

Comparação de Funções de Avaliação em Revisão de Teorias Probabilísticas de Primeira-Ordem

Aline Paes*, Kate Revoredo*, Gerson Zaverucha*, Vitor Costa*†

*Programa de Engenharia de Sistemas e Computação(COPPE)

Universidade Federal do Rio de Janeiro

Caixa Postal 68511, 21945-970, Rio de Janeiro, RJ, Brasil

{ampaes,kate,geron}@cos.ufrj.br

†Computer Sciences Department University of Wisconsin-Madison

1210 West Dayton Street, Madison, WI 53706, USA

vitor@biostat.wisc.edu

Abstract—Recentemente tem sido grande o interesse em integrar raciocínio probabilístico com representações lógicas de primeira-ordem. Os modelos propostos na literatura aprendem considerando como espaço de busca modificações em toda a teoria. Em um trabalho anterior argumentamos que quando a teoria é aproximadamente correta a utilização de técnicas de revisão para apenas alterar a teoria nos pontos em que ela falha na cobertura dos exemplos seria mais vantajoso. Para avaliar estas modificações e escolher a melhor foi utilizado log verossimilhança. Entretanto, em tarefas de classificação Bayesiana proposicionais foi mostrado que esta não se mostra adequada e por esse motivo log verossimilhança condicional deve ser utilizada. Neste artigo, comparamos os resultados experimentais da utilização de quatro funções de avaliação probabilísticas, incluindo log verossimilhança condicional, ao revisar uma teoria probabilística de primeira-ordem.

I. INTRODUÇÃO

A possibilidade de representação de indivíduos, suas propriedades e as relações entre eles, fizeram da lógica de primeira ordem um sistema de representação de conhecimento muito vantajoso, mas limitado por não representar incerteza. Recentemente tem sido grande o interesse em integrar estes sistemas com mecanismos de raciocínio probabilístico definindo uma teoria probabilística de primeira ordem, unindo o que eles têm de melhor. Alguns exemplos são: *Probabilistic Relational Models (PRM)* [Koller, 1999] [Friedman et al., 1999], *Independent Choice Logic (ICL)* [Poole, 1993], *Bayesian Logic Programs (BLP)* [Kersting and De Raedt,], *Constraint Logic Programming (CLP(BN))* [Costa et al., 2003] e *Stochastic Logic Program (SLP)* [Muggleton, 2002].

Tendo a diminuição do espaço de busca e conseqüentemente uma melhora em desempenho como motivação, foi proposto em [Revoredo and Zaverucha, 2002] um sistema de revisão de programas em lógica Bayesiano denominado RBLP, que espera receber um BLP inicial e através dos exemplos descobrir pontos que falham neste BLP propondo modificações para estes e escolhendo a melhor de acordo com uma função de avaliação. O BLP retornado será correto. Em [Paes et al., 2005] este sistema de revisão foi estendido para que as modificações também incluíssem especializações na teoria, mesmo não tendo exemplos negativos, com o objetivo

de melhorar o valor da função de avaliação probabilística.

Para avaliar as modificações propostas pelo sistema de revisão em [Revoredo and Zaverucha, 2002] foi utilizada a log verossimilhança. Entretanto, em tarefas de classificação Bayesiana proposicionais esta não se mostra adequada e por este motivo log verossimilhança condicional da classe dados os atributos deve ser utilizada, já que para propósitos de classificação somente a parcela da verossimilhança que corresponde a esta última função é relevante [Grossman and Domingos, 2004]. Neste artigo, comparamos os resultados experimentais da utilização de quatro funções de avaliação probabilísticas ao revisar uma teoria probabilística de primeira-ordem: log verossimilhança condicional, log verossimilhança, Minimum Description Length (MDL) [Lam and Bacchus, 1994] e Akaike's Information Criterion (AIC) [Stone, 1977].

O presente artigo está organizado da seguinte forma: na seção II será brevemente apresentado refinamento de teoria de primeira-ordem e revisão de teoria probabilística de primeira-ordem; na seção III serão descritas as funções de avaliação probabilísticas utilizadas; os resultados experimentais são descritos na seção IV; finalmente na seção V são discutidos conclusões e trabalhos futuros.

