais eficiente procurar por estes pontos na teoria e revisá-los, do que utilizar um algoritmo que aprende uma nova teoria do zero. Essa idéia foi aplicada pelo sistema de revisão PFORTE proposto por nós em [Paes et al. 2005, Paes et al. 2006] (onde PFORTE significa **P**robabilistic **F**irst **O**rders **R**evisão of **T**heories from **E**xamples). PFORTE avalia uma teoria inicial composta por cláusulas definidas probabilísticas de primeira-ordem, detectando lugares que evitam que esta teoria reflita a base de dados corretamente e propondo mudanças a estes pontos através da aplicação de *operadores de revisão*. Acreditamos que algoritmos de aprendizado de estrutura, como o Alchemy [Kok and Domingos 2005] e o de aprendizado de BLPs [Kersting and De Raedt 2002], que também podem aprender de uma teoria inicial, podem tirar vantagens da estratégia do PFORTE: ao contrário de propor modificações na estrutura inteira e levar um longo tempo concretizando isto, eles podem usar o algoritmo do PFORTE para detectar e revisar somente os pontos incorretos da teoria.

Freqüentemente, a tarefa de classificação pode beneficiar-se da aquisição de novos conceitos, que podem revelar conexões entre atributos ou auxiliar a simplificação da teoria. Dentro da abordagem de Programação em Lógica Indutiva (ILP), invenção de predicados é uma forma de extensão automática da linguagem com novas definições de predicados para solucionar ou melhorar o processo de aprendizado [Kramer 1995]. Do ponto de vista de aprendizado estatístico, variáveis não observadas podem ser introduzidas como um meio de melhorar o processo de aprendizado [Ramachandran and Mooney 1998, Elidan et al. 2001]. Neste trabalho, nos investigamos os benefícios de considerarmos técnicas de invenção de predicados e/ou introdução de variáveis não observadas durante a revisão de teorias probabilísticas de primeira-ordem. Nossa abordagem estende o sistema de revisão PFORTE com dois novos operadores de revisão que propõem novos predicados, aprendem sua definição e distribuição de probabilidade condicional associadas. Os experimentos mostram que existem situações onde somos capazes de representar aspectos não observados dos dados e conseqüentemente aprender um modelo melhor, que pode melhorar a acurácia de classificação.

Este trabalho está organizado da seguinte forma: na seção 2 revisamos revisão de teorias de primeira-ordem e na seção 3 revisão de teorias probabilísticas de primeira-ordem; na seção 4 os nossos operadores de revisão de teoria que utilizam invenção de predicados são introduzidos; os resultados experimentais são descritos na seção 5; e finalmente algumas conclusões e trabalhos futuros são apresentados na seção 6.

2. Revisão de Teorias de Primeira-ordem

A aquisição de conhecimento em ILP é uma tarefa, que consome tempo e com possibilidades de erro. Revisão de teorias de primeira-ordem visa melhorar uma base de conhecimento automaticamente usando métodos de aprendizado de máquina [Wrobel 1996].

A tarefa de revisão de teoria de primeira-ordem pode ser definida como:

- **Dada:** uma teoria inicial composta por: uma componente invariante, chamada *conhecimento preliminar (background knowledge (BK))*, e uma componente que pode ser modificada (H) e um conjunto consistente de exemplos (E) dividido em exemplos positivo (E^+) e exemplos negativos (E^-).
- **Encontre:** uma teoria "minimamente revista" que prove todos os exemplos positivos em E^+ e nenhum dos exemplos negativos em E^- , de forma que a teoria final seja consistente com a base de dados

Pode-se dizer então que aprendizado em ILP é um caso particular de revisão de teoria, onde H é inicialmente vazio.

Em uma teoria com múltiplos predicados e cláusulas, muitas cláusulas podem estar envolvidas na prova de um exemplo negativo e/ou podem ser generalizadas para que uma não prova de um exemplo positivo torne-se bem sucedida. A qualidade das revisões produzidas depende crucialmente de quais dessas cláusulas serão escolhidas para serem modificadas e onde, definindo então um conjunto de *pontos de revisão*. Após especificar onde a teoria deve ser modificada, é preciso determinar como modificá-la, definindo então *operadores de revisão*. Exemplos de operadores de revisão são: *exclusão de cláusula*, *adição de antecedente*, *exclusão de antecedente* e *adição de cláusula*. Um possível algoritmo de revisão que utiliza uma abordagem gulosa de subida de encosta (*greedy hill-climbing*) é exibido na figura 1.

