JOSÉ EDUARDO OCHOA LUNA

**LÓGICAS PROBABILÍSTICAS COM RELAÇÕES DE
INDEPENDÊNCIA: REPRESENTAÇÃO DE
CONHECIMENTO E APRENDIZADO DE MÁQUINA**

São Paulo
2011

JOSÉ EDUARDO OCHOA LUNA

**LÓGICAS PROBABILÍSTICAS COM RELAÇÕES DE
INDEPENDÊNCIA: REPRESENTAÇÃO DE
CONHECIMENTO E APRENDIZADO DE MÁQUINA**

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção do
Título de Doutor em Engenharia.

São Paulo
2011

JOSÉ EDUARDO OCHOA LUNA

**LÓGICAS PROBABILÍSTICAS COM RELAÇÕES DE
INDEPENDÊNCIA: REPRESENTAÇÃO DE
CONHECIMENTO E APRENDIZADO DE MÁQUINA**

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção do
Título de Doutor em Engenharia.

Área de concentração:
Engenharia de Controle e Automação
Mecânica

Orientador:
Prof. Dr. Fabio Gagliardi Cozman

São Paulo
2011

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, de julho de 2011.

Assinatura do autor _____

Assinatura do orientador _____

FICHA CATALOGRÁFICA

Luna, José Eduardo Ochoa

**Lógicas probabilísticas com relações de independência: representação de conhecimento e aprendizado de máquina / J.E.O. Luna. – ed.rev. – São Paulo, 2011.
132 p.**

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1.Inteligência artificial 2.Aprendizado computacional 3.Lógica 4.Probabilidade I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos II.t

Aos meus pais, Felicia e José Luis.

AGRADECIMENTOS

Agradeço a todos que contribuíram diretamente ou indiretamente na elaboração deste trabalho, especialmente:

- A Deus guia em cada passo.
- Ao Professor Fabio Cozman, pela oportunidade de realizar esta pesquisa, a orientação, ensinamentos e confiança durante todos estes anos.
- Aos meus pais, Felicia e José Luis, e irmãos: Nancy, Rocio, Margot e Luis. Pelo amor, e os ensinamentos que foram e serão sempre importantes na minha vida.
- A você, Nena, pela inspiração, carinho e companhia constantes.
- Aos colegas do Laboratório de Tomada de Decisão: Cassio, Ismênia, Waldemar, Daniel, Ricardo, Rodrigo, Denis, Victor; pela amizade, compartilhamento de experiências e pesquisas.
- À CAPES, pelo apoio financeiro.

RESUMO

A combinação de lógica e probabilidade (lógicas probabilísticas) tem sido um tópico bastante estudado nas últimas décadas. A maioria de propostas para estes formalismos pressupõem que tanto as sentenças lógicas como as probabilidades sejam especificadas por especialistas. Entretanto, a crescente disponibilidade de dados relacionais sugere o uso de técnicas de aprendizado de máquina para produzir sentenças lógicas e estimar probabilidades. Este trabalho apresenta contribuições em termos de representação de conhecimento e aprendizado. Primeiro, uma linguagem lógica probabilística de primeira ordem é proposta. Em seguida, três algoritmos de aprendizado de lógica de descrição probabilística *CRALC* são apresentados: um algoritmo probabilístico com ênfase na indução de sentenças baseada em classificadores Noisy-OR; um algoritmo que foca na indução de inclusões probabilísticas (componente probabilístico de *CRALC*); um algoritmo de natureza probabilística que induz sentenças lógicas ou inclusões probabilísticas. As propostas de aprendizado são avaliadas em termos de acurácia em duas tarefas: no aprendizado de lógicas de descrição e no aprendizado de terminologias probabilísticas em *CRALC*. Adicionalmente, são discutidas aplicações destes algoritmos em processos de recuperação de informação: duas abordagens para extensão semântica de consultas na Web usando ontologias probabilísticas são discutidas.

Palavras-chave: Lógica Probabilística. Aprendizado de Lógica de Descrição. Programação em Lógica Indutiva. Incerteza. Redes Bayesianas Relacionais.

ABSTRACT

The combination of logic and probabilities (probabilistic logics) is a topic that has been extensively explored in past decades. The majority of work in probabilistic logics assumes that both logical sentences and probabilities are specified by experts. As relational data is increasingly available, machine learning algorithms have been used to induce both logical sentences and probabilities. This work contributes in knowledge representation and learning. First, a first-order probabilistic logic is proposed. Then, three algorithms for learning probabilistic description logic CRALC are given: a probabilistic algorithm focused on learning logical sentences and based on Noisy-OR classifiers; an algorithm that aims at learning probabilistic inclusions (probabilistic component of CRALC) and; an algorithm that using a probabilistic setting, induces either logical sentences or probabilistic inclusions. Evaluation of these proposals has been performed in two situations: by measuring learning accuracy of both description logics and probabilistic terminologies. In addition, these learning algorithms have been applied to information retrieval processes: two approaches for semantic query extension through probabilistic ontologies are discussed.

Keywords: Probabilistic Logic. Description Logic Learning. Inductive Logic Programming. Uncertainty. Relational Bayesian Networks.

SUMÁRIO

.....

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

Lista de Símbolos

1	Introdução	p. 18
1.1	Objetivos	p. 19
1.2	Contribuições	p. 21
1.3	Organização do texto	p. 22
2	Fundamentos	p. 23
2.1	Linguagens formais	p. 23
2.1.1	Lógica de primeira ordem	p. 26
2.1.1.1	Semântica em lógica de primeira ordem	p. 27
2.1.2	Lógicas de descrição	p. 30
2.2	Modelos probabilísticos gráficos	p. 34
2.2.1	Conceitos preliminares	p. 35
2.2.2	Redes Bayesianas	p. 37

2.3	Considerações finais	p. 42
3	Lógicas probabilísticas e RPL	p. 44
3.1	Lógica probabilística clássica	p. 44
3.2	Lógica probabilística moderna	p. 47
3.3	Lógica probabilística proposicional	p. 49
3.4	Lógica probabilística relacional	p. 52
3.5	Considerações finais	p. 58
4	Lógicas de descrição probabilísticas e CR$\mathcal{ALC}$	p. 60
4.1	Lógicas de descrição probabilísticas	p. 60
4.2	Lógica de descrição CR $\mathcal{ALC}$	p. 62
4.3	Considerações finais	p. 66
5	Aprendizado de máquina de lógica probabilística	p. 68
5.1	Programação em lógica indutiva probabilística	p. 68
5.2	Aprendizado de lógica de descrição probabilística:CR $\mathcal{ALC}$	p. 73
5.3	Aprendizado das probabilidades em CR $\mathcal{ALC}$	p. 74
5.4	Aprendizado de CR $\mathcal{ALC}$: abordagem probabilística	p. 75
5.5	Algoritmo de aprendizado de inclusões probabilísticas	p. 80
5.5.1	Função de pontuação	p. 80
5.5.2	Aprendizado das inclusões probabilísticas	p. 80
5.6	Abordagem combinada	p. 83
5.6.1	A função de pontuação probabilística	p. 84
5.6.2	O algoritmo para aprender terminologias probabilísticas	p. 85
5.7	Consistência do aprendizado em CR $\mathcal{ALC}$	p. 87

5.7.1	Condições no algoritmo probabilístico	p. 88
5.7.2	Condições no algoritmo de inclusões probabilísticas	p. 88
5.7.3	Condições no algoritmo combinado	p. 89
5.8	Considerações finais	p. 89

6 Experimentos p. 91

6.1	Metodologia	p. 91
6.2	Dados	p. 93
6.2.1	Dados de repositórios	p. 93
6.2.2	Dados reais	p. 94
6.2.2.1	Conjunto de dados Lattes	p. 94
6.2.2.2	Conjunto de dados Wikipedia-YAGO	p. 95
6.3	Experimentos	p. 95
6.3.1	Experimentos com dados de repositórios	p. 96
6.3.2	Experimentos com dados reais	p. 97
6.3.2.1	Terminologia probabilística obtida pelo algoritmo Noisy-OR	p. 98
6.3.2.2	Terminologia probabilística obtida pelo algoritmo de inclusões probabilísticas	p. 100
6.3.2.3	Terminologias probabilísticas obtidas pelo algoritmo combinado	p. 103
6.4	Considerações finais	p. 106

7 Aplicações p. 107

7.1	Extensão semântica de consultas por meio de lógica de descrição probabilística	p. 107
7.1.1	Procedimento de busca	p. 108
7.1.2	Procedimento de extensão de consulta	p. 109

7.1.3	Procedimento de ranking	p. 109
7.1.4	Resultados preliminares	p. 109
7.2	Extensão semântica de consultas usando contexto semântico	p. 113
7.2.1	Procedimento de busca	p. 114
7.2.2	Extensão de consulta via contexto da consulta	p. 114
7.2.3	Resultados preliminares	p. 114
7.3	Considerações finais	p. 116

8 Conclusões p. 117

8.1	Lógica probabilística de primeira ordem	p. 117
8.2	Aprendizado de lógica de descrição probabilística	p. 118
8.3	Aplicações em Web semântica	p. 119
8.4	Trabalhos futuros	p. 120

Referências p. 122

Apêndice I – Inferências em RPL usando programação multilinear p. 131

LISTA DE FIGURAS

1	Cenário do mundo dos blocos (GENESERETH; NILSSON, 1987)	p. 28
2	Rede Bayesiana exemplo (NEAPOLITAN, 2003).	p. 40
3	Grafo da rede PPL exemplo. Ave (A), AveVoa (AV), AveAntartida (AA), Pinguim (P), Voa (V).	p. 51
4	Rede associada a RPL. Esquerda: grafo de independência dos nós lógicos (baseados nas relações instanciadas) e nós sentença. Direita: grafo com dependências entre sentenças e instanciações de suas relações.	p. 54
5	Grafo simplificado para o domínio Universidade.	p. 56
6	Grafo do domínio Universidade após processamento das variáveis livres.	p. 56
7	Uma versão proposicionalizada do domínio Universidade.	p. 58
8	Rede-t associada a ontologia probabilística.	p. 65
9	Modelo ICI.	p. 76
10	Consulta tradicional.	p. 111
11	Resultado final estendido.	p. 112
12	Resultados da consulta exata.	p. 115
13	Resultados da consulta aberta.	p. 115
14	Resultado final consolidado.	p. 116

LISTA DE TABELAS

1	Descrição de variáveis aleatórias no Exemplo 2.	p. 40
2	Probabilidade de um conceito particular (A , B ou C) dado a dificuldade do curso e a inteligência do estudante.	p. 55
3	Dados de repositórios.	p. 93
4	Conjunto de dados Lattes.	p. 94
5	Conjunto de dados Wikipedia-YAGO.	p. 96
6	Resultados do aprendizado de lógica de descrição. Acur. (Acurácia) e Comp. (comprimento da solução encontrada)	p. 96
7	Resultados do aprendizado $CR\mathcal{ALC}$ (Lattes). Acur. (acurácia) e # Nós (quantidade nós da rede Bayesiana relacional).	p. 98
8	Resultados do aprendizado $CR\mathcal{ALC}$ (Wikipedia-YAGO). Acur. (acurácia) e # Nós (quantidade nós da rede Bayesiana relacional).	p. 98

LISTA DE ABREVIATURAS

Abreviatura	Descrição
DL	lógica de descrição
DAG	grafo direcionado acíclico
PPL	lógica probabilística proposicional
RPL	lógica probabilística relacional
$CR\mathcal{ALC}$	$\mathcal{ALC}$ credal
RBN	rede Bayesiana relacional
rede-t	grafo direcionado acíclico associado a $CR\mathcal{ALC}$
ILP	programação em lógica indutiva
TAN	classificador probabilístico baseado em redes Bayesianas

LISTA DE SÍMBOLOS

Símbolo Descrição

$\mathcal{L}_{\mathcal{LP}}$	lógica proposicional
$\mathcal{V}(\cdot)$	função de valoração proposicional
$\models$	implicação lógica
Γ	teoria, conjunto de fórmulas proposicionais ou de primeira ordem
$\vdash$	dedução, sistema dedutivo
$\mathcal{P}$	conjunto de predicados
$\mathcal{F}$	símbolos função
$\mathcal{C}$	símbolos constantes em lógica de primeira ordem
ϕ	sentença / fórmula em lógica de primeira ordem
I	interpretação em lógica de primeira ordem
U	atribuição de variáveis em lógica de primeira ordem
$\models_I \phi$	interpretação, modelo de ϕ
$\models_I \Gamma$	interpretação, modelo de conjunto de sentenças
$\sqsubseteq$	subsunção de conceitos em lógica de descrição
$\equiv$	equivalência de conceitos em lógica de descrição
$\top$	conceito universal
$\perp$	conceito vazio
$\mathcal{T}$	terminologia em lógica de descrição
$\mathcal{I}$	interpretação em lógica de descrição
$\Delta^{\mathcal{I}}$	domínio de interpretação em lógica de descrição
Ω	espaço amostral
E	evento
X	variável aleatória
G	grafo direcionado acíclico
V	conjunto de variáveis aleatórias
$pa(X_i)$	variáveis pais de X_i

$P(X_i pa(X_i))$	probabilidade condicional de X_i dados seus pais
w	interpretação / mundo possível em lógica probabilística
θ	programa linear
$\mathcal{G}$	grafo direcionado acíclico em PPL e RPL
$\mathcal{G}'_D$	grafo proposicionalizado de $\mathcal{G}$ no domínio D
$\mathcal{L}$	lista de fórmulas em PPL e RPL
$\mathcal{A}$	lista de atribuições em PPL e RPL
$P(C D) = \alpha$	inclusão probabilística em $\text{CR}\mathcal{ALC}$
H	hipótese em ILP
B	conhecimento base em ILP
$\mathbb{E}$	exemplos em ILP
$\mathcal{K}$	base de conhecimento no aprendizado de lógica de descrição
$\mathcal{H}$	hipótese no aprendizado de lógica de descrição
$\mathcal{E}$	exemplos para o aprendizado de lógica de descrição
ρ	operador de refinamento no aprendizado de lógica de descrição

1 INTRODUÇÃO

Muitas tarefas do mundo real, desde identificar objetos em imagens a extrair fatos em textos, requerem raciocínio probabilístico sobre os objetos relacionados. O número de variáveis aleatórias para descrever tais cenários cresce com o número de objetos; desta forma, linguagens proposicionais como redes Bayesianas (PEARL, 1988) — que representam um conjunto fixo de variáveis aleatórias — são insuficientes. São necessárias linguagens probabilísticas nas quais vários objetos podem ser manipulados de forma compacta (MILCH; RUSSELL, 2007).

Lógicas probabilísticas são formalismos para representação de conhecimento que adicionam probabilidades a tipos diversos de lógicas. Do ponto de vista sintático uma lógica probabilística comporta as construções de uma dada lógica e também probabilidades associadas às fórmulas. Do ponto de vista semântico, há várias possibilidades para interpretação das probabilidades nas fórmulas (NILSSON, 1986). Esse formalismo composto permitiria tanto superar as limitações do determinismo associado à lógica quanto generalizar modelos probabilísticos (FAGIN; HALPERN; MEGIDDO, 1990). Na prática, porém, muitas dessas possíveis vantagens são atenuadas pela complexidade em conciliar os formalismos em uma única linguagem.

Apesar das constantes pesquisas o potencial das lógicas probabilísticas não tem sido totalmente explorado. Os principais motivos disto são (HAENNI *et al.*, 2008):

1. a diversidade de formalismos que foram propostos nas últimas décadas dificulta a tarefa de determinar qual deles se adequa melhor para um dado problema;
2. as lógicas probabilísticas são comumente difíceis de entender, e muito trabalho teórico é ainda necessário para determinar suas propriedades;
3. a complexidade computacional subjacente a muitas destas linguagens limita seu uso.

Algumas tentativas de superar essas limitações, principalmente a complexidade computacional, têm sido direcionadas a “estruturar” a lógica probabilística, por exemplo

mediante a introdução de julgamentos de independência e mecanismos de representação baseados na teoria de grafos, inspirados em redes Bayesianas e redes Bayesianas relacionais (ANDERSEN; HOOKER, 1994; JAEGER, 1997; OCHOA-LUNA; CAMPOS; COZMAN, 2007; CAMPOS; COZMAN; OCHOA-LUNA, 2009; COZMAN; POLASTRO, 2009). Procura-se transferir as excelentes propriedades computacionais destes modelos gráficos às lógicas probabilísticas.

Também na direção de encontrar linguagens relativamente simples e poderosas, lógicas de descrição probabilísticas surgiram como evolução de *lógicas de descrição* (BAADER; NUTT, 2003). Lógicas de descrição são usualmente fragmentos de lógica de primeira ordem que permitem especificar indivíduos, conceitos e relações binárias. Um dos domínios que emprega lógicas de descrição é a Web Semântica, área voltada à representação e manipulação de conteúdo Web na forma de ontologias (BIZER *et al.*, 2009).

Embora promissoras, a aplicabilidade das lógicas probabilísticas em domínios reais ainda é limitada porque existem dificuldades decorrentes na construção de modelos; mesmo para um especialista pode não ser uma tarefa trivial definir sentenças lógicas e probabilidades associadas. Por outro lado, a crescente disponibilidade de dados relacionais possibilita a real utilização de aprendizado para construção automática de sentenças e probabilidades em lógicas probabilísticas. No caso específico das lógicas de descrição, a quantidade de dados disponível na Internet tem propiciado a busca por métodos que de alguma forma permitam o aprendizado de ontologias.

1.1 Objetivos

O objetivo principal deste trabalho é desenvolver métodos para aprendizado da estrutura lógica (as sentenças) e de probabilidades de lógicas de descrição probabilísticas, quando dados relacionais estão disponíveis. Especificamente, pretende-se induzir, a partir de exemplos, definições de conceitos em lógica de descrição e suas probabilidades.

O aprendizado das sentenças lógicas proposto neste trabalho é influenciado por técnicas de programação em lógica indutiva (*Inductive Logic Programming*, ILP) (LAVRAC; DZEROSKI, 1994). ILP é uma área que combina aprendizado de máquina e programação em lógica, e cujos métodos têm sido aplicados com sucesso em diversos domínios, tais como bio-informática, processamento de linguagem natural e mineração de dados multirrelacional (DZEROSKI; LAVRAC, 2001). Um dos principais algoritmos em ILP é FOIL (QUINLAN, 1990; QUINLAN; CAMERON-JONES, 1993), no qual o aprendizado indutivo é realizado a

partir dos exemplos positivos e negativos associados a um predicado alvo de aprendizado. Em FOIL a construção da solução se baseia na definição de uma relação de cobertura que avalia uma hipótese de acordo com a quantidade de exemplos positivos “cobertos” por ela. Muitos dos algoritmos em ILP usam essa noção de cobertura e exploram extensões e heurísticas (DE RAEDT; DEHASPE, 1997).

O foco deste trabalho é aprendizado de lógica de descrição probabilística, em particular *CRALC* (COZMAN; POLASTRO, 2008, 2009). Três algoritmos para aprendizado de sentenças em *CRALC*, mencionados a seguir, constituem a contribuição fundamental deste trabalho.

1) Uma proposta inicial de aprendizado para *CRALC* foi apresentada em (OCHOA-LUNA; COZMAN, 2009). O foco do algoritmo é o aprendizado de sentenças usando métodos probabilísticos e adaptando métodos de ILP. A indução de sentenças é guiada pelo aprendizado de modelos de independência de influência causal (modelos ICI (VOMLEL, 2006)). O processo é visto como o problema de aprendizado de um classificador probabilístico que possibilita o discernimento entre exemplos positivos e negativos.

2) Uma segunda proposta focou no aprendizado das *inclusões probabilísticas* em *CRALC* (REVOREDO; OCHOA-LUNA; COZMAN, 2010). Basicamente, após a aplicação de algoritmos de aprendizado de lógica de descrição tradicionais (DL-Learner (LEHMANN; HITZLER, 2008b) ou DL-FOIL (FANIZZI; D’AMATO; ESPOSITO, 2008)) realiza-se a busca por inclusões probabilísticas (usualmente dadas por probabilidades condicionais entre conceitos) nos componentes da definição aprendida. Cada componente (ou combinação de componentes) aprendido na definição denota um candidato a elemento condicional na inclusão probabilística. De acordo com uma pontuação baseada em máxima verossimilhança escolhe-se a melhor inclusão probabilística a qual concorre com a definição determinística (sem probabilidades) para ser mantida ou não na terminologia (conjunto de sentenças) resultante.

3) A terceira proposta (OCHOA-LUNA; REVOREDO; COZMAN, 2010a) parte do pressuposto que tanto os componentes determinísticos quanto os probabilísticos são relevantes na definição de terminologias probabilísticas. Esta proposta pretende integrar a abordagem determinística e a probabilística em um único modelo. O argumento é que, em alguns casos, mesmo após a aplicação de algoritmos de aprendizado determinísticos, algumas definições permanecem inerentemente probabilísticas, e não conseguem cobrir todos os exemplos positivos. Para realizar aprendizado, o algoritmo proposto, de natureza probabilística, realiza a busca de conceitos determinísticos (adaptando um algoritmo tradi-

onal), e após uma série de iterações o conceito é considerado determinístico se a pontuação obtida estiver acima de um limiar dado; caso contrário é iniciada a busca por inclusões probabilísticas. Nesta segunda etapa, dados os k primeiros candidatos obtidos na busca anterior realiza-se o aprendizado de um classificador probabilístico TAN (FRIEDMAN; GEIGER; GOLDSZMIDT, 1997) para determinar as inclusões probabilísticas, sendo retornada a classe com o maior valor de probabilidade condicional no classificador aprendido.

Para demonstrar a utilidade das terminologias aprendidas, esses algoritmos foram usados em tarefas de recuperação de informação (MANNING; RAGHAVAN; SCHÜTZE, 2008). Dois tipos de problemas foram abordados: a extensão semântica e a identificação de contexto semântico em buscas na Web. Na primeira tarefa (OCHOA-LUNA; REVOREDO; COZMAN, 2010c), uma terminologia probabilística construída por meio de aprendizado foi usada para estender, com conteúdo semântico, buscas em textos de páginas HTML da plataforma Lattes. Partes do texto estão associadas (indexadas) a propriedades na terminologia probabilística, e constituem evidência na rede Bayesiana relacional que é gerada a partir da terminologia probabilística aprendida. Após inferências probabilísticas, são retornados os conceitos probabilisticamente mais relacionados com a consulta do usuário.

Na segunda tarefa (OCHOA-LUNA; REVOREDO; COZMAN, 2010b), textos de documentos são associados a propriedades na terminologia, e considera-se como contexto semântico de uma busca o conjunto de conceitos e relações associadas a consultas exatas (documentos que casam exatamente com as palavras chave da busca). Os resultados da busca referentes a consultas abertas (componentes da busca analisados de forma separada) são avaliados de acordo com sua aderência ao contexto semântico. Isto é, de acordo com sua aparição nas propriedades indexadas, cada termo da consulta é inserido como evidência na rede Bayesiana relacional (construída a partir da evidência dada pelo contexto semântico). De acordo com seu valor da probabilidade o resultado da consulta aberta é acrescentado ao conjunto de documentos associados ao contexto.

Em suma, esse trabalho procura dar contribuições do ponto de vista teórico (lógica probabilística de primeira ordem para representação de conhecimento) e do ponto de vista prático (algoritmos de aprendizado para aplicações em mineração de dados relacional e Web semântica).

1.2 Contribuições

Em síntese, este trabalho apresenta contribuições em três frentes:

- Proposta de uma linguagem para modelar problemas em lógica probabilística de primeira ordem.
- Proposta de três algoritmos para aprendizado usando a lógica de descrição probabilística *CRALC*: a) uma abordagem inteiramente probabilística visando o aprendizado da estrutura lógica; b) uma abordagem visando o aprendizado de inclusões probabilísticas; c) uma abordagem combinada onde tanto a estrutura e as probabilidades são aprendidas em um algoritmo que combina as vantagens do aprendizado de estruturas determinísticas e o aprendizado de classificadores probabilísticos. Em relação aos algoritmos de aprendizado de lógica de descrição probabilística *CRALC*, não existem outras propostas disponíveis na literatura.
- Aplicações dos algoritmos de aprendizado em tarefas de recuperação de informação.

1.3 Organização do texto

O Capítulo 2 realiza uma revisão de conceitos relacionados a lógica e probabilidade. O Capítulo 3 apresenta a definição e revisão bibliográfica de lógica probabilística. Em particular é apresentada a lógica probabilística de primeira ordem que faz parte da contribuição desse trabalho (Seção 3.2). O Capítulo 4 é devotado ao estudo da lógica de descrição probabilística e a linguagem *CRALC* foco dos algoritmos de aprendizado. O Capítulo 5 aborda modelos de aprendizado de lógicas probabilísticas baseados em técnicas de programação em lógica indutiva. Este capítulo apresenta as maiores contribuições desse trabalho, os três algoritmos para aprendizado de lógica de descrição probabilística (Seções 5.4, 5.5 e 5.6). No Capítulo 6 são relatados os resultados empíricos obtidos da comparação dos algoritmos que foram propostos. Os resultados procuram também demonstrar a aplicabilidade dos algoritmos em domínios de recuperação de informação, tema do Capítulo 7. Finalmente, o Capítulo 8 apresenta as considerações finais do trabalho, uma revisão das contribuições do trabalho e a discussão de trabalhos futuros.

2 FUNDAMENTOS

Neste capítulo são descritos formalismos para representação de conhecimento baseados em duas abordagens. A primeira abordagem, determinística, foca em linguagens formais lógicas. A segunda abordagem, probabilística, foca em modelos probabilísticos gráficos.

Esse capítulo apresenta as principais características de cada formalismo e familiariza o leitor com a notação e definições importantes. Embora sucinta, essa revisão deve ser útil para entender a abordagem combinada da lógica probabilística, tema do próximo capítulo.

O capítulo está dividido em duas seções bem definidas. A Seção 2.1 aborda as linguagens formais lógicas, incluindo a lógica proposicional, a lógica de primeira ordem e, finalmente, as lógicas de descrição. A Seção 2.2 discute a representação de incerteza através da ótica dos modelos probabilísticos gráficos. Representação e raciocínio sob incerteza são abordados mediante o estudo das redes Bayesianas. Considerações finais são apresentadas na Seção 2.3.

2.1 Linguagens formais

Linguagens formais são objetos matemáticos cujas regras de formação são precisamente definidas e às quais se pode atribuir um único sentido, sem ambigüidade. A seguir é revista a lógica proposicional. A maior parte desta seção foi resumida de (SILVA; FINGER; MELO, 2006).

A *linguagem proposicional* tem uma expressividade limitada, porém permite expressar uma série de relações lógicas interessantes. Nesse contexto, uma *proposição* é um enunciado ao qual se pode atribuir um *valor verdade* (verdadeiro ou falso).

A linguagem proposicional não permite expressar relações sobre elementos de um conjunto, como as noções de todos, algum ou nenhum. Tais relações são chamadas de *quantificadoras* e são encontradas no estudo da *lógica de primeira ordem*.

Para apresentar uma linguagem formal é preciso fornecer os componentes básicos da linguagem: o *alfabeto* e as *regras de formação* da linguagem (ou gramática).

No caso da lógica proposicional, o alfabeto é composto pelos seguintes elementos:

- (i) um conjunto infinito e contável de símbolos proposicionais, também chamados de átomos ou de variáveis proposicionais;
- (ii) o conectivo unário $\neg$ (negação);
- (iii) os conectivos binários $\wedge$ (conjunção), $\vee$ (disjunção) e $\rightarrow$ (implicação);
- (iv) elementos de pontuação, os parênteses ‘(’ e ‘)’.

Os elementos da linguagem $\mathcal{L}_{\mathcal{LP}}$ da lógica proposicional são chamados de *fórmulas* (ou fórmulas bem formadas). O conjunto $\mathcal{L}_{\mathcal{LP}}$ de fórmulas pode ser definido por indução, satisfazendo as seguintes regras de formação:

- caso básico: todos os símbolos proposicionais estão em $\mathcal{L}_{\mathcal{LP}}$. Os símbolos proposicionais são chamados de fórmulas atômicas ou átomos.
- caso indutivo 1: Se $A \in \mathcal{L}_{\mathcal{LP}}$, então $\neg A \in \mathcal{L}_{\mathcal{LP}}$.
- caso indutivo 2: Se $A, B \in \mathcal{L}_{\mathcal{LP}}$, então $(A \wedge B) \in \mathcal{L}_{\mathcal{LP}}$, $(A \vee B) \in \mathcal{L}_{\mathcal{LP}}$, $(A \rightarrow B) \in \mathcal{L}_{\mathcal{LP}}$.

Em geral, letras minúsculas p, q, r, s , representam os símbolos atômicos. As letras maiúsculas A, B, C, D representam fórmulas.

O estudo da semântica da lógica proposicional consiste em atribuir *valores verdade* às fórmulas da linguagem. Há apenas dois valores verdade: verdadeiro e falso (representados por 1 e 0).

Inicialmente, atribui-se valores verdade para os símbolos proposicionais por meio de uma função de *valoração*. Uma valoração proposicional $\mathcal{V}$ é uma função $\mathcal{V} : S \rightarrow \{0, 1\}$ que mapeia cada símbolo proposicional em S em um valor verdade. Essa valoração apenas diz quais são verdadeiros e quais falsos. Em seguida, estende-se a valoração para todas as formas da linguagem da lógica proposicional, de forma a obter uma valoração $\mathcal{V} : \mathcal{L}_{\mathcal{LP}} \rightarrow \{0, 1\}$. Essa extensão da valoração é feita sobre a estrutura das fórmulas.

Em geral, as fórmulas podem ser classificadas de acordo com o seu comportamento diante de todas as valorações possíveis de seus átomos.

- Uma fórmula A é dita *satisfazível* se existe uma valoração $\mathcal{V}$ de seus átomos tal que $\mathcal{V}(A) = 1$.
- Uma fórmula A é dita *insatisfazível* se toda valoração $\mathcal{V}$ de seus átomos é tal que $\mathcal{V}(A) = 0$.
- Uma fórmula A é dita *válida* ou uma *tautologia* se toda valoração $\mathcal{V}$ de seus átomos é tal que $\mathcal{V}(A) = 1$.
- Uma fórmula é dita *falsificável* se existe uma valoração $\mathcal{V}$ de seus átomos tal que $\mathcal{V}(A) = 0$.

Um dos grandes desafios da computação é encontrar métodos eficientes para decidir se uma fórmula é satisfazível/insatisfazível ou se é válida/falsificável.

Um dos primeiros métodos propostos na literatura para verificação de satisfazibilidade e validade de fórmulas é o método da tabela de verdade. A tabela de verdade é um método exaustivo de geração de valorações para uma dada fórmula A . Na prática, porém é um método inviável do ponto de vista computacional dada a quantidade exponencial de valorações possíveis para uma fórmula com N átomos.

Um dos temas mais estudados da lógica é a *consequência lógica*, isto é, determinar se uma fórmula é consequência de outra fórmula ou de um conjunto de fórmulas. No caso da lógica proposicional, a resposta é dada em termos de valorações. Diz-se que uma fórmula B é consequência lógica de outra fórmula A , representada por $A \models B$, se toda valoração v que satisfaz A também satisfaz B . Essa definição permite que B seja satisfeito por valorações que não satisfazem A . Nesse caso, também se diz que A *implica logicamente* B . Podem ser usadas as tabelas de verdade para verificar a consequência lógica.

Um conjunto de fórmulas é chamado de *teoria*. Uma fórmula A é consequência lógica de um conjunto de fórmulas Γ , representado por $\Gamma \models A$, se toda valoração v que satisfaz a todas as fórmulas de Γ também satisfaz A .

A partir de um conjunto de fórmulas Γ , pode-se *inferir* novas fórmulas que sejam consequência lógica de Γ . Essa é a tarefa de um sistema dedutivo.

Definição 1 (Inferência Lógica). *Inferir consiste em derivar ou deduzir as consequências lógicas de um conjunto de fórmulas, chamado de teoria, mediante o uso de um sistema dedutivo.*

Quando um sistema dedutivo infere uma fórmula A a partir de uma teoria Γ , então é representado por $\Gamma \vdash A$.

Existem vários procedimentos distintos que permitem realizar uma inferência, e cada procedimento dá origem a um distinto sistema dedutivo (também chamado de sistema de inferência). Não é desejável que o sistema de dedução produza fórmulas que não sejam consequência lógica da teoria usada como hipótese. Diz-se que um sistema dedutivo $\vdash$ é *correto* se isso nunca ocorre, ou seja, se $\Gamma \vdash A$ somente se $\Gamma \models A$.

Por outro lado, deseja-se que um sistema dedutivo consiga inferir todas as possíveis consequências lógicas de uma teoria. Diz-se que um sistema dedutivo é *completo* se ele for capaz de realizar todas essas inferências, ou seja, se sempre tem-se $\Gamma \vdash A$ se $\Gamma \models A$.

Exemplos de tipos de sistemas dedutivos completos e corretos incluem: axiomatizações, sistemas de dedução natural e o método dos tableaux analíticos (SILVA; FINGER; MELO, 2006).

2.1.1 Lógica de primeira ordem

Uma linguagem formal mais expressiva que a lógica proposicional é a lógica de primeira ordem. Essa seção foi resumida de (HUTH; RYAN, 2004) e (GENESERETH; NILSSON, 1987). Considere a sentença declarativa:

$$\text{Todo estudante é mais novo que algum instrutor.} \quad (2.1)$$

Em lógica proposicional, esta asserção poderia ser identificada com um átomo proposicional p . Porém, essa proposição não consegue refletir detalhes da estrutura lógica desta sentença. Não é possível saber se a sentença é sobre um estudante, instrutor, alguém mais novo, etc. Essas são propriedades de algum tipo, e é necessário um mecanismo para expressar essas propriedades com suas relações lógicas e suas dependências.