II. CONCEITOS PRELIMINARES

Nesta seção serão apresentados brevemente os conceitos de refinamento de teoria de primeira-ordem, e revisão de teoria probabilística de primeira-ordem.

A. Revisão de Teoria

A aquisição de conhecimento é uma tarefa difícil, demorada e com chances de erro. O processo de automaticamente melhorar uma base de conhecimento existente utilizando métodos de aprendizado é executada pelos sistemas de revisão de teoria [Wrobel, 1996].

No problema de revisão de teoria começa-se com uma teoria inicial a qual desejamos minimamente modificar, através de algum critério de minimalidade, para que seja consistente com o conjunto de exemplos. Esta teoria inicial pode ser dividida em duas partes: uma assumidamente correta (conhecimento preliminar) e outra que pode ser modificada pela revisão

através dos exemplos. Já o aprendizado em Inductive Logic Programming (ILP), pode ser visto como uma revisão de teoria onde a teoria inicial consiste apenas do conhecimento preliminar e portanto somente ocorre a adição de novas cláusulas.

O problema de revisão de uma teoria pode ser definido da seguinte maneira, [Wrobel, 1996]:

Definição 2.1: Dados:

- uma teoria inicial T
- um conjunto de exemplos positivos (C^+) e negativos (C^-).

Achar:

- uma teoria revista T'
- que implique logicamente todos os exemplos positivos (completo), $T' \models C^+$
- e nenhum dos exemplos negativos (consistência), $\forall c^- \in C^- : T' \not\models c^-$
- e satisfaz um critério de minimalidade.

[Towell and Shavlik, 1994],[Garcez and Zaverucha, 1999], [Ramachandran and Mooney, 1998], [Buntine, 1991] e [Wogulis and Pazzani, 1993], [Richards and Mooney, 1995], [Wrobel, 1996] mostram que sistemas de revisão de teorias, proposicionais e de primeira-ordem respectivamente, podem aprender teorias mais precisas e com menos dados do que sistemas puramente indutivos.

B. Pontos de Revisão

Quando a teoria a ser revista é constituída de apenas um predicado o exemplo selecionado, positivo ou negativo, determina que tipo de operador deve ser utilizado, se será um operador de generalização ou um operador de especialização.

No caso de uma teoria com múltiplas definições de predicados muitas cláusulas podem estar envolvidas na prova de um exemplo negativo ou na não prova de um exemplo positivo. Desta forma a indicação de que tipo de operador deve ser utilizado não é imediata tornando-se necessário a determinação dos pontos desta teoria que precisam ser corrigidos (pontos de revisão). Dependendo do tipo de exemplo que esteja sendo considerado podemos ter a definição de dois tipos de pontos de revisão:

- Generalização - o literal em uma cláusula responsável pela falha na prova de um exemplo positivo (ponto de falha) e outros antecedentes (pontos de contribuição) que podem ter contribuído para esta falha atribuindo valores incorretos para as variáveis;
- Especialização - cláusulas usadas em provas de exemplos negativos.

A especificação do ponto de revisão determina o tipo de operador de revisão que precisará ser aplicado para tornar a teoria consistente com a base de dados.

C. Operadores de Revisão

Para efetuar a revisão em cada ponto de revisão são utilizados operadores. A princípio qualquer operador utilizado em aprendizado de máquina (primeira-ordem) pode ser

usado em um sistema de revisão de teoria. Abaixo descrevemos alguns desses operadores os quais foram definidos em [Richards and Mooney, 1995].