Algorithm 1 Algoritmo de revisão de teorias de primeira-ordem

repita

gerar pontos de revisão;

para cada ponto de revisão

gerar possíveis revisões;

avaliar possíveis revisões de acordo com alguma métrica;

atualizar melhor revisão possível;

implementar melhor revisão;

até que nenhuma revisão melhore a teoria

3. Revisão de Teorias Probabilísticas de Primeira-ordem

Uma cláusula probabilística de primeira-ordem é uma cláusula definida, sem funções e com uma distribuição de probabilidade condicional associada. Dessa forma, o algoritmo de revisão exibido em 1 precisa de algumas alterações para revisar teorias probabilísticas de primeira-ordem. Estas alterações estão exibidas em *itálico* e serão descritas nesta seção.

3.1. Gerar pontos de revisão

Para cada exemplo, uma rede suporte [Kersting and De Raedt 2002] é construída. Esta rede suporte é um caso particular de rede Bayesiana que contém todas as cláusulas utilizadas na prova do exemplo em questão, onde os antecedentes da cláusula são os nós

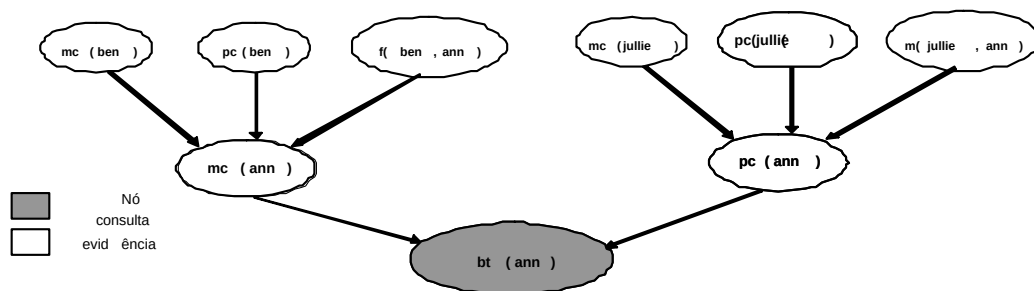


Figure 1. Rede suporte para o exemplo do tipo sanguíneo *blood-type* [Friedman et al. 1999]: "modelo genético de herança de um único gene que determina o tipo sanguíneo $bt(X)$ de uma pessoa X . Cada pessoa X tem duas cópias de um cromossomo contendo este gene, um $mc(Y)$, herdado da sua mãe $m(Y, X)$, e um, $pc(Z)$, herdado do seu pai $f(Z, X)$."

pais do nó correspondente ao conseqüente da cláusula e a distribuição de probabilidade condicional (CPD) associada a este nó é a CPD da cláusula. Dessa forma, o conjunto composto pelos nós que não foram classificados corretamente, o valor inferido para este nó difere do fornecido pela base de dados, e todos os nós pertencentes a sua cobertura de Markov¹ são coletados. Se algum desses nós não for observado, não tem seu valor especificado na base de dados, sua cobertura de Markov também é considerada e assim por diante. Em seguida, as cláusulas cuja a cabeça instanciada pertencem a este conjunto tornam-se pontos de revisão. Como exemplo, considere a rede suporte exibida na Figura 1, construída a partir da prova do exemplo $bt(ann) = b$ usando a teoria:

$bt(X) : -mc(X), pc(X).$

$mc(X) : -mc(Y), pc(Y), m(Y, X).$

$pc(X) : -mc(Y), pc(Y), f(Y, X). \}$

$mc(ben).pc(ben).f(ben, ann).mc(jullie).pc(jullie).m(jullie, ann).$

Suponha que o exemplo $bt(ann)$ foi classificado incorretamente. Já que, sua cobertura de Markov é composta pelos nós $mc(ann)$ e $pc(ann)$, o conjunto de nós criado é $\{bt(ann), mc(ann), pc(ann)\}$. Logo, somente a primeira cláusula é considerada como ponto de revisão.