Os *predicados* são usados para esse propósito. Por exemplo, $E(\text{pedro})$ denota que Pedro é um estudante e $I(\text{paul})$ denota que Paul é um instrutor. Do mesmo modo, $N(\text{pedro}, \text{paul})$ denota que Pedro é mais novo que Paul. Os símbolos E , I e N são chamados de predicados. Os predicados não são suficientes para expressar a sentença na Expressão (2.1). Não se deseja escrever todas as instâncias $E(\cdot)$, onde $\cdot$ é substituído por cada nome de estudante. Nesse caso, usa-se o conceito de *variável*. Variáveis são escritas $u, v, w, x, \dots$ e podem assumir diferentes valores concretos.

Ainda é necessário “todos” e “algum”; para tanto são usados *quantificadores* $\forall$ (para todos) e $\exists$ (existe ou para algum) os quais sempre estão associados com variáveis, como em $\forall x$ (para todo x). Agora a sentença pode ser escrita de maneira simbólica

$$\forall x(E(x) \rightarrow (\exists y(I(y) \wedge N(x, y)))).$$

O alfabeto em lógica de primeira ordem consiste de três conjuntos: um conjunto de símbolos predicados $\mathcal{P}$, um conjunto de símbolos função $\mathcal{F}$ e um conjunto de símbolos constantes $\mathcal{C}$. Cada símbolo predicado e função tem associado uma aridade, o número de argumentos esperado.

Definição 2 (Termos). *Os termos são definidos como segue*

- Qualquer variável é um termo.
- Se $c \in \mathcal{F}$ é uma função nula (de aridade 0), então c é um termo.
- Se $t_1, t_2, \dots, t_n$ são termos e $f \in \mathcal{F}$ tem aridade $n > 0$, então $f(t_1, t_2, \dots, t_n)$ é um termo.

Definição 3 (Fórmulas). *Define-se o conjunto de fórmulas sobre $(\mathcal{F}, \mathcal{P})$ de maneira indutiva*

- Se $P \in \mathcal{P}$ é um símbolo predicado de aridade $n \geq 1$, e se $t_1, t_2, \dots, t_n$ são termos sobre $\mathcal{F}$, então $P(t_1, t_2, \dots, t_n)$ é uma fórmula.
- Se ϕ é uma fórmula, então $(\neg\phi)$ também é.
- Se ϕ e φ são fórmulas, então também são $(\phi \wedge \varphi)$, $(\phi \vee \varphi)$ e $(\phi \rightarrow \varphi)$.
- Se ϕ é uma fórmula e x uma variável, então $(\forall x\phi)$ e $(\exists x\phi)$ são fórmulas.

2.1.1.1 Semântica em lógica de primeira ordem

A semântica em lógica de primeira ordem depende da noção de *interpretação*. Uma interpretação é um mapeamento entre elementos da linguagem e elementos da conceitualização. O mapeamento é representado pela função $I(\cdot)$. Usualmente denota-se $I(\sigma)$ por σ^I . O símbolo $|I|$ denota o universo de discurso (o domínio). Para I ser uma interpretação, deve satisfazer as seguintes propriedades:

- Se σ é um objeto, então $\sigma^I \in |I|$.

- Se π é uma função de aridade n , então $\pi^I : |I|^n \rightarrow |I|$.
- se ρ é uma relação de aridade n então $\rho^I \subseteq |I|^n$.

Figura 1: Cenário do mundo dos blocos (GENESERETH; NILSSON, 1987)

Exemplo 1. *Um exemplo de interpretação considerando o mundo dos blocos da Figura 1 é dado a seguir. O mapeamento corresponde à interpretação usual destes símbolos:*

$$\begin{aligned}
 A^I &= a \\
 B^I &= b \\
 C^I &= c \\
 D^I &= d \\
 E^I &= e \\
 Hat^I &= \{\langle b, a \rangle, \langle c, b \rangle, \langle e, d \rangle\} \\
 On^I &= \{\langle a, b \rangle, \langle b, c \rangle, \langle d, e \rangle\} \\
 Above^I &= \{\langle a, b \rangle, \langle b, c \rangle, \langle a, c \rangle, \langle d, e \rangle\} \\
 Table^I &= \{c, e\} \\
 Clear^I &= \{a, d\}
 \end{aligned}$$

Uma atribuição de variáveis U é uma função das variáveis de uma linguagem sobre os objetos do universo de discurso. Usa-se a abreviação σ^U para $U(\sigma)$. Por exemplo, $x^U = a, y^U = b$ são atribuições dos blocos a e b a variáveis x, y . Uma interpretação I e uma atribuição de variáveis U podem ser combinadas em uma atribuição conjunta T_{IU} que se aplica a termos em geral.

Diferentemente da lógica proposicional, as noções de interpretação e atribuições de variáveis são importantes porque permitem definir a noção de *satisfação* em lógica de primeira ordem. O fato de uma sentença ϕ ser satisfeita por uma interpretação I e uma atribuição de variáveis U é escrita como $\models_I \phi[U]$; afirma-se então que a sentença ϕ é verdadeira relativa à interpretação I e à atribuição U . Sob algumas interpretações, uma sentença pode ser verdadeira, enquanto sob outras pode ser falsa.

A satisfazibilidade de sentenças lógicas depende dos operadores lógicos envolvidos. Assim:

- $\models_I (\neg\phi)[U]$ se e somente se $\not\models_I \phi[U]$
- $\models_I (\phi_1 \wedge \dots \wedge \phi_n)[U]$ se e somente se $\models_I \phi_i[U]$ para todos $i = 1, \dots, n$.
- $\models_I (\phi_1 \vee \dots \vee \phi_n)[U]$ se e somente se $\models_I \phi_i[U]$ para algum $i, 1 \leq i \leq n$.
- $\models_I (\phi \rightarrow \psi)[U]$ se e somente se $\not\models_I \phi[U]$ ou $\models_I \psi[U]$.
- $\models_I (\phi \leftarrow \psi)[U]$ se e somente se $\models_I \phi[U]$ ou $\not\models_I \psi[U]$.
- $\models_I (\phi \leftrightarrow \psi)[U]$ se e somente se $\models_I (\phi \rightarrow \psi)[U]$ e $\models_I (\phi \leftarrow \psi)[U]$.

E no caso das sentenças quantificadas:

- $\models_I (\forall v\phi)[U]$ se e somente se para todo $d \in |I|$ é o caso que $\models_I \phi[V]$ onde $V(v) = d$ e $V(\mu) = U(\mu)$ para $\mu \neq v$.
- $\models_I (\exists v\phi)[U]$ se e somente se para algum $d \in |I|$ é o caso que $\models_I \phi[V]$ onde $V(v) = d$ e $V(\mu) = U(\mu)$ para $\mu \neq v$.

Se uma interpretação I satisfaz uma sentença ϕ para todas as atribuições das variáveis então I é chamada de *modelo* de ϕ , isto é, $\models_I \phi$.

Uma sentença é dita *satisfazível* se e somente se existe alguma interpretação e atribuição de variáveis que a satisfazem; caso contrário a sentença é *insatisfazível*. Uma sentença é *válida* se e somente se é satisfeita por toda interpretação e atribuição de variáveis. Sentenças válidas são aquelas que são verdadeiras em virtude de sua forma lógica e não fornecem informação acerca do domínio sendo descrito. A sentença $P(A) \vee \neg P(A)$ é válida, porque quaisquer interpretação satisfaz $P(A)$ ou $\neg P(A)$.

Estas definições podem se estendidas a conjuntos de sentenças. Um conjunto de sentenças Γ é satisfeito por uma interpretação I e uma atribuição de variáveis U (denotada como $\models_I \Gamma[U]$) se e somente se cada membro de Γ é satisfeito por I e U . Uma interpretação é um *modelo* de um conjunto de sentenças Γ (denotada $\models_I \Gamma$) se e somente se é um modelo de todo membro do conjunto. Um conjunto de sentenças é satisfazível se e somente se existe uma interpretação e atribuição de variáveis que satisfaz todos seus elementos, caso contrário, é insatisfazível. Um conjunto de sentenças é válido se e somente se todo elemento é válido.

Em lógica de primeira ordem existe um procedimento de *inferência* baseado em uma poderosa e simples regra de inferência: o *princípio de resolução*. Por meio deste procedimento pode ser determinada a satisfazibilidade de uma base de conhecimento, ou podem ser respondidas consultas do tipo verdadeiro-falso. Maiores detalhes desse e outros procedimentos de inferência para lógica proposicional e lógica de primeira ordem podem ser encontrados em (CHANDRU; HOOKER, 1999; GENESERETH; NILSSON, 1987; HUTH; RYAN, 2004).

A versatilidade da lógica de primeira ordem tem um preço. Parte desse preço está associado ao problema de determinar a satisfazibilidade, o qual não é somente difícil mas insolúvel em geral (CHANDRU; HOOKER, 1999). Em lógica proposicional, há um procedimento finito para verificar se uma fórmula é ou não satisfazível: tentar todas as atribuições de valores de verdade das proposições atômicas. Lógica de primeira ordem é *indecidível*, o que significa que não há tal procedimento que funcione para toda fórmula. Embora a lógica de primeira ordem seja indecidível, é *semidecidível* o que significa que há um procedimento que sempre terminará com uma prova de insatisfazibilidade caso a fórmula seja insatisfazível. Se a fórmula for satisfazível, porém, os procedimentos podem nunca terminar.

2.1.2 Lógicas de descrição

Uma lógica de descrição (DL) é um formalismo de representação de conhecimento que descreve o conhecimento de um domínio de aplicação (o “mundo”) por meio da definição de conceitos relevantes (sua terminologia), e então usa esses conceitos para especificar propriedades de objetos e indivíduos que ocorrem no domínio (a descrição do mundo) (BAADER; NUTT, 2003). DLs representam conhecimento em termos de *objetos*, *conceitos* e *papéis*.

Em uma base de conhecimento qualquer existe uma clara distinção entre conhecimento intensional, ou geral, acerca do domínio, e conhecimento extensional, o qual é específico a um problema particular. Analogamente, uma base de conhecimento DL típica é dividida em dois componentes, uma TBox e uma ABox. A TBox contém conhecimento intensional na forma de uma terminologia e é construída mediante declarações que descrevem propriedades gerais de conceitos. Por sua vez, a ABox contém conhecimento extensional também chamado de conhecimento assertional, conhecimento que é específico a indivíduos do domínio de discurso.

Descrições elementares são *conceitos atômicos* e *papéis atômicos*. Descrições com-

plexas podem ser construídas indutivamente com *construtores de conceitos*. Na notação abstrata dada a seguir usa-se A e B para conceitos atômicos, r para papéis atômicos, e C e D representam descrições de conceitos. Linguagens de descrição se diferenciam de acordo com os construtores que disponibilizam. Por exemplo a linguagem $\mathcal{AL}$ é uma linguagem mínima de interesse prático. As outras linguagens desta família são extensões de $\mathcal{AL}$.

A título de exemplo, o alfabeto da linguagem $\mathcal{AL}$ é mostrado por meio de uma regra sintática:

$$\begin{array}{ll}
 C, D \rightarrow A | & (\text{conceito atômico}) \\
 & \top | \quad (\text{conceito universal}) \\
 & \perp | \quad (\text{conceito vazio}) \\
 & \neg A | \quad (\text{negação atômica}) \\
 & C \sqcap D | \quad (\text{interseção}) \\
 & \forall r. C | \quad (\text{restrição de valor}) \\
 & \exists r. \top | \quad (\text{quantificação existencial limitada})
 \end{array}$$

No caso mais geral, *axiomas terminológicos* tem a forma:

$$C \sqsubseteq D (r \sqsubseteq s) \text{ ou } C \equiv D (r \equiv s)$$

onde C, D são conceitos (e r, s são papéis). Axiomas do primeiro tipo são chamados de *inclusões*, enquanto que axiomas do segundo tipo são chamados de *igualdades*.

Uma igualdade cujo lado esquerdo é um conceito atômico é uma *definição*. Definições são usadas para introduzir *nomes simbólicos* para descrições complexas. Por exemplo por intermédio do axioma $\text{Mae} \equiv \text{Mulher} \sqcap \exists \text{temFilho. Pessoa}$ associa-se a descrição do lado direito ao nome **Mae**.

Um conjunto finito de definições $\mathcal{T}$ é uma *terminologia* ou TBox se nenhum nome simbólico é definido mais de uma vez, isto é, para cada conceito atômico A apenas há um axioma em $\mathcal{T}$ cujo lado esquerdo é A .

Por outro lado, conhecimento extensional sobre o domínio de interesse é contido na ABox, nas asserções sobre indivíduos. Por exemplo:

$$\text{Feminino} \sqcap \text{Pessoa}(\text{Ana})$$

denota que o indivíduo **Ana** é uma pessoa do sexo feminino. Dada a definição de mulher ($\text{Mulher} \equiv \text{Pessoa} \sqcap \text{Feminino}$), pode ser afirmado que **Ana** é uma instância do conceito

Mulher. De maneira similar:

temFilho(Ana, Juan)

denota que Ana tem Juan como filho.

Para definir uma semântica formal de conceitos em linguagens como $\mathcal{AL}$, consideram-se *interpretações* $\mathcal{I}$ que consistem de um conjunto não vazio $\Delta^{\mathcal{I}}$ (o domínio da interpretação) e uma função interpretação, que atribui a cada conceito atômico A um conjunto $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ e a cada papel atômico r uma relação binária $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

Em geral, a semântica de conceitos identifica linguagens de descrição como fragmentos de lógica de primeira ordem. Dado que uma interpretação $\mathcal{I}$ atribui, respectivamente, a cada conceito e papel atômico, uma relação unária e binária sobre $\Delta^{\mathcal{I}}$. Conceitos e papéis atômicos podem ser vistos como predicados unários e binários. Deste modo, qualquer conceito C pode ser traduzido efetivamente em uma fórmula de lógica de primeira ordem $\phi_C(x)$ com uma variável livre x tal que para cada interpretação $\mathcal{I}$ o conjunto de elementos de $\Delta^{\mathcal{I}}$ satisfazendo $\phi_C(x)$ é exatamente $C^{\mathcal{I}}$: um conceito atômico A é traduzido na fórmula $A(x)$; os construtores interseção, união e negação são traduzidos em conjunção, disjunção e negação lógicos; se C é traduzido a $\phi_C(x)$ e r é um papel atômico então a restrição de valor e a quantificação existencial são capturadas pelas fórmulas:

$$\begin{aligned}\phi_{\exists r.C}(y) &= \exists x \, r(y, x) \wedge \phi_C(x), \\ \phi_{\forall r.C}(y) &= \forall x \, r(y, x) \rightarrow \phi_C(x),\end{aligned}$$

onde y é uma nova variável.

Inferência em lógicas de descrição

Uma base de conhecimento formada por uma TBox e ABox tem uma semântica que a torna equivalente a um conjunto de axiomas em lógica de primeira ordem. Deste modo, como qualquer outro conjunto de axiomas, contém conhecimento implícito que pode ser feito explícito por meio de inferência. Os diferentes tipos de raciocínio realizados por um sistema DL são definidos como inferências lógicas.

Quando se modela um domínio, uma terminologia, denotada por $\mathcal{T}$, se constrói pela definição de novos conceitos, possivelmente em termos de outros que foram definidos com antecedência. Durante este processo, é importante determinar se um novo conceito definido faz sentido ou é contraditório. Do ponto de vista lógico, um conceito faz sentido se existe alguma interpretação que satisfaz os axiomas de $\mathcal{T}$ (isto é, um modelo de $\mathcal{T}$) de tal forma que o conceito denota um conjunto não vazio nesta interpretação. Um conceito

com esta propriedade é dito de *satisfazível* em relação a $\mathcal{T}$ e *insatisfazível* caso contrário.

Definição 4. *Inferência em uma terminologia $\mathcal{T}$ TBox consiste em verificar a satisfazibilidade dos seus conceitos.*

Muitas outras inferências para conceitos podem ser reduzidas a satisfazibilidade. Por exemplo, para verificar se um modelo de domínio é correto ou otimizar consultas que são formuladas como conceitos, pode ser de interesse saber se algum conceito é mais geral que outro: este é o problema da *subsunção*. Um conceito C é *subsumido* por outro conceito D se em cada modelo de $\mathcal{T}$ o conjunto denotado por C é um subconjunto do conjunto denotado por D . Algoritmos que verificam subsunção são também utilizados para organizar conceitos de uma TBox em uma taxonomia de acordo com a sua generalidade. Outras relações interessantes entre conceitos são *equivalência* e *disjunção*.

Estas propriedades são formalmente definidas a seguir, seja $\mathcal{T}$ uma TBox.

- **Satisfazibilidade** Um conceito C é *satisfazível* em relação a $\mathcal{T}$ se existe um modelo $\mathcal{I}$ de $\mathcal{T}$ tal que $C^{\mathcal{I}}$ é não vazio. Neste caso diz-se que $\mathcal{I}$ é um *modelo* de C .
- **Subsunção** Um conceito C é *subsumido* por um conceito D em relação a $\mathcal{T}$ se $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ para cada modelo $\mathcal{I}$ de $\mathcal{T}$. Neste caso denota-se $C \sqsubseteq_{\mathcal{T}} D$ ou $\mathcal{T} \models C \sqsubseteq D$.
- **Equivalência** Dois conceitos C e D são *equivalentes* em relação a $\mathcal{T}$ se $C^{\mathcal{I}} = D^{\mathcal{I}}$ para cada modelo $\mathcal{I}$ de $\mathcal{T}$. Neste caso denota-se $C \equiv_{\mathcal{T}} D$ ou $\mathcal{T} \models C \equiv D$.
- **Disjunção** Dois conceitos C e D são *disjuntos* em relação a $\mathcal{T}$ se $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ para cada modelo $\mathcal{I}$ de $\mathcal{T}$.

Tradicionalmente, o mecanismo de raciocínio básico fornecido por sistemas DL verifica a subsunção de conceitos. Isto, de fato, é suficiente para implementar as outras inferências, como pode ser visto nas seguintes reduções.

Proposição 1 (Redução a Subsunção). *Para conceitos C, D :*

- (i) C é *insatisfazível* $\Leftrightarrow C$ é *subsumido* por $\perp$;
- (ii) C e D são *equivalentes* $\Leftrightarrow C$ é *subsumido* por D e D é *subsumido* por C ;
- (iii) C e D são *disjuntos* $\Leftrightarrow C \sqcap D$ é *subsumido* por $\perp$.

Por outro lado, um sistema que permite formar a negação de uma descrição, pode reduzir subsunção, equivalência e disjunção de conceitos ao problema de satisfazibilidade.

Proposição 2 (Redução a Insatisfazibilidade). *Para conceitos C, D :*

- (i) C é subsumido por $D \Leftrightarrow C \sqcap \neg D$ é insatisfazível;
- (ii) C e D são equivalentes $\Leftrightarrow$ ambos $(C \sqcap \neg D)$ e $(\neg C \sqcap D)$ são insatisfazíveis;
- (iii) C e D são disjuntos $\Leftrightarrow C \sqcap D$ é insatisfazível.

A tarefa de raciocínio básica em uma ABox é *verificação de instâncias*, a qual testa se um indivíduo dado é uma instância (ou pertence) a um conceito específico.

A importância do estudo de lógicas de descrição está relacionada ao desenvolvimento de ontologias (BAADER; NUTT, 2003). Nos últimos anos ontologias se tornaram padrões para representação de conhecimento na Web semântica (SCHARRENBACH; BERNSTEIN, 2009). O mecanismo fundamental para raciocínio em ontologias é baseado em lógica de descrição.

2.2 Modelos probabilísticos gráficos

Os modelos apresentados na seção precedente são inerentemente determinísticos e não são capazes de representar incerteza. O raciocínio de senso comum é usualmente associado a informação incompleta; por isso, a incerteza subjacente em processos de raciocínio tem sido alvo de pesquisa desde os primórdios da Inteligência Artificial. Os modelos gráficos (baseados em grafos) se tornaram uma ferramenta popular para modelar incerteza. A forma como é realizada esta modelagem é mediante o uso da teoria de probabilidades e da teoria dos grafos (GETOOR; TASKAR, 2007). Os dois tipos mais comuns de modelos gráficos são as redes Bayesianas e as redes de Markov (PEARL, 1988).

O objetivo nestes modelos é representar a *distribuição conjunta* P de um conjunto de *variáveis aleatórias* $X = \{X_1, \dots, X_n\}$ de forma eficiente. Mesmo no caso mais simples, quando estas variáveis denotam valores binários, uma distribuição conjunta requer a especificação de 2^n números. Porém, é comum encontrar algum tipo de estrutura na distribuição, o que permite fatorá-la em componentes modulares. A estrutura que os modelos gráficos exploram são as propriedades de independência estocástica que existem em muitos fenômenos do mundo real.

2.2.1 Conceitos preliminares

Antes de explorar modelos probabilísticos gráficos, alguns conceitos de teoria de probabilidade devem ser definidos. Essa seção foi resumida de (NEAPOLITAN, 2003).

A teoria de probabilidade está relacionada a experimentos que têm um conjunto de resultados distintos. Um espaço amostral é um conjunto, cujos elementos são os resultados dos experimentos. Por simplicidade, considere o espaço amostral finito, nesse caso, cada subconjunto do espaço amostral é chamado de *evento*. Um subconjunto contendo exatamente um elemento é chamado de *evento elementar*. Assim que o espaço amostral é identificado, uma função de probabilidade é definida como:

Definição 5. *Suponha um espaço amostral Ω contendo n elementos distintos. Isto é, $\Omega = \{e_1, e_2, \dots, e_n\}$. Uma função que atribui um número real $P(E)$ a cada evento $E \subseteq \Omega$ é chamada de **função de probabilidade** no conjunto de subconjuntos de Ω .*

(Ω, P) é chamado de **espaço de probabilidade**.

Definição 6 (Probabilidade Condicional e Independência). *Sejam E e F eventos tais que $P(F) \neq 0$. Então a probabilidade condicional de E dado F , denotada $P(E|F)$, é dada por*

$$P(E|F) = \frac{P(E \cap F)}{P(F)}.$$

Dada a definição de espaço de probabilidade, os axiomas da teoria de probabilidade são dados por:

Teorema 1 (Axiomas da teoria de probabilidade). *Seja (Ω, P) o espaço de probabilidade. Então*

1. $P(\Omega) = 1$.
2. $0 \leq P(E) \leq 1$ para cada $E \subseteq \Omega$
3. Para E e $F \subseteq \Omega$ tal que $E \cap F = \emptyset$,

$$P(E \cup F) = P(E) + P(F).$$

Independência e probabilidade condicional são definidos a seguir.

Definição 7 (Independência). *Dois eventos E e F são independentes se uma das condições é válida*

1. $P(E|F) = P(E)$ e $P(E) \neq 0, P(F) \neq 0$.
2. $P(E) = 0$ ou $P(F) = 0$.

Definição 8. *Dois eventos E e F são condicionalmente independentes dado G se $P(G) \neq 0$ e uma das condições é válida:*

1. $P(E|F \cap G) = P(E|G)$ e $P(E|G) \neq 0, P(F|G) \neq 0$
2. $P(E|G) = 0$ ou $P(F|G) = 0$

Uma regra útil que envolve probabilidades condicionais é discutida a seguir. Suponha n eventos $E_1, E_2, \dots, E_n$ tal que $E_i \cap E_j = \emptyset$ para $i \neq j$ e $E_1 \cup E_2 \cup \dots \cup E_n = \Omega$. Tais eventos são chamados de mutuamente exclusivos e exaustivos. Então a lei da probabilidade total denota que para cada evento F ,

$$P(F) = \sum_{i=1}^n P(F \cap E_i).$$

Se $P(E_i) \neq 0$, então $P(F \cap E_i) = P(F|E_i)P(E_i)$. Deste modo, se $P(E_i) \neq 0$ para todo i , a lei é usualmente aplicada da seguinte maneira:

$$P(F) = \sum_{i=1}^n P(F|E_i)P(E_i).$$

Probabilidades condicionais de interesse geralmente são computadas a partir de probabilidades conhecidas usando o teorema de Bayes, definido a seguir:

Teorema 2 (Bayes). *Dados dois eventos E e F tais que $P(E) \neq 0$ e $P(F) \neq 0$,*

$$P(E|F) = \frac{P(F|E)P(E)}{P(F)}.$$

A seguir é definido o conceito de variável aleatória.

Definição 9 (Variável Aleatória). *Dado um espaço de probabilidade (Ω, P) , uma variável aleatória X é uma função de Ω para $\mathbb{R}$.*

Uma variável aleatória atribui um único valor a cada elemento (resultado) no espaço amostral. O conjunto de valores que a variável aleatória pode assumir é chamado de espaço de X . Uma variável aleatória é discreta se seu espaço é finito e contável.

Para uma variável X , $X = x$ denota o conjunto de todos os elementos $e \in \Omega$ que X mapeia ao valor de x , isto é, $X = x$ representa o evento $\{e \text{ tal que } X(e) = x\}$.

Claramente, duas variáveis aleatórias induzem uma função de probabilidade no produto cartesiano de seus espaços. Esta função de probabilidade, referenciada como $P(X, Y)$ é chamada de *distribuição de probabilidade conjunta* de X e Y . Este conceito se estende para três ou mais variáveis.

Dada uma distribuição de probabilidade conjunta, a lei de probabilidade conjunta implica que a distribuição de probabilidade de quaisquer variáveis aleatórias pode ser obtida pela soma das outras variáveis. Por exemplo, dada a distribuição de probabilidade conjunta $P(X, Y)$. Então

$$P(X = x) = \sum_y P(X = x, Y = y),$$

onde $\sum_y$ indica a soma de y por intermédio de todos os valores de Y . A distribuição de probabilidade $P(X = x)$ é chamada de distribuição de probabilidades marginal de X .

Nem sempre é possível representar uma crença mediante um valor preciso de probabilidade. Modelos baseados em conjuntos de distribuições de probabilidades, ou também conhecidos como *conjuntos credais*, permitem superar esta limitação.

Definição 10 (Conjunto Credal). *Dada uma variável aleatória X , um conjunto fechado e convexo de distribuições de probabilidades definidas sobre X é chamado de conjunto credal e indicado por $K(x)$.*

2.2.2 Redes Bayesianas

Imagine a situação onde a característica de uma entidade tem influência direta em outra característica daquela entidade. Por exemplo, a presença ou ausência de uma doença em um ser humano tem influência direta no resultado positivo ou negativo no teste daquela doença. Por décadas o teorema de Bayes tem sido usado para realizar *inferência probabilística* nesta situação. No exemplo, o teorema de Bayes seria utilizado para calcular a probabilidade condicional de um indivíduo ter uma doença quando um teste para aquela doença deu positivo. Considere a situação onde diversas características estão relacionadas por meio de cadeias de inferências. Por exemplo, o histórico de fumante tem uma influência direta na possibilidade de um indivíduo ter ou não bronquite e ter ou não câncer no pulmão.

Por outro lado, a presença ou ausência destas doenças têm uma influência direta na probabilidade dele experimentar ou não fadiga. A presença ou ausência de câncer de pulmão também tem uma influência direta no resultado positivo ou negativo do exame de

raio-X no peito. Nesta situação, seria desejável fazer *inferência probabilística* envolvendo características que não estão relacionadas via influência direta. Poderiam ser requeridas, por exemplo, as probabilidades condicionais de bronquite e câncer de pulmão quando é conhecido que o indivíduo fuma, está com fadiga, e tem resultado positivo de raio-X. Estas probabilidades condicionais não podem ser calculadas com uma aplicação simples do teorema de Bayes.

As redes Bayesianas foram desenvolvidas para simplificar a representação de probabilidades. Em resumo as redes Bayesianas focam nos problemas de a) representar a distribuição de probabilidade conjunta de um grande número de variáveis aleatórias; e b) realizar *inferência Bayesiana* nestas variáveis.

Para poder definir as redes Bayesianas e algumas das suas propriedades, algumas noções de teoria dos grafos são necessárias. Um grafo direcionado é um par (V, E) , onde V é um conjunto finito não vazio cujos elementos são nós (ou vértices), e E é um conjunto ordenado de pares de elementos de V . Os elementos de E são chamados de arestas (ou arcos), e se $(X, Y) \in E$, diz-se que há uma aresta de X a Y e que X e Y são incidentes na aresta. Se há uma aresta de X a Y ou de Y a X , diz-se que X e Y são adjacentes. Suponha que há um conjunto de nós $[X_1, \dots, X_k]$ onde $k \geq 2$, tal que $(X_{i-1}, X_i) \in E$ para $2 \leq i \leq k$. O conjunto de arestas que conectam os k nós é denominado **caminho** de X_1 a X_k . Os nós $X_2, \dots, X_{k-1}$ são chamados de nós interiores no caminho $[X_1, \dots, X_k]$. O subcaminho do caminho $[X_1, \dots, X_k]$ de X_i a X_j é o caminho $[X_i, X_{i+1} \dots X_j]$ onde $1 \leq i < j \leq k$. Um ciclo direcionado é um caminho de um nó a si próprio. Um caminho simples é um caminho que não contém subcaminhos com ciclos direcionados. Um grafo direcionado G é chamado de grafo direcionado acíclico (DAG) se não contém ciclos direcionados. Dado um DAG $G = (V, E)$ e nós X e Y em V , Y é chamado de pai de X se existe uma aresta de Y para X ; Y é chamado de descendente de X e X é chamado de ancestral de Y se há um caminho de X a Y ; Y é chamado de não descendente de X caso não seja descendente de X .

Baseado nas definições anteriores é possível definir a condição de Markov, fundamental para definir rede Bayesianas:

Definição 11. *Suponha uma distribuição de probabilidade conjunta P de variáveis aleatórias em um conjunto V e um DAG $G = (V, E)$. Diz-se que (G, P) satisfaz a condição de Markov se para cada variável $X \in V$, X é condicionalmente independente do conjunto de todos seus não descendentes dado o conjunto de seus pais. Independência entre dois conjuntos de variáveis X, Y é comumente denotada como $I_P(X, Y)$. Se o conjunto de pais e*

não descendentes de X é denotado por PA_X e ND_X , respectivamente, então a definição anterior significa que

$$I_P(\{X\}, ND_X | PA_X).$$

Deste modo:

Teorema 3. *Se (G, P) satisfaz a condição de Markov, então P é igual ao produto das distribuições condicionais de todos seus nós dado os valores de seus pais, quando estas distribuições condicionais existem.*

O Teorema 3 implica que os valores da distribuição conjunta são determinados de forma única pelas distribuições condicionais. Se cada nó no DAG não possui muitos pais, não há muitos valores nessas distribuições. Em geral, se cada variável tem dois valores possíveis e cada nó tem até k pais, precisa-se até $2^k n$ (quando o DAG tem n nós) valores para determinar as distribuições condicionais.

Teorema 4. *Seja um DAG G em que todos os nós representam variáveis aleatórias, e sejam definidas distribuições de probabilidade condicional discretas para cada nó dados seus pais em G . Então o produto destas distribuições condicionais produz a distribuição de probabilidades conjunta P das variáveis, e (G, P) satisfaz a condição de Markov.*

Note que o Teorema 4 requer que as distribuições condicionais especificadas sejam discretas. Em alguns casos com algumas distribuições contínuas o teorema ainda é válido.

Definição 12 (Redes Bayesianas). *Seja P a distribuição de probabilidades conjunta de um conjunto de variáveis aleatórias em algum conjunto V e $G = (V, E)$ um DAG. Então (G, P) é uma rede Bayesiana se (G, P) satisfaz a condição de Markov. De acordo com o Teorema 3, P é o produto de suas distribuições condicionais em G , e esta é a forma como P é sempre representado em uma rede Bayesiana. Além disso, de acordo com o Teorema 4, se um DAG G e quaisquer distribuições condicionais discretas são especificados, obtém-se uma rede Bayesiana.*

Exemplo 2. *A Figura 2 mostra uma rede Bayesiana que representa as relações probabilísticas entre as características do exemplo discutido no início da seção. Os valores das características, agora denotadas como variáveis aleatórias, são descritos na Tabela 1.*

Na rede Bayesiana pode ser observado que há uma aresta de H a L devido a que o histórico de fumante tem influência direta na presença de câncer de pulmão, e existe uma aresta de L a C porque a presença de câncer de pulmão tem influência direta no

Figura 2: Rede Bayesiana exemplo (NEAPOLITAN, 2003).

Tabela 1: Descrição de variáveis aleatórias no Exemplo 2.

Característica	Valor	Quando assume o valor
H	h_1	Há histórico de fumante
	h_2	Não há histórico de fumante
B	b_1	Bronquite presente
	b_2	Bronquite ausente
L	l_1	Câncer de pulmão presente
	l_2	Câncer de pulmão ausente
F	f_1	Fadiga presente
	f_2	Fadiga ausente
C	c_1	Teste de raio-X no peito positivo
	c_2	Teste de raio-X no peito negativo

resultado de raio-X no peito. Não há aresta de H a C porque o histórico de fumante tem influência no resultado de raio-X apenas por meio de sua influência na presença de câncer de pulmão. As probabilidades na rede são probabilidades condicionais dos estados de cada variável dado cada combinação dos valores dos pais das variáveis na rede, excetuando o caso das raízes onde são probabilidades a priori.

Inferência probabilística entre variáveis pode ser realizada usando a rede Bayesiana:

Definição 13 (Inferência Probabilística). *Inferência probabilística consiste em determinar a probabilidade de alguma variável de consulta dada evidência em outras variáveis, isto é, $P(\text{Consulta}|\text{Evidência})$.*

Por exemplo, podem ser computadas as probabilidades condicionais de bronquite e câncer de pulmão quando é conhecido que um indivíduo fuma, tem fadiga e tem raio-X positivo do peito, isto é, $P(b_1|h_1, f_1, c_1)$ e $P(l_1|h_1, f_1, c_1)$. Ainda, bronquite não tem influência no valor do teste raio-X. Essa probabilidade condicional não pode ser calculada usando aplicação simples do teorema de Bayes.