Os operadores para especialização são:

- Exclusão de regra - existem duas restrições para este operador. Ele não pode excluir uma cláusula que seja ou a cláusula base de um predicado recursivo ou a única cláusula que define um determinado predicado. Neste último caso substitui-se a cláusula a ser excluída pela regra conceito :- fail.
- Adição de antecedente - adiciona-se antecedentes a uma cláusula na tentativa de fazer com que não sejam provadas todos os exemplos negativos. Se a adição destes antecedentes fazer com que exemplos positivos deixem de ser provados, adiciona-se esta cláusula especializada à teoria e recomeça-se a especialização com a cláusula original, procurando especializações alternativas que retenham a prova dos outros exemplos positivos enquanto eliminam os negativos.

Os operadores para generalização são:

- Exclusão de antecedente - exclui antecedentes em uma cláusula, tornando-a mais geral, enquanto não prova nenhum exemplo negativo. O algoritmo não pára quando todas os exemplos positivos foram provados, continua excluindo tantos antecedentes quanto puder.
- Adiciona regra - é uma revisão baseada em cláusula. Deixa a cláusula original na teoria e gera novas baseadas nesta. O processo é feito em duas etapas. Primeiro copia a cláusula original e usando o algoritmo de exclusão de antecedente *hill-climbing*, exclui antecedentes sem permitir que nenhum exemplo negativo seja provado e permitindo também que alguns positivos, que antes não eram provados passem a ser (mesmo que isto permita a prova de negativos). Então cria uma ou mais especializações desta regra, utilizando o operador de adição de antecedentes, para permitir a prova dos exemplos positivos desejados enquanto elimina os negativos.
- Identificação - constrói uma cláusula nova para generalizar a definição de um antecedente que falhou na prova de um exemplo positivo. Melhor que desenvolver a cláusula do nada, executa um passo de resolução inversa [Muggleton, 1992] usando duas regras existentes no domínio da teoria.
- Absorção - Vai substituir uma cláusula já existente por uma nova a partir da substituição dos seus antecedentes falhos pela cabeça de uma cláusula, cujo o conjunto dos seus antecedentes contém os antecedentes definidos como sendo falhos.

D. Revisão de Teorias Probabilísticas de Primeira-Ordem

A teoria probabilística de primeira-ordem (TPPO) pode ser dividida em duas partes: uma lógica que são as cláusulas e outra probabilística (parâmetros) que são as distribuições de probabilidade (CPD) de cada uma dessas cláusulas. Esta distribuição de probabilidade associa uma probabilidade para

cada valor do domínio do predicado em questão. Como exemplo considere a TPPO definida pelas cláusulas

1. class(A):-obj(A,B),in(A,B,C),obj(A,C)
2. obj(A,B):-dom(A,B)

e pelas distribuições de probabilidades apresentadas na tabela II-D onde o domínio (D) de todos os predicados é binário ($D_{dom} = D_{obj} = \{triangulo(\text{ abreviado por } T), circulo(\text{ abreviado por } C)\}$, $D_{in} = \{verdadero(\text{ abreviado por } V), falso(\text{ abreviado por } F)\}$, $D_{class} = \{positivo(\text{ abreviado por } P), negativo(\text{ abreviado por } N)\}$).

TABLE I
CPDS DAS REGRAS

P(dom(A,B))	P(in(A,B,C))	dom(A,B)	P(obj(A,B))
< 0.40, 0.60 >	< 0.20, 0.80 >	T	0.99
		C	0.01

obj(A,B)	in(A,B,C)	obj(A,C)	P(class(A))
T	F	T	0.70
C	F	T	0.70
T	V	T	0.005
C	V	T	0.001
T	F	C	0.06
C	F	C	0.8
T	V	C	0.999
C	V	C	0.001

Quando usamos TPPO os exemplos que na abordagem lógica (vide definição 2.1) eram considerados negativos serão transformados em exemplos positivos, já que a distribuição de probabilidade é que passará a repetir esta diferença de acordo com o domínio dos predicados.

O problema de revisão de uma TPPO pode então ser definido da seguinte maneira:

Definição 2.2: Dados:

- uma teoria probabilística de primeira-ordem inicial T .
- um conjunto de exemplos C .
- uma função de avaliação probabilística.

Achar:

- uma teoria probabilística de primeira-ordem revista T'
- que implique logicamente todos os exemplos $T' \models C^+$
- maximize a função de avaliação probabilística
- e satisfaz um critério de minimalidade.