3.2. Gerar possíveis revisões

Depois de encontrar os pontos de revisão todos os operadores utilizados em aprendizado de máquina podem ser considerados para propor modificações, mas com algumas modificações. Por exemplo, quando adicionando antecedentes, qualquer antecedente da

¹Lembramos que a cobertura de Markov de um nó em uma rede Bayesiana consiste dos seus nós pais, dos seus nós filho e dos nós pai dos seus nós filho.

linguagem pode ser adicionado se melhorar o valor da função de avaliação considerada. De forma similar, quando excluindo um antecedente qualquer antecedente da cláusula escolhida como ponto de revisão pode ser removido desde que melhore o valor da função de avaliação. Em ambos os casos o antecedente escolhido para ser adicionado/ excluído é aquele que fornece a melhor avaliação.

3.3. Avaliar possíveis revisões

Geralmente em uma abordagem Bayesiana a função de avaliação utilizada para avaliar um modelo proposto são: verossimilhança, verossimilhança condicional ou critério de informação bayesiana (*Bayesian Information Criteria* (BIC)). Na abordagem de revisão probabilística, a rede suporte de todos os exemplos é considerada para calcular qualquer uma dessas funções de avaliação. Dessa forma, para cada revisão proposta a rede suporte de todos os exemplos é construída, e então as CPDs são re-aprendidas utilizando um algoritmo de aprendizado de parâmetro e por fim a modificação proposta é avaliada.

4. Invenção de predicados como operadores de revisão

Algumas vezes durante o processo de revisão, de forma similar durante o aprendizado em ILP, o algoritmo não é capaz de propor uma modificação útil usando a linguagem existente. Vários trabalhos foram propostos em ILP para estender a linguagem com novos predicados de forma a solucionar a limitação da linguagem ou melhorar o processo de aprendizado, conhecido na área como *invenção de predicados* [Kramer 1995]. Do ponto de vista de aprendizado estatístico invenção de predicados pode ser vista como a introdução de variáveis não observadas [Ramachandran and Mooney 1998, Elidan et al. 2001], o que revela conexões entre variáveis aleatórias ou ajuda a simplificar a rede Bayesiana.

Nesta seção nós descrevemos dois novos operadores adicionais de revisão para serem considerados quando revisando uma teoria probabilística de primeira-ordem: operadores de *compactação* e *incremento*. Ambos inventam novos predicados, aprendem sua definição e CPD.

4.1. Operador de compactação

A motivação para o uso do *operador de compactação* é a simplificação da teoria com consequente melhora na acurácia. Para isto, este operador recebe um ponto de revisão, que é uma cláusula, inventa um novo predicado e substitui o corpo do ponto de revisão por ele. Depois, uma cláusula para definir este novo predicado é criada com o corpo substituído. Este operador é relacionado a abordagem utilizada por Elidan et al [Elidan et al. 2001] para redes Bayesianas, diferindo na detecção de onde o novo predicado (variável não observada) deve ser inserido.

Considere por exemplo a cláusula em 1 originalmente obtida de uma aplicação de seguros [Binder et al. 1997].

$$custo(X, Y) \leftarrow habilidadeDirecao(X), antilock(Y), milhas(Y) \quad (1)$$

O operador de compactação propõe a seguinte modificação, onde $acidente(X, Y)$ é o predicado inventado

$$\begin{aligned} custo(X, Y) &\leftarrow acidente(X, Y). \\ acidente(X, Y) &\leftarrow habilidadeDirecao(X), antilock(Y), milhas(Y). \end{aligned}$$

Outras cláusulas da teoria que contenham em seu corpo os antecedentes $habilidadeDirecao(X)$, $antilock(Y)$, $milhas(Y)$ também são alteradas para substituir esse grupo de antecedentes pelo novo predicado $acidente(X, Y)$. Dessa forma todas as cláusulas que foram alteradas têm as suas CPDs simplificadas e conseqüentemente a inferência na rede Bayesiana resultante também.

4.2. Operador de incremento

O segundo operador de revisão, *incremento*, segue a idéia apresentada em [Ramachandran and Mooney 1998] e assim como o operador de compactação, difere na detecção de onde o novo predicado deve ser inserido. A motivação para este operador é a detecção de possíveis conexões entre variáveis aleatórias além de apenas simplificar a teoria. Este operador recebe um ponto de revisão, inventa um novo predicado e então substitui um dos antecedentes do ponto de revisão por ele. Uma cláusula para definir o novo predicado é então criada com o antecedente substituído no corpo. Esta cláusula é especializada na tentativa de melhorar o valor da função de avaliação. Ao final, para cada antecedente (A_i) na cláusula original (C) nós definimos um par de cláusulas (C_{mod} e C_{new}). A primeira é C com A_i substituído pelo predicado inventado (new_i), e a segunda a cláusula definindo new_i com A_i e talvez outros antecedentes no corpo. O par que fornecer o melhor valor para a função de avaliação será considerado como a modificação proposta pelo operador de incremento.