Se fosse conhecida a distribuição conjunta destas cinco variáveis aleatórias, poderia ser computada a probabilidade de um indivíduo ter bronquite dada a evidência da seguinte forma:

$$P(b_1|h_1, f_1, c_1) = \frac{P(b_1, h_1, f_1, c_1)}{P(h_1, f_1, c_1)} = \frac{\sum_l P(b_1, h_1, f_1, c_1, l)}{\sum_{b,l} P(b, h_1, f_1, c_1, l)},$$

onde $\sum_{b,l}$ denota a marginalização ou eliminação das variáveis b e l . Por outro lado, dada a condição de Markov, existe uma forma fatorada para essa distribuição conjunta. De acordo com o Teorema 3 a distribuição de probabilidade conjunta das variáveis aleatórias na rede Bayesiana é dada por $P(h, b, l, f, c) = P(h)P(b|h)P(l|h)P(f|b, l)P(c|l)$, deste modo:

$$P(b_1|h_1, f_1, c_1) = \frac{\sum_l P(h_1)P(b_1|h_1)P(l|h_1)P(f_1|b_1, l)P(c_1|l)}{\sum_{b,l} P(h_1)P(b|h_1)P(l|h_1)P(f_1|b, l)P(c_1|l)}.$$

Uma das estratégias usadas para realizar esse cálculo consiste em reorganizar os somatórios de acordo com o escopo das variáveis eliminadas:

$$P(b_1|h_1, f_1, c_1) = \frac{P(h_1)P(b_1|h_1) \sum_l P(l|h_1)P(f_1|b_1, l)P(c_1|l)}{P(h_1) \sum_{b,l} P(b|h_1)P(l|h_1)P(f_1|b, l)P(c_1|l)}.$$

E após simplificações:

$$P(b_1|h_1, f_1, c_1) = \frac{0.25 \sum_l P(l|h_1)P(f_1|b_1, l)P(c_1|l)}{\sum_{b,l} P(b|h_1)P(l|h_1)P(f_1|b, l)P(c_1|l)},$$

$$P(b_1|h_1, f_1, c_1) = 0.37.$$

A forma como é realizada a eliminação de variáveis, ou como são agrupadas as variáveis, ou como são enviadas as mensagens entre variáveis são alguns dos fatores associados a algoritmos que foram propostos na literatura para realizar inferência em redes Bayesianas. Uma revisão exaustiva das diversas propostas de algoritmos de inferência exatos assim como aproximados pode ser consultada em (HUANG; DARWICHE, 1996).

Suponha que houvesse valores das cinco características discutidas no exemplo (histórico de fumante, bronquite, câncer de pulmão, fadiga e raio-X no peito) para um grande número de indivíduos. Nesse sentido, algoritmos de aprendizado poderiam ser utilizados para construir a rede Bayesiana da Figura 2 (COOPER; HERSKOVITS, 1992; CHENG *et al.*, 2002; LAM; BACCHUS, 1994).

O aprendizado em redes Bayesianas consiste em determinar a estrutura gráfica e/ou as probabilidades associadas a partir de dados de treinamento. Em geral, existem diversos tipos de aprendizado que dependem em fatores tais como se a estrutura é ou não conhecida e se a observação dos dados é parcial ou completa (LAURITZEN, 1995; FRIEDMAN, 1997; RAMONI; SEBASTIANI, 1997). Pode haver instâncias de dados nas quais o valor de uma ou mais variáveis não foi observado e casos extremos com variáveis que nunca foram observadas. Existem na literatura algoritmos de aprendizado exatos e aproximados que se adequam a cada caso (HECKERMAN; GEIGER; CHICKERING, 1995).

2.3 Considerações finais

A primeira parte desse capítulo abordou modelos formais de representação e raciocínio baseados em lógica. Na segunda parte foram descritos, sucintamente, os modelos probabilísticos gráficos e foi enfatizado o estudo das redes Bayesianas. Cada formalismo apresenta vantagens e desvantagens em termos de representação e raciocínio. Em particular, para certas tarefas alguns formalismos se adequam melhor que outros. A esse respeito cabe ressaltar as dificuldades por parte de formalismos lógicos de lidar com incerteza, o que limita seu uso em domínios onde existe informação incompleta.

Por outro lado, os modelos probabilísticos gráficos, representados aqui pelas redes Bayesianas, permitem lidar com incerteza. Do ponto de vista lógico uma rede Bayesiana pode ser considerada como um tipo de lógica proposicional a qual fornece uma forma limitada de representação de conhecimento.

Seria interessante ter formalismos lógicos que ofereçam suporte a variáveis e quantificadores nos quais também pudesse se raciocinar de maneira probabilística. Nas últimas décadas, principalmente, houve interesse na combinação desses formalismos na forma de lógica probabilística. Porém, como será detalhado no próximo capítulo, essa não é uma tarefa trivial pois existem dificuldades decorrentes dessa integração.

3 LÓGICAS PROBABILÍSTICAS E RPL

Neste capítulo a combinação das duas formas de representação de conhecimento (lógica e probabilística), apresentadas de forma separada no capítulo anterior, são estudadas de forma integrada sob o nome de lógica probabilística. Uma lógica probabilística é um formalismo que combina raciocínio lógico com teoria de probabilidades. Um formalismo com essa flexibilidade permite superar as limitações do determinismo associado à lógica e por sua vez possibilita a generalização dos modelos probabilísticos (FAGIN; HALPERN; MEGIDDO, 1990).

O capítulo é organizado da seguinte forma: as Seções 3.1 e 3.2 apresentam, respectivamente, algumas das principais propostas clássicas e modernas para lógica probabilística. A seguir são descritas linguagens lógico-probabilísticas que utilizam relações de independências (foco da pesquisa de doutorado). Em particular, lógica probabilística proposicional (PPL) (Seção 3.3) e lógica probabilística relacional (RPL), uma das contribuições desse trabalho de doutorado (Seção 3.4).

3.1 Lógica probabilística clássica

Propostas para união de lógica e teoria de probabilidades remontam ao trabalho de George Boole no século passado (BOOLE, 1958), e têm sido discutidas várias vezes nas últimas décadas (HAILPERIN, 1996; NILSSON, 1986). Boole pesquisou atribuições da forma $P(\phi) = \alpha$, onde ϕ é uma fórmula em lógica proposicional (BOOLE, 1958). Hailperin apresentou uma solução para este modelo baseada em programação linear (HAILPERIN, 1996).

Cada probabilidade $P(\phi)$ é então uma combinação linear de probabilidades sobre atribuições verdadeiras ω : $P(\phi)$ é interpretada como $\sum_{\omega \models \phi} P(\omega)$, onde $\omega \models \phi$ denota a satisfazibilidade da fórmula ϕ por ω (HANSEN; JAUMARD, 1996)¹. Uma *inferência* é o

¹Em outras palavras, a probabilidade de uma sentença é a soma das probabilidades dos conjuntos de mundos possíveis nos quais aquela sentença é verdadeira.

lógicas probabilísticas proposicionais cresceu rapidamente após sua proposta (FAGIN; HALPERN; MEGIDDO, 1990; HALPERN, 1990; ABADI; HALPERN, 1994; HALPERN, 2003; FRISCH; HADDAWY, 1994; GEORGAKOPOULOS; KAVVADIAS; PAPADIMITRIOU, 1988; JAUMARD; HANSEN; ARAGAO, 1991; VAN DER GAAG, 1991). Diversas abordagens proposicionais são revistas por Hansen e Jaumard (HANSEN; JAUMARD, 1996).

Lógicas probabilísticas de primeira ordem têm sido estudadas por diversos filósofos da ciência (HOWSON; URBACH, 1993; KYBURG JR.; TENG, 2001). Em particular, Gaifman e Snir consideraram a probabilidade como uma função sobre sentenças de uma linguagem de primeira ordem (GAIFMAN; SNIR, 1982). Eles adotaram uma semântica *baseada em modelos*, interpretando $P(\phi) \geq \alpha$ como $P(\text{Mod}(\phi)) \geq \alpha$, onde $\text{Mod}(\alpha)$ é o conjunto de todas as interpretações para a sentença ϕ . Outras abordagens são as lógicas probabilísticas de Keisler (KEISLER, 1985) e Hoover (HOOVER, 1978), os quais exploraram semânticas *baseadas em domínios* onde valores de probabilidade são especificados no domínio dos indivíduos, e não nas interpretações.

Recentemente, pesquisa considerável em lógica probabilística de primeira ordem tem sido motivada pela lógica probabilística de Nilsson (1986). A perspectiva foi um pouco diferente dos predecessores: ao invés de usar a linguagem como uma forma para expressar fatos sobre probabilidade, Nilsson trata a linguagem como uma generalização da lógica de primeira ordem onde os valores de verdade são números reais ao invés de 0 e 1. Nilsson considera atribuições $P(\phi) = \alpha$, onde ϕ é uma sentença, e a semântica é baseada em modelos com valores de probabilidade definidos nas interpretações. Inferências são resolvidas por programação linear; é definido um método de máxima entropia que sempre produz um único valor de probabilidades durante inferência.

O exemplo descrito a seguir (baseado em (JAUMARD *et al.*, 2006)) denota alguns dos conceitos em lógica probabilística de primeira ordem de Nilsson.

Exemplo 4. *Dada uma base de conhecimento lógica probabilística, o objetivo de uma inferência consiste em determinar os valores atribuídos à fórmula ϕ_5 de tal forma que as fórmulas remanescentes continuem consistentes:*

$$P(\forall x \exists y \ p(x, y) \vee q(y)) = 0.9 \quad (\phi_1)$$

$$P(\exists x \ \neg r(x)) = 0.6 \quad (\phi_2)$$

$$P(\exists y \ q(y)) = 0.6 \quad (\phi_3)$$

$$P(\forall x \forall y \ \neg p(x, y) \vee r(x) \vee q(y)) = 0.7 \quad (\phi_4)$$

$$\underline{\pi} \leq P(\exists x \exists y \ \neg p(x, y)) \leq \bar{\pi} \quad (\phi_5)$$

Dadas essas 5 sentenças existem $2^5 = 32$ interpretações (mundos) possíveis/ impossíveis². Em lógica de primeira ordem o princípio de resolução pode ser usado para determinar a inconsistência de cada interpretação possível. Por exemplo, dada a interpretação 11110, resolução permite estabelecer se o sistema $\{\phi_1, \phi_2, \phi_3, \phi_4, \overline{\phi_5}\}$ é inconsistente e portanto uma interpretação impossível³. Deste modo são identificadas 12 interpretações possíveis as quais levam à seguinte formulação linear:

$$\max / \min w_1 + w_2 + w_3 + w_5 + w_7 + w_9 + w_{11} + w_{12}$$

sujeito a:

$$\begin{array}{l} w_1 + w_2 + w_3 + w_4 + w_5 + w_6 + w_7 + w_8 + w_9 + w_{10} + w_{11} + w_{12} = 1 \\ w_4 + w_5 + w_6 + w_7 + w_8 + w_9 + w_{10} + w_{11} + w_{12} = 0.9 \\ w_2 + w_3 \phantom{+ w_4 + w_5 + w_6 + w_7 + w_8 + w_9 + w_{10} + w_{11} + w_{12}} = 0.6 \\ w_1 + w_2 + w_3 \phantom{+ w_4 + w_5 + w_6 + w_7 + w_8 + w_9 + w_{10} + w_{11} + w_{12}} = 0.6 \\ w_1 + w_3 + w_4 + w_5 + w_6 + w_7 + w_8 + w_9 \phantom{+ w_{10} + w_{11} + w_{12}} = 0.7 \end{array}$$

A solução do programa linear devolve o seguinte resultado $P(\phi_5) = [0.1, 1]$.

3.2 Lógica probabilística moderna

O que diferencia uma abordagem clássica de uma moderna em lógica probabilística, é a inserção, por parte desta última, de elementos que visam diminuir a complexidade computacional de inferências (LUKASIEWICZ, 2001).

Outra dificuldade que surge durante processos de inferência é que um número reduzido de atribuições pode levar a *vazio inferencial*; isto é, à situação onde inferências de interesse produzem intervalos de probabilidade muito grandes. Vazio inferencial pode ser atenuado ou eliminado pelo uso de julgamentos de independência estocástica.

É natural imaginar que independência estocástica deva ser inserida em lógica probabilística não apenas para evitar o vazio inferencial mas também para ajudar a “decompor” bases de conhecimento grandes. Nesse sentido, pesquisas têm sido realizadas em lógica probabilística de primeira ordem baseada em “teoria de grafos”, atrelada à teoria de redes Bayesianas.

As primeiras tentativas de combinar redes Bayesianas e elementos de lógica de primeira

²O número de interpretações possíveis/impossíveis é computado a partir do número de fórmulas disponíveis.

³Neste exemplo em particular, a aplicação de resolução encontra uma contradição no mundo 11110 o que faz com que essa interpretação seja impossível.

ordem parecem ter sido motivadas pelo desejo de estender o escopo das redes Bayesianas, e não estender lógicas de primeira ordem. A meta era especificar redes Bayesianas sobre conjuntos de indivíduos, pela replicação de blocos básicos de proposições ou variáveis, em cenários de interesse para estatística (THOMAS; SPIEGELHALTER; GILKS, 1992) e inteligência artificial (WELLMAN; BREESE; GOLDMAN, 1992). Estas regras de replicação foram em alguns casos esquemáticas (ALMOND, 1994; KOLLER; PFEFFER, 1997; MAHONEY; LASKEY, 1996; THOMAS; SPIEGELHALTER; GILKS, 1992), e algumas vezes baseadas explicitamente em lógica de primeira ordem (BACCHUS, 1993; GLESNER; KOLLER, 1995; JAEGER, 1997; NGO; HADDAWY, 1997; POOLE, 1993).

Muitas lógicas probabilísticas de primeira ordem baseadas em teoria de grafos têm surgido na última década; o interesse neste tipo de lógicas talvez seja devido às conexões com aplicações de aprendizado de máquina. A disponibilidade de grandes quantidades de dados relacionais parece justificar o interesse em modelos estatísticos com relações entre indivíduos (FRIEDMAN *et al.*, 1999; GETOOR *et al.*, 2001; HECKERMAN; MEEK; KOLLER, 2004). Note que, enquanto algumas lógicas probabilísticas estão estendendo diretamente as redes Bayesianas, outras estão preocupadas em gerar redes Bayesianas a partir de regras (COSTA *et al.*, 2003; LASKEY; COSTA, 2005; FIERENS *et al.*, 2005; KERSTING; DICK, 2004; KERSTING; DE RAEDT; KRAMER, 2000; MILCH *et al.*, 2005; SATO; KAMEYA, 2001). Uma excelente revisão destas e outras linguagens relacionadas é relatada por Milch e Russell (MILCH; RUSSELL, 2007). Outras classificações recentes no escopo do aprendizado relacional estocástico (SRL) (GETOOR; TASKAR, 2007) e a Programação Lógica Indutiva Probabilística (PILP) (DE RAEDT; KERSTING, 2008) também são encontradas na literatura.

Deve ser enfatizado que todas estas lógicas baseadas em teoria de grafos garantem, usando várias suposições, que as atribuições são sempre satisfeitas por uma única distribuição de probabilidades. Assim, é possível dividir as lógicas probabilísticas de primeira ordem modernas existentes em duas classes. A primeira classe, formada por lógicas com semânticas de conjuntos credais⁴ e sem suposições de independência estocástica. A segunda classe, formada por lógicas nas quais grafos e independências estocásticas são usadas para produzir uma única distribuição de probabilidades para algum conjunto de fórmulas bem formadas. Com notáveis exceções (ANDERSEN; HOOKER, 1994), poucas lógicas tentaram combinar características destas duas classes.

Nas próximas seções duas lógicas que combinam características dessas duas classes

⁴Semântica credal em lógica probabilística significa a existência de *ao menos* um valor satisfazendo as atribuições.

são descritas com maiores detalhes: a lógica probabilística proposicional (PPL) (COZMAN; CAMPOS; ROCHA, 2008) e a sua extensão para lógica de primeira ordem ou relacional (RPL). A lógica RPL é uma contribuição deste trabalho (CAMPOS; COZMAN; OCHOA-LUNA, 2009).

3.3 Lógica probabilística proposicional

Nesta seção exploram-se formas de introduzir “estrutura” a atribuições em lógica probabilística proposicional usando ferramentas da teoria dos grafos. A lógica probabilística proposicional pode ser definida da seguinte forma:

Definição 14. *A linguagem Lógica Probabilística Proposicional (PPL) (COZMAN; CAMPOS; ROCHA, 2008) consiste de uma tripla $(\mathcal{G}, \mathcal{L}, \mathcal{A})$, onde $\mathcal{G}$ é um grafo direcionado acíclico com n nós, cada nó identifica uma variável X_i ; $\mathcal{L}$ é uma lista de fórmulas proposicionais; $\mathcal{A}$ é uma lista de atribuições $\alpha_j P(\varphi_j) \leq P(\phi_j \wedge \varphi_j) \leq \beta_j P(\varphi_j)$ ⁵, onde ϕ_j e φ_j são fórmulas; todas as fórmulas contém variáveis em $\mathcal{G}$. Além disso, $\mathcal{G}$ está sujeito a condição de Markov: uma proposição é estocasticamente independente de seus não descendentes dados seus pais.*

Em suma, nas redes PPL:

- (i) valores de probabilidades são associados às fórmulas;
- (ii) independência estocástica é parte da linguagem;
- (iii) grafos são usados para representar independências estocásticas em uma forma modular;
- (iv) grafos não são extraídos sintaticamente de valores e fórmulas lógicas;
- (v) fórmulas e valores podem ser inconsistentes, ou podem ser satisfeitos por um ou mais valores de probabilidade.

Inferência na linguagem PPL é definida da seguinte maneira:

Definição 15 (Inferência-PPL). *Dada uma linguagem PPL $\mathcal{N} = (\mathcal{G}, \mathcal{L}, \mathcal{A})$, inferência-PPL é o problema de decidir se existe uma distribuição de probabilidades d (para o conjunto de nós em $\mathcal{G}$) que satisfaz todas as restrições em $\mathcal{L}$ e $\mathcal{A}$.*

⁵Isto é, $\alpha_j \leq P(\phi_j | \varphi_j) \leq \beta_j$.

O exemplo mostrado a seguir ilustra alguma das características da linguagem PPL.

Exemplo 5. *Considere as seguintes sentenças em Lógica Probabilística Proposicional que ilustram conhecimentos hipotéticos sobre aves (COZMAN; CAMPOS; ROCHA, 2008).*

$$\begin{aligned}
& AveAntartida \rightarrow Ave \\
& Pinguim \rightarrow Ave \\
& AveVoa \rightarrow Voa \\
& Pinguim \rightarrow \neg Voa \\
& P(AveVoa|Ave) = 0.95 \\
& P(AveAntartida|Ave) = 0.01 \\
& P(Ave) \geq 0.2 \\
& P(AveVoa \vee Pinguim|AveAntartida) \geq 0.2 \\
& P(Voa|Ave) \geq 0.8
\end{aligned}$$

O grafo $\mathcal{G}$ na Figura 3 denota as relações de independência entre algumas das sentenças, por exemplo, $AveVoa$ e $Pinguim$ são independentes dado qualquer conjunção de literais $AveAntartida$ e Ave . As relações de independência levam a restrições não lineares na formulação de programas lineares para resolver inferências em redes PPL. Felizmente estes programas não lineares podem ser resolvidos por técnicas de **programação multilinear** ⁶. A aplicação de técnicas de programação multilinear sujeito às suposições de independência leva à inferência de valores de probabilidade para: $P(AveVoa|Ave \wedge \neg AveAntartida) \in [0.95, 0.96]$.

Programas multilineares são programas não lineares onde a função objetivo e as restrições envolvem variáveis em soma de produtos, isto é, $\sum_1^n \prod_{j=1}^m x_j$. A representação de relações de independência gera restrições multilineares. Portanto, neste trabalho usa-se um algoritmo desenvolvido em Campos (CAMPOS, 2004) que resolve programas multilineares por fornecer uma representação generalizada e eficiente.

A resolução de programas multilineares utiliza uma técnica de Reformulação-Linearização (RL) (SHERALI; TUNCBILEK, 1992) que gera algumas restrições multilineares (implícitas ao programa original) e substitui cada produto de variáveis por uma variável artificial, obtendo assim um subproblema linear aproximado. A solução deste subproblema aproximado é um limite superior para o problema multilinear. Este método tem sido aplicado com sucesso na computação de limites inferiores e superiores de probabilidades condicionais em diversos modelos (CAMPOS; COZMAN, 2004).

⁶Técnicas de programação multilinear podem ser revistas em (CAMPOS, 2004).

Figura 3: Grafo da rede PPL exemplo. Ave (A), AveVoa (AV), AveAntartida (AA), Pinguim (P), Voa (V).

A principal questão na análise de redes PPL é que a complexidade da inferência depende das interações entre o grafo, sentenças e valores (COZMAN; CAMPOS; ROCHA, 2008). Complexidade depende da esparsidade do grafo e a “proximidade” (no grafo, aumentado com arcos que representam fórmulas lógicas) de proposições em sentenças e valores. A situação é parecida a redes Bayesianas: quanto mais esparsa o grafo menor o esforço necessário para as *inferências*. Note que se uma proposição não é desenhada explicitamente no grafo, então é estocasticamente independente de todas as outras (é estocasticamente independente de seus não descendentes dados seus pais).

As redes PPL estão diretamente relacionadas à *Lógica Bayesiana* de Andersen e Hooker (ANDERSEN; HOOKER, 1994). Essa lógica é baseada em grafos acíclicos direcionados associados com a condição de Markov e com medidas que podem conter sentenças proposicionais. Entretanto, nas redes PPL foi feita uma separação entre $\mathcal{L}$ e $\mathcal{A}$; e o algoritmo de inferência (COZMAN; CAMPOS; ROCHA, 2008) é significativamente diferente.

A vantagem de usar teoria dos grafos em redes PPL está na redução do programa multilinear a ser resolvido por satisfazibilidade e inferência. Enquanto uma lógica probabilística sem estrutura requer a manipulação de 2^n atribuições de verdade, um modelo baseado em teoria dos grafos deve lidar apenas com probabilidades na decomposição $\prod_i P(X_i|pa(X_i))$. A idéia básica consiste em executar um algoritmo de inferência em redes Bayesianas para produzir uma descrição compacta de um programa multilinear—um programa onde os valores de probabilidade $P(X_i|pa(X_i))$ na rede auxiliar aparecem como variáveis de otimização. Esta idéia foi usada anteriormente na literatura de redes credais (COZMAN, 2000).

3.4 Lógica probabilística relacional

A linguagem PPL descrita na seção anterior é restrita à lógica proposicional; essa é uma linguagem muito limitada para representação de conhecimento. Um dos objetivos da pesquisa de doutorado e, portanto, uma das contribuições desse trabalho foi estender a linguagem PPL permitindo o uso de construções de primeira ordem, tais como relações e quantificadores. Esta tarefa não é totalmente óbvia: por um lado, o grafo em uma linguagem PPL representa relações de independência entre variáveis; por outro lado, as sentenças em lógica de primeira ordem representam dependências fortes entre variáveis.

A linguagem proposta é formada por *fórmulas* que são combinações de *constantes*, *relações*, operadores Booleanos, quantificadores $\exists$ e $\forall$, e *variáveis lógicas*. A semântica é dada pelo uso de um domínio (conjunto de indivíduos), o qual é finito e conhecido, e uma *interpretação* atribui indivíduos a constantes e relações no domínio respectivamente a constantes e relações no vocabulário. De maneira mais formal a linguagem denominada Lógica Probabilística Relacional é definida como:

Definição 16. A linguagem Lógica Probabilística Relacional (RPL) é definida por uma tupla $(\mathcal{D}, \mathcal{S}, \mathcal{G}, \mathcal{L}, \mathcal{A})$, onde:

1. $\mathcal{D}$ é um domínio;
2. $\mathcal{S}$ é um vocabulário de relações (funcionais/tipificadas)⁷;
3. $\mathcal{G}$ é um grafo direcionado acíclico onde cada nó é uma sentença ou uma fórmula aberta em $\mathcal{S}$;
4. $\mathcal{L}$ é uma lista de sentenças em $\mathcal{S}$;
5. $\mathcal{A}$ é uma lista de atribuições de valores especificados por tuplas $(\phi_j, \varphi_j, \alpha_j, \beta_j)$, onde ϕ_j, φ_j são sentenças em $\mathcal{S}$ e $\alpha_j \leq \beta_j$ são números em $[0, 1]$, interpretados como $\alpha_j P(\varphi_j) \leq P(\phi_j \wedge \varphi_j) \leq \beta_j P(\varphi_j)$.

O grafo $\mathcal{G}'_{\mathcal{D}}$ (obtido por proposicionalização de $\mathcal{G}$ no domínio $\mathcal{D}$) está sujeito à condição de Markov: sentenças são condicionalmente independentes de seus não descendentes dadas as instancicações de suas relações e pais.

⁷Uma relação *tipificada* restringe os valores do seus argumentos. Suponha que há uma relação *conceito*(x, y, x) e domínio dado por $\{Maria, Joao, Math1, Phil1, Eng1, A, B, C\}$. Assim, existem 512 possíveis instancicações. Porém se a relação for tipificada, isto é, se $x \in \{Maria, Joao\}$, $y \in \{Math1, Phil1, Eng1\}$ e $z \in \{A, B, C\}$ as instancicações são apenas 18. De maneira similar, uma relação é *funcional* se é permitida apenas uma instancicação para alguns de seus argumentos, dadas as instancicações dos argumentos restantes.

Inferências podem ser muito complexas em uma linguagem com esse grau de flexibilidade; naturalmente, um grafo esparso provavelmente demandará menos esforço computacional que um grafo denso com muitas sentenças de primeira ordem associadas.

Definição 17 (Inferência em RPL). *Inferência em uma linguagem RPL consiste em encontrar um limite superior/inferior para a probabilidade de uma sentença θ . O enfoque proposto para inferência é da proposicionalização, isto é, usa-se a condição de Markov na Definição 16 para gerar uma PPL a partir de uma linguagem RPL. Uma vez proposicionalizada, a inferência ocorre na rede associada à PPL.*

O primeiro passo na proposicionalização de uma linguagem RPL consiste em produzir um grafo $\mathcal{G}'_{\mathcal{D}}$ a partir do grafo $\mathcal{G}$ e o domínio $\mathcal{D}$. O segundo passo consiste em processar $\mathcal{L}$ e $\mathcal{A}$ ligando sentenças às relações instanciadas:

- Para cada sentença ϕ em $\mathcal{L}$ e com valores atribuídos em $\mathcal{A}$:
 1. Inserir um nó para a sentença ϕ e um nó da relação instanciada $r_{\phi}(x)$ para cada possível instanciiação x das relações que aparecem em ϕ . Estes nós relação serão pais de ϕ no grafo. Cada r_{ϕ} deve ser instanciado de acordo com restrições de tipo e possíveis propriedades funcionais.
 2. Verificar se ϕ ou $\neg\phi$ são aplicáveis, dada cada configuração de seus pais, e codificar a tabela de verdade para a sentença ϕ como uma tabela de probabilidade condicional do nó ϕ dado seus pais. Para uma atribuição em $\mathcal{A}$ definir φ como uma conjunção de pais instanciados de ϕ e especificar a probabilidade de ϕ dado cada φ (igual a 1 quando φ implica ϕ , e igual a 0 caso contrário).

Seja $\mathcal{G}'_p$ e $\mathcal{A}'_p$ respectivamente o grafo e a lista de valores após proposicionalização. Note que $\mathcal{G}'_p$ pode ter muito mais dependências entre sentenças que $\mathcal{G}$, porque instanciações de relações podem aparecer como pais de várias sentenças contendo aquela relação. Valores para $\mathcal{A}'_p$ foram criados em ambos passos do algoritmo (quando fórmulas abertas foram instanciadas e então quando tabelas de probabilidade condicional foram criadas para codificar tabelas de verdade de sentenças dadas as relações instanciadas). $(\mathcal{G}'_p, \mathcal{L}, \mathcal{A}'_p)$ é uma linguagem PPL e $\mathcal{D}, \mathcal{S}$ não são mais necessárias pois toda a informação está disponível na linguagem PPL.

Após a proposicionalização, podem ser aplicadas as mesmas técnicas desenvolvidas para linguagens PPL (COZMAN; CAMPOS; ROCHA, 2008). Maiores detalhes sobre a geração do programa multilinear usado para inferência podem ser consultados no Apêndice I.

Figura 4: Rede associada a RPL. Esquerda: grafo de independência dos nós lógicos (baseados nas relações instanciadas) e nós sentença. Direita: grafo com dependências entre sentenças e instâncias de suas relações.

Exemplo 6. A Figura 4 (esquerda) mostra uma grafo ligando umas poucas sentenças. Suponha que o domínio seja $\mathcal{D} = \{u, v, w\}$ e as relações $\{r_1(x) : x \neq w\}$ e $r_2(x)$. Além disso, suponha que $\mathcal{L} = \{\phi_1 \doteq \forall x r_1(x) \wedge r_2(x), \phi_2 \doteq \exists x r_1(x)\}$, $\mathcal{A} = \{P(\phi_1) \leq 0.6, P(\phi_2) = 0.3, P(\phi_2 \wedge \phi_1) \geq 0.3 \cdot P(\phi_1)\}$. Note que o grafo contém arcos entre relações instanciadas (de $r_1(u)$ e $r_2(v)$). Desse modo, durante a proposicionalização, não são inseridos novos nós para estes dois casos. A Figura 4 (direita) mostra a rede proposicionalizada.

Para ilustrar alguns dos aspectos definidos sobre linguagens RPL foram realizados experimentos no domínio “Universidade” (FRIEDMAN *et al.*, 1999; GETOOR *et al.*, 2001). Nesta base de conhecimento estão representadas relações entre professores, alunos, cursos e matrículas.

Um aluno está tipicamente inscrito em diversos cursos. Suponha que o conceito do aluno dependa da inteligência dele e a dificuldade do curso, uma distribuição de probabilidades é definida na Tabela 2.

Suponha que o domínio esteja formado por dois alunos, $\mathcal{D}_1 = \{Maria, Joao\}$; dois cursos, $\mathcal{D}_2 = \{Math1, Phil1\}$; três conceitos, $\mathcal{D}_3 = \{A, B, C\}$; e as descrições qualitativas $\mathcal{D}_4 = \{Alta, Baixa\}$. As relações de interesse são *estáInscrito*(x, y), com $x \in \mathcal{D}_1$, $y \in \mathcal{D}_2$; *inteligência*(x, y), com $x \in \mathcal{D}_1$, $y \in \mathcal{D}_4$; *dificuldade*(x, y), com $x \in \mathcal{D}_2$, $y \in \mathcal{D}_4$; e *conceito*(x, y, z), com $x \in \mathcal{D}_1$, $y \in \mathcal{D}_2$, $z \in \mathcal{D}_3$. Suponha que *inteligência* e *dificuldade* sejam funcionais. Finalmente, suponha que *conceito* seja tipificado para *conceito*(*Joao*, *Math1*, z), *conceito*(*Joao*, *Phil1*, z), e *conceito*(*Maria*, *Phil1*, z) seja funcional (z é definido de maneira única).

Tabela 2: Probabilidade de um conceito particular (A , B ou C) dado a dificuldade do curso e a inteligência do estudante.

<i>dificuldade</i>	<i>inteligência</i>	A	B	C
<i>Alta</i>	<i>Alta</i>	0.5	0.4	0.1
<i>Alta</i>	<i>Baixa</i>	0.1	0.5	0.4
<i>Baixa</i>	<i>Alta</i>	0.8	0.1	0.1
<i>Baixa</i>	<i>Baixa</i>	0.3	0.6	0.1

Considere as seguintes atribuições de valores:

$$P(\text{estáInscrito}(\text{Joao}, \text{Phil1})) \leq 0.5 \quad (A_1)$$

$$P(\exists x \forall y \text{ conceito}(x, y, A)) \leq 0.001 \quad (A_2)$$

$$0.1 \leq P(\exists x \forall y \text{ estáInscrito}(x, y)) \leq 0.15 \quad (A_3)$$

$$0.05 \leq P(\exists y \forall x \text{ estáInscrito}(x, y)) \leq 0.1 \quad (A_4)$$

Atribuições (A_1) a (A_4) expressam:

- *Joao* fará sua inscrição em *Phil1* com probabilidade não maior a 0.5.
- É pouco provável que um aluno consiga tirar A em todos os cursos.
- Com probabilidade baixa existe um aluno inscrito em todos os cursos.
- Com probabilidade baixa existe um curso no qual estejam inscritos todos os alunos.

Denote por ϕ_i a sentença lógica que aparece na atribuição A_i (por exemplo, ϕ_2 é $\exists x \forall y \text{ conceito}(x, y, A)$). Considere adicionalmente as sentenças lógicas:

$$\forall x, y (\text{estáInscrito}(x, y) \iff \exists z \text{ conceito}(x, y, z)) \quad (\phi_5)$$

$$\text{estáInscrito}(\text{Joao}, \text{Math1}) \quad (\phi_6)$$

$$\text{estáInscrito}(\text{Maria}, \text{Phil1}) \quad (\phi_7)$$

$$\text{inteligência}(\text{Joao}, \text{Baixa}) \quad (\phi_8)$$

$$\text{inteligência}(\text{Maria}, \text{Alta}) \quad (\phi_9)$$

$$\text{dificuldade}(\text{Math1}, \text{Alta}) \quad (\phi_{10})$$

$$\text{dificuldade}(\text{Phil1}, \text{Baixa}) \quad (\phi_{11})$$

Finalmente, define-se o grafo $\mathcal{G}$ para o domínio, ilustrado na Figura 5. Este grafo contém nós com variáveis livres. A primeira parte do procedimento de proposiciona-lização consiste em usar estes nós e as funções de probabilidade como *templates*. Suponha

Figura 5: Grafo simplificado para o domínio Universidade.

Figura 6: Grafo do domínio Universidade após processamento das variáveis livres.

que não haja restrições particulares nas funções de probabilidade (sem restrições nas probabilidades condicionais das relações). Deste modo, o grafo expandido $\mathcal{G}'_D$ é ilustrado na Figura 6 e nenhum parâmetro probabilístico adicional precisa ser criado durante o processamento do *template*.

Cabe salientar que o grafo na Figura 6 não é ainda o grafo proposicionalizado final, porque os nós para as sentenças lógicas e as atribuições probabilísticas ainda não foram incluídos.

