

JAIME SHINSUKE IDE

**ALGORITMOS PARA INFERÊNCIA APROXIMADA
EM REDES CREDAIS COM VARIÁVEIS BINÁRIAS**

São Paulo

2005

OK

JAIME SHINSUKE IDE

**ALGORITMOS PARA INFERÊNCIA APROXIMADA
EM REDES CREDAIS COM VARIÁVEIS BINÁRIAS**

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

Área de concentração:
Engenharia de Controle e Automação
Mecânica

Orientador:
Prof. Dr. Fabio Gagliardi Cozman

São Paulo

2005

DEDICATÓRIA

Ao Tadeu Nakamura, falecido companheiro de pesca.

AGRADECIMENTOS

Dedico apenas algumas palavras que mostram só parte dos meus sinceros agradecimentos para:

- os meus pais por me terem dado vida e formação;
- o Prof. Dr. Fabio Gagliardi Cozman pelo imenso e constante apoio prestado ao longo do trabalho;
- os Professores Dr. Newton Maruyama e Dr. Alexandre Kawano pelas valiosas críticas recebidas no Exame de Qualificação;
- o Prof. Dr. Paulo Eigi Miyagi pelas orientações recebidas desde os tempos de graduação;
- os Professores Dr. Marcos Hiroyuki Tsuchida e Dra. Jolanda Ygosse Batista pelos auxílios prestados no planejamento de projetos futuros;
- a Dra. Denise Draper por gentilmente enviar uma cópia impressa de sua tese;
- o Dr. José Carlos da Rocha pelas inúmeras orientações recebidas;
- o Dr. Cássio Polpo de Campos pelo auxílio prestado aos experimentos conduzidos no trabalho;
- o amigo Daniel Kikuti pela ajuda na revisão do texto e valiosas conversas;
- o amigo de pesquisa Alessandro Antonucci pelas discussões que abriram novos caminhos;
- o amigo Fábio Koichi Nakabayashi pela ajuda indireta prestada na fase final da tese;
- os colegas de pesquisa Haipeng Guo, Christof Baumgen, Honghai Liu pela troca de experiência internacional;

- o colega André Hideaki Saheki pelos inúmeros auxílios de informática recebidos;
- os membros do Laboratório de Telecomando e Tomada da Decisão da USP (Clayton, Fábio Fagundes, Thiago, João Vítor, Fábio Ramos, Ismênia, Mathias, Marcelo, Marko e Waldemar);
- a FAPESP pelo suporte financeiro;
- o Instituto de Pesquisas Eldorado e a HP Labs pelo suporte material;
- os inúmeros amigos e colegas de pós-graduação pelas incontáveis trocas de experiência;
- todos os amigos pelas ajudas indiretas prestadas nos momentos difíceis, sem as quais não teria perseverado no trabalho.

EPÍGRAFE

“As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality.”

Albert Einstein (1879 - 1955)

RESUMO

Modelos baseados em grafos associados a distribuições de probabilidade são amplamente empregados em diversas áreas. *Redes Bayesianas* são os modelos probabilísticos baseados em grafos mais populares na literatura. Uma das dificuldades encontradas no uso de redes Bayesianas ocorre nas situações em que há falta de dados, crenças incompletas, ou divergência entre opiniões de especialistas. Nestas circunstâncias, *incerteza* pode ser representada por conjuntos de medidas de probabilidade, denominado *conjuntos credais*. A associação de conjuntos credais com um *grafo acíclico direcionado* recebe o nome de *rede credal*. Nesta tese, três novos algoritmos para inferência aproximada em redes credais binárias são propostos: *Loopy 2U* (L2U), *Iterated Partial Evaluation* (IPE) e *Structured Variational 2U* (SV2U). O primeiro deles, algoritmo L2U, é uma extensão do algoritmo de propagação de crenças em ciclos (*Loop Belief Propagation*). O segundo, algoritmo IPE, é inspirado na técnica de avaliação parcial localizada (*Localized partial Evaluation*) empregada em redes Bayesianas. O terceiro, algoritmo SV2U, implementa técnicas variacionais; para tanto, são formuladas as aproximações de campo médio em redes credais, na sua forma *primitiva* (totalmente fatorada) e *estruturada* (estrutura de árvore). Os algoritmos foram implementados e estão disponíveis no software 2UBayes. Experimentos foram realizados e uma análise comparativa entre os algoritmos foi feita. Resultados obtidos mostraram que aproximações precisas são obtidas com menor esforço computacional que demais algoritmos existentes.