Como exemplo considere o ponto de revisão exibido em 1. O operador de incremento irá propor 3 pares de cláusulas como possíveis revisões cada um baseado em um dos três antecedentes do ponto de revisão. Após avaliar cada um, o par mostrado abaixo para o antecedente $habilidadeDirecao(X)$ retornou a melhor avaliação e foi o escolhido como revisão proposta.

$$\begin{aligned} custo(X, Y) &\leftarrow acidente(X, Y), antilock(Y), milhas(Y). \\ acidente(X, Y) &\leftarrow habilidadeDirecao(X). \end{aligned}$$

Assim como é feito pelo algoritmo de compactação, outras cláusulas da teoria que contenham o corpo da cláusula que define o predicado inventado são alteradas para substituir esse grupo de antecedentes pelo novo predicado, podendo assim simplificar a CPD das cláusulas envolvidas.

Primeiramente, nós apenas consideramos novos predicados com aridade definida pelo número de variáveis distintas que aparecem na cláusula original. O novo predicado assumirá o mesmo domínio do predicado que está substituindo na cláusula original.

5. Resultados Experimentais

Os experimentos foram conduzidos usando 3 problemas, que foram inicialmente representados através de uma teoria BLP e são exibidas aqui na forma de redes. Em seguida, nós retiramos alguns predicados intermediários da teoria (e da linguagem) e então os antecedentes das cláusulas que os definem (pais) foram conectados aos conseqüentes das cláusulas onde ele pertence ao corpo (filhos). Depois, rodamos o sistema de revisão PFORTE estendido com os dois novos operadores de revisão de forma a verificar se eles seriam capazes de recuperar a teoria original. Os resultados foram validados usando 10-fold cross-validation. Nós selecionamos a teoria que forneceu o maior valor para a verossimilhança condicional dos exemplos. As três bases de dados consideradas foram:

- uma pequena base de dados artificial onde a consideração de invenção de predicados é importante para encontrar uma boa classificação (figura 2 a)). Nos experimentos o nó *H* foi retirado (figura 2 b));
- o problema do Alarme, desenvolvido para monitorar pacientes em uma unidade de tratamento intensivo. A rede original pode ser visualizada em [Beinlich et al. 1989], onde nos nossos experimentos os nós 11, 27 e 32 foram retirados;
- o *aromatic amino acid pathway of yeast* [Angelopoulos and Muggleton 2002]. A base de dados foi originalmente obtida da base de dados do KEGG pathway [int 1992] e pode ser visualizada na figura 3. Os círculos representam metabolitos, referenciados por seu número de ascensão KEGG (ex: C00631 e C03356). O pathway pertence a biosíntese dos aromatic amino-ácidos fenilalanina, tirosina e triptofano que são os círculos em vermelho. Os caminhos de maior reação são exibidos em preto. Moléculas que estão envolvidas em reações, mas não se relacionam com os pathways principais são exibidas em azul. Cada retângulo, junto com os metabolitos que são ligados a ele, representam reações químicas. Reagentes são exibidos entrando em um lado de cada retângulo e produtos saindo pelo outro. Cada retângulo é rotulado pelo número-EC da enzima catalizadora da reação (ex: C00631 é um reagente, 4.2.1.11 a enzima catalizadora e C00074 o produto). Para simplificação setas vão de intermediadores para produtos.

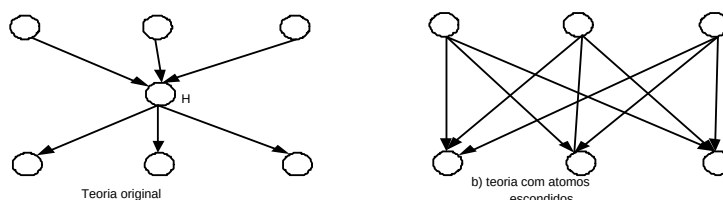


Figure 2. Teoria original e teoria com o átomo H retirado para base de dados a)

No experimento *a*) as variáveis de classe correspondem aos três nós do nível mais baixo da rede (*a*). Nossa modificação é exibida em (*b*). Apesar do grafo modificado ter