Neste ponto, pode parecer que algumas relações são independentes das outras. Porém, elas ainda podem ter dependências lógicas que ficarão explícitas no grafo proposicionalizado final (por exemplo, em um caso extremo quando nenhum grafo foi especificado, tudo é estocásticamente independente). Portanto, para finalizar a proposicionalização global da rede associada à RPL: são incluídos nós para as sentenças, assim como dependências entre as sentenças e suas relações instanciadas.

O grafo proposicionalizado $\mathcal{G}'_p$ é ilustrado na Figura 7, onde:

- As sentenças ϕ_1 e ϕ_6 a ϕ_{11} não foram incluídas no grafo porque já existem nós para tais sentenças. Por exemplo, ϕ_1 já é um nó $estáInscrito(Joao, Phil1)$, ϕ_6 é o nó

estáInscrito(*Joao*, *Math1*), e assim por diante.

- Os seguintes nós poderiam ter sido removidos *estáInscrito*(*Joao*, *Math1*), *estáInscrito*(*Maria*, *Phil1*), *inteligência*(*Joao*), *inteligência*(*Maria*), *dificuldade*(*Math1*), *dificuldade*(*Phil1*) correspondentes de ϕ_6 até ϕ_{11} , respectivamente, caso tivesse sido notado que eles foram implicados pelas sentenças de $\mathcal{L}$. Esses nós foram mantidos no grafo para mostrar suas dependências e por compatibilidade com o procedimento de proposicionalização.
- As sentenças ϕ_3 e ϕ_4 têm como pais todos os nós *estáInscrito*, enquanto que ϕ_5 tem todos os nós *estáInscrito* e *conceito* como pais. A sentença ϕ_2 tem apenas nós pais *conceito* que são possíveis instanciações de ϕ_2 (assim os nós *conceito*(*Maria*, *Math1*, *B*) e *conceito*(*Maria*, *Math1*, *C*) não são pais de ϕ_2).
- Existem relações funcionais *inteligência* e *dificuldade*, que possuem um único argumento. A mesma coisa acontece com os nós *conceito*(*Joao*, *Phil1*), *conceito*(*Joao*, *Math1*) e *conceito*(*Maria*, *Phil1*). Dado que *conceito*(*Maria*, *Math1*, *z*) não é funcional, a relação *conceito* quando restrita a esses argumentos ainda precisa de três argumentos.
- Dado que a relação pode ser funcional apenas para certas combinações de argumentos, uma relação pode aparecer na rede proposicionalizada com número diferente de argumentos. Esta idéia pode tornar a rede proposicionalizada consideravelmente menor do que poderia chegar a ser.

Inferências são produzidas por programas multilineares gerados da rede proposicionalizada. Algumas inferências interessantes são relatadas a seguir. Primeiro, note que a rede inteira é inconsistente, porque A_2 denota que a probabilidade de obter um conceito *A* em todos os cursos é muito baixa, porém a Tabela 2 e as outras restrições na base de dados dão uma probabilidade maior para este evento.

A remoção da atribuição A_2 torna a rede consistente. Inferências são realizadas mediante a inserção de nós consulta às sentenças consultas desejadas e então usa-se técnicas de programação multilinear descritas anteriormente. Alguns resultados ilustram melhor essas inferências:

1. A probabilidade de *Joao* obter um conceito *A* em *Phil1* dado que ele está inscrito em todos os cursos:

$$P(\text{conceito}(\text{Joao}, \text{Phil1}, A) | \forall y \text{ estáInscrito}(\text{Joao}, y)) \in [0.0, 0.03].$$

Figura 7: Uma versão proposicionalizada do domínio Universidade.

2. A probabilidade que *Phil1* seja escolhido apenas por alunos inteligentes:

$$P(\forall x \text{ estáInscrito}(x, \text{Phil1}) \rightarrow \text{inteligência}(x, \text{Alta})) \in [0.9, 1.0].$$

3. A probabilidade de um aluno conseguir o conceito *A* em todos os cursos:

$$P(\exists x \forall y \text{ conceito}(x, y, A)) \in [0.003, 0.04].$$

Esta última inferência demonstra porque a atribuição A_2 levou a inconsistências, pois A_2 conflita com a inferência. Este último exemplo também ilustra como as inferências são processadas. Como o nó ϕ_2 no grafo da Figura 7 é exatamente o nó que deve ser inserido para realizar esta inferência, as consultas são então os valores máximo e mínimo para a probabilidade marginal $P(\phi_2)$.

3.5 Considerações finais

Neste capítulo foram apresentados diversos modelos para lógica probabilística. Na lógica probabilística clássica, por exemplo, a realização de inferências é baseada na formulação e solução de programas lineares. A lógica moderna, por outro lado, mediante a inserção de “estrutura” visa atenuar a complexidade das inferências. O uso de independência probabilística é uma forma de inserir estrutura em lógicas probabilísticas e foi focado em dois formalismos. O primeiro, denominado lógica probabilística proposicional

e o segundo, lógica probabilística relacional.

A linguagem lógico probabilística RPL é uma contribuição, em termos de representação de conhecimento, do trabalho realizado (OCHOA-LUNA; CAMPOS; COZMAN, 2007; CAMPOS; COZMAN; OCHOA-LUNA, 2009). Esta linguagem, porém, não será explorada nas tarefas de aprendizado, onde é dada prioridade ao aprendizado de fragmentos de lógica de primeira ordem tais como lógicas de descrição probabilísticas (descritas no próximo capítulo) visando aplicações em domínios reais.

4 LÓGICAS DE DESCRIÇÃO PROBABILÍSTICAS E $\text{CR}\mathcal{ALC}$

As lógicas de descrição apresentam considerável poder de representação a um custo computacional atrativo (BAADER; NUTT, 2003). Devido a essa característica, na pesquisa em lógica probabilística tem havido significativo interesse em lógicas de descrição probabilísticas (LUKASIEWICZ; STRACCIA, 2008): linguagens que associam terminologias e asserções com probabilidades.

Neste capítulo discutem-se lógicas de descrição probabilísticas. Uma revisão sucinta da literatura é dada na Seção 4.1. Posteriormente, foca-se o estudo da lógica de descrição probabilística $\text{CR}\mathcal{ALC}$ (Seção 4.2), linguagem escolhida para aprendizado. Considerações finais são dadas na Seção 4.3.

4.1 Lógicas de descrição probabilísticas

As lógicas de descrição probabilísticas surgiram como extensão das linguagens de ontologias (D'AMATO; FANIZZI; LUKASIEWICZ, 2008). Probabilidades foram associadas a terminologias e em alguns casos a asserções.

Heinsohn (HEINSOHN, 1994), Jaeger (JAEGER, 1994) e Sebastiani (SEBASTIANI, 1994) consideram axiomas de inclusão probabilística tais como $P_{\mathcal{D}}(\text{Planta}) = \alpha$, um indivíduo escolhido de forma aleatória é uma **Planta** com probabilidade α . Isto é, probabilidades são atribuídas a subconjuntos do domínio $\mathcal{D}$, o que caracteriza uma semântica *baseada no domínio*. Sebastiani permite atribuições tais como $P(\text{Planta}(\text{Tweety})) = \alpha$, especificando probabilidades sobre as interpretações. Por exemplo, pode ser dada a interpretação a $P(\text{AveVoa}(\text{Tweety})) = 0.001$ como a atribuição de 0.001 à probabilidade de todas as interpretações onde Tweety for uma ave que voa. Isto caracteriza a semântica *baseada em interpretações*.

Em geral, muitas propostas para lógicas de descrição probabilísticas adotaram semântica

baseada em domínio (DING; PENG; PAN, 2006; DÜRIG; STUDER, 2005; GIUGNO; LUKASIEWICZ, 2002; HEINSOHN, 1994; JAEGER, 1994; KOLLER; PFEFFER, 1997; LUKASIEWICZ, 2008; SEBASTIANI, 1994; YCART; ROUSSET, 2000), enquanto relativamente poucas adotaram semântica baseada em interpretações (COSTA; LASKEY, 2006; SEBASTIANI, 1994). A dificuldade com semântica baseada em domínio é o problema da *inferência direta* (KYBURG JR.; TENG, 2001): informação estatística sobre o domínio não se traduz em informação sobre os indivíduos. Por exemplo, suponha que $P(\text{AveVoa}) = 0.3$ foi aprendido; em uma semântica baseada em domínio, uma interpretação e um domínio são fixados e 0.3 indica a probabilidade de um elemento do domínio ser uma ave que voa. Porém, não é possível determinar $P(\text{AveVoa}(\text{Tweety}))$, dado que a interpretação é fixa e Tweety é uma ave que pode ou não voar. Por este motivo, muitas propostas para lógicas de descrição probabilísticas com semântica baseada em domínio simplesmente não lidam com asserções.

Note que Durig e Studer evitam inferência direta mediante a inserção de probabilidades nas asserções (DÜRIG; STUDER, 2005). Lukasiewicz propôs outra estratégia, onde probabilidades nas terminologias e asserções se combinam por meio de uma relação de implicação com propriedades não monotônicas, *implicação lexicográfica* (LUKASIEWICZ; STRACCIA, 2008; LUKASIEWICZ, 2008). Lukasiewicz considera versões probabilísticas de lógicas de descrição expressivas; sua lógica $P\text{-}\mathcal{SHOIN}(D)$ é atualmente a lógica de descrição probabilística de maior expressividade na literatura (LUKASIEWICZ, 2008).

As lógicas de descrição probabilísticas discutidas até agora compartilham a propriedade de que um conjunto de fórmulas poder ser satisfeita por um ou mais valores de probabilidade. Outra das características compartilhadas pelas lógicas de descrição probabilísticas discutidas é que elas não exprimem julgamentos de independência. No entanto, nos últimos quinze anos têm havido esforços significativos na combinação de construções lógicas com redes de Markov e Bayesianas, com o intuito de se beneficiar de julgamentos de independência ao invés de lidar com suas complexidades (GETOOR; TASKAR, 2007; MILCH; RUSSELL, 2007). Várias lógicas de descrição probabilísticas recentes têm adotado a semântica baseada em redes Bayesianas. A primeira lógica em adotar essa semântica, P-CLASSIC, acrescenta a lógica CLASSIC com conjuntos de redes Bayesianas (“p-classes”) para especificar um único valor de probabilidade no domínio (KOLLER; PFEFFER, 1997). Uma limitação disto é que P-CLASSIC não lida com asserções.

Outras lógicas que combinam terminologias com redes Bayesianas são a lógica de descrição *Tiny* de Yelland (YELLAND, 1999), a linguagem BayesOWL (DING; PENG; PAN,

2006) e a lógica de Staker (STAKER, 2002) (nenhuma delas lida com asserções). A linguagem PR-OWL de Costa e Laskey (COSTA; LASKEY, 2006) adota uma semântica baseada em interpretações herdada da redes Bayesianas multi-entidade (MEBNs) (LASKEY; COSTA, 2005). A linguagem PR-OWL é expressiva porém sem garantias em relação a inferência e domínios infinitos; seus algoritmos de inferência são baseados em proposicionalização incremental. Existem também trabalhos relevantes em bases de dados de conhecimento (HUNG; GETOOR; SUBRAHMANIAN, 2007) e lógicas de descrição nebulosas (LUKASIEWICZ; STRACCIA, 2008).

Uma lógica de descrição probabilística proposta recentemente é $\mathcal{CRALC}$ (COZMAN; POLASTRO, 2008, 2009). Maiores detalhes dessa linguagem serão dados na próxima seção (aprendizado dessa linguagem será explorado no próximo capítulo). $\mathcal{CRALC}$ é similar a PR-OWL em termos de semântica, baseada em interpretações, porém diferente em termos de expressividade. Muitas construções em $\mathcal{CRALC}$ podem ser emuladas na versão probabilística da linguagem OWL de Nottelmann e Fuhrs (NOTTELMANN; FUHR, 2006), entretanto métodos de inferência são completamente diferentes.

4.2 Lógica de descrição $\mathcal{CRALC}$

Para começar a falar em $\mathcal{CRALC}$ é importante descrever sucintamente elementos da sua lógica de descrição base, $\mathcal{ALC}$ (BAADER; NUTT, 2003). Assuma a, b indivíduos; A, B, C conceitos; r, s papéis. Uma *terminologia* contém *inclusões* $C \sqsubseteq D$ e *definições* $C \equiv D$. Um conceito pode ser um nome de conceito ou, recursivamente, uma *conjunção* ($C \sqcap D$), *disjunção* ($C \sqcup D$), *negação* ($\neg C$), *restrição existencial* ($\exists r.C$), *restrição de valor* ($\forall r.C$). Um conceito C *usa diretamente* D se eles aparecem respectivamente nos lados esquerdo e direito de uma inclusão/definição. A relação *usa* é sempre o fecho transitivo de *usa diretamente*. Uma terminologia é *acíclica* se não existem conceitos que *usam* a si próprio em uma inclusão ou uma definição. Uma ABox contém asserções tais como $C(a), r(a, b)$.

A semântica de $\mathcal{ALC}$ é dada por um conjunto $\mathcal{D}$, o *domínio*, e um mapeamento $\mathcal{I}$, a *interpretação*. Este mapeamento atribui a cada indivíduo um elemento do domínio, cada nome de conceito a um subconjunto do domínio, cada nome de papel a uma relação binária em $\mathcal{D} \times \mathcal{D}$. Uma interpretação é estendida a outros conceitos:

1. $\mathcal{I}(C \sqcup D) = \mathcal{I}(C) \cup \mathcal{I}(D)$.
2. $\mathcal{I}(C \sqcap D) = \mathcal{I}(C) \cap \mathcal{I}(D)$.

3. $\mathcal{I}(\neg C) = \mathcal{D} \setminus \mathcal{I}(C)$.
4. $\mathcal{I}(\exists r.C) = \{x \in \mathcal{D} \mid \exists y : (x, y) \in \mathcal{I}(r) \wedge y \in \mathcal{I}(C)\}$.
5. $\mathcal{I}(\forall r.C) = \{x \in \mathcal{D} \mid \forall y : (x, y) \in \mathcal{I}(r) \rightarrow y \in \mathcal{I}(C)\}$.
6. $C \sqsubseteq D$ se e somente se $\mathcal{I}(C) \subseteq \mathcal{I}(D)$, e $C \equiv D$ se e somente se $\mathcal{I}(C) = \mathcal{I}(D)$.

Existem traduções destas construções em lógica de primeira ordem e modal (ANDERSEN; HOOKER, 1994). Por isso é comum considerar um conceito C como um predicado unário $C(x)$, um papel r como um predicado binário $r(x, y)$, e as restrições como quantificadores. Na linguagem $\text{CR}\mathcal{ALC}$, todas as construções de $\mathcal{ALC}$ estão presentes porém somente são aceitos nomes de conceitos no lado esquerdo das inclusões/definições. Além disso, são permitidos três tipos de atribuições de probabilidade, onde C é um nome de conceito, D é um conceito, r é um nome de papel:

$$P(C) \in [\underline{\alpha}, \bar{\alpha}], \quad (4.1)$$

$$P(C|D) \in [\underline{\alpha}, \bar{\alpha}], \quad (4.2)$$

$$P(r) \in [\underline{\beta}, \bar{\beta}]. \quad (4.3)$$

Na Expressão (4.2), C usa *diretamente* conceitos condicionantes. Escreve-se $P(C|D) = \alpha$ (inclusão probabilística) quando $\underline{\alpha} = \bar{\alpha}$, $P(C|D) \geq \underline{\alpha}$ quando $\underline{\alpha} < \bar{\alpha} = 1$, e assim por diante. Nenhum conceito pode usar a si próprio, nem por intermédio de inclusões/definições determinísticas ou probabilísticas; isto garante a estrutura acíclica.

A semântica é baseada em probabilidades nas interpretações; isto é, os valores de probabilidade são atribuídos ao conjunto de todas as interpretações. A semântica da Expressão (4.1) indica que: para qualquer $x \in \mathcal{D}$, a probabilidade que x pertença à interpretação de C está no intervalo $[\underline{\alpha}, \bar{\alpha}]$. Isto é,

$$\forall x \in \mathcal{D} : P(\{x : x \in \mathcal{I}(C)\}) \in [\underline{\alpha}, \bar{\alpha}].$$

Uma forma intuitiva e informal para expressar a semântica é

$$\forall x \in \mathcal{D} : P(C(x)) \in [\underline{\alpha}, \bar{\alpha}].$$

A semântica das Expressões (4.2) e (4.3) é então:

$$\begin{aligned} \forall x \in \mathcal{D} : P(C(x)|D(x)) &\in [\underline{\alpha}, \bar{\alpha}], \\ \forall (x, y) \in \mathcal{D} \times \mathcal{D} : P(r(x, y)) &\in [\underline{\beta}, \bar{\beta}]. \end{aligned}$$

Como usual em lógica probabilística baseada em interpretações (BACCHUS, 1990), CRALC requer que todos os indivíduos sejam *rígidos* (um indivíduo corresponde ao mesmo elemento do domínio através das interpretações). De maneira similar a outras lógicas de descrição probabilísticas (LASKEY; COSTA, 2005; DING; PENG; PAN, 2006; JAEGER, 1997), CRALC adota uma condição de Markov que é melhor formulada usando funções indicadoras¹. Primeiro, para cada conceito C e para cada $x \in \mathcal{D}$, a função indicadora de $C(x)$ é independente da função indicadora de cada asserção de um conceito que não usa $C(x)$, dada as funções indicadoras de asserções de conceitos que C usa diretamente. Segundo, para cada papel r e para cada $(x, y) \in \mathcal{D} \times \mathcal{D}$, a função indicadora de $r(x, y)$ é independente de cada função indicadora de asserções, com exceção daquelas asserções de conceitos que usam $r(x, y)$. Isto fecha a especificação de CRALC .

Define-se a *rede-t* de uma terminologia como o grafo acíclico direcionado onde cada nó é um nome de conceito ou uma restrição ou um nome de papel. Se o conceito C usa diretamente outros nomes de conceitos/papéis, esses nomes aparecem como *pais* de C na rede-t. Para simplificar a apresentação, suponha que apenas um nome de conceito C possa aparecer em uma restrição $\forall r.C$ ou $\exists r.C$ (sem perda de generalidade, como C pode ser definido por intermédio de outros conceitos). Uma restrição $\forall r.C$ ou $\exists r.C$ tem pais r e C na rede-t.

O exemplo dado a seguir ilustra algumas definições dadas até agora.

Exemplo 7. *Considere a seguinte terminologia probabilística a qual é uma adaptação probabilística da ontologia Kangaroo (COZMAN; POLASTRO, 2009):*

$$\begin{aligned}
 P(\text{Animal}) &= 0.9 \\
 P(\text{Racional}) &= 0.6 \\
 P(\text{temFilho}) &= 0.3 \\
 \text{Humano} &\equiv \text{Animal} \sqcap \text{Racional} \\
 \text{Besta} &\equiv \text{Animal} \sqcap \neg \text{Racional} \\
 \text{Pai} &\equiv \text{Humano} \sqcap \exists \text{temFilho.Humano} \\
 P(\text{Canguru} | \text{Besta}) &= 0.4 \\
 P(\text{Canguru} | \neg \text{Besta}) &= 0.0 \\
 \text{CanguruPai} &\equiv \text{Canguru} \sqcap \exists \text{temFilho.Canguru}.
 \end{aligned}$$

A Figura 8 mostra a rede-t associada à ontologia probabilística. Por exemplo, o nó

¹A função indicadora de uma relação instanciada $C(x)$ ou $r(x, y)$ produz 1 se a relação instanciada é verdadeira e 0 caso contrário.

Humano tem como pais os nós *Animal* e *Racional* porque usa esses conceitos na sua definição. Do mesmo modo, um dos nós $\exists$ tem como pais *temFilho* e *Canguru* pois fazem parte da restrição. Os outros nós são construídos de maneira similar.

Figura 8: Rede-t associada a ontologia probabilística.

A tarefa básica em CRALC consiste em calcular a probabilidade de uma asserção, possivelmente condicionada a outras asserções. No Exemplo 7, poderia ser de interesse calcular a probabilidade de $P(\text{CanguruPai}(\text{Tweety}))$ dada a ontologia probabilística Kangaroo.

Em geral para poder responder essa questão é necessário explicar como é definida a distribuição de probabilidade conjunta em CRALC . Considere uma terminologia $\mathcal{T}$, um conjunto $\mathcal{C}$ com os nomes de conceito e as restrições em $\mathcal{T}$, um conjunto $\mathcal{R}$ formado pelos nomes de papel em $\mathcal{T}$, e um domínio finito $\mathcal{D}$. Denote por $\text{pa}(C(a))$ o conjunto de todas as funções indicadoras de asserções que $C(a)$ usa diretamente (C pode ser uma restrição). Sob a condição de Markov, qualquer distribuição de probabilidade conjunta para o conjunto $\mathbf{X}$ de funções indicadoras de asserções de $\mathcal{T}$ e $\mathcal{D}$ tem a forma

$$P(\mathbf{X}) = \prod_{C \in \mathcal{C}; x \in \mathcal{D}} P(C(x) | \text{pa}(C(x))) \times \prod_{r \in \mathcal{R}; (x,y) \in \mathcal{D} \times \mathcal{D}} P(r(x,y)). \quad (4.4)$$

Note que não se diferencia uma relação instanciada de sua função indicadora: $C(x)$ e $r(x,y)$ significam funções indicadoras de relações binárias e unárias na Expressão (4.4). Adota-se a suposição de *unicidade de nomes*: se a e b são nomes distintos para indivíduos, suas interpretações são distintas. São contempladas duas outras suposições. A suposição *domínio confinado* denota que $\mathcal{D}$ tem cardinalidade finita conhecida, denotada por N . A suposição de *unicidade* denota a existência de uma atribuição precisa para cada nome de papel ($P(r) = \beta$), e para cada nome de conceito tem-se uma definição $C \equiv D$, ou uma atribuição simples $P(C) = \alpha$, ou um par de atribuições $\{P(C|C') = \alpha', P(C|\neg C') = \alpha''\}$

onde C' deve ser um nome de conceito (que pode ser definido em outro lugar). Sob estas três suposições, coletivamente denominadas de *suposição Bayesiana*, cada terminologia $\text{CR}\mathcal{ALC}$ define uma *rede Bayesiana relacional (RBN) única* (HUNG; GETOOR; SUBRAHMANIAN, 2007) cuja instancição é uma rede Bayesiana dada pela Expressão (4.4).²

Inferência em $\text{CR}\mathcal{ALC}$ pode ser definida da seguinte maneira:

Definição 18. *Inferência em $\text{CR}\mathcal{ALC}$ é o cálculo de $P(A_0(a_0)|\mathcal{A})$ dada uma terminologia e uma *ABox* $\mathcal{A} = \{A_j(a_j)\}_{j=1}^M$.*

Dada uma terminologia probabilística, a rede-t e a expressão da distribuição de probabilidade conjunta, bastaria proposicionalizar a rede Bayesiana resultante de acordo com as asserções dadas. Essa, porém, não é uma estratégia escalável. Uma estratégia adotada em $\text{CR}\mathcal{ALC}$ foi usar técnicas de eliminação de variáveis de primeira ordem na hora de realizar inferências. Em resumo, o processo consiste em quebrar³ a rede-t parcialmente instanciada em uma rede Bayesiana proposicional associada a indivíduos da terminologia e uma rede Bayesiana relacional associada a indivíduos *genéricos*, que representam conjuntos de instâncias que se repetem em uma proposicionalização. Essa estratégia, aliada à utilização de métodos variacionais⁴ para realizar inferência aproximada possibilitam a realização de inferência de forma escalável. Essa última característica torna as redes $\text{CR}\mathcal{ALC}$ apropriadas para aplicações do mundo real. Entre outras coisas essa foi uma das razões que motivou a adoção de $\text{CR}\mathcal{ALC}$ como linguagem alvo de aprendizado no doutorado, tema que será detalhado no próximo capítulo.

4.3 Considerações finais

Embora a inserção de estrutura por parte de propostas recentes em lógicas probabilísticas de primeira ordem tenha trazido avanços em termos de expressividade, representação e inferência, quando se lida com aplicações em domínios reais existe um compromisso maior entre expressividade e complexidade. Por esse motivo, nos últimos anos lógicas de descrição probabilísticas têm sido estudadas.

A linguagem de descrição probabilística denominada $\text{CR}\mathcal{ALC}$, além de adotar independência estocástica, apresenta uma semântica baseada em interpretações, o que pos-

²Lógicas tais como P-CLASSIC e PR-OWL adotam suposições parecidas.

³O procedimento é conhecido como *shattering*.

⁴Maiores detalhes na realização de inferência em $\text{CR}\mathcal{ALC}$ podem ser consultados em (COZMAN; POLASTRO, 2008, 2009).

sibilita a realização de inferências sobre indivíduos. A inferência, é realizada de maneira eficiente nesta linguagem, onde métodos aproximados que possibilitam inferências escaláveis também têm sido explorados. Essas características influenciaram na escolha dessa linguagem para aprendizado, tema abordado no próximo capítulo.

5 APRENDIZADO DE MÁQUINA DE LÓGICA PROBABILÍSTICA

Nos dois capítulos precedentes introduziram-se linguagens lógico-probabilísticas com diversos graus de expressividade. Visando uma aplicação maior desses formalismos, neste capítulo explora-se o aprendizado de máquina, tanto da estrutura lógica como as probabilidades de lógicas de descrição probabilísticas. Em particular, são apresentados algoritmos de aprendizado para a lógica de descrição probabilística $\text{CR}\mathcal{ALC}$ — as principais contribuições deste trabalho.

Aprendizado de máquina é o estudo de sistemas que melhoram seu comportamento com a experiência (DE RAEDT, 2008). Em muitos casos, especialmente no contexto de mineração de dados, a experiência é dada por um conjunto de observações ou exemplos nos quais procura-se por padrões, regularidades ou regras de classificação e que, idealmente, possam ser passíveis de interpretação pelo usuário.

As abordagens de aprendizado de máquina usuais são capazes de lidar ou com incerteza ou com representações relacionais mas tipicamente não com ambas. Neste capítulo aborda-se o aprendizado de lógicas probabilísticas inicialmente do ponto de vista da programação em lógica indutiva probabilística (Seção 5.1). Posteriormente, foca-se no aprendizado da lógica de descrição probabilística. As principais contribuições desta tese são três algoritmos para o aprendizado da lógica de descrição probabilística $\text{CR}\mathcal{ALC}$. Cada algoritmo é detalhado nas Seções 5.4, 5.5 e 5.6. Em seguida, é feita uma discussão sobre consistência dos algoritmos (Seção 5.7) e são colocadas as considerações finais (Seção 5.8).

5.1 Programação em lógica indutiva probabilística

A Programação em Lógica Indutiva Probabilística lida com *aprendizado de máquina e mineração de dados* em *domínios relacionais* onde as observações podem ser ausentes, parcialmente observadas, e/ou com ruído (DE RAEDT; KERSTING, 2008). A programação em lógica indutiva probabilística se baseia, principalmente, em conceitos de *programação em*

lógica indutiva (ILP) (LAVRAC; DZEROSKI, 1994) ou mineração de dados multirrelacional (DZEROSKI; LAVRAC, 2001) posteriormente incrementados com aspectos probabilísticos.

ILP é uma área de pesquisa resultante da combinação de aprendizado de máquina e programação em lógica (LAVRAC; DZEROSKI, 1994). Em ILP visa-se o desenvolvimento de algoritmos para o aprendizado indutivo de descrições relacionais; porém, não se lida explicitamente com incerteza associada a informação com ruído ou faltante. Nesse sentido, ILP probabilístico estuda como os formalismos, características e métodos de programação em lógica indutiva podem ser estendidos para lidar com incerteza.

Em síntese, em ILP tenta-se encontrar uma hipótese H (um programa lógico, um programa clausal definido) que melhor defina o conjunto de exemplos positivos e negativos fornecidos para o aprendizado.

Exemplo 8. *Considere o problema de aprendizado do predicado `filha` de aridade 2 dados as seguintes asserções como exemplos:*

```
filha(maria, ana)    +
filha(maria, pedro)  +
filha(jose, pedro)   -
```

O símbolo $+$ ($-$) denota o exemplo positivo (negativo). O conhecimento base B também é dado de maneira explícita e neste exemplo descreve as relações familiares e o sexo de cada pessoa:

```
feminino(maria).
feminino(ana).
mae(ana, maria).
mae(ana, jose).
pai(pedro, maria).
pai(pedro, jose).
```

A partir desta informação pode ser induzida a seguinte hipótese H :

$$\begin{aligned} \text{filha}(C, P) &: \neg \text{feminino}(C), \text{mae}(P, C) \\ \text{filha}(C, P) &: \neg \text{feminino}(C), \text{pai}(P, C) \end{aligned} \tag{5.1}$$

Esta definição explica de forma correta os exemplos em termos do conhecimento base. Os exemplos positivos ($\mathbb{E}+$) são implicados por H e B , isto é, $B \wedge H \models \mathbb{E}+$.

Definição 19. Problema de Aprendizado ILP. *Dado um conjunto de exemplos positivos ($\mathbb{E}+$) e negativos ($\mathbb{E}-$) sobre a linguagem $\mathcal{L}_E$; uma teoria base B na forma de*

um conjunto de cláusulas definidas; uma linguagem de hipóteses $\mathcal{L}_H$, que determina as cláusulas que são permitidas nas hipóteses e uma relação cobertura $(e, H, B) \in \{0, 1\}$, que devolve a classificação de um exemplo e em relação a H e B . O problema de aprendizado em ILP consiste em encontrar uma hipótese H em $\mathcal{H}$ que cubra (em relação à teoria base B) todos os exemplos positivos $\mathbb{E}+$ (completude) e nenhum dos exemplos negativos $\mathbb{E}-$ (consistência).

A linguagem $\mathcal{L}_E$ escolhida para representar os exemplos junto com a relação de cobertura determina o tipo de ILP. Na literatura são conhecidos dois tipos de aprendizado: a partir de exemplos de *implicações* e *interpretações*. Uma terceira, baseada em *provas* foi definida por De Raedt e Kersting (DE RAEDT; KERSTING, 2008).

Em síntese, no aprendizado a partir de implicações, os exemplos são cláusulas definidas e uma hipótese H com base de conhecimento B cobre um exemplo e se e somente se $B \cup H \models e$. No aprendizado a partir de interpretações, os exemplos são interpretações *Herbrand* (conjunto de asserções que descrevem completamente uma situação) e uma hipótese H com base de conhecimento B cobre um exemplo e se e somente se e é um modelo de $B \cup H$. Finalmente, no aprendizado a partir de provas, os exemplos são árvores de provas instanciadas e um exemplo e é coberto por uma hipótese H junto com base B se e somente se e é uma árvore de prova para $H \cup B$.

Quando se trabalha com representações ILP probabilísticas existem fundamentalmente duas mudanças em relação a ILP:

- As cláusulas são anotadas com informação probabilística tais como probabilidades condicionais, e
- A relação de cobertura se torna probabilística.

Uma relação de cobertura probabilística é mais fraca que a relação de cobertura tradicional de ILP e é definida como a probabilidade de um exemplo dado a hipótese e a teoria base. O problema ILP probabilístico é definido a seguir.

Definição 20. Problema ILP Probabilístico. Dado um conjunto $\mathbb{E} = \mathbb{E}_p \cup \mathbb{E}_n$ de exemplos observados e não observados em uma linguagem $\mathcal{L}_E$; uma relação de cobertura probabilística $\text{cobertura}(e, H, B) = P(e|H, B)$; uma linguagem lógica probabilística $\mathcal{L}_H$ para as hipóteses; e uma teoria base B . Procura-se encontrar uma hipótese H^* em $\mathcal{L}_H$ tal que $H^* = \arg \max_H \text{pontuação}(\mathbb{E}, H, B)$ e as seguintes restrições são dadas: $\forall e_p \in \mathbb{E}_p : \text{cobertura}(e_p, H^*, B) > 0$ e $\forall e_i \in \mathbb{E}_n : \text{cobertura}(e_i, H^*, B) = 0$. A pontuação

é usualmente uma função objetivo envolvendo a relação de cobertura probabilística dos exemplos observados tais como verossimilhança $\prod_{e_p \in \mathbb{E}_p} \text{cobertura}(e_p, H^*, B)$ ou alguma variante penalizada.

Aprendizado de lógica probabilística de primeira ordem foi abordado sucintamente durante a pesquisa de doutorado (OCHOA-LUNA; COZMAN, 2009). Basicamente, foi desenvolvido um algoritmo ILP probabilístico baseado em FOIL e nFOIL (LANDWEHR; KERSTING; DE RAEDT, 2007) para o aprendizado de regras lógico probabilísticas em tarefas de classificação. O foco, porém, tem sido por linguagens com maiores possibilidades de aplicação em problemas reais tais como lógicas de descrição. Muitos dos conceitos usados em ILP probabilístico serão usados no aprendizado de lógica de descrição probabilística. No entanto, embora o aprendizado de formalismos em lógica probabilística de primeira ordem tenha sido abordado anteriormente (DE RAEDT; KERSTING, 2008; GETOOR; TASKAR, 2007), o aprendizado de lógicas de descrição probabilísticas é um assunto que tem sido pouco explorado.

As próximas seções fornecem detalhes do aprendizado de lógica de descrição e principalmente lógica de descrição probabilística onde são discutidas as maiores contribuições desta tese.

Aprendizado de lógica de descrição

O aprendizado de lógicas de descrição tem sido motivado, principalmente, pelo uso de tais linguagens na construção de ontologias. O uso de ontologias para representação de conhecimento tem sido fundamental nas propostas para Web Semantica (ANTONIOU; VAN HARMELEN, 2004). No entanto, construir ontologias de forma manual pode se transformar em uma tarefa complicada e com muita exigência de tempo (FANIZZI; D'AMATO; ESPOSITO, 2008). Hoje, devido principalmente à disponibilidade de dados, o aprendizado de ontologias tem se tornado uma alternativa interessante e esforços consideráveis estão sendo investidos para desenvolver meios automatizados para aquisição de ontologias (LEHMANN, 2007).