A base de treinamento \mathbf{C} é composta de átomos básicos. Para cada átomo básico provado é construída uma rede Bayesiana usando a noção de Construção de Modelo Baseado em Conhecimento (KBMC)[Haddawy, 1999]. Os nós dessa rede são as variáveis aleatórias onde o seu domínio é o domínio do predicado correspondente. Ao final temos uma nova base de treinamento \mathbf{B} composta das redes Bayesianas resultantes. É esta base que será utilizada no aprendizado dos parâmetros e para determinar o valor da função de avaliação probabilística. Qualquer algoritmo de inferência Bayesiano pode ser aplicado para encontrar a probabilidade do exemplo.

A revisão pode ser dividida em duas partes: revisão da parte lógica da TPPO (revisão da estrutura) e da parte probabilística (revisão dos parâmetros).

Ao revisarmos os parâmetros probabilísticos de uma TPPO examinamos a estrutura corrente e buscamos as distribuições de probabilidade que maximizam uma determinada função de avaliação probabilística. Os algoritmos de aprendizado de parâmetros propostos na literatura tais como o EM [Kersting and De Raedt, 2002], [Koller and Pfeffer, 1997] e o gradiente [Kersting and De Raedt, 2001] podem ser utilizados.

1) *Revisão da estrutura:* Assim como em revisão de teoria de primeira-ordem, na revisão de TPPO é preciso identificar os pontos falhos. Os pontos que falharam na cobertura (prova) de algum exemplo consideramos como pontos de revisão lógicos e os pontos que cobriram os exemplos, mas não os classificaram corretamente, ou seja cobriram com probabilidade baixa, consideramos como pontos de revisão probabilísticos. Para os primeiros são aplicados operadores de generalização e para os segundos operadores de especialização.

Como não existem exemplos negativos os operadores de especialização não seriam necessários, mas eles devem ser aplicados, pois podem melhorar a função de avaliação probabilística. Eles são os mesmos descritos na seção II-A sendo que o operador de adição de antecedentes especializa a regra enquanto melhorar a função de avaliação probabilística. Para ambos os operadores deve ser mantida a prova dos exemplos.

Os operadores para generalização também são os mesmos descritos na seção II-A com exceção do que adiciona uma regra, pois o operador de especialização para adição de antecedentes foi alterado conforme mencionado acima.

Toda vez que um operador é aplicado a um determinado ponto de revisão os parâmetros probabilísticos são aprendidos e o operador é avaliado pela função de avaliação probabilística.

III. FUNÇÕES DE AVALIAÇÃO PROBABILÍSTICAS

Como foi previamente mencionado quatro funções de avaliação são consideradas:

1. Verossimilhança - a função de verossimilhança é definida como:

$$L(H : \mathbf{B}) = \mathbf{P}(\mathbf{B}|H, \Theta) \quad (1)$$

onde H é a TPPO corrente, ou seja, é a teoria resultante após a aplicação de um operador de revisão, \mathbf{B} o conjunto de exemplos de treinamento e θ são as distribuições de probabilidade. Considerando que os exemplos são independentes, temos:

$$P(\mathbf{B}|\Theta, H) = \prod_{i=1}^m P(B_i|\Theta, H) \quad (2)$$

onde m é o número de exemplos no conjunto de treinamento. Por ser de mais fácil manipulação comumente utiliza-se $\log L(H : \mathbf{B})$:

$$LL(H : \mathbf{B}) = \sum_{i=1}^m \log P(B_i|\Theta, H) \quad (3)$$

A log verossimilhança negativa é uma medida padrão de erro de treinamento e é definida como: $NLL(H|B) = -LL(H|B)$. A log-verossimilhança pode ser vista como medindo quantos bits são necessários para descrever D , baseando-se na distribuição de probabilidade. Além disso, olhando do ponto de vista estatístico, quanto maior a log-verossimilhança, mais próximo H está de modelar a distribuição sobre os dados B [Friedman et al., 1997]. Devido a isso, essa função é amplamente utilizada no aprendizado de redes Bayesianas.