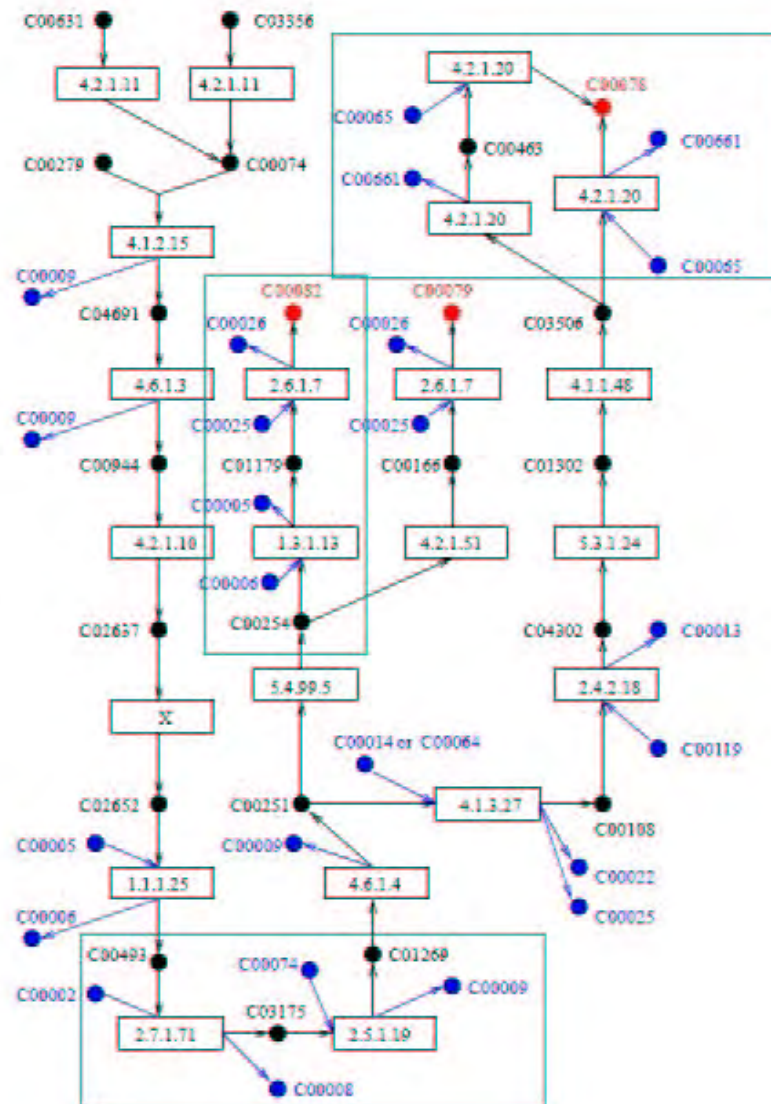


Figure 3. Teoria original para a base de dados c).

arestas conectando todos os nós do topo aos nós de baixo, é mais complicado considerar a correlação entre as variáveis de baixo. Aplicando o nosso sistema de revisão, PFORTE, a este exemplo várias possíveis revisões foram consideradas. A melhor avaliação na base de treinamento foi obtida aplicando o operador de compactação, o que neste caso recupera o nó retirado, logo PFORTE foi capaz de recuperar a teoria original. A melhora também foi obtida no conjunto de teste.

No experimento b) PFORTE percebeu a necessidade de vários novos átomos. PFORTE usou o operador de incremento para introduzir um nó conectando o nó 35 aos

pais 22 e 23 e também incluiu o nó 17 como um pai do nó novo. Ele também introduziu novos nós em outros pontos da teoria usando o mesmo operador, mas a modificação anterior foi a melhor de acordo com a função de avaliação. Melhora também obtida no conjunto de teste.

Para a base de dados *c*) consideramos três subconjuntos do pathway, que estão marcados em retângulos na Figura 3. Para validação da nossa proposta assumimos que toda a base estava disponível, argumentavelmente uma proposição muito otimista. O primeiro experimento removeu as reações contendo a enzima 2.5.1.19 e os metabolitos *C03175* e *C00074* presentes no subconjunto mais abaixo na figura. O segundo experimento removeu a reação contendo a enzima 2.6.1.7 e o metabolito *C00025* no subconjunto do meio. O terceiro experimento removeu a enzima 4.2.1.20 do subconjunto do topo. No primeiro experimento PFORTE foi capaz de encontrar a reação retirada, entretanto ligando apenas ao metabolito *C01269*. No segundo e no terceiro experimentos PFORTE encontrou as reações retiradas, mas elas não foram ligadas a todos os metabolitos como originalmente. No segundo experimento o novo nó foi ligado somente ao metabolito *C00082* e no terceiro, somente ao metabolito *C00661*. Todas as revisões foram propostas pelo operador de compactação. Novamente, a melhora também foi obtida no conjunto de teste.