ABSTRACT

Graphical models associated with probabilities find use in many fields. Bayesian networks are the most popular probabilistic graphical model in the literature. In situations where we have lack of data, incomplete beliefs and divergence between expert opinions, uncertainty can be represented by sets of probability measures called credal sets. Such sets, when associated with directed acyclic graphs, result in credal networks. Inference algorithms in credal networks generally display high complexity, and approximate inference seems to be a natural solution for large networks. In this thesis, we present three new approximate algorithms for inference in binary credal networks: Loopy 2U (L2U), Iterated Partial Evaluation (IPE) and Structured Variational 2U (SV2U). The first one, the L2U algorithm, is an extension of the Loopy Belief Propagation algorithm for Bayesian network inference. The second one, the IPE algorithm, is directly based on the Localized Partial Evaluation (LPE) technique. Finally, the SV2U algorithm implements a variational approach; in this work, it is shown how to formulate mean field approximations for credal sets using naive (fully factorized) and structured (tree-like) schemes. The algorithms were implemented and a software package (2UBayes) has been made available. Experiments were conducted and a comparative analysis between algorithms was performed. These empirical results showed that accurate approximations with low computational cost are achieved.

SUMÁRIO

Lista de Figuras

Lista de Tabelas

Lista de Siglas

Lista de Símbolos

1	Introdução	p. 20
2	Redes Bayesianas e seus Métodos de Aproximação	p. 24
2.1	Terminologia básica sobre grafos	p. 24
2.2	Redes Bayesianas	p. 27
2.3	Algoritmos para inferência em redes Bayesianas	p. 30
2.4	Propagação de mensagens (algoritmo BP) e “Loopy Belief Propagation” (algoritmo LBP)	p. 31
2.4.1	Propagação local	p. 32
2.4.2	Propagação global	p. 33
2.4.3	Propagação em redes com ciclo	p. 34
2.5	Método de avaliação parcial local (algoritmo LPE)	p. 36
2.6	Métodos variacionais em redes Bayesianas	p. 39

2.6.1	Aproximação variacional	p. 40
2.6.2	Aproximação de campo médio	p. 40
2.6.3	Importância da computação local	p. 43
2.6.4	Aproximação de campo médio estruturado	p. 44
2.6.5	Exemplo: aproximação de campo médio estruturado na rede Pyramid	p. 46
3	Redes Credais	p. 48
3.1	Conjuntos credais	p. 49
3.2	Redes credais com extensão forte	p. 52
3.3	Inferência em redes credais	p. 54
3.4	Inferência em redes credais binárias	p. 56
3.5	Algoritmo 2U: propagação de intervalos exatos para poli-árvores binárias	p. 58
3.5.1	Propagação local de intervalos	p. 58
3.5.2	Propagação global de intervalos	p. 62
4	Novos Algoritmos para Inferência Aproximada em Redes Credais	p. 66
4.1	“Loopy Belief Propagation” em redes credais: algoritmo L2U	p. 68
4.1.1	Descrição do algoritmo L2U	p. 68
4.1.2	Exemplo numérico de aplicação do algoritmo L2U	p. 71
4.2	“Localized Partial Evaluation” em redes credais: algoritmo IPE	p. 74
4.2.1	Descrição do algoritmo IPE	p. 74
4.2.2	Exemplo de aplicação do algoritmo IPE	p. 78