- Verossimilhança condicional - se a tarefa é de classificação, o ideal seria encontrar uma teoria com o menor erro de classificação possível. Em [Friedman et al., 1997] foi mostrado que maximizar a verossimilhança condicional (vide fórmula abaixo) da classe é equivalente a minimizar o erro de classificação. Isto ocorre porque na tarefa de classificação, somente a parcela da verossimilhança relativa a verossimilhança condicional da classe dados os atributos é relevante. Portanto esta função é preferível à verossimilhança em problemas de classificação.

$$CLL(H|B) = \sum_{i=1}^m \log P(y_i | x_{i,1}, \dots, x_{i,v-1}) \quad (4)$$

onde $B_i = \{y_i, x_{i,1}, \dots, x_{i,v-1}\}$ e y_i representa a classe no exemplo i . De forma semelhante a log verossimilhança condicional negativa pode ser definida como:

$$NCLL(H|B) = -CLL(H|B) \quad (5)$$

- Minimum Description Length (MDL) - Tanto a verossimilhança como a verossimilhança condicional são funções monotônicas quando adicionamos antecedentes em uma regra e por isso favorecerão teorias mais complexas. Para contornar este problema pode ser adicionada uma penalidade de complexidade à verossimilhança (ou à verossimilhança condicional), como no MDL [Lam and Bacchus, 1994].

$$MDL(H|B) = \frac{1}{2} m \log n + NLL(H|B) \quad (6)$$

Se desejado for é possível trocar a log verossimilhança por log verossimilhança condicional:

$$MDL_{CLL}(H|B) = \frac{1}{2} m \log n + NCLL(H|B) \quad (7)$$

- Akaike Information Criterion (AIC) - O método AIC [Stone, 1977] adiciona uma complexidade de penalidade menor à verossimilhança com o objetivo de evitar problemas de *underfitting* que podem surgir com o MDL.

$$AIC(H|B) = m \log e - LL(H|B) \quad (8)$$

Assim como no MDL é possível trocar a log verossimilhança por log verossimilhança condicional:

$$AIC_{CLL}(H|B) = m \log e + NCLL(H|B) \quad (9)$$

IV. RESULTADOS EXPERIMENTAIS

Nós aplicamos as quatro funções de avaliação probabilísticas citadas na seção III ao sistema de revisão e comparamos experimentalmente os resultados em três domínios. O primeiro é o domínio de Bongard¹. A base de dados é composta de 60 exemplos onde utilizamos o mesmo conceito mencionado em [Kersting and De Raedt, 2002]: "existe um triângulo em um círculo". Consideramos que 30% das variáveis aleatórias não são observadas. Os outros domínios considerados são o domínio da Família [Quinlan, 1990] e o domínio da Escola [Costa et al., 2003]. As bases de dados são observadas e têm 353 e 554 exemplos, respectivamente. Na tabela IV as TPPOs originais são mostradas. Para o domínio de Bongard esta TPPO1 foi utilizada para gerar os exemplos.

TABLE II
TPPOS ORIGINAIS

Domínio	TPPO1
bongard	class(A):-obj(A,B),in(A,B,C),obj(A,C) obj(A,B):-dom(A,B)
Família	wife(A, B) :- gender(A), married(A,B). husband(A,B) :- gender(A), married(A, B). mother(A, B) :- gender(A),parent(A, B). father(A, B) :- gender(A), parent(A, B). daughter(A, B) :- gender(A), parent(B, A). son(A, B) :- gender(A), parent(B, A). sister(A, B) :- gender(A), sibling(A,B). brother(A, B) :- gender(A), sibling(A, B). sibling(A,B) :- parent(C, A), parent(C, B), A \= B.
Escola	student_ranking(S) :- registration_grade(R), registration(R, C, S). course_rating(C) :- registration_satisfaction(R), registration(R, C, S). registration_satisfaction(R) :- registration_grade(R), course(C, P), professor_ability(P), registration(R, C, S). registration_grade(R) :- course_diffculty(C), registration(R, C, S), student_intelligence(S). professor_popularity(P) :- professor_ability(P).