6. Conclusão

Neste trabalho descrevemos dois operadores de revisão que usam técnicas de invenção predicados durante o processo de revisão de uma teoria probabilística de primeira ordem. Estes novos operadores de revisão propõem novos predicados, aprendendo suas definições e CPDs associadas. Resultados experimentais demonstram que realmente há situações onde nós somos capazes de representar aspectos não observados dos dados e conseqüentemente aprender um modelo mais preciso, o que pode promover uma melhor em classificação.

Dado os estudos apresentados aqui pretendemos estender os experimentos feitos com o problema do *aromatic amino acid pathway of yeast* para bases de dados mais realistas;

Atualmente estamos trabalhando para permitir que mais de um antecedente seja incluído pelo operador de incremento; e pretendemos expandir os operadores para escolherem o melhor número de variáveis e para determinar o melhor domínio dos predicados inventados [Elidan et al. 2001].

Agradecimentos

O segundo autor é parcialmente financiado pela CAPES e os outros pelo CNPq. Vítor Santos Costa foi parcialmente financiado por fundos concedidos ao LIACC pela Fundação para a Ciência e Tecnologia, Program POSI.

References

- (1992). International union of biochemistry and molecular biology. enzyme nomenclature: recommendations (1992) of the nomenclature committee of the international union of biochemistry and molecular biology. In *Academic Press*, New York.
- Angelopoulos, N. and Muggleton, S. (2002). Machine learning metabolic pathway descriptions using a probabilistic relational representation. *Electronic Transactions in Artificial Intelligence*, 6.
- Beinlich, I., Suermondt, G., Chavez, R., and Cooper, G. (1989). The alarm monitorim system. In *Proc. of the 2nd European Conf. on AI and Medicine*.
- Binder, J., Koller, D., Russell, S., and Kanazawa, K. (1997). Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29:213–244.
- Costa, V., Page, D., Qazi, M., and Cussens., J. (2003). CLP(BN): Constraint logic programming for probabilistic knowledge. In *Proc. 19th UAI*, pages 517–524.
- Elidan, G., Lotner, N., Friedman, N., and Koller, D. (2001). Discovering hidden variables: A structure-based approach. In *Advances in Neural Information Processing Systems 14*, pages 479–485.
- Friedman, N., Getoor, L., Koller, D., and Pfeffer, A. (1999). Learning probabilistic relational models. In *Proc. 16th IJCAI*, pages 1300–1309.
- Kersting, K. and De Raedt, L. (2002). Basic Principles of Learning Bayesian Logic Programs. Technical Report 174, University of Freiburg.
- Kok, S. and Domingos, P. (2005). Learning the structure of Markov Logic Networks. In *Proc. 22nd ICML*, pages 441–448.
- Kramer, S. (1995). Predicate invention: A comprehensive view. Technical Report ÖFAI-TR-95-32, Austrian Research Institute for Artificial Intelligence.
- Muggleton, S. (2002). Learning structure and parameters of stochastic logic programs. In *Proc. 12th ILP/LNAI 2583, Springer Verlag*, pages 198–206.
- Paes, A., Revoredo, K., Zaverucha, G., and Costa, V. S. (2005). Probabilistic first-order theory revision from examples. In *Proc. 15th ILP, LNAI 3625, Springer Verlag*, pages 295–311.
- Paes, A., Revoredo, K., Zaverucha, G., and Costa, V. S. (2006). PFORTE: Revising probabilistic FOL theories. In *Proc. IBERAMIA-SBIA 2006, LNAI 4140, Springer Verlag*, pages 441–450.
- Ramachandran, S. and Mooney, R. (1998). Theory refinement of bayesian networks with hidden variables. In *Proc. 15th ICML*, pages 454–462.
- Wrobel, S. (1996). First-order theory refinement. In Raedt, L. D., editor, *Advances in Inductive Logic Programming*, pages 14–33. IOS Press.