4.3	Métodos variacionais em redes credais	p. 80
4.3.1	Abordagem baseada em conjuntos	p. 81
4.3.2	Exemplo: Campo médio baseado em conjuntos em rede com 2 nós	p. 84
4.3.3	Algoritmo SV2U	p. 86
4.3.4	Exemplo: SV2U em rede com 3 nós	p. 89
5	Experimentos e Análise Comparativa	
	entre os Algoritmos	p. 90
5.1	Metodologia dos experimentos conduzidos	p. 90
5.2	Experimentos em redes credais geradas aleatoriamente	p. 92
5.3	Experimentos em redes credais com estrutura conhecida	p. 93
5.3.1	Testes com o algoritmo L2U	p. 97
5.3.2	Testes com o algoritmo IPE	p. 98
5.3.3	Testes com o algoritmo SV2U	p. 100
5.4	Discussão dos resultados	p. 101
6	Considerações Finais	p. 107
	Referências	p. 110
	Apêndice I – <i>Software 2UBayes</i>	p. 116

LISTA DE FIGURAS

- 1 Tipos de grafos utilizados em redes Bayesianas. p. 25
- 2 *Markov blanket* do nó H_j . O conjunto $\{pa_k \setminus H_j\}$ representa os pais de X_k , excluindo H_j p. 26
- 3 A rede *Dog-Problem* (CHARNIAK, 1991). p. 28
- 4 Rede Bayesiana com 4 nós e suas distribuições de probabilidade. Devemos ter $p_i + q_i = 1$ ($1 \leq i \leq 10$), por se tratarem de valores de probabilidade. p. 29
- 5 Esquema de propagação de mensagens (PEARL, 1988). $\pi_X(U_i)$ e $\lambda_{Y_i}(X)$ são mensagens que o nó X recebe (setas com linha cheia), para assim atualizar os valores $\pi(X)$ e $\lambda(X)$ (Equações (2.4) e (2.5)). $\pi_{Y_i}(X)$ e $\lambda_X(U_i)$ são mensagens que o nó X envia aos seus pais e filhos (setas com linha tracejada). p. 32
- 6 Algoritmo LBP em uma rede simples com 4 nós. p. 35
- 7 Ilustração do método LPE: escolha de um *active set* (área hachurada) e inserção de mensagens *vazias* (setas). p. 37
- 8 Representação das mensagens propagadas no algoritmo LPE. O *active set* é composto pela área hachurada mais o resto da rede. O arco pontilhado representa o arco ausente inserido no lugar do arco original. p. 38
- 9 Ilustração esquemática de métodos variacionais. A entropia relativa KL (*Kullback-Leibler*) é utilizada para medir a “distância” entre a distribuição desejada e aproximada. p. 41
- 10 Ilustração de como a distância KL é minimizada, adotando-se distribuições aproximadas totalmente fatoradas, através de computações locais. p. 44

11	Exemplo de aplicação da aproximação de campo médio na rede Pyramid. Os arcos tracejados são os arcos removidos da rede original para obter uma poli-árvore.	p. 46
12	Representação gráfica do conjunto credal do Exemplo 4.	p. 50
13	Representação gráfica do conjunto credal do Exemplo 5.	p. 50
14	Um exemplo de rede credal com 2 variáveis binárias, associadas a conjuntos credais especificados em separado.	p. 53
15	Exemplo de uma rede credal: rede <i>Dog-Problem</i> associada a variáveis binárias com os respectivos conjuntos credais locais. . .	p. 54
16	Intervalos exatos, aproximações internas e aproximações externas para a probabilidade de um evento.	p. 55
17	Exemplo de rede credal com conjuntos credais especificados em separado. As variáveis são binárias.	p. 57
18	Esquema de propagação de mensagens do algoritmo 2U (FAGIOLI; ZAFFALON, 1998). $\pi_X(U_i)$ e $\Lambda_{Y_i}^X$ são mensagens que o nó X recebe de seus pais e filhos (setas com linha cheia), para assim atualizar os valores $\pi(X)$ e Λ^X (Equações (3.6) e (3.7)). $\pi_{Y_i}(x)$ e $\Lambda_X^{U_i}$ são mensagens que o nó X envia aos seus pais e filhos (setas com linha tracejada).	p. 60
19	Resumo esquemático dos algoritmos desenvolvidos.	p. 67
20	Algoritmo L2U.	p. 69
21	Ilustração de como os <i>arcos ausentes</i> (arcos com linha pontilhada) são inseridos. Esquerda: a rede multi-conectada original; direita: rede poli-árvore com <i>arcos ausentes</i> e suas mensagens vazias. . .	p. 75
22	Ilustração do processo de refinamento dos intervalos externos obtidos. Em termos gráficos, o intervalo externo final consiste na interseção dos intervalos.	p. 76
23	Descrição do algoritmo <i>Iterated Partial Evaluation</i> (IPE).	p. 77
24	Aproximação de <i>limite externo</i> . O intervalo $I_Q^*(H_0 = h_0)$ é aproximado por uma borda externa $I_Q(H_0 = h_0)$	p. 82