Foram introduzidos alguns erros na TPPO1s, tais como exclusão de regra, exclusão de antecedente, adição de antecedentes, mudança de antecedente e mudança de variável, gerando teorias corrompidas (TPPO2s).

Como em [Baião et al., 2003], para evitar *overfitting* nós aplicamos k-fold validação cruzada, separando os dados em conjuntos disjuntos de treinamento e teste e t-fold validação cruzada para separar os dados de treinamento em conjuntos disjuntos de treinamento e validação [Kohavi, 1995], [Mitchell, 1997], com k=4 e t=3. Em cada k-fold a melhor TPPO (de acordo com a função de avaliação probabilística) é guardada e aplicada ao conjunto de teste.

As tabelas IV, IV e IV apresentam os resultados da utilização das funções de avaliação probabilísticas citadas na seção anterior, onde TPPO3 indica a teoria revista, VFAP representa o valor da função de avaliação probabilística, AL a Acurácia Lógica (porcentagem de exemplos cobertos) e AP a Acurácia Probabilística (porcentagem de exemplos

¹O domínio de Bongard foi desenvolvido por M. Bongard e é um problema conhecido em ILP

classificados corretamente). Os melhores valores de acurácia probabilística estão representados em negrito.

Quando a função era a log verossimilhança (LL) ou log verossimilhança condicional (CLL), o número de antecedentes das regras foi limitado a 3 no domínio de Bongard e família e 4 no domínio da Escola para evitar *overfitting* [Grossman and Domingos, 2004].

TABLE III
RESULTADOS EXPERIMENTAIS PARA O DOMÍNIO DE BONGARD

Função	Teoria	VFAP	AL(%)	AP(%)
NLL	TPPO2	0	0	0
	TPPO3	1.18	100	60
NCLL	TPPO2	0	0	0
	TPPO3	0.27	100	87
MDL_LL	TPPO2	0	0	0
	TPPO3	1.81	100	63
MDL_CLL	TPPO2	0	0	0
	TPPO3	4.8	100	67
AIC_LL	TPPO2	0	0	0
	TPPO3	5.26	100	73
AIC_CLL	TPPO2	0	0	0
	TPPO3	1.12	100	79

TABLE IV
RESULTADOS EXPERIMENTAIS PARA O DOMÍNIO DA FAMÍLIA

Função	Teoria	VFAP	AL(%)	AP(%)
NLL	TPPO2	3.55	38	18
	TPPO3	2.47	100	71
NCLL	TPPO2	3.17	36	16
	TPPO3	1.32	100	73
MDL_LL	TPPO2	5.64	38	16
	TPPO3	3.27	100	72
MDL_CLL	TPPO2	5.24	39	19
	TPPO3	2.59	100	76
AIC_LL	TPPO2	4.38	35	16
	TPPO3	2.49	100	73
AIC_CLL	TPPO2	4.20	36	15
	TPPO3	2.02	100	76

TABLE V
RESULTADOS EXPERIMENTAIS PARA O DOMÍNIO DA ESCOLA

Função	Teoria	VFAP	AL(%)	AP(%)
NLL	TPPO1	3.81	37	10
	TPPO2	2.41	100	53
NCLL	TPPO1	1.98	36	10
	TPPO2	1.40	100	70
MDL_LL	TPPO1	6.65	37	10
	TPPO2	3.33	100	53
MDL_CLL	TPPO1	4.65	39	10
	TPPO2	12.99	100	68
AIC_LL	TPPO1	4.32	40	10
	TPPO2	2.25	100	53
AIC_CLL	TPPO1	2.73	39	10
	TPPO2	6.86	100	67

Já que estamos considerando a log verossimilhança (condicional) negativa, a melhor TPPO é aquela que minimiza a função de avaliação probabilística.

A acurácia lógica é 100% em todos os domínios mostrando que as teorias revistas são consistentes com as bases de dados.