Muitas das abordagens iniciais focaram apenas o aprendizado de ontologias simples tais como hierarquias taxonômicas. Entretanto, propostas recentes tais como YINYANG (IANNONE; PALMISANO; FANIZZI, 2007), DL-FOIL (FANIZZI; D'AMATO; ESPOSITO, 2008), DL-Learner (LEHMANN; HITZLER, 2008b) e TermiTIS (FANIZZI; D'AMATO; ESPOSITO, 2010) têm focado no aprendizado de terminologias expressivas. De alguma maneira, todos esses enfoques têm sido inspirados por técnicas ILP. Nesse sentido, tenta-se transferir

características de ILP ao aprendizado de lógicas de descrição.

O objetivo de aprendizado em lógica de descrição é geralmente encontrar um conceito correto em relação aos exemplos dados. Uma definição formal segue:

Definição 21. *Dada uma base de conhecimento $\mathcal{K}$, um conceito alvo Alvo tal que $\text{Alvo} \notin \mathcal{K}$, um conjunto $\mathcal{E} = \mathcal{E}_p \cup \mathcal{E}_n$ de exemplos positivos e negativos dados como asserções para Alvo , o objetivo de aprendizado é encontrar uma definição de conceito $C(\text{Alvo} \equiv C)$ tal que $\mathcal{K} \cup C \models \mathcal{E}_p$ e $\mathcal{K} \cup C \not\models \mathcal{E}_n$.*

Uma definição correta para o conceito Alvo deve cobrir todos os exemplos positivos e nenhum dos negativos. Um algoritmo de aprendizado pode ser construído mediante a combinação de:

1. um operador de refinamento, o qual define como a árvore de busca pode ser construída;
2. um algoritmo de busca que controla como a árvore é percorrida;
3. uma função de pontuação para avaliar os nós na árvore e possibilitar a escolha do melhor candidato.

O operador de refinamento

Operadores de refinamento permitem encontrar definições de conceito candidatas por meio de duas operações básicas: generalização e especialização (LEHMANN; HITZLER, 2008a). Tais operadores tanto em ILP quanto em aprendizado de lógica de descrição estão baseados no denominado operador θ -subsunção que possibilita a construção de caminhos ordenados no espaço de busca. Se um conceito C subsume um conceito D ($D \sqsubseteq C$), então C cobre todos os exemplos que são cobertos por D , o qual torna a subsunção uma ordem apropriada. Um dos operadores de refinamento mais importantes para aprendizado de lógica de descrição está disponível no sistema DL-Learner (LEHMANN; HITZLER, 2008a, 2008b). Tal operador é *próprio*, *completo* e *completo fraco*. Essas propriedades permitem avaliar a utilidade do operador de refinamento no aprendizado de hipóteses.

Um operador de refinamento ρ para $\mathcal{ALC}$ é chamado de próprio, se e somente se para todos os conceitos C e D , $D \in \rho(C)$ implica que $C \not\equiv D$. O operador é chamado de completo se e somente se para todos os conceitos $C \sqsubset D$ é possível alcançar o conceito E com $E \equiv C$ a partir de D pela aplicação de ρ . O operador é chamado de completo fraco se e somente se para todos os conceitos $C \sqsubset \top$ é possível alcançar E com $E \equiv C$ a partir

de $\top$ pela aplicação de ρ . Mais detalhes sobre esse operador podem ser encontrados em (LEHMANN; HITZLER, 2008a).

A função de pontuação

Em um cenário determinístico uma relação de cobertura simplesmente testa se para uma definição de conceito candidata (C), um exemplo dado e é coberto; isto é, $\mathcal{K} \cup C \models e$ onde $e \in \mathcal{E}_p$ ou $e \in \mathcal{E}_n$. Nesse sentido uma relação de cobertura $cobertura(e, \mathcal{K}, C)$ indica se um conceito candidato cobre ou não um exemplo dado. Uma relação de cobertura é comumente avaliada por verificação de instâncias (FANIZZI; D'AMATO; ESPOSITO, 2008).

Em aprendizado de lógica de descrição é comum comparar candidatos por meio de funções de pontuação baseadas no número de exemplos positivos/negativos cobertos. Para evitar *overfitting* em conceitos, expansões horizontais¹ têm sido também exploradas (LEHMANN; HITZLER, 2008b). Por exemplo, em DL-Learner uma relação *fitness* leva em consideração o número de exemplos positivos bem como o comprimento de soluções quando os candidatos são expandidos na árvore de busca.

O algoritmo para percorrer o espaço de busca

O algoritmo de aprendizado depende basicamente da forma como o espaço de conceitos candidatos obtidos após a aplicação de operadores de refinamento é percorrido. Neste cenário determinístico a busca pelo melhor conceito candidato é comumente baseada no algoritmo FOIL (QUINLAN; CAMERON-JONES, 1993). Existem também abordagens alternativas. DL-Learner, por exemplo, usa uma estratégia baseada em algoritmos genéticos (LEHMANN, 2007) e uma proposta que se baseia na expansão horizontal e a verificação de redundância para percorrer a árvore de busca (LEHMANN; HITZLER, 2008b).

5.2 Aprendizado de lógica de descrição probabilística:CRALC

Não existem na literatura trabalhos visando o aprendizado de lógicas de descrição probabilísticas. Nesta seção são apresentadas três abordagens para o aprendizado de estrutura e probabilidades da linguagem de descrição probabilística CRALC. Estes algoritmos são contribuições deste trabalho.

Uma discussão preliminar sobre o aprendizado das probabilidades em CRALC é dada

¹Dado um nó em uma árvore de busca, a expansão horizontal é a maior largura dos seus conceitos filhos.

na Seção 5.3. Em seguida, o primeiro algoritmo de aprendizado de CR $\mathcal{ALC}$ (uma abordagem probabilística para o aprendizado de terminologias) é descrito na Seção 5.4. O segundo algoritmo (Seção 5.5) foca o aprendizado das inclusões probabilísticas. Finalmente, o terceiro (Seção 5.6) é uma abordagem que integra o aprendizado da estrutura e as inclusões probabilísticas em um único modelo.

5.3 Aprendizado das probabilidades em CR $\mathcal{ALC}$

Dada uma terminologia em $\mathcal{ALC}$, uma forma direta para definir uma terminologia probabilística CR $\mathcal{ALC}$ seria estabelecer relações de independência estocástica entre alguns conceitos e suas definições e estimar probabilidades. Nesta seção explica-se brevemente como é realizada a estimativa de probabilidades das inclusões probabilísticas. Cabe ressaltar que algumas das propostas para o aprendizado de terminologias probabilísticas apresentadas nas próximas seções usam este tipo de estimativas.

Suponha uma terminologia completa em $\mathcal{ALC}$ na qual são definidas relações de independência entre conceitos, isto é, transforma-se $C \equiv D$ ou $C \sqsubseteq D$ em $P(C|D) = \alpha$. Neste caso D pode assumir qualquer definição complexa. O objetivo é estimar a probabilidade desta inclusão probabilística.

Dado que CR $\mathcal{ALC}$ adota uma semântica baseada em interpretações, idealmente seriam necessários dados para cada indivíduo a fim de aprender as suas descrições. Por exemplo, se fosse desejado aprender a probabilidade de $P(\text{Professor}(a))$, seria necessário um conjunto de dados com informação acerca do objeto a ter ou não a propriedade *Professor*. Diferentes objetos podem ter diferentes probabilidades; muitas lógicas de descrição probabilísticas que adotam semântica baseada em interpretações combinam estas probabilidades diferentes em um único valor a ser adotado por todos os objetos (isto é, $\forall x : P(C(x)) = \alpha$).

Muitos conjuntos de dados disponíveis contêm informações sobre todos (ou ao menos de muitos) objetos do domínio, ao invés de informações sobre um objeto específico. Aprendizado com tais conjuntos de dados significa aprender probabilidades com semântica baseada em domínio. Neste trabalho assume-se que as probabilidades aprendidas com semântica baseada em domínio são uma boa aproximação para probabilidades com semântica baseada em interpretações; isto é, realiza-se *inferência direta* pela transferência das probabilidades aprendidas do domínio às interpretações. As probabilidades nas inclusões probabilísticas são aprendidas por meio da aproximação de probabilidades estimadas com

semântica baseada em domínio.

Para estimar probabilidades baseadas em domínio, usa-se estimativas de máxima verossimilhança ou o algoritmo *Expectation Maximization (EM)* (DEMPSTER; LAIRD; RUBIN, 1977) quando os dados para aprendizado são incompletos.

5.4 Aprendizado de CRALC: abordagem probabilística

A primeira proposta para aprendizado de CRALC adota uma estratégia inteiramente probabilística para a indução de terminologias probabilísticas. Basicamente, o procedimento de busca por definições de conceitos no aprendizado de lógica de descrição é substituído por um método probabilístico.

Entretanto, as mesmas suposições iniciais para o aprendizado em lógica de descrição são mantidas. Desse modo, a base de conhecimento é dada por $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. Os exemplos positivos e negativos, dados como indivíduos $\mathcal{E} = Ind_{\mathcal{C}}^+(\mathcal{A}) \cup Ind_{\mathcal{C}}^-(\mathcal{A}) \subseteq Ind(\mathcal{A})$. O conceito alvo é C . Hipóteses candidatas são dadas por definições válidas em CRALC.

A idéia principal desta proposta consiste em considerar todo o processo de busca de conceitos como um problema de classificação². Induzir uma definição para um conceito alvo consiste em encontrar um classificador que cubra todos os exemplos positivos e nenhum negativo.

Cada hipótese candidata $\mathcal{H}$ para o conceito alvo pode ser considerada uma variável probabilística; pode assumir os valores de verdadeira, falsa ou desconhecida caso o resultado de verificação de instâncias $\mathcal{K} \cup \mathcal{H} \models C(e)$ seja respectivamente verdadeiro, falso ou desconhecido. Nesse contexto, a tarefa de aprendizado se resume a aprender um classificador probabilístico para o conceito alvo.

Dentre os muitos classificadores existentes na literatura, consideram-se os classificadores que usam uma rede Bayesiana para representar relações entre atributos e classes. Neste grupo encontra-se o classificador Noisy-OR, inserido nos denominados *modelos de independência de influência causal* (ICI) (VOMLEL, 2006). Em um classificador Noisy-OR visa-se o aprendizado de uma classe C dado um grande número de atributos. Para o problema de aprendizado em CRALC, esta característica torna o classificador Noisy-OR preferível sobre outros classificadores probabilísticos³ dada a quantidade de hipóteses

²Em geral, um classificador é uma função que atribui instâncias representadas por atributos a uma classe.

³Uma revisão de classificadores probabilísticos existentes pode ser consultada em (FRIEDMAN; GEIGER;

Figura 9: Modelo ICI.

candidatas (atributos) que podem ser geradas no processo de indução de terminologias probabilísticas.

Em geral, em um classificador ICI cada variável atributo $A_j, j = 1, \dots, k$ ⁴ tem definida uma variável filha A'_j com uma distribuição de probabilidade associada $P(A'_j|A_j)$. As variáveis $A'_j, j = 1, \dots, k$ são os pais da variável classe C . $P_M(C|\mathbf{A}')$ representa uma função determinística f que atribui a cada combinação de valores $(a'_1, \dots, a'_k)$ uma classe c . Um classificador ICI genérico é ilustrado na Figura 9.

Por ser uma rede Bayesiana, a distribuição de probabilidades do modelo P_M é

$$P_M(c, \mathbf{a}', \mathbf{a}) = P_M(c|\mathbf{a}') \prod_{j=1}^k P_M(a'_j|a_j) \cdot P_M(a_j),$$

onde a probabilidade condicional $P_M(c|\mathbf{a}')$ é 1 se $c = f(\mathbf{a}')$ e 0 caso contrário.

O modelo Noisy-OR foi inicialmente introduzido em (PEARL, 1988) e é uma generalização da função determinística OR. Também é um classificador ICI onde f é uma função OR:

$$P_M(C = 0|\mathbf{A}' = 0) = 1 \text{ e } P_M(C = 0|\mathbf{A}' \neq 0) = 0.$$

A distribuição de probabilidades conjunta do modelo Noisy-OR é:

$$P_M(\cdot) = P_M(C|A'_1, \dots, A'_k) \cdot \left(\prod_{j=1}^k P_M(A'_j|A_j) \cdot P_M(A_j) \right).$$

GOLDSZMIDT, 1997; SU; ZHANG, 2006).

⁴ $\mathbf{A}$ denota a variável multidimensional $(A_1, \dots, A_k)$ e $\mathbf{a} = (a_1, \dots, a_k)$ seus estados.

e segue que

$$P_M(C = 0 | \mathbf{A} = \mathbf{a}) = \prod_j P_M(A'_j = 0 | A_j = a_j), \quad (5.2)$$

$$P_M(C = 1 | \mathbf{A} = \mathbf{a}) = 1 - \prod_j P_M(A'_j = 0 | A_j = a_j). \quad (5.3)$$

Usando o limiar $0 \leq t \leq 1$ todos os vetores de dados $\mathbf{a} = (a_1, \dots, a_k)$ tais que $P_M(C = 0 | \mathbf{A} = \mathbf{a}) < t$ são classificados como classe $C = 1$.

O classificador Noisy-OR tem a seguinte semântica: se um atributo A_j está em um estado a_j então a instância $(a_1, \dots, a_j, \dots, a_k)$ é classificada como $C = 1$ a menos que haja uma efeito inibidor, com probabilidade $P_M(A'_j = 0 | A_j = a_j)$. Por suposição, todos os efeitos inibidores são independentes. Por conseguinte, a probabilidade de uma instância não pertencer à classe C ($C = 0$), é o produto de todos os efeitos inibidores $\prod_j P_M(A'_j = 0 | A_j = a_j)$.

O processo de aprendizado em um modelo ICI M visa estimar os parâmetros que maximizem a capacidade de prever corretamente a classe C . Para o aprendizado dos parâmetros do modelos ICI, Vomlel (2006) propõe o uso do algoritmo EM. Uma implementação eficiente deste algoritmo depende da transformação do modelo ICI mediante o uso de variáveis ocultas (maiores detalhes podem ser encontrados em (VOMLEL, 2006)). A seguir é dada uma descrição sucinta do algoritmo EM adaptado às funções de combinação Noisy-OR.

Cada iteração do algoritmo EM consiste de duas etapas: o passo da estimativa (passo E) e passo da maximização (passo M). Em um modelo transformado e passível de decomposição o passo E corresponde ao cálculo de contagens esperadas marginais $n(A'_l, A_l)$ baseado nos dados $D = \{\mathbf{e}^1, \dots, \mathbf{e}^n\}$ ⁵ e o modelo M:

$$n(A'_l, A_l) = \sum_{i=1}^n P_M(A'_l, A_l | \mathbf{e}^i) \text{ para todos } l = 1, \dots, k,$$

onde para cada estado (a'_l, a_l)

$$P_M(A'_l = a'_l, A_l = a_l | \mathbf{e}^i) = \begin{cases} P_M(A'_l = a'_l | \mathbf{e}^i) & \text{se } a_l = a'_l, \\ 0 & \text{caso contrário.} \end{cases}$$

Para o classificador Noisy-OR P_M e uma evidência $C = c, \mathbf{A} = \mathbf{a}$. As probabilidades estimadas (o passo E) de A'_l para $l = 1, \dots, k$ podem ser calculadas como se segue (VOM-

⁵Cada $\mathbf{e}^i$ é um vetor de dados correspondente à evidência $\mathbf{e}^i = \{c^i, a_1^i, \dots, a_k^i\}$.

LEL, 2006):

$$P_M(A'_l = a'_l | C = c, \mathbf{a}) = \begin{cases} 1 & \text{se } c = 0 \text{ e } a'_l = 0, \\ 0 & \text{se } c = 0 \text{ e } a'_l = 1, \\ \frac{1}{z} \cdot \left(P_M(A'_l = 0 | A_l = a_l) - \prod_j P_M(A'_j = 0 | A_j = a_j) \right) & \text{se } c = 1 \text{ e } a'_l = 0, \\ \frac{1}{z} \cdot P_M(A'_l = 1 | A_l = a_l) & \text{se } c = 1 \text{ e } a'_l = 1. \end{cases}$$

onde z é uma constante de marginalização. O passo de maximização corresponde ao cálculo:

$$P_M^*(A'_l | A_l) = \frac{n(A'_l, A_l)}{n(A_l)}, \text{ para todos } l = 1 \dots, k.$$

Algoritmo 1: Algoritmo de aprendizado usando Noisy-OR.

Entrada: Um conceito alvo C , conhecimento base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, um conjunto de exemplos $\mathcal{E} = \text{Ind}_C^+(\mathcal{A}) \cup \text{Ind}_C^-(\mathcal{A}) \subseteq \text{Ind}(\mathcal{A})$ com asserções positivas e negativas.

Saída: Definição do conceito induzido C .

```

1  iniciar  $C' = \top$ ;
2  repita
3      determinar hipóteses  $H_1, \dots, H_n \sqsubseteq C'$  obtidas após a aplicação de operadores
      de refinamento para a lógica  $\mathcal{ALC}$ ;
4      sejam  $h_1, \dots, h_n$  (correspondentes às hipóteses  $H_1, \dots, H_n$ ) atributos do
      classificador probabilístico Noisy-OR;
5      realizar o aprendizado do classificador probabilístico usando o algoritmo EM;
6      para cada  $h_i$  faça
7          calcular a pontuação  $\prod_{e_p \in E_p} \text{cobertura}(e_p, h_i, \mathcal{K})$ ;
8      seja  $h'$  a hipótese com a melhor pontuação;
9      de acordo com  $h'$  adicionar  $H'$  a  $C$  (também  $C' \equiv H'$ );
10 até  $\text{pontuação}(\{h_1, \dots, h_i\}, \lambda_i, E) > \text{pontuação}(\{h_1, \dots, h_{i+1}\}, \lambda_{i+1}, E)$  ;
11 devolve  $C$ ;
```

Definido o classificador Noisy-OR, o algoritmo de aprendizado completo é ilustrado no Algoritmo 1. O algoritmo recebe como entrada uma base conhecimento $\mathcal{K}$. O objetivo é devolver a definição de conceito para C . Inicialmente o conceito C' é $\top$ (linha 1). A aplicação de operadores de refinamento para $\mathcal{ALC}$ (LEHMANN; HITZLER, 2008b) no conceito alvo C gera hipóteses $H_1, \dots, H_n$ (linha 3). Cada hipótese gerada H_i se torna uma variável probabilística h_i a qual pode assumir os valores verdadeiro, falso se para cada

exemplo disponível cobre ou não o conceito alvo C (linha 4). Cada uma dessas variáveis probabilísticas passa a fazer parte do modelo ICI Noisy-OR e é necessário estimar as probabilidades do modelo via EM (linha 5). Dado o classificador Noisy-OR construído é possível calcular a pontuação de cada hipótese em relação ao conjunto de todos os exemplos disponíveis (linhas 6 e 7)⁶. A hipótese com melhor pontuação, h' , é adicionada na definição objetivo (linha 9). O procedimento é repetido até que a adição da hipótese h_{i+1} não traga melhoras na pontuação global da definição de C , formada por $h_1, \dots, h_i$ (linha 10). O símbolo λ denota os parâmetros de máxima verossimilhança.

Após o aprendizado dos conceitos, o aprendizado das probabilidades associadas à terminologia é realizada mediante estimativas de máxima verossimilhança ou é usado o algoritmo EM (Seção 5.3).

Exemplo 9. *Imagine que deseja-se encontrar uma definição para o conceito Professor. Dados os seguintes exemplos positivos e negativos:*

Professor(Joao)	+
Professor(Maria)	−
Professor(Alberto)	−

e parte da base de conhecimento dada em forma explícita (asserções):

Pessoa(Joao)
Pessoa(Maria)
Pessoa(Alberto)
Artigo(Pub1)
Artigo(Pub2)
orienta(Joao, Maria)
publica(Joao, Pub1)
publica(Maria, Pub1)
publica(Alberto, Pub2)
⋮

Inicialmente $C' = \top$ e após a aplicação do operador de refinamento $H_1 = \forall \text{publica}.\top$, $H_2 = \exists \text{orienta}.\top$, $H_3 = \text{Pessoa} \dots$

⁶A relação de cobertura probabilística utiliza a verificação de instâncias para determinar se uma hipótese candidata e o exemplo fazem parte de um dado conceito.

Após várias iterações a definição aprendida é a seguinte:

$$\text{Professor} \equiv \text{Pessoa} \sqcap (\exists \text{orienta.Pessoa} \sqcap \exists \text{publica.Artigo}).$$

5.5 Algoritmo de aprendizado de inclusões probabilísticas

Diferentemente do algoritmo apresentado na seção precedente, nesta seção foca-se no aprendizado de inclusões probabilísticas em $\text{CR}\mathcal{ALC}$. Lembrando que as inclusões probabilísticas são atribuições probabilísticas de tipo $P(C|D) = \alpha$.

Basicamente a ideia é a seguinte. Suponha que um conceito C é aprendido (sem probabilidades) e sua definição seja dada por $C \equiv D \sqcap F$. Suponha também que o ruído inerente nos dados indique a possibilidade de uma inclusão probabilística relacionando os conceitos na definição.

Deste modo, inclusões probabilísticas da forma $P(C|X) = \alpha$ são examinadas, onde X são conceitos baseados em D e F , e α é uma constante aprendida por estimativas de máxima verossimilhança. O objetivo é a busca pelo melhor conceito condicionante para C . Assim, consideram-se todos os componentes da definição do conceito; no exemplo dado, X pode assumir os conceitos: domínio completo, D , F ou $D \sqcap F$. Assim, as probabilidades $P(C)$, $P(C|D)$, $P(C|F)$ e $P(C|D \sqcap F)$ são avaliadas. Finalmente, examina-se, baseado em uma função de avaliação, se é melhor considerar a inclusão probabilística encontrada ou manter a definição lógica original (isto é, $C \equiv D \sqcap F$).

5.5.1 Função de pontuação

A função de pontuação é útil para avaliar candidatos a inclusões probabilísticas. Esta função é baseada na verossimilhança dos exemplos disponíveis para aprendizado:

$$L = \prod_{e \in E} P(e|\mathcal{K}).$$

5.5.2 Aprendizado das inclusões probabilísticas

Considere uma terminologia determinística em $\mathcal{ALC}$, que pode ter sido aprendida usando algoritmos de aprendizado para lógica de descrição. O aprendizado de inclusões probabilísticas inicialmente assume independência probabilística entre os nomes de con-

ceitos e estima inclusões probabilísticas $P(C)^7$, onde $C \in N_C$. A seguir, é realizado o cálculo da pontuação correspondente para esta terminologia.

Posteriormente, o algoritmo verifica se a inclusão de probabilidades condicionais em outros conceitos ou em descrições de conceitos melhora a pontuação inicial. Nesta etapa, o algoritmo considera como candidatos a conceitos condicionantes os diversos componentes da definição lógica do conceito sob análise (é natural considerar, por exemplo, que o conceito C possa ser condicionado apenas por conceitos aos quais está relacionado via sua definição). Adicionalmente, pode ser usado subsunção para testar outros conceitos condicionantes. No final do processo são mantidos apenas conceitos condicionantes ou descrições que fornecem a maior pontuação.

Quando uma inclusão probabilística candidata usa apenas parte de uma descrição lógica de conceito original, maior análise é necessária. Uma decisão deve ser tomada sobre a adição da inclusão probabilística ou a eliminação da definição de conceito original. Note que se a descrição de conceitos original e a inclusão probabilística fossem mantidas juntas, então não seria possível construir uma distribuição de probabilidades única para a terminologia probabilística aprendida. O procedimento completo é mostrado no Algoritmo 2.

O Algoritmo 2 recebe uma base de conhecimento $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ e um conjunto de treinamento $\mathcal{E}$, composto por exemplos. Para cada conceito C na terminologia $\mathcal{T}$ o algoritmo devolve os componentes de sua definição. Todos os conceitos subsumidos por esta definição são considerados de forma separada e incluídos no conjunto H (linha 4); o conceito vazio e a definição completa são também inseridos em H . Se um conceito inserido for subsumido por outro conceito, este último conceito é também incluído em H (linha 5). A variável $pontuação_c$ é iniciada com a pontuação de $\mathcal{K}$ (linha 7), e uma variável A_c é iniciada com vazio (linha 8). Cada conceito incluído em H é considerado separadamente como condicionante do conceito de interesse (linha 10). A probabilidade desta nova inclusão probabilística é estimada usando estimativas de máxima verossimilhança e então a nova pontuação é computada (linhas 11 e 12). Se esta nova pontuação melhora a atual, a inclusão probabilística é considerada como a melhor e mantida em A_c (linha 15). Após o aprendizado de uma inclusão probabilística para cada conceito, o algoritmo retorna uma base de conhecimento com uma nova terminologia (linha 16). Se a pontuação inicial não for melhorada por nenhuma inclusão probabilística proposta para um conceito, este conceito não é associado com nenhuma inclusão probabilística. Adicionalmente são

⁷Isto é, $P(C|\mathcal{T})$.

Algoritmo 2: Algoritmo para aprendizado de inclusões probabilísticas.

Entrada: Uma base de conhecimento $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ e um conjunto de exemplos $\mathcal{E}$.

Saída: Uma base de conhecimento nova $\mathcal{K} = \langle \mathcal{T}_f, \mathcal{A} \rangle$ com inclusões probabilísticas.

```

1  $\mathcal{T}_f = \mathcal{T}$ ;
2 calcular pontuação  $pontuação_f$  usando  $\mathcal{E}$ ;
3 para cada conceito  $C \in \mathcal{T}$  faça
4   busca em  $\mathcal{T}$  a definição de conceito  $C$  e inclui os componentes da sua definição
   em um conjunto  $H$ ;
5   se conceito  $C$  é subsumido por algum outro conceito  $D$  então
6     incluir  $D$  em  $H$ ;
7    $pontuação_c = pontuação_f$ ;
8   iniciar  $A_c$ ;
9   para cada  $h \in H$  faça
10    atribuir  $\mathcal{T}_h = \mathcal{T} \cup P(C|h)$ ;
11    aprender probabilidades usando  $\mathcal{E}$ ;
12    calcular  $pontuação_o_h$  usando  $\mathcal{E}$ ;
13    se  $pontuação_o_h > pontuação_c$  então
14       $pontuação_c = pontuação_o_h$ ;
15       $A_c = P(C|h)$ ;
16  $\mathcal{T}_f = \mathcal{T}_f \cup A_c$  (restrições de inserção são examinadas);
```

verificadas as restrições de inserção, isto é, se a inclusão probabilística substitui ou não a definição de conceito original.

Após a finalização do algoritmo uma rotina de poda pode verificar se a terminologia é passível de simplificação. Por exemplo, a inclusão $\text{Pesquisador} \sqsubseteq \text{Professor}$ pode ser automaticamente apagada da terminologia na presença de inclusões probabilísticas $P(\text{Pesquisador}|\text{Professor}) = 0.4$ e $P(\text{Pesquisador}|\neg\text{Professor}) = 0.0$.

Exemplo 10. Tomando como base as condições iniciais do Exemplo 9 e a definição do conceito de Professor dada por:

$$\text{Professor} \equiv \text{Pessoa} \sqcap (\exists \text{publica.Artigo} \sqcap \exists \text{orienta.Pessoa})$$

O algoritmo avalia cada uma das opções para $P(\text{Professor}|\text{componentes}(\text{Professor}))$, isto

é:

$$P(\text{Professor}|\text{Pessoa} \sqcap (\exists \text{publica.Artigo} \sqcap \exists \text{orienta.Pessoa})) = \beta$$

$$P(\text{Professor}|\text{Pessoa}) = \lambda$$

$$P(\text{Professor}|\exists \text{publica.Artigo}) = \iota$$

$$P(\text{Professor}|\exists \text{orienta.Pessoa}) = \theta$$

$$P(\text{Professor}|\text{Pessoa} \sqcap \exists \text{publica.Artigo}) = \gamma$$

$$P(\text{Professor}|\text{Pessoa} \sqcap \exists \text{orienta.Pessoa}) = \alpha$$

$$P(\text{Professor}|\exists \text{publica.Artigo} \sqcap \exists \text{orienta.Pessoa}) = \varphi$$

A avaliação da pontuação entre os diversos candidatos, determina a escolha de uma inclusão probabilística:

$$P(\text{Professor}|\text{Pessoa} \sqcap \exists \text{orienta.Pessoa}) = \alpha.$$

5.6 Abordagem combinada

Nas duas abordagens precedentes ou foi dada ênfase ao aprendizado de estruturas por métodos probabilísticos ou foi focado o aprendizado de inclusões probabilísticas a partir de conceitos aprendidos por métodos tradicionais. Nesta seção, adota-se uma abordagem combinada. Dado que uma terminologia probabilística consiste de definições de conceitos e inclusões probabilísticas, uma escolha fundamental no aprendizado sob uma abordagem combinada é dar a cada componente a relevância adequada.

A natureza determinística de alguns conceitos permite estabelecer suas definições de forma clara, por exemplo para $\text{Pai} \equiv \text{Masculino} \sqcap \text{temFilho}. seria desnecessário encontrar uma interpretação probabilística. Por outro lado, existem conceitos para os quais uma descrição probabilística pode ser de maior interesse, por exemplo, $P(\text{AveVoa}|\text{Ave}) = \alpha$. A princípio, um algoritmo de aprendizado deve ser capaz de lidar com essas duas possibilidades.$

Pode ser argumentado que os exemplos positivos e negativos definem a escolha por uma definição de conceito ou uma inclusão probabilística. Em um cenário determinístico espera-se encontrar definições de conceitos que cubram todos os exemplos positivos, o que nem sempre é possível. É comum encontrar heurísticas que flexibilizam essas restrições. Além disso, existem exemplos para os quais não é possível determinar se fazem parte de hipóteses candidatas. Incerteza surge deste tipo de informação incompleta. Deste modo, quando não é possível encontrar uma definição de conceito que cubra todos os exemplos positivos assume-se que tais hipóteses são candidatas a inclusões probabilísticas e inicia-se

uma busca pela melhor inclusão probabilística que se ajuste aos exemplos.

Da mesma forma que o aprendizado de lógica de descrição, três tarefas são importantes e devem ser consideradas: (1) definição de operadores de refinamento, (2) definição da função de pontuação e (3) algoritmo para percorrer o espaço de busca.

O operador de refinamento descrito na Seção 5.1 é usado para aprender o componente determinístico das terminologias probabilísticas. As outras duas tarefas foram adaptadas para aprendizado de lógica de descrição probabilística da seguinte forma.

5.6.1 A função de pontuação probabilística

Nesta proposta, dado que se deseja aprender terminologias probabilísticas, adota-se uma relação de cobertura probabilística (LANDWEHR; KERSTING; DE RAEDT, 2007):

$$\text{cobertura}(e, \mathcal{K}, C_h) = P(e|\mathcal{K}, C_h).$$

Cada hipótese candidata aliada a um exemplo resulta em uma variável aleatória. Essa variável é verdadeira se o exemplo é coberto e falsa caso contrário, para garantir a correção do processo ILP (isto é, para cobrir exemplos positivos e não os negativos). Desta maneira uma relação de cobertura probabilística é uma generalização da cobertura determinística, e é adequada para uma abordagem combinada. As probabilidades podem ser calculadas pela aplicação do teorema de Bayes:

$$P(e|\mathcal{K}, C_{h_1}, \dots, C_{h_k}) = \frac{P(C_{h_1}, C_{h_2}, \dots, C_{h_k}|C_a)P(C_a)}{P(C_{h_1}, \dots, C_{h_k})},$$

onde $C_{h_1}, \dots, C_{h_k}$ são candidatos de definições de conceitos, e C_a denota a variável conceito alvo. Existem três possibilidades para modelar $P(C_{h_1}, \dots, C_{h_k}|C_a)$: (1) a suposição *Naive Bayes* pode ser adotada (LANDWEHR; KERSTING; DE RAEDT, 2007) (cada conceito candidato é independente dado o alvo), e então $P(C_{h_1}, \dots, C_{h_k}|C_a) = \prod_i P(C_{h_i}|C_a)$; (2) a função Noisy-OR também poderia ser usada (OCHOA-LUNA; COZMAN, 2009); (3) uma opção menos restritiva baseada nas redes de árvores aumentadas *Naive Bayes* (TAN) pode ser útil (LANDWEHR; KERSTING; DE RAEDT, 2007). Esta última possibilidade foi adotada para a relação de cobertura probabilística. Em cada caso, as probabilidades são estimadas por parâmetros de máxima verossimilhança (condicionais). A definição de conceito candidato C_{h_i} com a maior probabilidade $P(C_{h_i}|C_a)$ é a escolhida como melhor candidata.

Dado que a escolha foi por uma relação de cobertura probabilística, a pontuação

probabilística é definida de forma parecida:

$$pontuação(\mathcal{K}|C_h) = \prod_{e_i \in \mathcal{E}_p} P(e_i|\mathcal{K}, C_h).$$

Na pontuação probabilística definida, um limiar previamente estabelecido permite diferenciar entre um candidato determinístico e uma inclusão probabilística. Maiores detalhes são dados a seguir.

5.6.2 O algoritmo para aprender terminologias probabilísticas

Nesta abordagem, adota-se uma estratégia combinada onde um algoritmo tradicional é usado para percorrer o espaço de conceitos determinísticos e um procedimento probabilístico é usado para gerar as inclusões probabilísticas. A escolha entre uma definição determinística ou uma inclusão probabilística é baseada em uma pontuação probabilística. O processo inicia pela busca da definição do conceito determinístico. Se após uma série de iterações a pontuação do melhor candidato fica abaixo de um limiar dado, a busca por uma inclusão probabilística é iniciada — a busca determinística é preterida. Os k -melhores candidatos são considerados como ponto inicial para a busca pela inclusão probabilística. O procedimento de aprendizado completo é mostrado no Algoritmo 3.

O algoritmo inicia com a definição de conceito geral na raiz da árvore de busca (linha 1). Este nó é expandido de acordo com operadores de refinamento e critério de expansão horizontal (linha 4), isto é, nós filhos obtidos por operadores de refinamento são adicionados à árvore de busca (linha 6). Os parâmetros probabilísticos desses nós filhos são aprendidos (linha 7) e então são avaliados para uma nova expansão (nós alternativos baseados no fator de expansão horizontal são também considerados (linha 9)). Este processo continua até o critério de parada ser atingido (diferença para pontuação é insignificante); após isto, o melhor nó obtido é avaliado e se estiver acima de um limiar dado, a definição de conceito determinística encontrada é retornada (linha 13). Caso contrário, o procedimento para o aprendizado da inclusão probabilística é chamado (linha 15).