25	<i>Campo médio baseado em conjuntos: uma aplicação do método variacional de campo médio para inferência em redes credais.</i> . . .	p. 83
26	Rede credal com 2 nós associados a variáveis binárias (visualizada no sistema JavaBayes, pacote que permite representar conjuntos credais locais e extensão forte).	p. 85
27	Descrição de implementação do algoritmo SV2U.	p. 87
28	Esquerda: rede credal multi-conectada com 3 variáveis binárias. Direita: uma estrutura poli-árvore aproximada, diferindo no conjunto credal local associado à variável <i>C</i>	p. 88
29	A rede <i>Pyramid</i> , grafo multi-camadas com 28 variáveis binárias, 34 arcos e densidade de arcos 1.21.	p. 93
30	A rede <i>Alarm</i> possui 37 variáveis binárias, 46 arcos e densidade de arcos 1.24. O console mostra a variável “Disconnect” sendo editada.	p. 97
31	Correlação entre os valores extremos aproximados (obtidos pelo algoritmo L2U) e exatos, em dois exemplos: (a) rede <i>Pyramid</i> , com 28 variáveis binárias, sem evidências, e convergência em 4 iterações;(b) a rede <i>Alarm</i> “binarizada”, com 37 variáveis, sem evidências, e convergência em 9 iterações.	p. 98
32	Correlação entre os valores extremos aproximados (obtidos pelo algoritmo IPE) e exatos, em dois exemplos: (a) rede <i>Pyramid</i> , com 28 variáveis binárias, sem evidências, 100 iterações e MSE = 5.1%; (b) rede <i>Alarm</i> “binarizada”, com 37 variáveis, sem evidências, 100 iterações e MSE=7.2%.	p. 99
33	Rede <i>Multidog</i> (uma versão modificada da <i>Dog-Problem</i>), uma rede credal com 5 nós binários.	p. 100
34	Correlação entre os valores extremos aproximados (obtidos pelo algoritmo SV2U) e exatos na rede <i>Pyramid</i> . As probabilidades inferior/superior de todas as variáveis são computadas de modo exato e aproximado, obtendo-se MSE=2%, em 0.078 segundos. .	p. 102

35	Correlação entre os valores extremos aproximados (obtidos pelo algoritmo SV2U) e exatos na rede <i>Alarm</i> “binarizada”. As probabilidades inferior/superior de todas as variáveis são computadas de modo exato e aproximado, obtendo-se $MSE = 2.9\%$, em 0.422 segundos.	p. 102
36	Ilustração do console do programa 2UBayes.	p. 117
37	Ilustração da rede credal sendo carregada no programa 2UBayes.	p. 118
38	Ilustração da execução do algoritmo L2U no programa 2UBayes.	p. 118