Além disso, a acurácia probabilística melhorou e o valor da função de avaliação probabilística foi minimizada na maioria das TPPOs, o qual é um bom resultado já que a classificação é a tarefa principal.

Como os resultados mostram, as funções de avaliação probabilísticas que usam log verossimilhança condicional retornaram melhor acurácia probabilística, inclusive, a NCLL foi a melhor escolha nos domínios de Bongard e Escola. Estes resultados confirmam que a utilização da log verossimilhança condicional quando o problema é de classificação é uma escolha mais adequada, também em abordagens de primeira ordem.

A limitação do número de antecedentes que uma regra pode ter mostrou ser uma boa escolha de penalidade de complexidade, o que é um resultado compatível com trabalhos anteriores [Grossman and Domingos, 2004]. O MDL, apesar de utilizado em muitos sistemas de aprendizado de máquina, não apresentou resultados significativos em comparação com as outras funções, sugerindo que esta função não é muito adequada para a tarefa de classificação probabilística de primeira ordem, o que é uma conclusão consistente com resultados anteriores voltados para a classificação utilizando redes Bayesianas proposicionais (por exemplo [Allen and Greiner, 2000], [?], [Grossman and Domingos, 2004]). Os resultados obtidos com a função AIC foram melhores do que os resultados do MDL, confirmando que a penalidade do AIC, por ser menor que a do MDL, favorece menos *underfitting*.

V. CONCLUSÃO

Em [Grossman and Domingos, 2004] foi mostrado que a função de log verossimilhança não era uma função adequada em tarefas de classificação Bayesiana proposicionais e por este motivo log verossimilhança condicional deveria ser utilizada. Nesse artigo comparamos os resultados experimentais obtidos quando utilizando diferentes funções de avaliação probabilísticas, incluindo log verossimilhança condicional em um sistema de revisão de teorias probabilísticas de primeira ordem. Neste trabalho os resultados experimentais também apontaram para uma melhora em acurácia probabilística quando utilizando log verossimilhança condicional como função de avaliação probabilística.

Como trabalho futuro pretendemos estender os resultados para outras bases de dados em particular [Baião et al., 2003]. É importante mencionar que como a área de aprendizado probabilístico relacional ainda é nova não existem bases de dados disponíveis.

VI. AGRADECIMENTOS

O primeiro autor é parcialmente financiado pela CAPES e os outros pelo CNPq.

REFERENCES

- [Allen and Greiner, 2000] Allen, T. and Greiner, R. (2000). Model selection criteria for learning belief nets: An empirical comparison. In *Proceedings of the 17th International Conference on Machine Learning (ICML-04)*, pages 1047–1054.