O Algoritmo 4 aprende inclusões probabilísticas; inicia recuperando os k melhores nós na árvore de busca e calculando a informação condicional mútua para cada par de nós (linhas 1 e 2). Em seguida, um grafo não direcionado é construído onde os vértices são os k nós e as arestas são ponderadas com o valor da informação condicional mútua para cada par de vértices (linhas 3 e 4). Uma árvore *maximum weight spanning tree* deste grafo é construída (linha 5) e o conceito alvo é adicionado como pai para cada vértice (linha 6).

Algoritmo 3: Algoritmo para aprendizado de terminologias probabilísticas.

Entrada: Uma base de conhecimento inicial $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, conceito alvo C_a e um conjunto de treinamento $\mathcal{E}$.

Saída: Definição ou inclusão probabilística para C_a

```

1  ArvoreBusca com um nó  $\{C_h = \top, h = 0\}$ ;
2  enquanto critério de parada faça
3      escolha o nó  $N = \{C_h, h\}$  com a melhor pontuação probabilística na
        ArvoreBusca;
4      expandir nó a largura  $h + 1$ ;
5      início
6          adicionar todos os nós  $D \in (\text{OperadorRefinamento}(C_h))$  com largura
             $= h + 1$ ;
7          aprender parâmetros para todos os nós  $D$ ;
8           $N = \{C_h, h + 1\}$ ;
9          expandir nós alternativos de acordo com fator de expansão horizontal e
             $h + 1$  (LEHMANN; HITZLER, 2008b);
10     fim
11   $N' =$  melhor nó em ArvoreBusca;
12  se pontuação( $N'$ ) > limiar então
13      retornar conceito determinístico  $C'_h \in N'$ ;
14  senão
15      chamar InclusãoProbabilística(ArvoreBusca,  $C_a$ );

```

Os parâmetros probabilísticos são aprendidos (linha 7). Este classificador TAN aprendido é usado para avaliar os candidatos possíveis para inclusões probabilísticas e o melhor é retornado (linha 8).

Algoritmo 4: Algoritmo para aprendizado inclusões probabilísticas.

Entrada: ArvoreBusca previamente calculado e conceito alvo C_a

Saída: Inclusão probabilística.

- 1 **para cada** cada par de candidatos C_{h_i}, C_{h_j} nos primeiros k nós da ArvoreBusca **faça**
 - 2 └ calcular a informação mútua condicional $I(C_{h_i}, C_{h_j} | C_a)$;
 - 3 construir um grafo não direcionado no qual vértices são os k candidatos;
 - 4 anotar o peso na aresta que conecta C_{h_i} a C_{h_j} pelo valor $I(C_{h_i}, C_{h_j} | C_a)$;
 - 5 construir um maximum weight spanning tree deste grafo;
 - 6 adicionar C_a como pai para cada C_{h_i} ;
 - 7 aprender probabilidades para $P(C_{h_i} | pa(C_{h_i}))$;
 - 8 retornar a maior inclusão probabilística $P(C_a | C'_h) = \alpha$;
-

Exemplo 11. *Dando continuidade ao Exemplo 9, sob as mesmas condições. O algoritmo inicia a busca pela descrição do conceito alvo. Após uma série de iterações os melhores candidatos para o conceito Professor, são os seguintes:*

$$\exists \text{orienta.T} \quad (1)$$

$$\text{Pessoa} \quad (2)$$

$$\exists \text{publica.Artigo} \quad (3)$$

$$\text{Pessoa} \sqcup \exists \text{publica.Artigo} \quad (4)$$

Neste caso, suponha que a melhor definição encontrada (1) não cubra todos os exemplos positivos. Portanto, é iniciada a busca pela inclusão probabilística correspondente. A esse respeito cada um dos quatro candidatos participam no aprendizado do classificador TAN, isto é, independência mútua entre eles é calculada, o as probabilidades estimadas. Em seguida, as probabilidades $P(\text{Professor} | C_{h_i}) = \alpha$ para $1 \leq i \leq 4$ são avaliadas e a inclusão probabilística retornada:

$$P(\text{Professor} | \text{Pessoa} \sqcap \exists \text{orienta.T}) = \alpha.$$

5.7 Consistência do aprendizado em $\text{CR}\mathcal{ALC}$

Diferentemente de lógicas de descrição onde a consistência esta associada ao aprendizado de terminologias que sejam satisfazíveis, isto é, $\mathcal{K} \not\models \perp$, a consistência dos algoritmos

propostos é dada pelo aprendizado efetivo de terminologias em $CR\mathcal{ALC}$. Isto é, o algoritmo é consistente se consegue especificar construções da lógica de descrição $CR\mathcal{ALC}$. Em síntese, as condições que devem ser seguidas para garantir a consistência da linguagem aprendida são:

- (i) As construções lógicas aprendidas são da lógica $\mathcal{ALC}$.
- (ii) Não existem definições de conceitos cíclicas.
- (iii) No lado esquerdo das inclusões probabilísticas devem ser permitidas apenas nomes de conceitos.
- (iv) Um conceito pode ser definido ou por uma descrição lógica ou por inclusão probabilística mas não por ambas (salvo quando ambas definições são idênticas o que não interfere na construção da rede Bayesiana relacional)
- (v) A linguagem assim definida deve permitir, após a construção da rede-t, a definição de uma rede Bayesiana relacional e portanto a definição de uma distribuição de probabilidades.

5.7.1 Condições no algoritmo probabilístico

Apesar do algoritmo ser baseado em um processo probabilístico, o algoritmo constrói as soluções baseado em um operador de refinamento (LEHMANN; HITZLER, 2008b) para $\mathcal{ALC}$. Portanto a condição (i) é cumprida. Os candidatos de conceitos retornados por esse mesmo operador não são cíclicos e o classificador Noisy-OR usa essas hipóteses como nós. O fato de usar uma rede Bayesiana de independência causal impossibilita a aciclidade entre hipóteses, portanto a condição (ii) é cumprida. As inclusões probabilísticas neste algoritmo são simples, apenas algumas definições lógicas são transformadas em inclusões probabilísticas e portanto as condições (iii) e (iv) são atingidas.

5.7.2 Condições no algoritmo de inclusões probabilísticas

A primeira etapa deste algoritmo usa um algoritmo de aprendizado de lógica de descrições tradicional. Portanto as condições (i) e (ii) dependem da escolha do algoritmo para esta tarefa. Em particular, o algoritmo apresentado em (LEHMANN; HITZLER, 2008b) satisfaz essas condições. Na etapa da construção das inclusões probabilísticas são avaliados apenas conceitos aprendidos ou presentes numa terminologia consistente. Nesse sentido,

o lado esquerdo da inclusão probabilística é sempre um nome de conceito portanto a condição (iii) é atingida. É possível que após a análise de candidatos para inclusões probabilísticas existam definições e inclusões probabilísticas conflitantes (diferentes definições para um mesmo conceito). No algoritmo é avaliada a melhor opção a ser mantida na terminologia probabilística e portanto a condição (iv) é atingida.

5.7.3 Condições no algoritmo combinado

Neste algoritmo, embora a construção de definições ou inclusões probabilísticas seja feita de modo integrado as duas formas de aprendizado são identificáveis. O objetivo inicial é a busca por uma definição de conceito, a qual usa operadores de refinamento de $\mathcal{ALC}$ (condição (i) atingida). O algoritmo que percorre os diferentes candidatos obtidos pelo operador de refinamento constrói definições acíclicas (apesar do procedimento usar cobertura e pontuação probabilística), e portanto a condição (ii) é satisfeita. O algoritmo de inclusões probabilísticas baseado no classificador TAN avalia possíveis candidatos para definições aprendidas na primeira parte do algoritmo. Deste modo apenas nomes de conceitos são considerados na parte condicional e portanto a condição (iii) é atingida. A escolha entre uma definição determinística ou o aprendizado das probabilidades para uma inclusão probabilística é realizada internamente pelo algoritmo via a aplicação de um limiar. Portanto, é impossível que existam nomes de conceitos conflitantes em definições e inclusões probabilísticas assim a condição (iv) é atingida.

5.8 Considerações finais

O aprendizado de máquina de formalismos lógico-probabilísticos foi o foco desse capítulo. De maneira sucinta foi descrita a abordagem da programação em lógica indutiva probabilística em tarefas de aprendizado. Alguns dos conceitos apresentados foram adaptados para o aprendizado da lógica de descrição probabilística $\text{CR}\mathcal{ALC}$ — principal contribuição deste trabalho. Três algoritmos foram propostos para o aprendizado de $\text{CR}\mathcal{ALC}$.

O primeiro (OCHOA-LUNA; COZMAN, 2009), enfatizou o aprendizado de terminologias de maneira probabilística. O processo de aprendizado foi transformado no processo de aprendizado de um classificador que induz a definição de conceitos por meio da classificação dos exemplos positivos e negativos fornecidos. O classificador Noisy-OR foi usado como classificador probabilístico.

A segunda proposta (REVOREDO; OCHOA-LUNA; COZMAN, 2010), focou no aprendizado das inclusões probabilísticas a partir de uma terminologia determinística, a qual poderia ter sido aprendida ou construída por um especialista. Dada as definições de conceitos, esse algoritmo busca por condicionantes entre as definições de conceitos, e avalia possíveis componentes via uma função de pontuação. No final deste procedimento, caso a inserção de inclusões probabilísticas melhore a pontuação global, as inclusões probabilísticas são mantidas; caso contrário são descartadas.

A terceira proposta (OCHOA-LUNA; REVOREDO; COZMAN, 2010a), parte do pressuposto que os dois tipos de componentes, probabilístico e determinístico, têm a mesma relevância e a natureza probabilística ou determinística dos conceitos está ligada aos dados. Isto é, os exemplos positivos e negativos dão indícios da natureza dos componentes da terminologia. O aprendizado é realizado sob uma base probabilística e usam-se funções de cobertura e pontuações probabilísticas, e inicia-se a busca pela definição determinística de conceitos. Após uma série de iterações decide-se pelo fim da busca ou pelo início do processo de aprendizado de inclusão probabilística.

Uma análise sucinta de cada algoritmo de ponto de vista da aderência em relação à linguagem $\text{CR}\mathcal{ALC}$ foi realizada. Cabe ressaltar que o aprendizado de lógicas de descrição probabilísticas é um tópico pouco ou nada explorado na literatura. Em particular, os algoritmos para o aprendizado de $\text{CR}\mathcal{ALC}$ foram as primeiras propostas até o momento. No próximo capítulo serão apresentados alguns resultados empíricos da aplicação destes algoritmos em conjuntos de dados artificiais e reais.

6 EXPERIMENTOS

Neste capítulo são apresentados os experimentos conduzidos para verificar os benefícios dos algoritmos propostos. Em particular, aprendizado de lógica de descrição probabilística é um tópico que tem sido pouco explorado na literatura.

Dois tipos de experimentos foram realizados. O primeiro avalia os algoritmos propostos em relação ao aprendizado de lógicas de descrição (sem inclusões probabilísticas). O segundo avalia os algoritmos propostos em relação ao aprendizado de terminologias probabilísticas em CRALC . Conjuntos de dados coletados de repositórios de informações públicas disponíveis na Internet foram usados.

O capítulo é organizado da seguinte maneira. A Seção 6.1 detalha a metodologia usada na realização dos experimentos. Os conjuntos de dados usados são descritos na Seção 6.2. A comparação entre os algoritmos propostos para aprendizado é detalhado na Seção 6.3. Considerações finais são apresentadas na Seção 6.4.

6.1 Metodologia

Avaliar empiricamente a acurácia de hipóteses é fundamental em aprendizado de máquina (MITCHELL, 1997). Nesse sentido, dois tipos de experimentos foram conduzidos.

O objetivo do primeiro tipo de experimento consiste em testar as propostas para aprendizado de terminologias probabilísticas em aprendizado de lógicas de descrição. Entretanto, mesmo no aprendizado de ontologias a partir de exemplos, há falta de padrões de avaliação (LEHMANN, 2007)¹. Os algoritmos propostos foram comparados em termos de *acurácia* com os resultados obtidos por Lehmann e Hitzler (LEHMANN; HITZLER, 2008b)². Em todos os casos foram usados *5-folds* validação cruzada³. O teste de significância usado

¹Em alguns casos, conhecimento base de diversos problemas de aprendizado geral foram adaptados para o aprendizado de ontologias (LEHMANN; HITZLER, 2008b).

²Os melhores resultados disponíveis na literatura.

³Validação cruzada estratificada (mantendo-se o balanceamento das classes) consiste em dividir o conjunto de treinamento em k partes aproximadamente iguais. Uma dessas partes será o subconjunto a

foi t-test com intervalos de confiança de 95%.

O segundo tipo de experimentos avaliou a capacidade de aprendizado de terminologias probabilísticas. Cada uma das propostas de aprendizado foi testada em termos de *acurácia* usando conjunto de dados reais. Estes resultados possibilitaram a comparação dos algoritmos propostos. Validação cruzada foi usada e intervalos de confiança de 95%.

Para avaliar essas propostas podem ser usadas métricas básicas em recuperação de informação, tais como *precisão*, *revocação* e *acurácia*. Em particular, *acurácia* foi adotada nas comparações por ser uma métrica de efetividade usada frequentemente para avaliar problemas de classificação em aprendizado de máquina (MANNING; RAGHAVAN; SCHÜTZE, 2008). A *acurácia* mede a fração de classificações corretas. A noção de *acurácia* fica clara se examinada a seguinte tabela:

	relevante	não relevante
recuperado	verdadeiros positivos (vp)	falsos positivos (fp)
não recuperado	falsos negativos (fn)	verdadeiros negativos (vn)

Nesse sentido, a *acurácia* é dada por:

$$\text{acurácia} = \frac{vp + vn}{vp + fp + fn + vp}.$$

Os algoritmos de aprendizado e os cenários para validação cruzada foram implementados na linguagem Java. Muitas das rotinas foram construídas usando como base as APIs das ferramentas DL-Learner ⁴ e JavaBayes ⁵.

O banco de dados MySQL ⁵ ⁶ foi usado para armazenar padrões encontrados durante o *parsing* de páginas HTML usadas posteriormente na construção dos exemplos positivos e negativos durante o treinamento e teste dos algoritmos de aprendizado. Os experimentos foram executados em um computador CORE 2 DUO 2.2 GHz com 4GB de memória na plataforma Ubuntu Linux.

ser utilizado para teste. A cada execução do experimento (*fold*) esse conjunto muda. O erro médio da validação cruzada é computado a partir da média aritmética dos erros fornecidos por cada conjunto de testes.

⁴<http://dl-learner.org>

⁵<http://www.pmr.poli.usp.br/ltd/Software/javabayes/Home/>

⁶www.mysql.com/

6.2 Dados

Dois tipos de conjuntos de dados foram usados nas experimentações. O primeiro tipo, denominado de repositório, contém conjuntos de dados previamente usados na comparação de algoritmos de aprendizado de lógica de descrição. O segundo tipo, denominado de dados reais, contém dados coletados de fontes públicas disponíveis na Internet e foram usados no aprendizado de terminologias probabilísticas e na realização de comparações entre as propostas. Esses dados passaram pelas etapas de seleção, pre-processamento (limpeza dos dados e normalização) e transformação (por exemplo, nomes transformados em códigos para identificar instâncias). Cada tipo de conjunto de dados é detalhado a seguir.

6.2.1 Dados de repositórios

Os dados de repositórios são conjuntos de dados usados no aprendizado de lógicas de descrições por trabalhos precedentes e que apresentam resultados empíricos. Duas fontes principais foram consultadas: DL-FOIL (FANIZZI; D'AMATO; ESPOSITO, 2008) e DL-Learner (LEHMANN; HITZLER, 2010). Esses conjuntos incluem: o problema dos trens, o problema de aprendizado de mão de poker e raciocínio moral em seres humanos.

A Tabela 3 mostra as características dos dados de repositórios usados nos experimentos.

Tabela 3: Dados de repositórios.

Problema	axiomas	conceitos	papéis	exemplos
trains	252	8	5	10
moral I	2176	43	4	43
moral II	2107	40	4	43
poker I	1335	2	6	49
poker II	1419	2	6	55

Na tabela 3 estão indicadas a quantidade de conceitos, papéis, exemplos e *axiomas* disponíveis no conjunto de dados. Em problemas de aprendizado de lógica de descrição (assim como em ILP) é comum a base de conhecimento ser dada de maneira explícita, isto é, na forma de asserções. Por isso, a quantidade de axiomas (quantidade de definições e asserções) é significativa.

6.2.2 Dados reais

Dados reais foram extraídos de fontes públicas disponíveis na Internet o que possibilitou a comparação dos algoritmos de aprendizado de *CRACC*, a seguir é detalhado cada conjunto de dado usado.

6.2.2.1 Conjunto de dados Lattes

Este conjunto de dados foi construído a partir de informação disponível na plataforma de currículos Lattes ⁷. Este repositório foi escolhido por fornecer uma quantidade considerável de dados relacionais reais e de acesso público.

Como os currículos estão em HTML, foi realizado um procedimento preliminar de *parsing* para extrair conteúdo na forma de asserções sobre conceitos e relacionamentos. Em particular, foram extraídos dados de cada pesquisador tais como: nome, instituição, idiomas, endereço, etc. Foi dada ênfase à extração de informação relacional entre pesquisadores mediante a análise de publicações, supervisão de trabalhos, participação em bancas examinadores, etc. Posteriormente, esse conjunto foi normalizado e cada instância foi identificada de maneira única. Por exemplo, a cada pesquisador foi atribuído um código único que possibilita sua identificação nos diferentes papéis.

Embora o processo para seleção dos currículos tenha sido aleatório, foi focado um universo restrito a pesquisadores da USP⁸. Para efeito das comparações realizadas foram selecionados um total de 1050 pesquisadores. A Tabela 4 detalha algumas das características dos conjuntos de dados extraídos da plataforma Lattes.

Tabela 4: Conjunto de dados Lattes.

Problema	axiomas	conceitos	papéis	exemplos
lattes I	2494	14	12	208
lattes II	2603	14	10	146
lattes III	2650	13	10	230
lattes IV	2350	2	13	34

⁷<http://lattes.cnpq.br>.

⁸Embora os currículos Lattes sejam de acesso público não é uma tarefa trivial coletar currículos individuais específicos.

6.2.2.2 Conjunto de dados Wikipedia-YAGO

Os conjuntos de dados usados na comparação dos algoritmos propostos no aprendizado de $\mathcal{CRALC}$ são baseados em informações disponíveis na Wikipedia⁹ coletadas na ontologia YAGO (SUCHANEK; KASNECI; WEIKUM, 2007).

Artigos de Wikipedia consistem fundamentalmente de texto livre, porém também é possível encontrar diversos tipos de informação estruturada na forma de marcadores Wiki. Tal informação inclui estruturas denominadas *infobox*, categorização, imagens geocoordenadas, ligações para páginas Web externas, páginas de desambiguação, redirecionamentos entre páginas, e ligações entre diferentes edições da Wikipedia.

Nos últimos anos, houve diversos projetos que tentaram estruturar essa enorme fonte de conhecimento. Um exemplo é o projeto DBpedia (BIZER *et al.*, 2009) o qual extrai informação estruturada de Wikipedia e a transforma em uma rica base de conhecimento. Outro exemplo é o projeto YAGO, uma enorme base de conhecimento baseada tanto em Wikipedia quanto em WordNet¹⁰. Atualmente, YAGO é composta por mais de 2 milhões de entidades (tais como pessoas, organizações, cidades, etc.). Também disponibiliza cerca de 20 milhões de asserções sobre essas entidades. Diferentemente de muitas bases construídas de forma automática, YAGO tem sido confirmada manualmente e tem uma acurácia de 95%. Diversos domínios desde filmes, locais, eventos históricos, vinhos, etc. foram considerados nesta ontologia. Existem aproximadamente 92 relações disponíveis. Exemplos incluem *atuoEm*, *nasceuEm*, *criou*, *descobriu*, *descreve*, *morreuEm*, *aconteceuEm*, *temOrientadorAcademico*, *temFilho*, *ganhouPremio*, *influencia*, *casadoCom*, *moraEm*, *eParteDe*, *politicoDe*, *trabalhaEm*.

Nos experimentos não foi usada a totalidade destes dados. Foram construídos subconjuntos de dados YAGO associados a dois domínios: o primeiro, relacionado a cientistas (1335); o segundo, relacionado a diretores de filmes (2000). A Tabela 5 descreve algumas das características dos conjuntos de dados extraídos.

6.3 Experimentos

Nesta seção são mostrados os resultados dos experimentos realizados com os dados de repositórios e com os dados reais.

⁹<http://www.wikipedia.org/>

¹⁰wordnet.princeton.edu/

Tabela 5: Conjunto de dados Wikipedia-YAGO.

Problema	axiomas	conceitos	papéis	exemplos
cientistas I	7180	11	32	438
cientistas II	6534	11	32	378
diretores I	8006	12	35	626
diretores II	7234	12	35	389

6.3.1 Experimentos com dados de repositórios

O objetivo desta primeira série de experimentos é verificar se os algoritmos de aprendizado propostos para o aprendizado de $\text{CR}\mathcal{ALC}$ também são competitivos em relação a abordagens tradicionais no aprendizado de lógica de descrição — embora esse não seja o objetivo final dos algoritmos propostos. Deste modo, foi possível realizar comparações com trabalhos próximos ao aprendizado de terminologias probabilísticas dado que ainda não existem casos de uso padrões para aprendizado de lógicas de descrição probabilísticas. Com os conjuntos de dados da Seção 6.2.1 foram executados 5-*fold* validação cruzada¹¹. Foram medidas a acurácia das soluções assim como o comprimento das definições induzidas. A Tabela 6 mostra os resultados da execução de dois algoritmos propostos (probabilístico baseado em Noisy-OR e abordagem combinada) comparado com o melhor algoritmo (DL-Learner) para os diversos conjuntos de dados.

Tabela 6: Resultados do aprendizado de lógica de descrição. Acur. (Acurácia) e Comp. (comprimento da solução encontrada)

Problema	DL-Learner		Noisy		Combinado	
	Acur.	Comp.	Acur.	Comp.	Acur.	Comp.
trains	100	5	99.5	7	100	6
moral I	97.8	3	96.5	5	97.7	4
moral II	97.8	8	96.0	8	97.6	8
poker I	100	5	98.8	6	100	5
poker II	100	11	99.7	12	100	11

A melhor solução é a que fornece a maior acurácia e o menor comprimento. Da Tabela 6 pode ser observado que o algoritmo Noisy (probabilístico Noisy-OR) em termos de acurácia teve um desempenho ligeiramente inferior a DL-Learner¹², Por outro lado, o

¹¹Para efeitos de comparação pois essas foram as características usadas por DL-Learner.

¹²Diferenças estatisticamente significativas.

algoritmo combinado (abordagem combinada) teve desempenho muito próximo ao ideal — sem diferenças significativas em relação ao algoritmo DL-Learner. Talvez a diferença maior seja notada no comprimento das soluções encontradas. Existe uma preocupação maior na construção de soluções curtas do DL-Learner comparado ao algoritmo *Noisy* — o algoritmo DL-Learner penaliza soluções compridas. Por outro lado, o algoritmo *Combinado* encontra soluções adequadas pois utiliza também mecanismos de penalização.

Embora o objetivo dos algoritmos propostos seja pelo aprendizado de terminologias probabilísticas nota-se que os dois algoritmos testados (principalmente o algoritmo combinado) são competitivos para aprendizado de lógicas de descrições. Cabe ressaltar, porém, que em alguns casos as definições retornadas pelos algoritmos propostos podem não ser significativas em lógica de descrição e que apenas façam sentido quando acompanhadas do componente probabilístico.

6.3.2 Experimentos com dados reais

Nesta seção são comparados três algoritmos propostos para aprendizado de lógica de descrição probabilística *CRALC* usando dados reais definidos na Seção 6.2.2. Dado que não existem na literatura propostas para este tipo de aprendizado as comparações são realizadas em termos de acurácia e aderência dos modelos aprendidos em relação aos dados.

Na Tabela 7 são apresentados resultados experimentais dos conjuntos de dados Lattes. Após 10-*fold* validação cruzada, as comparações foram realizadas em termos de acurácia (Acur.) e a quantidade de nós gerados (# Nós) ao construir a rede Bayesiana relacional. O melhor modelo aprendido tem a maior acurácia e uma quantidade menor de nós. O algoritmo Noisy apesar de ter os menores valores de acurácia retornou a menor quantidade de nós possíveis. O algoritmo Inclusão (foca o aprendizado de inclusões probabilísticas) obteve melhores resultados em termos de acurácia do que o algoritmo Noisy, porém com uma quantidade de nós maior. Finalmente, o algoritmo Combinado obteve os melhores resultados em termos de acurácia, com o quantidade de nós proxima do algoritmo Noisy.

A Tabela 8 exhibe os resultados obtidos no segundo conjunto de dados (Wikipedia-YAGO). Foram realizados 10-*folds* de validação cruzada onde foram avaliados a acurácia e a quantidade de nós obtidos após a geração da rede Bayesiana relacional. O problema de aprendizado neste conjunto de dados é mais complexo, o qual se reflete nos valores de acurácia (inferiores ao conjunto de dados Lattes) e ao número maior de nós obtido. A tendência porém é mantida em relação aos algoritmos avaliados. Isto é, o algoritmo

Tabela 7: Resultados do aprendizado $\text{CR}\mathcal{ALC}$ (Lattes). Acur. (acurácia) e # Nós (quantidade nós da rede Bayesiana relacional).

Problema	Noisy		Inclusão		Combinado	
	Acur.	# Nós	Acur.	# Nós	Acur.	# Nós
lattes I	76.74	48	76.84	51	77.13	49
lattes II	69.52	54	72.77	58	74.42	56
lattes III	73.48	56	74.68	62	78.45	58
lattes IV	71.5	78	71.56	110	72.34	75

Combinado obteve os melhores resultados em relação à acurácia e a quantidade de nós próxima ao algoritmo Noisy. O algoritmo Inclusão, por outro lado, teve melhor acurácia e quantidade de nós próxima do algoritmo Noisy.

Tabela 8: Resultados do aprendizado $\text{CR}\mathcal{ALC}$ (Wikipedia-YAGO). Acur. (acurácia) e # Nós (quantidade nós da rede Bayesiana relacional).

Problema	Noisy		Inclusão		Combinado	
	Acur.	# Nós	Acur.	# Nós	Acur.	# Nós
cientistas I	81.4	53	82.33	56	83.44	55
cientistas II	82.3	65	82.46	69	84.55	68
diretores I	65.67	67	67.14	69	73.56	66
diretores II	69.45	74	71.3	79	74.75	76

Mesmo após a aplicação de dois conjuntos de dados diferentes o algoritmo Combinado mostrou melhor desempenho no aprendizado de terminologias probabilísticas. O algoritmo apresentou um balanceamento entre acurácia e complexidade estrutural.

A seguir são apresentadas algumas das terminologias probabilísticas que foram aprendidas durante os experimentos. Alguns *folds* foram selecionados para cada algoritmo assim como conjuntos de dados particulares.

6.3.2.1 Terminologia probabilística obtida pelo algoritmo Noisy-OR

Nesta seção é apresentado um exemplo de aprendizado de terminologia probabilística realizado usando dados extraídos da plataforma de currículos Lattes e o algoritmo probabilístico Noisy-OR. O conjunto de dados usado foi o Lattes II (Seção 6.2.2.1).

Neste exemplo as relações estiveram restritas a modelar conceitos como **Professor**, **Pessoa**, **Publicação** e **Organização**. O aprendizado realizado teve como base as asserções sobre tais conceitos e papéis definidos entre eles. As inclusões probabilísticas foram adicionadas de acordo com a sintaxe de $\text{CR}\mathcal{ALC}$ e seguindo as diretrizes dadas na Seção 5.3.

Por exemplo, para identificar de forma apropriada um professor, a seguinte descrição do conceito foi induzida:

Professor $\equiv$ **Pessoa**

$\sqcap(\exists \text{temPublicação.Publicação} \sqcup \exists \text{orienta.Pessoa} \sqcup \exists \text{trabalhaEm.Organização}).$

Quando não existe evidência disponível, o valor de probabilidade é $P(\text{Professor}(0)) = 0.68^{13}$ (uma quantidade maior de currículos de professores foi considerado nos experimentos), à luz de novas evidências o valor de probabilidade muda:

$$P(\text{Professor}(0)|\exists \text{temPublicação}(1)) = 0.72,$$

e

$$P(\text{Professor}(0)|\exists \text{temPublicação}(1) \sqcup \exists \text{orienta}(1)) = 0.75.$$

Outra consulta, relacionada à classificação relacional, permite identificar publicações duplicadas. Pode haver interesse em recuperar o número de publicações de um determinado grupo de pesquisa. Embora essa tarefa possa parecer trivial, dificuldades podem surgir devido a documentos elaborados em co-autoria. A princípio, cada co-autor poderia fazer o cadastro da mesma publicação na plataforma Lattes, talvez com diferente formato ou até com a possibilidade de inserção de erros. A esse respeito, um conceito probabilístico para publicações duplicadas foi aprendido:

PublicaçãoDuplicada $\equiv$ **Publicação**

$\sqcap(\exists \text{temTítuloParecido.Publicação} \sqcup \exists \text{temMesmoAno.Publicação} \sqcup \exists \text{temMesmoTipo.Publicação}).$

Esta definição denota que uma publicação duplicada está relacionada a publicações que possuem um título parecido, o mesmo ano e tipo (artigo de periódico, capítulo de livro, etc). No início, a probabilidade *a priori* é baixa: $P(\text{PublicaçãoDuplicada}(0)) = 0.05$. Evidência na similaridade de títulos aumenta consideravelmente o valor de probabilidade:

$$P(\text{PublicaçãoDuplicada}(0)|\exists \text{temTítuloParecido}(0, 1)) = 0.77.$$

¹³Nestes cálculos, índices $0, 1 \dots n$ representam indivíduos.

Maior evidência relacionada ao tipo da publicação quase garante a existência de um conceito duplicado:

$$P(\text{PublicaçãoDuplicada}(0) | \exists \text{itemTítuloParecido}(1) \sqcap \exists \text{itemMesmoTipo}(1)) = 0.99.$$

Cabe salientar que a similaridade entre títulos não garante a existência de duplicidade de documentos. Dois documentos podem partilhar o mesmo título (mesmo autor), mas nada impede que tenham sido publicados em meios diferentes (por exemplo, um artigo de conferência e um artigo de periódico estendido). Raciocínio probabilístico é útil para lidar com tais questões.

6.3.2.2 Terminologia probabilística obtida pelo algoritmo de inclusões probabilísticas

O seguinte exemplo demonstra algumas das características do algoritmo de aprendizado de inclusões probabilísticas. O conjunto de dados usados foi o Lattes II (6.2.2.1). Em particular, foi focada a informação bibliográfica e a participação em bancas dos pesquisadores. Descrições de conceitos, ou seja a terminologia inicial foi aprendida usando DL-Learner¹⁴. Algumas das informações que foram consideradas são as seguintes.

Pesquisador(r1), Pesquisador(r2), Pesquisador(r4), ...
 foiOrientado(r8, r179), foiOrientado(r30, r83), foiOrientado(r33, r1), ...
 mesmaPublicação(r1, r32), mesmaPublicação(r4, r12), mesmaPublicação(r5, r115), ...
 mesmaBanca(r1, r32), mesmaBanca(r4, r12), ...
 mesmaInstituição(r1, r27), mesmaInstituição(r1, r28), ...
 orienta(r1, r33), orienta(r1, r171), orienta(r1, r81), ...

O objetivo é aprender uma terminologia probabilística que melhor explique um conjunto de padrões de colaboração entre pesquisadores. O aprendizado foi realizado em duas etapas. Na primeira, foram aprendidas as definições de conceitos (tais como *Colaborador Proximo*, *PesquisadorMobilidadeNula*, *PesquisadorFortementeRelacionado*) dados exemplos positivos e negativos. Os dois indivíduos r_1 e r_4 poderiam ser exemplos positivos do conceito *ColaboradorProximo*. Intuitivamente, um colaborador próximo é um pesquisador que tem uma publicação em comum com um colega da mesma instituição. Por outro lado, uma instituição pode estar interessada em saber sobre a mobilidade de seus antigos estudantes; isto é, se eles permaneceram ou não como membros da instituição após a conclusão do

¹⁴<http://dl-learner.org/Projects/DLLearner>

doutorado. O conceito `PesquisadorMobilidadeNula` pretende definir um pesquisador com mobilidade nula. Os conceitos `PesquisadorFortementeRelacionado` e `PesquisadorLegado` definem categorias de relacionamentos entre pesquisadores.

Como especificado anteriormente, DL-Learner foi utilizado para gerar os conceitos iniciais usando conhecimento base e exemplos. Uma terminologia preliminar é:

<code>ColaboradorProximo</code>	$\equiv$	<code>Pesquisador</code> $\sqcap (\exists \text{mesmaInstituição}.\exists \text{mesmaPublicação}.$ $\exists \text{mesmaPublicação}.\exists \text{orienta}.\text{Pesquisador}) \sqcap$ $(\exists \text{mesmaPublicação}.\exists \text{mesmaInstituição}.$ $\exists \text{mesmaPublicação}.\text{Pesquisador})$
<code>ColaboradorProximoFaculdade</code>	$\equiv$	<code>ColaboradorProximo</code> $\sqcap \exists \text{mesmaBanca}.\text{Pesquisador}$
<code>PesquisadorMobilidadeNula</code>	$\equiv$	<code>Pesquisador</code> $\sqcap (\exists \text{mesmaInstituição}.\exists \text{mesmaBanca}.$ $\exists \text{foiOrientado}.\forall \text{orienta}.\exists \text{mesmaInstituição}.$ $\text{Pesquisador}) \sqcap (\exists \text{foiOrientado}.$ $\exists \text{mesmaInstituição}.\text{Pesquisador})$
<code>PesquisadorFortementeRelacionado</code>	$\equiv$	<code>Pesquisador</code> $\sqcap (\exists \text{mesmaPublicação}.\text{Pesquisador} \sqcap$ $\exists \text{foiOrientado}.\text{Pesquisador})$
<code>PesquisadorLegado</code>	$\equiv$	<code>Pesquisador</code> $\sqcap (\exists \text{mesmaBanca}.\text{Pesquisador} \sqcap$ $\exists \text{foiOrientado}.\text{Pesquisador}).$

O próximo passo foca no aprendizado de inclusões probabilísticas. Baseado nas definições de conceitos, as possíveis inclusões probabilísticas para o conceito `ColaboradorProximo` são:

$$\begin{aligned}
 &P(\text{ColaboradorProximo} | \text{Pesquisador} \sqcap (\exists \text{mesmaInstituição}.\exists \text{mesmaPublicação}.\exists \text{mesmaPublicação}.\exists \text{orienta}.\text{Pesquisador}) \sqcap (\exists \text{mesmaPublicação}.\exists \text{mesmaInstituição}.\exists \text{mesmaPublicação}.\text{Pesquisador})) = 0.145, \\
 &P(\text{ColaboradorProximo} | \text{Pesquisador}) = 0.56, \\
 &P(\text{ColaboradorProximo} | \text{Pesquisador} \sqcap \exists \text{mesmaInstituição}.\exists \text{mesmaPublicação}.\exists \text{mesmaPublicação}.\exists \text{orienta}.\text{Pesquisador}) = 0.487 \\
 &P(\text{ColaboradorProximo} | \text{Pesquisador} \sqcap \exists \text{mesmaPublicação}.\exists \text{mesmaInstituição}.\exists \text{mesmaPublicação}.\text{Pesquisador}) = 0.78.
 \end{aligned}$$

Baseado na pontuação probabilística, o último candidato foi escolhido. Esta escolha implica que a descrição de conceito original deve ser removida. A terminologia probabilística completa é dada por:

$P(\text{Pesquisador}) = 1.0$	$P(\text{foiOrientado}) = 0.29$
$P(\text{mesmaInstituição}) = 0.83$	$P(\text{mesmaPublicação}) = 0.73$
$P(\text{mesmaBanca}) = 0.41$	
$P(\text{ColaboradorProximo}$	$ \text{Pesquisador} \sqcap \exists \text{mesmaPublicação}.\exists \text{mesmaInstituição}.$
	$\exists \text{mesmaPublicação}.\text{Pesquisador}) = 0.78$
$\text{ColaboradorProximoFaculdade} \equiv$	$\text{ColaboradorProximo}$
	$\sqcap \exists \text{mesmaBanca}.\text{Pesquisador}$
$P(\text{PesquisadorMobilidadeNula}$	$ \text{Pesquisador} \sqcap \exists \text{foiOrientado}.$
	$\exists \text{mesmaInstituição}.\text{Pesquisador}) = 0.98$
$\text{PesquisadorFortementeRelacionado} \equiv$	Pesquisador
	$\sqcap (\exists \text{mesmaPublicação}.\text{Pesquisador} \sqcap$
	$\exists \text{foiOrientado}.\text{Pesquisador})$
$\text{PesquisadorLegado} \equiv$	Pesquisador
	$\sqcap (\exists \text{mesmaBanca}.\text{Pesquisador} \sqcap$
	$\exists \text{foiOrientado}.\text{Pesquisador})$

Pode-se observar que *PesquisadorFortementeRelacionado* e *PesquisadorLegado* não têm inclusões probabilísticas. Por outro lado, os conceitos *ColaboradoProximo* e *PesquisadorMobilidadeNula* foram substituídos por inclusões probabilísticas adequadas.

Para investigar a aplicação dos conceitos aprendidos, algumas inferências foram realizadas na terminologia resultante. Um algoritmo exato (COZMAN; POLASTRO, 2008) foi utilizado. Quando *Pesquisador*(0) e *Pesquisador*(1) são dados como evidência, o valor de probabilidade obtido é $P(\text{PesquisadorMobilidadeNula}(0)) = 0.46$. Quando maior evidência é dada o valor muda:

$$P(\text{PesquisadorMobilidadeNula}(0)|\exists \text{foiOrientado}(0,1)) = 0.83,$$

e para:

$$P(\text{PesquisadorMobilidadeNula}(0)|\exists \text{foiOrientado}(0,1) \sqcap \exists \text{mesmaInstituição}(0,1)) = 0.9.$$

Este último valor de probabilidade indica que o indivíduo 0 poderia ser classificado como *PesquisadorMobilidadeNula*. (Note que dada a mesma evidência, a ontologia deter-

minística original classificaria de forma errada o indivíduo 0.)

6.3.2.3 Terminologias probabilísticas obtidas pelo algoritmo combinado

O algoritmo Combinado foi aplicado em dois conjuntos de dados da Wikipedia-YAGO (Cientistas I e Diretores II, Seção 6.2.2.2) para o aprendizado de terminologias probabilísticas. Em ambos casos o limiar usado foi de 0.85 e os 20 melhores candidatos foram considerados na etapa do aprendizado de inclusões probabilísticas.

A partir do primeiro conjunto de dados, Cientistas, a seguinte terminologia probabilística foi aprendida:

	$P(\text{Pessoa}) = 0.9$
	$P(\text{Tópico}) = 0.4$
	$P(\text{Ano}) = 0.35$
	$P(\text{Premio}) = 0.2$
	$P(\text{Texto}) = 0.25$
	$P(\text{InstituiçãoEducativa}) = 0.3$
	$P(\text{escreveu}) = 0.4$
	$P(\text{temOrientadorAcadêmico}) = 0.80$
	$P(\text{interessadoEm}) = 0.6$
	$P(\text{morreuNoAno}) = 0.7$
	$P(\text{ganhouPremio}) = 0.4$
	$P(\text{trabalhaEm}) = 0.85$
	$P(\text{influencia}) = 0.6$
Cientista $\equiv$	Pessoa
	$\sqcap (\exists \text{temOrientadorAcadêmico.Pessoa}$
	$\sqcap \exists \text{escreveu.Texto} \sqcap \exists \text{trabalhaEm.InstituiçãoEducativa})$
$P(\text{CientistaInfluente}$	$\mid \text{Cientista} \sqcap \exists \text{influencia.}$
	$\exists \text{morreuEmAno.Ano}) = 0.85$
$P(\text{Musico}$	$\mid \text{Pessoa} \sqcap \exists \text{temOrientadorAcadêmico.} \exists \text{escreveu.Texto}) = 0.1$
CientistaHomenageado $\equiv$	Cientista
	$\sqcap \exists \text{ganhouPremio.Premio}$

A terminologia $\text{CR}\mathcal{ALC}$ resultante foi explorada por inferência probabilística. Tarefas de classificação foram pesquisadas. Suponha que existe o interesse em classificar um

potencial cientista dado que é conhecido que ele/ela escreveu um livro e tem um orientador acadêmico:

$$P(\text{Cientista}(0) | \text{Pessoa}(0) \sqcap \exists \text{escreveu.Texto}(1) \sqcap \text{temOrientadorAcadêmico.Pessoa}(2)) = 0.5.$$

Quando maior evidência é disponível o valor de probabilidade é atualizado para:

$$\begin{aligned} P(\text{Cientista}(0) | \text{Pessoa}(0) \\ \sqcap (\exists \text{escreveu.Texto}(1) \sqcap \exists \text{temOrientadorAcadêmico.} \\ \exists \text{influencia.Pessoa}(3))) = 0.65. \end{aligned}$$

No segundo conjunto de dados foram coletados fatos sobre diretores de filmes abrangendo desde os clássicos até os contemporâneos. A terminologia probabilística completa é mostrada a seguir.

	$P(\text{Pessoa}) = 0.9$
	$P(\text{Premio}) = 0.1$
	$P(\text{Ano}) = 0.25$
	$P(\text{Filme}) = 0.3$
	$P(\text{casadoCom}) = 0.1$
	$P(\text{influencia}) = 0.35$
	$P(\text{ganhouPremio}) = 0.28$
	$P(\text{temFilho}) = 0.05$
	$P(\text{morreuNoAno}) = 0.5$
	$P(\text{dirigiu}) = 0.8$
	$P(\text{atuouEm}) = 0.4$
$\text{Ator} \equiv$	$\text{Pessoa} \sqcap \forall \text{atuou.Filme}$
$P(\text{Diretor}$	$\mid \text{Pessoa} \sqcap (\exists \text{dirigiu.Filme} \sqcap \exists \text{influencia.}$
	$\exists \text{atuou.Filme}) = 0.75$
$P(\text{AtorNoPassado}$	$\mid \text{Diretor} \sqcap \exists \text{atuou.Filme}) = 0.6$
$\text{DiretorHomenageado} \equiv$	$\text{Diretor} \sqcap \exists \text{ganhouPremio.Premio}$
$\text{FamiliaDiretor} \equiv$	$\text{Diretor} \sqcap (\exists \text{casadoCom.Diretor} \sqcup \exists \text{temFilho.Diretor})$
$P(\text{DiretorInfluente}$	$\mid \text{Diretor} \sqcap \exists \text{ganhouPremio.Premio} \sqcap \exists \text{influencia.}$
	$\exists \text{casadoCom.Diretor}) = 0.7$
$P(\text{DiretorMaiorInfluencia}$	$\mid \text{Diretor} \sqcap \exists \text{morreuNoAno.Ano} \sqcap \exists \text{influencia.}$
	$\exists \text{ganhouPremio.Premio}) = 0.8$

Componentes aprendidos abrangem desde definições de conceitos básicos tais como **Ator** até inclusões probabilísticas para descrever diretores mais influentes. Suponha que há interesse em classificar uma pessoa dado que é sabido que ele/ ela tem atuado e dirigido filmes. De acordo com a evidência disponível:

$$P(\text{Ator}(0) \mid \text{Pessoa}(0) \sqcap \exists \text{atuouEm.Filme}(1) \sqcap \exists \text{dirigiu.Filme}(2)) = 0.4,$$

$$P(\text{Diretor}(0) \mid \text{Pessoa}(0) \sqcap \exists \text{atuouEm.Filme}(1) \sqcap \exists \text{dirigiu.Filme}(2)) = 0.55,$$

Quando maior evidência é dada, o valor de probabilidade muda:

$$\begin{aligned} P(\text{Ator}(0) \mid \text{Pessoa}(0) \\ \sqcap (\exists \text{atuouEm.Filme}(1) \sqcap \exists \text{dirigiu.Filme}(2) \\ \sqcap \exists \text{influencia.Pessoa}(3))) = 0.3. \end{aligned}$$

6.4 Considerações finais

Neste capítulo foram apresentados resultados empíricos da avaliação dos algoritmos de aprendizado da lógica de descrição *CRALC*.

Dado que não existem trabalhos prévios para realizar comparações, a estratégia adotada foi realizar, inicialmente, avaliações dos algoritmos propostos no aprendizado de lógicas de descrições. Posteriormente foram realizadas comparações dos algoritmos em tarefas de aprendizado de terminologias probabilísticas.

No primeiro caso, os algoritmos obtiveram resultados em termos de acurácia que os tornam competitivos com propostas para o aprendizado de ontologias. No segundo caso, os três algoritmos conseguiram aprender terminologias probabilísticas com diversos graus de acurácia. O algoritmo que realiza o aprendizado de forma combinada obteve os melhores resultados nos dois conjuntos de dados reais testados.

Visando a utilização real dos algoritmos propostos, no próximo capítulo serão discutidos resultados da aplicação destes em tarefas de recuperação de informação.

Nos últimos anos, principalmente em encontros dedicados ao estudo de propostas de raciocínio probabilística para Web semântica, a criação de casos de uso que possibilitem a comparação entre abordagens *fuzzy* e probabilísticas tem sido discutida. Determinar para quais tarefas uma determinada abordagem é mais apropriada tem sido um dos objetivos desses grupos de pesquisa. Espera-se que a definição desses padrões possibilite no futuro comparações entre as diversas propostas para aprendizado de terminologias probabilísticas.

7 APLICAÇÕES

Este capítulo explora a aplicabilidade dos algoritmos de aprendizado propostos. Aprendizado e inferência de ontologias probabilísticas definidas em $\text{CR}\mathcal{ALC}$ são utilizadas em tarefas de recuperação de informação visando a *extensão semântica* de buscas na Web.

A extensão semântica visa melhorar os resultados de buscas na Internet mediante a identificação de conceitos. Nesse sentido, duas abordagens são propostas neste capítulo. Na primeira (Seção 7.1), a extensão semântica é realizada por meio da definição de uma ontologia probabilística em $\text{CR}\mathcal{ALC}$. Na segunda (Seção 7.2), a construção de contexto semântico baseado em $\text{CR}\mathcal{ALC}$ possibilita a extensão semântica de consultas. Considerações finais são dadas na Seção 7.3.

7.1 Extensão semântica de consultas por meio de lógica de descrição probabilística

Na última década diversos trabalhos focando recuperação de informação semântica foram propostos. Procedimentos Booleanos e espaço de vetores, por exemplo, têm suas versões semânticas correspondentes (SHETH *et al.*, 2002; R.; MCCOOL; MILLER, 2003; POPOV *et al.*, 2004; DING *et al.*, 2005) e (STOJANOVIC; STUDER; STOJANOVIC, 2003; ANYANWU; MADUKO; SHETH, 2005; FERNANDEZ *et al.*, 2008) respectivamente. Para uma revisão detalhada veja (SCHEIR; PAMMER; LINDSTAEDT, 2007).

Extensão de consultas (ou *sugestão de consultas*) é uma estratégia frequentemente usada por motores de busca para derivar consultas que são capazes de retornar resultados de maior utilidade que as consultas originais (MANNING; RAGHAVAN; SCHÜTZE, 2008). Muitos motores de busca fornecem facilidades que possibilitam aos usuários a especificação, reformulação ou completamento de suas consultas. *Extensão semântica de consultas* é um tipo especial de extensão de consultas baseado na identificação de conceitos semânticos contidos nas consultas dos usuários (MEIJ *et al.*, 2009). Por exemplo,

o resultado da consulta “Publicações de lógica de descrição probabilística” poderia ser melhorado por um sistema de busca que estendesse a consulta pela inclusão do conceito **Autor** ao invés de apenas o conceito **Publicação**.

Uma linguagem de lógica de descrição probabilística *CRALC* pode ser útil para extensão semântica de consultas. A idéia consiste em associar uma ontologia probabilística *CRALC* a um conjunto de documentos. Note que a partir de uma terminologia *CRALC* é possível definir uma rede Bayesiana relacional que interliga conceitos e papéis. Deste modo, se palavras chave em buscas fossem, de alguma maneira, ligadas aos conceitos e papéis na rede Bayesiana relacional então seria possível, via inferência probabilística, encontrar conceitos mais próximos às consultas iniciais. Esses conceitos, que em alguns casos podem não ter relação direta com a consulta original, possibilitam a sua extensão semântica.

O primeiro passo consiste em construir uma ontologia probabilística *CRALC* que modele o domínio de um conjunto de documentos. Essa ontologia probabilística é aprendida usando os algoritmos apresentados no Capítulo 5. Em seguida, os documentos são ligados à ontologia por meio de índices. Textos nos documentos são indexados; estes textos são propriedades na ontologia correspondente. Assim, os documentos e a ontologia estão separados mas ao mesmo tempo relacionados pelo compartilhamento do mesmo texto indexado. A ontologia e os documentos indexados são a entrada no processo de busca semântica proposto. O processo de busca semântica é dividido em três etapas: (i) busca, (ii) extensão da consulta e (iii) ranking dos resultados baseados na sua relevância. As escolhas chave em cada tarefa são detalhadas a seguir.

7.1.1 Procedimento de busca

Dada uma consulta na forma de um conjunto de palavras chave, os conceitos e papéis relacionados a ela são encontrados por meio de três etapas. Na primeira etapa, uma busca baseada em palavras chave é realizada e o conjunto de documentos relacionados às palavras fornecidas pelo usuário é retornado. Em seguida, os conceitos e papéis relacionados a estes documentos são retornados de acordo com os índices correspondentes (deste modo, as propriedades dos conceitos são também identificadas). Finalmente, uma rede Bayesiana relacional proposicionalizada é construída, isto é, conceitos e papéis identificados via indexação passam a ser evidência na rede. A rede Bayesiana relacional proposicionalizada é a entrada para a etapa de extensão de consulta.

7.1.2 Procedimento de extensão de consulta

Estender a consulta envolve adicionar termos e/ou operadores à consulta original para melhorar os resultados. Nesta proposta, a ontologia fornece termos que podem ser adicionados à consulta. Dada a rede Bayesiana relacional proposicionalizada na etapa precedente, por meio de inferência probabilística é possível identificar conceitos e termos que podem estender a consulta. Para tanto, a probabilidade de todos os conceitos que não são evidência na rede Bayesiana relacional é inferida. Conceitos com probabilidades superiores a um limiar dado são selecionados para estender a consulta, e servem como entrada para a etapa de ranking.

7.1.3 Procedimento de ranking

Nesta etapa os documentos relacionados a conceitos selecionados na etapa de extensão de consultas são recuperados e classificados de acordo com sua probabilidade. Esses documentos são mostrados juntamente com os documentos selecionados inicialmente na etapa de busca. Deve ser salientado que os documentos selecionados no processo de busca são reordenados de acordo com suas probabilidades; isto é, uma lista consolidada e ordenada de documentos é mostrada ao usuário.

Existem dois problemas identificados nessa proposta. O primeiro é o tamanho das ontologias e o segundo, a quantidade de instâncias que são obtidas após a proposicionalização. Em princípio, essas dificuldades impossibilitariam a realização de inferências em domínios reais e deste modo a proposta estaria limitada a domínios de tamanho limitado. Felizmente, podem ser usados métodos variacionais para realizar inferência aproximada (COZMAN; POLASTRO, 2009) possibilitando a aplicação desta proposta.

7.1.4 Resultados preliminares

Experimentos foram realizados em conjunto de dados da plataforma Lattes. O algoritmo de aprendizado de inclusão probabilística apresentado no Capítulo 5 foi utilizado para aprender a ontologia probabilística a partir de dados. Ao todo foram selecionados 1940 documentos Web. A terminologia probabilística completa é dada por:

	$P(\text{Pessoa}) = 0.9$
	$P(\text{Publicação}) = 0.5$
	$P(\text{Banca}) = 0.33$
	$P(\text{Orientação}) = 0.35$
	$P(\text{temPublicação}) = 0.85$
	$P(\text{temOrientação}) = 0.6$
	$P(\text{temParticipação}) = 0.78$
	$P(\text{foiOrientado}) = 0.15$
	$P(\text{mesmaInstituição}) = 0.4$
	$P(\text{mesmaPublicação}) = 0.22$
	$P(\text{mesmaBanca}) = 0.19$
Pesquisador $\equiv$	Pessoa
	$\sqcap (\exists \text{temPublicação.Publicação}$
	$\sqcap \exists \text{temOrientação.Orientação}$
	$\sqcap \exists \text{temParticipação.Banca})$
$P(\text{ColaboradorProximo})$	$ \text{Pesquisador} \sqcap \exists \text{mesmaPublicação.}$
	$\exists \text{mesmaInstituição.}$
	$\exists \text{mesmaPublicação.Pesquisador}) = 0.95$
ColaboradorProximoFaculdade $\equiv$	ColaboradorProximo
	$\sqcap \exists \text{mesmaBanca.Pesquisador}$
$P(\text{PesquisadorMobilidadeNula})$	$ \text{Pesquisador} \sqcap \exists \text{foiOrientado.}$
	$\exists \text{mesmaInstituição.Pesquisador}) = 0.98$
PesquisadorFortementeRelacionado $\equiv$	Pesquisador
	$\sqcap (\exists \text{mesmaPublicação.Pesquisador} \sqcap$
	$\exists \text{foiOrientado.Pesquisador})$
PesquisadorLegado $\equiv$	Pesquisador
	$\sqcap (\exists \text{mesmaBanca.Pesquisador} \sqcap$
	$\exists \text{foiOrientado.Pesquisador})$

Textos nos documentos Web foram indexados a propriedades ligadas na ontologia. Quando uma palavra chave ocorre em uma propriedade dada, essa palavra traz consigo evidência sobre instâncias de propriedades de um conceito dado. A terminologia probabilística mostrada anteriormente atua como um *template* para instâncias de conceitos e propriedades.

O processo geral é detalhado a seguir. Suponha que é fornecida uma consulta por “redes Bayesianas” (o motor de busca Lucene ¹ foi usado nesta etapa), o sistema retorna uma lista ordenada de 20 pesquisadores com links ao seus currículos Lattes como mostrado na Figura 10.

Figura 10: Consulta tradicional.

O usuário deve seguir cada link e investigar onde a palavra “redes Bayesianas” é localizada para, deste modo, poder determinar a relevância do documento recuperado. No algoritmo proposto esses 20 resultados são documentos candidatos passíveis de extensão. Esses resultados, na verdade, são instâncias de conceitos candidatos na terminologia probabilística.

Além disso, dado que houve indexamento no texto das propriedades, é possível instanciar propriedades específicas onde a consulta ocorre. Este passo permite “proposicionalizar” a rede Bayesiana relacional inerente com a ontologia probabilística. Neste cenário probabilístico, cada ocorrência da consulta dentro de propriedades denota evidência nos nós correspondentes. Por exemplo, se *Pesquisador*(0) contém a palavra de consulta em uma publicação dada então é atribuído o valor de verdade ao nó correspondente, *temPublicação*(0, 1). Alguns papéis também denotam relações entre instâncias de conceitos (o papel *mesmaPublicação*(0, 2) relaciona *Pesquisador*(0) e *Pesquisador*(2) por meio de uma publicação compartilhada) e deste modo reforça a verossimilhança de conceitos relacionados que levam a extensões da consulta original.

Inferência probabilística é realizada na rede Bayesiana relacional para obter extensões semânticas da consulta; isto é, os conceitos e os pesquisadores mais relacionados à consulta

¹<http://lucene.apache.org/>

são adicionados aos resultados. A página de resultados estendida é mostrada na Figura 11. Algumas novas entradas foram adicionadas à página de resultados iniciais (por exemplo, o pesquisador P. E. M. foi adicionado dado seu forte relacionamento com um pesquisador em “redes Bayesianas”). Além disso, a lista de pesquisadores final possui informação estendida com links a propriedades específicas e conceitos ao invés de textos curtos sem informação.

Figura 11: Resultado final estendido.

Raciocínio probabilístico possibilita obter um ranking probabilístico. Intuitivamente, maior evidência em um tópico dado resulta em uma melhor posição no ranking. O ranking da Figura 10 retornou os três primeiros pesquisadores: I.B. de M., F. T. R. e F.G.C. Alternativamente, o algoritmo proposto retorna a ordem modificada: F.G.C., I.B. de M. e A. C. F. O. Uma rede Bayesiana relacional possibilita investigar o porque desses resultados. A posição no ranking foi atribuído ao pesquisador F.G.C. dada a evidência no tópico da consulta em publicações, trabalhos de orientação e participação em bancas ($P(\text{Pesquisador}(\text{F.G.C.}) \mid \text{temPublicação.P, orienta.S, participa.B}) = \alpha$). O resto do ranking foi obtido de maneira parecida.

Para avaliar os resultados obtidos pelo algoritmo proposto, dois tipos de teste foram realizados. O primeiro tipo foca na busca de pesquisadores que melhor se adequam a diversos tópicos (dados como palavras chave). O objetivo desse teste é avaliar se a busca semântica retorna resultados significativos. Para tanto, foram usados tópicos tais como “redes Bayesianas”, “lógica probabilística”, entre outros 20 bem estabelecidos grupos de pesquisa no Brasil. Listas de pesquisadores e conceitos relacionados foram avaliados de forma qualitativa. Foi verificado se documentos relevantes associados aos conceitos estavam contidos nas consultas. Em cada caso, os resultados obtidos foram considerados

relevantes. Note que a análise de resultados para buscas semânticas é um tema em aberto; não há padrões de avaliação que contenham todas as informações requeridas para julgar a qualidade dos métodos de busca semântica (FERNANDEZ *et al.*, 2008).

O segundo teste está direcionado ao problema de ranking; isto é, são realmente os primeiros pesquisadores da lista os mais relevantes para cada tópico? Esta questão é relacionada aos valores de probabilidade que denotam as forças dos relacionamentos entre instâncias, o resultado foi de 90% de análise positivo.

7.2 Extensão semântica de consultas usando contexto semântico

Nesta seção apresenta-se uma proposta para uso de *contexto semântico* que possibilita a extensão de consultas visando a obtenção de resultados mais significativos nas buscas Web. De forma parecida à proposta dada na seção precedente também é usada uma ontologia probabilística aprendida a partir de dados. Diferentemente da proposta anterior, essa ontologia probabilística em *CRALLC* é usada na definição de contexto semântico. Esse contexto semântico assim definido possibilita a extensão semântica de consultas.

Documentos Web são ligados a essa ontologia por meio de índices. Textos em documentos são ligados a propriedades na ontologia. Em princípio, a ontologia e os documentos indexados são entradas para o processo de busca semântica. Além disso, o processo recebe uma consulta que pode ser (i) *exata* se retorna documentos que casam exatamente com as palavras chave da busca, ou (ii) *aberta* quando retorna documentos contidos nos *n*-gramas da consulta (as palavras chave de busca têm análise separada). A proposta é uma versão balanceada entre essas duas abordagens. Assume-se que uma consulta exata define contexto semântico (um conjunto de conceitos e papéis probabilísticos ligados a palavras chave exatas denotam contexto). Diversos documentos retornados por uma consulta aberta poderiam estar relacionados com o contexto semântico. Deste modo, uma consulta exata é estendida pela adição de documentos associados às consultas abertas que são mais relacionados em termos de probabilidades. O processo de busca semântica é então dividido em duas partes: (i) busca e (ii) extensão de consulta por meio do contexto da consulta e *CRALLC*. Os documentos são classificados de acordo com sua relevância. As escolhas chave para cada tarefa são mostradas a seguir.

7.2.1 Procedimento de busca

Dada uma consulta como um conjunto de palavras chave, os conceitos e papéis relacionados a ela são encontrados em três etapas. Na primeira, é realizada a **busca exata**, baseada em palavras chave da consulta do usuário, e são retornados o conjunto de documentos relacionados. Posteriormente, os papéis e conceitos relacionados a esses documentos são encontrados via os índices correspondentes (deste modo, as propriedades de conceitos são também identificados). Finalmente, uma RBN é construída, os conceitos selecionados são inseridos como evidência na rede. Essa RBN é entrada para a etapa de extensão e também engloba o contexto semântico dado pela consulta exata.

7.2.2 Extensão de consulta via contexto da consulta

O algoritmo proposto encontra extensões a consultas exatas mediante a realização de inferência probabilística na RBN obtida no procedimento anterior. Cada documento retornado pela consulta aberta é avaliado probabilisticamente para investigar sua adequação com o contexto semântico. Para tanto, um indivíduo é instanciado na RBN representando um dos documentos no conjunto de consultas abertas (se qualquer propriedade é instanciada pelo documento atual então essa evidência é também adicionada). Logo, inferência é realizada no nó correspondente a esse documento. O processo é repetido para cada documento. Deste modo, os documentos mais relacionados são retornados. Estes documentos são mostrados juntamente com os documentos obtidos pela consulta exata. Deve ser notado que os documentos selecionados no processo de busca são reordenados de acordo com suas probabilidades, isto é, uma lista combinada é exibida ao usuário.

7.2.3 Resultados preliminares

Experimentos foram realizados em conjuntos de dados obtidos da plataforma Lattes. Documentos selecionados aleatoriamente foram utilizados para aprender a terminologia probabilística em *CRACC*, para esta tarefa foi utilizado o algoritmo de aprendizado combinado definido no capítulo 5.

Suponha que há interesse em mapear pesquisadores em “lógica probabilística”. Uma busca com um motor de busca tradicional, tal como Lucene, retornaria como resultado da *consulta exata* um conjunto restrito de quatro pesquisadores com links ao seus currículos Lattes como mostrado na Figura 12. Por outro lado, se as palavras “probabilística” e “lógica” são analisadas de forma separada, maiores resultados (Figura 13) são obtidos,

Figura 12: Resultados da consulta exata.

como resultado da *consulta aberta*. Muitos desses últimos documentos não têm significado para o objetivo da busca. Uma busca probabilística semântica com conhecimento de contexto possibilita um balanceamento entre estes dois extremos.

Figura 13: Resultados da consulta aberta.

Os primeiros quatro resultados ligados à consulta exata trazem evidência sobre o contexto semântico. Isto é, propriedades e conceitos específicos onde a consulta exata ocorre são identificadas nesta etapa (texto nas propriedades está indexado). Este passo proposicionaliza a rede Bayesiana associada com a ontologia probabilística. Deste modo, a rede Bayesiana relacional proposicionalizada codifica o contexto semântico.

Em seguida, cada documento retornado pela consulta aberta deve ser probabilisticamente avaliado para investigar sua proximidade com o contexto “verdadeiro” definido pela consulta exata. Assim, um indivíduo a mais é instanciado na RBN. Este indivíduo representa cada documento candidato no conjunto da consulta aberta. Cada candidato é avaliado de forma separada, isto é, evidência associada a esse documento é inserida (de

acordo com palavras chaves em propriedades) e é realizado um teste de verossimilhança² ao contexto da consulta exata.

Figura 14: Resultado final consolidado.

Apenas os pesquisadores mais relacionados (e seus conceitos relacionados) são adicionados aos resultados. A lista final de resultados é mostrada na Figura 14. Algumas novas entradas dos resultados da consulta aberta foram adicionados. Por exemplo, o último resultado foi corretamente adicionado dado seu forte relacionamento em termos de informação contextual a pesquisadores de “lógica probabilística”. Além disso, a lista final de pesquisadores têm informação estendida com links a propriedades e conceitos específicos ao invés de textos simples sem informação.

7.3 Considerações finais

Neste capítulo, os algoritmos propostos para aprendizado foram usados em aplicações em Web Semântica, em particular, em processos de recuperação de informação. O uso de lógica de descrição probabilística neste tipo de aplicações é um tópico que não havia sido explorado até então.

Algoritmos de aprendizado para *CRALLC* foram usados no aprendizado de ontologias probabilísticas. Essas ontologias são usadas na modelagem de domínio de um conjunto de documentos Web. O conhecimento codificado na terminologia probabilística possibilita a extensão semântica de consultas de duas maneiras. A primeira, via adição de conceitos próximos aos termos da consulta. A segunda, via a adição de conceitos próximos ao contexto semântico da consulta.

Embora interessantes, os resultados foram avaliados de forma qualitativa, a avaliação de recuperação de informação semântica é um tema em aberto na literatura.

²Inferência probabilística efetuada na RBN.

8 CONCLUSÕES

Propostas para representação de conhecimento e aprendizado de máquina foram discutidas nesta tese. Com relação à representação de conhecimento, foi proposta uma linguagem lógico-probabilística de primeira ordem que usa noções de independência probabilística (Seção 3.4). Contudo, as maiores contribuições foram dadas em aprendizado de máquina.¹ Visando uma aplicação mais efetiva em problemas reais, algoritmos para o aprendizado de fragmentos de lógicas de primeira ordem probabilísticas (lógicas de descrição probabilísticas) foram propostos (Seções 5.4, 5.5 e 5.6). Algumas destas propostas foram inseridas em aplicações de recuperação de informação (Seções 7.1 e 7.2). As principais contribuições são detalhadas nas seções seguintes bem como os possíveis trabalhos futuros.

8.1 Lógica probabilística de primeira ordem

A lógica probabilística é um formalismo para representação de conhecimento composto de sentenças lógicas e valores de probabilidade associados. O uso de uma linguagem combinada permite aproveitar as vantagens de generalização da lógica e o raciocínio sob incerteza dos modelos probabilísticos. Nesta tese foi proposta uma linguagem lógico-probabilística de primeira ordem que usa julgamentos de independência das redes Bayesianas para estruturar e facilitar a realização de inferências (CAMPOS; COZMAN; OCHOA-LUNA, 2009; OCHOA-LUNA; CAMPOS; COZMAN, 2007). Em resumo, o conjunto de fórmulas que compõem a base de conhecimento têm associado um grafo onde estão embutidas as relações de dependências dos predicados e as sentenças. O processo de inferência usa a condição de Markov subjacente no grafo e mediante um processo de proposicionalização gera uma rede probabilística proposicional. Inferências lógico-probabilísticas são possíveis nesse modelo via algoritmos de programação multilinear.

¹Inicialmente, foi desenvolvido um algoritmo de aprendizado para linguagens lógico-probabilísticas gerais (OCHOA-LUNA; COZMAN, 2009).

A linguagem proposta é flexível e suficientemente expressiva para representar conhecimento e realizar inferências; no entanto, fragmentos de lógica de primeira ordem (principalmente as lógicas de descrição) são mais eficientes em aplicações reais. Por esse motivo, ao invés de investigar algoritmos de aprendizado para este tipo de modelos optou-se pelo aprendizado de lógicas de descrição probabilísticas. Nesse sentido, a lógica de descrição probabilística *CRALC* foi estudada. Duas das principais vantagens de *CRALC* em relação a outras linguagens é a sua semântica baseada em interpretações e a capacidade de realizar inferências probabilísticas escaláveis.

8.2 Aprendizado de lógica de descrição probabilística

Definir modelos lógico-probabilísticos de maneira manual não é uma tarefa trivial: é necessário estruturar as sentenças lógicas e também estimar as probabilidades. Atualmente, graças à disponibilidade de dados, especialmente na Internet, é possível automatizar a eliciação de modelos lógico-probabilísticos.

Neste trabalho, embora tenha sido inicialmente focado o aprendizado de lógicas de primeira ordem probabilísticas (OCHOA-LUNA; COZMAN, 2009), logo foi notada a importância de desenvolver algoritmos de aprendizado para linguagens com mais possibilidades de aplicação em domínios reais, tais como lógicas de descrição probabilísticas. Nesse sentido, foram propostos três algoritmos para o aprendizado da lógica de descrição probabilística *CRALC* que abrangem tanto as sentenças lógicas bem como as suas probabilidades.