- [Baião et al., 2003] Baião, F., Mattoso, M., Shavlik, J., and Zaverucha, G. (2003). Applying theory revision to the design of distributed databases. In *Proceedings of the 13th Int. Conference on Inductive Logic Programming, LNAI 2835, Springer Verlag*, pages 57–74.
- [Buntine, 1991] Buntine, W. (1991). Theory refinement on bayesian networks. In *Proceedings 17th Conference Uncertainty in Artificial Intelligence*, pages 52–60, San Mateo, CA.
- [Costa et al., 2003] Costa, V., Page, D., Qazi, M., and Cussens., J. (2003). Clp(bn): Constraint logic programming for probabilistic knowledge. In *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)*, pages 517–524.
- [Friedman et al., 1997] Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29:131–163.
- [Friedman et al., 1999] Friedman, N., Getoor, L., Koller, D., et al. (1999). Learning probabilistic relational models. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 1300–1309, Stockholm, Sweden.
- [Garcez and Zaverucha, 1999] Garcez, A. and Zaverucha, G. (1999). The connectionist inductive learning and logic programming system. *Applied Intelligence*, 11:59–77.
- [Grossman and Domingos, 2004] Grossman, D. and Domingos, P. (2004). Learning bayesian network classifiers by maximizing conditional likelihood. In *Proceedings of the 21th International Conference on Machine Learning (ICML-04)*, pages 361–368.
- [Haddawy, 1999] Haddawy, P. (1999). An overview of some recent developments on bayesian problem solving techniques. *AI Magazine - Special issue on Uncertainty in AI*, 20(2):11–29.
- [Kersting and De Raedt,] Kersting, K. and De Raedt, L. Towards combining inductive logic programming with bayesian networks. In *Proceedings of the 12th Int. Conference on Inductive Logic Programming, LNAI 2157 Springer Verlag*.
- [Kersting and De Raedt, 2001] Kersting, K. and De Raedt, L. (2001). Bayesian logic programs. Technical Report 151, University of Freiburg, Institute for Computer Science, Freiburg, German. April.
- [Kersting and De Raedt, 2002] Kersting, K. and De Raedt, L. (2002). Basic principles of learning bayesian logic programs. Technical Report 174, University of Freiburg, Institute for Computer Science, Freiburg, German.
- [Kohavi, 1995] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1137–1145.
- [Koller, 1999] Koller, D. (1999). Probabilistic relational models. In *Proceedings of the 9th Int. Conference on Inductive Logic Programming, LNAI 1634, Springer Verlag*, pages 3–13.
- [Koller and Pfeffer, 1997] Koller, D. and Pfeffer, A. (1997). Learning probabilities for noisy first-order rules. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 1316–1323.
- [Lam and Bacchus, 1994] Lam, W. and Bacchus, F. (1994). Learning bayesian belief networks: an approach based on the mdl principle. *Computational Intelligence*, 10(4):269–293.
- [Mitchell, 1997] Mitchell, T. (1997). *Machine Learning*. McGraw-Hill, New York.
- [Muggleton, 1992] Muggleton, S. (1992). *Inductive logic programming*. Academic Press, New York.
- [Muggleton, 2002] Muggleton, S. (2002). Learning structure and parameters of stochastic logic programs. In *Proceedings of the 12th Int. Conference on Inductive Logic Programming, LNAI 2583, Springer Verlag*, pages 198–206.
- [Paes et al., 2005] Paes, A., Revoredo, K., Zaverucha, G., and Costa, V. (2005). Revisão de teorias probabilísticas de primeira-ordem. In *To appear in ENIA 2005*.
- [Poole, 1993] Poole, D. (1993). Probabilistic horn abduction and bayesian networks. *Artificial Intelligence*, 64(1):81–129.
- [Quinlan, 1990] Quinlan, J. (1990). Learning logical definitions from relations. *Machine Learning*, 5:239–266.
- [Ramachandran and Mooney, 1998] Ramachandran, S. and Mooney, R. (1998). Theory refinement of bayesian networks with hidden variables. In *Proceedings of the 15th International Conference on Machine Learning (ICML)*, pages 454–462.
- [Revoredo and Zaverucha, 2002] Revoredo, K. and Zaverucha, G. (2002). Revision of first-order bayesian classifiers. In *Proceedings of the 12th Int. Conference on Inductive Logic Programming, LNAI 2583, Springer Verlag*, pages 223–237.
- [Richards and Mooney, 1995] Richards, B. L. and Mooney, R. J. (1995). Automated refinement of first-order horn-clause domain theories. *Machine Learning*, 19:95–131.
- [Stone, 1977] Stone, M. (1977). An asymptotic equivalence of choice of model by cross-validation and akaike’s criterion. *Journal of the Royal Statistical Society series B*, 39:44–47.
- [Towell and Shavlik, 1994] Towell, G. and Shavlik, J. (1994). Knowledge-based artificial neural networks. *Artificial Intelligence*, 70(1–2):119–165.
- [Wogulis and Pazzani, 1993] Wogulis, J. and Pazzani, M. (1993). A methodology for evaluation theory revision systems: results with Audrey II. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1128–1134, Chambery, France.
- [Wrobel, 1996] Wrobel, S. (1996). First-order theory refinement. In Raedt, L. D., editor, *Advances in Inductive Logic Programming*, pages 14–33. IOS Press.