Na primeira proposta (OCHOA-LUNA; COZMAN, 2009), foi enfatizado o problema de aprendizado de sentenças sob uma perspectiva probabilística, baseado em técnicas de programação lógica indutiva. Isto é, exemplos positivos e negativos guiam o aprendizado de conceitos. A indução das sentenças lógicas passou a ser tratada como um problema de classificação probabilística, onde o aprendizado de um classificador probabilístico baseado em Noisy-OR possibilita discernir entre conceitos candidatos. Finda a indução da melhor estrutura, as probabilidades podem ser inseridas de maneira ad-hoc.

Na segunda proposta (REVOREDO; OCHOA-LUNA; COZMAN, 2010), foi explorado o problema do aprendizado das inclusões probabilísticas de *CRALC*. Entretanto, o aprendizado de terminologias probabilísticas completas também foi contemplado. Numa primeira etapa usa-se um algoritmo de aprendizado de lógicas de descrição tradicional para aprender definições de conceitos candidatos. As definições aprendidas são decompostas nos seus componentes básicos (conceitos e papéis). Em seguida, cada um desses componentes e a

combinação deles é testado como candidato condicional do conceito alvo. Deste modo, encontra-se o melhor condicionante para o conceito alvo (a inclusão probabilística). Logo, uma escolha é realizada pela manutenção do conceito determinístico original ou a sua substituição pela inclusão probabilística aprendida.

Finalmente, na terceira proposta (OCHOA-LUNA; REVOREDO; COZMAN, 2010a) foi realizado o aprendizado integrado tanto da estrutura lógica quanto das inclusões probabilísticas. A idéia principal é a constatação da existência de conceitos inerentemente determinísticos e outros inerentemente probabilísticos; o algoritmo proposto tenta identificar ambos componentes de forma integrada. Inicialmente define-se um cenário probabilístico; basicamente a relação de cobertura é probabilística e é usada na busca por conceitos determinísticos. Após algumas iterações o melhor candidato pode ser retornado como definição do conceito alvo ou pode ser considerado probabilístico. Neste último caso, os k -primeiros candidatos são usados para aprender as inclusões probabilísticas. Para tanto são realizados testes de independência condicional entre o conceito alvo e os candidatos condicionantes e um classificador TAN é aprendido, o qual retorna a inclusão probabilística mais provável.

Experimentos iniciais foram realizados com dados de repositórios visando sua comparação com propostas semelhantes. Os algoritmos se mostraram acurados quando testados em tarefas de aprendizado de lógica de descrição (sem probabilidades). Posteriormente, foram conduzidos experimentos sobre conjunto de dados extraídos de repositórios de informações públicas da Internet. Os três algoritmos propostos foram avaliados na capacidade de aprendizado de terminologias probabilísticas. Baseado na acurácia obtida, os algoritmos conseguiram aprender os modelos. Contudo, os melhores resultados foram obtidos pelo algoritmo que realiza o aprendizado integrado. Os resultados indicam que os algoritmos podem ser promissores para elicitar terminologias probabilísticas a partir de conjuntos de dados reais. Essas propostas constituem contribuições para a literatura de lógica probabilística. Cabe salientar que, mesmo em áreas como aprendizado em lógica de descrição, estudos comparativos são escassos; no caso de aprendizado em lógica de descrição probabilística são praticamente inexistentes na literatura.

8.3 Aplicações em Web semântica

Embora os algoritmos de aprendizado tenham sido avaliados empiricamente usando dados reais, duas aplicações, orientadas ao efetivo uso dos resultados do aprendizado,

também foram desenvolvidas. Ambas estão inseridas na área de Web semântica. Em particular, foi focada a recuperação de informações usando informação semântica.

A primeira proposta (OCHOA-LUNA; REVOREDO; COZMAN, 2010b) fez o uso de uma ontologia probabilística aprendida com um dos algoritmos propostos para realizar extensão de consultas semânticas de buscas na Web. A segunda proposta (OCHOA-LUNA; REVOREDO; COZMAN, 2010c) usou ontologias aprendidas para modelar o contexto semântico de buscas na Web. Deste modo é possível estender as buscas com conceitos e pápeis associados ao contexto.

Propostas para extensão semântica de consultas usando terminologias probabilísticas são escassas na literatura. No entanto, os resultados obtidos ainda devem ser testados em domínios maiores.

8.4 Trabalhos futuros

Em termos de representação de conhecimento, uma contribuição futura seria o estudo mais detalhado de métodos de comparação entre as diversas lógicas probabilísticas, tanto em termos de representação como de aprendizado.

Uma das possíveis soluções seria desenvolver casos de uso que sejam passíveis de comparação e, deste modo, investigar quais tipos de aplicações são mais adequadas com as características de cada linguagem. Essa é uma questão que já foi discutida em eventos da área (URSW2008, URSW2009 e URSW2010) porém, pouco foi produzido nesse sentido.

Aprendizado de lógica de descrição probabilística é um problema pouco pesquisado e com resultados empíricos prévios inexistentes na literatura. Embora promissores, os algoritmos propostos precisam ser classificados de acordo com o tipo de aplicações nos quais podem trazer melhores resultados. Além disso, ainda resta um estudo mais detalhado de formas de comparação entre os algoritmos propostos.

Ainda nesse ponto, os algoritmos propostos foram testados especificamente na linguagem *CRALLC* dado que não existem linguagens gerais em lógica de descrição probabilística. Entretanto, poderia ser mais explorada a forma de adaptar os algoritmos de aprendizado propostos a lógicas de descrição probabilísticas tais como PR-OWL (COSTA; LASKEY, 2006) ou lógicas de descrição Bayesianas (D'AMATO; FANIZZI; LUKASIEWICZ, 2008). A princípio, a base lógica é semelhante, contudo, questões referentes à semântica e a forma como as probabilidades são estimadas precisariam ser aprofundadas.

Trabalhos futuros em recuperação de informação em Web Semântica devem considerar a adoção de novas métricas de avaliação. Métricas tradicionais como precisão e revocação precisam ser adaptadas em ambientes que lidam com probabilidades.

REFERÊNCIAS

- ABADI, M.; HALPERN, J. Decidability and expressiveness for first-order logics of probability. *Inform. Comput.*, v. 112 (1), p. 1–36, 1994.
- ALMOND, R. G. Hypergraph grammars for knowledge based model construction. Technical Report 23, StatSci, 1994.
- ANDERSEN, K. A.; HOOKER, J. N. Bayesian logic. *Decision Support Systems*, v. 11, p. 191–210, 1994.
- ANTONIOU, G.; VAN HARMELEN, F. *A Semantic Web Primer*. Cambridge, MA, USA: MIT Press, 2004. ISBN 0262012103.
- ANYANWU, K.; MADUKO, A.; SHETH, A. SemRank: ranking complex relationship search results on the semantic web. In: *Proceedings of the 14th international conference on World Wide Web*. New York, NY, USA: ACM, 2005. p. 117–127. ISBN 1-59593-046-9.
- BAADER, F.; NUTT, W. The description logic handbook. In: BAADER, F. *et al.* (Ed.). New York, NY, USA: Cambridge University Press, 2003. cap. Basic description logics, p. 43–95. ISBN 0-521-78176-0.
- BACCHUS, F. *Representing and Reasoning with Probabilistic Knowledge: A Logical Approach*. Cambridge: MIT Press, 1990.
- BACCHUS, F. Using first-order probability logic for the construction of Bayesian networks. In: *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI-93)*. Providence, Washington, DC, USA: Morgan Kaufmann, 1993. p. 219–226.
- BIZER, C. *et al.* DBpedia - a crystallization point for the web of data. *Web Semant.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 7, n. 3, p. 154–165, 2009. ISSN 1570-8268.
- BOOLE, G. *The Laws of Thought*. New York, NY: Dover, 1958.
- CAMPOS, C. P. de. *Redes Credais e Qualitativas: Complexidade e Algoritmos*. Tese (Doutorado) — Escola Politécnica, Universidade de São Paulo, São Paulo, Brasil, 2004.
- CAMPOS, C. P. de; COZMAN, F. G. Inference in credal networks using multilinear programming. In: *Proceedings of 2nd European Starting AI Researcher Symposium*. Valencia, Spain: IOS Press, 2004. p. 50–61.

- CAMPOS, C. P. de; COZMAN, F. G.; OCHOA-LUNA, J. Assembling a consistent set of sentences in relational probabilistic logic with stochastic independence. *Jornal of Applied Logic*, v. 7, p. 131–135, 2009.
- CHANDRU, V.; HOOKER, J. *Optimization Methods for Logical Inference*. New York, NY: Wiley, 1999.
- CHENG, J. *et al.* Learning Bayesian networks from data: an information-theory based approach. *Artif. Intell.*, Elsevier Science Publishers Ltd., Essex, UK, v. 137, p. 43–90, May 2002. ISSN 0004-3702.
- COOPER, G.; HERSKOVITS, E. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, v. 9, p. 309–347, 1992.
- COSTA, P. C. G.; LASKEY, K. B. PR-OWL: A framework for probabilistic ontologies. In: *Proceeding of the 2006 conference on Formal Ontology in Information Systems*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2006. p. 237–249. ISBN 1-58603-685-8.
- COSTA, V. S. *et al.* CLP(BN): Constraint logic programming for probabilistic knowledge. In: KJAERULFF, U.; MEEK, C. (Ed.). *Conference on Uncertainty in Artificial Intelligence*. San Francisco, California: Morgan-Kaufmann, 2003. p. 517–524.
- COZMAN, F. G. Credal networks. *Artificial Intelligence*, v. 120, p. 199–233, 2000.
- COZMAN, F. G.; CAMPOS, C. P. de; ROCHA, J. C. da. Probabilistic logic with independence. *International Journal of Approximate Reasoning*, v. 49, n. 1, p. 3–17, 2008.
- COZMAN, F. G.; POLASTRO, R. B. Loopy propagation in a probabilistic description logic. In: *SUM '08: Proceedings of the 2nd International Conference on Scalable Uncertainty Management*. Berlin, Heidelberg: Springer-Verlag, 2008. p. 120–133. ISBN 978-3-540-87992-3.
- COZMAN, F. G.; POLASTRO, R. B. Complexity analysis and variational inference for interpretation-based probabilistic description logics. In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. Arlington, Virginia, United States: AUAI Press, 2009. (UAI '09), p. 117–125. ISBN 978-0-9749039-5-8.
- D'AMATO, C.; FANIZZI, N.; LUKASIEWICZ, T. Tractable reasoning with Bayesian description logics. In: *Proceedings of the 2nd international conference on Scalable Uncertainty Management*. Berlin, Heidelberg: Springer-Verlag, 2008. (SUM '08), p. 146–159. ISBN 978-3-540-87992-3.
- DE RAEDT, L. *Logical and Relational Learning*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2008.
- DE RAEDT, L.; DEHASPE, L. Clausal discovery. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 26, p. 99–146, March 1997. ISSN 0885-6125.
- DE RAEDT, L.; KERSTING, K. Probabilistic inductive logic programming. In: RAEDT, L. D. *et al.* (Ed.). Berlin, Heidelberg: Springer-Verlag, 2008. cap. Probabilistic inductive logic programming, p. 1–27. ISBN 3-540-78651-1, 978-3-540-78651-1.

- DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the EM algorithm. *Journal Royal Statistical Society B*, v. 44, p. 1–38, 1977.
- DING, L. *et al.* Search on the semantic web. *Computer*, IEEE Computer Society, Los Alamitos, CA, USA, v. 38, p. 62–69, 2005. ISSN 0018-9162.
- DING, Z.; PENG, Y.; PAN, R. BayesOWL: Uncertainty Modeling in Semantic Web Ontologies. *Soft Computing in Ontologies and Semantic Web*, p. 3–29, 2006.
- DZEROSKI, S.; LAVRAC, N. *Relational Data Mining*. Berlin: Springer, 2001.
- DÜRIG, M.; STUDER, T. Probabilistic abox reasoning: Preliminary results. In: *In Proceedings DL-2005*. Edinburgh, Scotland, 2005. p. 104–111.
- FAGIN, R.; HALPERN, J. Y.; MEGIDDO, N. A logic for reasoning about probabilities. *Information and Computation*, v. 87, p. 78–128, 1990.
- FANIZZI, N.; D'AMATO, C.; ESPOSITO, F. DL-FOIL concept learning in description logics. In: *ILP '08: Proceedings of the 18th International Conference on Inductive Logic Programming*. Berlin, Heidelberg: Springer-Verlag, 2008. p. 107–121. ISBN 978-3-540-85927-7.
- FANIZZI, N.; D'AMATO, C.; ESPOSITO, F. Induction of concepts in web ontologies through terminological decision trees. In: BALCAZAR, J. *et al.* (Ed.). *Machine Learning and Knowledge Discovery in Databases*. Berlin: Springer/Heidelberg, 2010, (Lecture Notes in Computer Science, v. 6321). p. 442–457.
- FERNANDEZ, M. *et al.* Semantic search meets the web. In: *Proceedings of the 2nd IEEE International Conference on Semantic Computing*. Washington, DC, USA: IEEE Computer Society, 2008. p. 253–260. ISBN 978-0-7695-3279-0.
- FIERENS, D. *et al.* Logical Bayesian networks and their relation to other probabilistic logical models. In: KRAMER, S.; PFAHRINGER, B. (Ed.). *Inductive Logic Programming*. Berlin: Springer / Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3625). p. 121–135.
- FRIEDMAN, N. Learning belief networks in the presence of missing values and hidden variables. In: *Proceedings of the Fourteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997. (ICML '97), p. 125–133. ISBN 1-55860-486-3.
- FRIEDMAN, N.; GEIGER, D.; GOLDSZMIDT, M. Bayesian network classifiers. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 29, p. 131–163, 1997. ISSN 0885-6125.
- FRIEDMAN, N. *et al.* Learning probabilistic relational models. In: *16th International Joint Conference in Artificial Intelligence (IJCAI99)*. Berlin: Springer-Verlag, 1999. p. 1300–1309.
- FRISCH, A. M.; HADDAWY, P. Anytime deduction for probabilistic logic. *Artificial Intelligence*, v. 69, p. 93–122, 1994.

- GAIFMAN, H.; SNIR, M. Probabilities over rich languages, testing and randomness. *The Journal of Symbolic Logic*, v. 47(3), p. 495–548, 1982.
- GENESERETH, M. R.; NILSSON, J. N. *Logical Foundations of Artificial Intelligence*. Palo Alto: Morgan Kaufmann Publishers, 1987.
- GEORGAKOPOULOS, G.; KAVVADIAS, D.; PAPADIMITRIOU, C. H. Probabilistic Satisfiability. *Journal of Complexity*, v. 4, p. 1–11, 1988.
- GETOOR, L. *et al.* Learning probabilistic models of relational structure. In: *Proceedings of the Eighteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001. (ICML '01), p. 170–177. ISBN 1-55860-778-1.
- GETOOR, L.; TASKAR, B. *Introduction to Statistical Relational Learning*. Cambridge: MIT Press, 2007.
- GIUGNO, R.; LUKASIEWICZ, T. P-*SHOQ*(D): A probabilistic extension of *SHOQ*(D) for probabilistic ontologies in the semantic web. In: *JELIA '02: Proceedings of the European Conference on Logics in Artificial Intelligence*. London, UK: Springer-Verlag, 2002. p. 86–97. ISBN 3-540-44190-5.
- GLESNER, S.; KOLLER, D. Constructing flexible dynamic belief networks from first-order probabilistic knowledge bases. In: *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*. 1995. p. 217–226.
- HAENNI, R. *et al.* Probabilistic logics and probabilistic networks. Technical Report, Prolog net, 2008.
- HAILPERIN, T. *Sentential Probability Logic*. Bethlehem, United States: Lehigh University Press, 1996.
- HALPERN, J. Y. An analysis of first-order logics of probability. *Artificial Intelligence*, v. 46, p. 311–350, 1990.
- HALPERN, J. Y. *Reasoning about Uncertainty*. Cambridge, Massachusetts: MIT Press, 2003.
- HANSEN, P.; JAUMARD, B. Probabilistic satisfiability. Technical Report G-96-31, Les Cahiers du GERAD, École Polytechnique de Montréal, 1996.
- HECKERMAN, D.; GEIGER, D.; CHICKERING, D. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, v. 20, p. 197–243, 1995.
- HECKERMAN, D.; MEEK, C.; KOLLER, D. Probabilistic models for relational data. Technical Report MSR-TR-2004-30, Microsoft Research, 2004.
- HEINSOHN, J. Probabilistic description logics. In: *Proceedings of the Tenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-94)*. San Francisco, CA: Morgan Kaufmann, 1994. p. 311–31.
- HOOVER, D. N. Probability logic. *Annals of Mathematic Logic*, v. 14, p. 287–313, 1978.

- HOWSON, C.; URBACH, P. *Scientific Reasoning: the Bayesian Approach*. Chicago, Illinois: Open Court Publishing Company, 1993.
- HUANG, C.; DARWICHE, A. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, v. 15, p. 225–263, 1996.
- HUNG, E.; GETOOR, L.; SUBRAHMANYAN, V. S. Probabilistic interval XML. *ACM Trans. Comput. Logic*, ACM, New York, NY, USA, v. 8, n. 4, p. 24, 2007. ISSN 1529-3785.
- HUTH, M.; RYAN, M. *Logic in Computer Science: Modelling and Reasoning about Systems*. Cambridge: Cambridge University Press, 2004.
- IANNONE, L.; PALMISANO, I.; FANIZZI, N. An algorithm based on counterfactuals for concept learning in the semantic web. *Applied Intelligence*, Kluwer Academic Publishers, Hingham, MA, USA, v. 26, n. 2, p. 139–159, 2007. ISSN 0924-669X.
- JAEGER, M. Probabilistic reasoning in terminological logics. In: *Principles of Knowledge Representation and Reasoning: Proceedings of the 4th International Conference (KR94)*. San Mateo, CA: Morgan Kaufmann, 1994. p. 305–316.
- JAEGER, M. Relational Bayesian networks. In: GEIGER, D.; SHENOY, P. P. (Ed.). *Conference on Uncertainty in Artificial Intelligence*. San Francisco, California: Morgan Kaufmann, 1997. p. 266–273.
- JAUMARD, B. *et al.* First order probabilistic logic. In: *Fuzzy Information Processing Society, 2006. NAFIPS 2006. Annual meeting of the North American Volume*. Montreal, Quebec, 2006. p. 341–346.
- JAUMARD, B.; HANSEN, P.; ARAGAO, M. P. de. Column generation methods for probabilistic logic. *ORSA Journal on Computing*, v. 3, n. 2, p. 135–148, Spring 1991.
- KEISLER, H. J. Probabilistic quantifiers. In: *J. Barwise, S. Feferman (eds.), Model-Theoretic Logic*. New York: Springer, 1985. p. 509–556.
- KERSTING, K.; DE RAEDT, L.; KRAMER, S. Interpreting Bayesian logic programs. In: *AAAI-2000 Workshop on Learning Statistical Models from Relational Data*. 2000.
- KERSTING, K.; DICK, U. Balios - the engine for Bayesian logic programs. In: BOULICAUT, J.-F. *et al.* (Ed.). *PKDD*. Berlin: Springer, 2004. (Lecture Notes in Computer Science, v. 3202), p. 549–551.
- KOLLER, D.; PFEFFER, A. Object-oriented Bayesian networks. In: *Proceedings of the Thirteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*. San Francisco, CA: Morgan Kaufmann, 1997. p. 302–31.
- KYBURG JR., H. E.; TENG, C. M. *Uncertain Inference*. Cambridge: Cambridge University Press, 2001.
- LAM, W.; BACCHUS, F. Learning Bayesian belief networks: An approach based on the MDL principle. *Computational Intelligence*, v. 10(4), p. 269–293, 1994.

- LANDWEHR, N.; KERSTING, K.; DE RAEDT, L. Integrating Naïve Bayes and FOIL. *J. Mach. Learn. Res.*, MIT Press, Cambridge, MA, USA, v. 8, p. 481–507, 2007. ISSN 1533-7928.
- LASKEY, K.; COSTA, P. da. Of starships and Klingons: Bayesian logic for the 23rd century. In: *Proceedings of the Proceedings of the Twenty-First Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*. Arlington, Virginia: AUAI Press, 2005. p. 346–353.
- LAURITZEN, S. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, v. 19, p. 191–201, 1995.
- LAVRAC, N.; DZEROSKI, S. *Inductive Logic Programming: Techniques and Applications*. New York: Ellis Horwood, 1994.
- LEHMANN, J. Hybrid learning of ontology classes. In: *Proceedings of the 5th International Conference on Machine Learning and Data Mining*. Berlin: Springer, 2007. (Lecture Notes in Computer Science, v. 4571), p. 883–898.
- LEHMANN, J.; HITZLER, P. Foundations of refinement operators for description logics. In: BLOCKEEL, H.; SHAVLIK, J. W.; TADEPALLI, P. (Ed.). *ILP '08: Proceedings of the 17th International Conference on Inductive Logic Programming*. Berlin: Springer, 2008. (Lecture Notes in Computer Science, v. 4894), p. 161–174.
- LEHMANN, J.; HITZLER, P. A refinement operator based learning algorithm for the *ALC* description logic. In: BLOCKEEL, H.; SHAVLIK, J. W.; TADEPALLI, P. (Ed.). *ILP '08: Proceedings of the 17th International Conference on Inductive Logic Programming*. Berlin: Springer, 2008. (Lecture Notes in Computer Science, v. 4894), p. 147–160.
- LEHMANN, J.; HITZLER, P. Concept learning in description logics using refinement operators. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 78, p. 203–250, January 2010. ISSN 0885-6125.
- LUKASIEWICZ, T. Probabilistic logic programming with conditional constraints. *ACM Transactions on Computational Logic*, v. 2(3), p. 289–339, 2001.
- LUKASIEWICZ, T. Expressive probabilistic description logics. *Artif. Intell.*, Elsevier Science Publishers Ltd., Essex, UK, v. 172, n. 6-7, p. 852–883, 2008. ISSN 0004-3702.
- LUKASIEWICZ, T.; STRACCIA, U. Managing uncertainty and vagueness in description logics for the semantic web. *Web Semant.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 6, n. 4, p. 291–308, 2008. ISSN 1570-8268.
- MAHONEY, S.; LASKEY, K. Network engineering for complex belief networks. In: *Proceedings of the Proceedings of the Twelfth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-96)*. San Francisco, CA: Morgan Kaufmann, 1996. p. 389–39.
- MANNING, C.; RAGHAVAN, P.; SCHÜTZE, H. *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008.

- MEIJ, E. *et al.* Learning semantic query suggestions. In: *8th International Semantic Web Conference*. Berlin: Springer, 2009. p. 424–440.
- MILCH, B. *et al.* BLOG: Probabilistic models with unknown objects. In: *19th International Joint Conference in Artificial Intelligence (IJCAI05)*. Edinburgh, Scotland: Professional Book Center, 2005. p. 1352–1359.
- MILCH, B.; RUSSELL, S. First-order probabilistic languages: Into the unknown. In: MUGGLETON, S.; OTERO, R.; TAMADDONI-NEZHAD, A. (Ed.). *Inductive Logic Programming*. Berlin, Germany: Springer / Heidelberg, 2007, (Lecture Notes in Computer Science, v. 4455). p. 10–24.
- MITCHELL, T. *Machine Learning*. New York: McGraw-Hill, 1997.
- NEAPOLITAN, R. E. *Learning Bayesian Networks*. Upper Saddle River, NJ, USA: Prentice Hall, 2003.
- NGO, L.; HADDAWY, P. Answering queries from context-sensitive probabilistic knowledge bases. *Theoretical Computer Science*, v. 171, n. 1–2, p. 147–177, 1997.
- NILSSON, N. J. Probabilistic logic. *Artificial Intelligence*, v. 28, p. 71–87, 1986.
- NOTTELMANN, H.; FUHR, N. Adding probabilities and rules to OWL lite subsets based on probabilistic datalog. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, v. 14, n. 1, 2006.
- OCHOA-LUNA, J.; COZMAN, F. An algorithm for learning with probabilistic description logics. In: *Bobillo, F., et al. (eds.) Proceedings of the 5th International Workshop on Uncertainty Reasoning for the Semantic Web*. Chantilly, USA: CEUR-WS.org, 2009. v. 527, p. 63–74.
- OCHOA-LUNA, J.; REVOREDO, K.; COZMAN, F. Learning sentences and assessments in probabilistic description logics. In: *Bobillo, F., et al. (eds.) Proceedings of the 6th International Workshop on Uncertainty Reasoning for the Semantic Web*. Shanghai, China: CEUR-WS.org, 2010. v. 654, p. 85–96.
- OCHOA-LUNA, J.; REVOREDO, K.; COZMAN, F. Semantic query extension through probabilistic description logics. In: *Bobillo, F., et al. (eds.) Proceedings of the 6th International Workshop on Uncertainty Reasoning for the Semantic Web*. Shanghai, China: CEUR-WS.org, 2010. v. 654, p. 49–60.
- OCHOA-LUNA, J.; REVOREDO, K.; COZMAN, F. Semantic query extension using query contexts and probabilistic description logics. In: *Proceedings of the 3rd International Workshop on Web and Text Intelligence*. Sao Bernardo do Campo: Sociedade Brasileira da Computação SBC, 2010. p. 856–864.
- OCHOA-LUNA, J. E.; CAMPOS, C. P. de; COZMAN, F. G. Independence in relational languages with finite domains. In: *Anais do XXVII Congresso da SBC*. Rio de Janeiro, Brasil, 2007. p. 1421–1429.
- OCHOA-LUNA, J. E.; COZMAN, F. G. A probabilistic algorithm for multi-relational data mining. In: *Anais do V Workshop em Algoritmos e Aplicações de Mineração de Dados - WAAMD 2009*. Fortaleza, Brasil, 2009. p. 34–41.

- PEARL, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Los Altos, CA: Morgan Kaufmann Publishers, 1988.
- POOLE, D. Probabilistic Horn abduction and Bayesian networks. *Artificial Intelligence*, v. 64, p. 81–129, 1993.
- POPOV, B. *et al.* Kim – a semantic platform for information extraction and retrieval. *Nat. Lang. Eng.*, Cambridge University Press, New York, NY, USA, v. 10, n. 3-4, p. 375–392, 2004. ISSN 1351-3249.
- QUINLAN, J.; CAMERON-JONES, R. FOIL: A midterm report. In: BRAZDIL, P. (Ed.). *Machine Learning: ECML-93*. Berlin: Springer / Heidelberg, 1993, (Lecture Notes in Computer Science, v. 667). p. 1–20.
- QUINLAN, J. R. Learning logical definitions from relations. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 5, p. 239–266, 1990. ISSN 0885-6125.
- R., R. G.; MCCOOL; MILLER, E. Semantic search. In: *Proceedings of the 12th International Conference on World Wide Web*. New York, NY, USA: ACM, 2003. p. 700–709. ISBN 1-58113-680-3.
- RAMONI, M.; SEBASTIANI, P. Learning bayesian networks from incomplete databases. In: *Proceedings of the Thirteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*. San Francisco, CA: Morgan Kaufmann, 1997. p. 401–40.
- REVOREDO, K.; OCHOA-LUNA, J.; COZMAN, F. Learning terminologies in probabilistic description logics. In: COSTA, A. da R.; VICARI, R.; TONIDANDEL, F. (Ed.). *Advances in Artificial Intelligence SBIA 2010*. Berlin: Springer / Heidelberg, 2010, (Lecture Notes in Computer Science, v. 6404). p. 41–50.
- SATO, T.; KAMEYA, Y. Parameter learning of logic programs for symbolic-statistical modeling. *Journal of Artificial Intelligence Research*, v. 15, p. 391–454, 2001.
- SCHARRENBACH, T.; BERNSTEIN, A. On the evolution of ontologies using probabilistic description logics. In: *Proceedings of the First ESWC Workshop on Inductive Reasoning and Machine Learning on the Semantic Web*. Naples, Italy: Editor(s): Claudia d’Amato, Nicola Fanizzi, Marko Grobelnik, Agnieszka Lawrynowicz, Vojtech Svátek, 2009. (CEUR Workshop Proceedings).
- SCHEIR, P.; PAMMER, V.; LINDSTAEDT, S. Information retrieval on the semantic web - does it exist. In: *In LWA 2007, Lernen - Wissensentdeckung - Adaptivität*. Halle/Saale. 2007.
- SEBASTIANI, F. A probabilistic terminological logic for modelling information retrieval. In: *ACM Conf. on Research and Development in Information Retrieval (SIGIR)*. 1994. p. 122–130.
- SHERALI, H. D.; TUNCBILEK, C. H. A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique. *Journal of Global Optimization (Historical Archive)*, v. 2, n. 1, p. 101–112, Março 1992.

- SHETH, A. *et al.* Managing semantic content for the web. *IEEE Internet Computing*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 6, n. 4, p. 80–87, 2002. ISSN 1089-7801.
- SILVA, F. S. da; FINGER, M.; MELO, A. *Lógica para Computação*. Sao Paulo: Thomson Learning, 2006.
- STAKER, R. Reasoning in expressive description logics using belief networks. In: *Proceedings of the International Conference on Information and Knowledge Engineering*. Las Vegas, USA: CSREA Press, 2002. p. 489–495.
- STOJANOVIC, N.; STUDER, R.; STOJANOVIC, L. An approach for the ranking of query results in the semantic web. In: *The SemanticWeb - ISWC 2003*. Berlin: Springer / Heidelberg, 2003, (Lecture Notes in Computer Science, v. 2870). p. 500–516.
- SU, J.; ZHANG, H. Full Bayesian network classifiers. In: *Proceedings of the 23rd international conference on Machine learning*. New York, NY, USA: ACM, 2006. (ICML '06), p. 897–904. ISBN 1-59593-383-2.
- SUCHANEK, F. M.; KASNECI, G.; WEIKUM, G. Yago: a core of semantic knowledge. In: *WWW '07: Proceedings of the 16th international conference on World Wide Web*. New York, NY, USA: ACM, 2007. p. 697–706. ISBN 978-1-59593-654-7.
- THOMAS, A.; SPIEGELHALTER, D. J.; GILKS, W. R. BUGS: A program to perform Bayesian inference using Gibbs sampling. In: *Bayesian Statistics 4*. 1992. p. 837–842.
- VAN DER GAAG, L. Computing probability intervals under independency constraints. In: BONISSONE, P. P.; HENRION, M.; LEMMER, J. F. (Ed.). *Uncertainty in Artificial Intelligence 6*. Amsterdam, NL: Elsevier Science, 1991. p. 457–466.
- VOMLEL, J. Noisy-OR classifier: Research articles. *Int. J. Intell. Syst.*, John Wiley & Sons, Inc., New York, NY, USA, v. 21, n. 3, p. 381–398, 2006. ISSN 0884-8173.
- WELLMAN, M. P.; BREESE, J. S.; GOLDMAN, R. P. From knowledge bases to decision models. *Knowledge Engineering Review*, v. 7(1), p. 35–53, 1992.
- YCART, B.; ROUSSET, M.-C. A zero-one law for random sentences in description logics. In: *Colloquium on Mathematics and Computer Science*. Birkhäuser, Basel: editors: I, D. Gardy, A. Mokkadem., 2000. p. 329–340.
- YELLAND, P. M. Market analysis using a combination of Bayesian networks and description logics. Technical Report, Sun Microsystems Laboratories, 1999.

APÊNDICE I – INFERÊNCIAS EM RPL USANDO PROGRAMAÇÃO MULTILINEAR

Uma rede proposicionalizada associada a RPL (isto é, uma linguagem PPL) é uma rede Bayesiana onde os valores de probabilidade não são conhecidos de forma precisa. Para produzir uma inferência, cada sentença/valor é considerada como uma subconsulta na rede, ou seja, uma versão simbólica de um algoritmo de inferência em redes Bayesianas deve ser executado (CAMPOS; COZMAN, 2004) para produzir uma coleção de problemas de otimização multilinear. O fato da distribuição conjunta fatorar de acordo com a topologia do grafo é chave para produzir programas multilineares de tamanho razoável.

Para gerar o programa multilinear, para cada sentença/valor uma por vez:

- Para cada $\phi_j \in \mathcal{L}$, atribuir $P(\phi_j) = 1$. Como ϕ_j já é um nó na rede, realizar uma inferência simbólica para a probabilidade marginal de ϕ_j na rede proposicionalizada. Isto deve criar um conjunto de restrições multilineares que definem $P(\phi_j)$ em termos das outras variáveis da rede. Em seguida, inserir a restrição $P(\phi) = 1$ e este conjunto de restrições no problema multilinear.
- Para cada $A_j \in \mathcal{A}'_p$, denotar $\alpha_j P(\varphi_j) \leq P(\phi_j \wedge \varphi_j) \leq \beta_j P(\varphi_j)$, onde ϕ_j e φ_j já têm nós correspondentes na rede. Executar as inferências simbólicas para a probabilidade de φ_j e para a probabilidade de $\phi_j \wedge \varphi_j$ (isto é justamente uma consulta conjunta em uma rede Bayesiana). Estas consultas simbólicas produzem dois conjuntos de restrições multilineares. Inserir estas restrições no problema multilinear, junto com a restrição que $\alpha_j P(\varphi_j) \leq P(\phi_j \wedge \varphi_j) \leq \beta_j P(\varphi_j)$.
- Incluir um nó para θ (a consulta) na rede, tendo como pais todas as interpretações que ocorrem em θ . Então realizar a inferência simbólica para a probabilidade marginal de θ usando o nó inserido. Restrições geradas devem ser incluídas no problema multilinear.
- No passo final, o programa multilinear deve ser resolvido — maximização e minimização da probabilidade de θ . Se houver uma solução, esta deve satisfazer todos

os valores. Caso contrário não existe uma solução possível para este problema multilinear e a rede RPL é inconsistente.

Portanto, para cada sentença, executa-se uma versão simbólica de um algoritmo de inferência em redes Bayesianas quaisquer, o qual produz um conjunto de restrições multilineares. O programa multilinear total é obtido a partir de todas as restrições multilineares. Usando este programa multilinear, é possível verificar a satisfazibilidade ou ainda avaliar o máximo/mínimo de $P(\theta)$ (CAMPOS; COZMAN, 2004).