LISTA DE TABELAS

- 1 Vértices da extensão forte da rede credal da Figura 14. p. 53
- 2 Fórmulas para obter os valores $\widehat{\Lambda}_X^{U_i}(\Lambda^X)$ das Equações (3.19) e (3.20). p. 62
- 3 Tabela contendo mensagens intervalares do algoritmo L2U para a rede credal da Figura 17, para $t=0, 1$ e 2 p. 74
- 4 Tabela contendo parâmetros de entrada do BNGenerator para geração de amostras de redes credais. $nNodes$ denota o número de nós das $nBNs$ redes geradas. $maxArcs$ é o número arcos da rede. $nval = 2$ denota variável binária e $structure = multi$ denota a estrutura multi-conectada das redes geradas. p. 92
- 5 Resultados obtidos pelos algoritmos desenvolvidos, executando-os em redes credais binárias com 10 nós e 12 arcos (densidade de arcos 1.2). p. 94
- 6 Resultados obtidos pelos algoritmos desenvolvidos, executando-os em redes credais binárias com 10 nós e 14 arcos (densidade de arcos 1.4). p. 95
- 7 Resultados obtidos pelos algoritmos desenvolvidos, executando-os em redes credais binárias com 10 nós e 16 arcos (densidade de arcos 1.6). p. 96
- 8 MSE e tempo médio de processamento (em segundos) obtidos pelos algoritmos no conjunto de redes geradas aleatoriamente. O símbolo d denota a densidade das redes. p. 103
- 9 Média dos MSE e tempos de processamento obtidos pelos algoritmos em redes com estrutura conhecida. p. 103

- 10 Análise comparativa entre os algoritmos L2U, IPE e SV2U. A notação foi simplificada na tabela, por questões de espaço. Ver as respectivas observações para maiores detalhes. p. 106

LISTA DE SIGLAS

Siglas utilizadas no documento:

<i>Sigla</i>	<i>Descrição</i>
<i>2U</i>	algoritmo de propagação de intervalos exatos
<i>L2U</i>	algoritmo <i>Loopy 2U</i>
<i>IPE</i>	algoritmo <i>Iterated Partial Evaluation</i>
<i>LPE</i>	algoritmo <i>Localized Partial Evaluation</i>
<i>BP</i>	algoritmo <i>Belief Propagation</i>
<i>LBP</i>	algoritmo <i>Loopy Belief Propagation</i>
<i>SV2U</i>	algoritmo <i>Structured Variational 2U</i>

LISTA DE SÍMBOLOS

Símbolos utilizados no documento:

<i>Símbolo</i>	<i>Descrição</i>
$cc(\cdot)$	operação de casco convexo
\mathbf{C}	rede credal qualquer
\mathbf{C}_P	rede credal com estrutura poli-árvore
c_i	conjunto de variáveis compostas por X_i e pa_i (<i>cluster</i> de X_i na rede \mathbf{C}_P)
C_{X_j}	conjunto de c_i que dependem de X_j , excluindo c_j
\mathbf{E}	conjunto de variáveis observadas
g_k	conjunto de variáveis compostas por X_k e pa_k (<i>cluster</i> de X_k na rede \mathbf{C})
G_{X_j}	conjunto de g_k que dependem de X_j
\mathbf{H}	conjunto de variáveis não observadas (ocultas)
$I(\cdot)$	intervalo de probabilidade
i_{max}	número máximo de iterações
$KL(q p)$	entropia relativa de <i>Kullback-Leibler</i> entre as distribuições q e p
$K(\cdot)$	conjunto credal local
$\mathbf{K}(\cdot)$	conjunto de $K(\cdot)$ associado a uma variável da rede credal
\mathbf{K}	conjunto de $\mathbf{K}(\cdot)$ associado a uma rede credal
k_λ	constante de normalização na atualização variacional
n_{pa}	número de pais
$p(\cdot)$	distribuição de probabilidade
$P(\cdot)$	valor de probabilidade
$P(x_i)$	valor de probabilidade assumida por $p(X_i = x_i)$
$\underline{P}(\cdot)$	valor de probabilidade inferior
$\overline{P}(\cdot)$	valor de probabilidade superior
$\underline{P}^*(\cdot)$	valor de probabilidade inferior aproximado
$\overline{P}^*(\cdot)$	valor de probabilidade superior aproximado

pa_i	conjunto de nós pais de X_i
$q(\cdot)$	distribuição de probabilidade aproximada
$q^*(\cdot)$	distribuição de probabilidade aproximada atualizada
\mathfrak{R}^+	conjunto dos números reais positivos
\mathbf{V}_K	conjunto de vértices de um conjunto credal K
\mathbf{X}	conjunto de variáveis aleatórias
X_i	variável aleatória contido em \mathbf{X}
x_i	estado (categoria) assumido pela variável X_i
X_Q	variável de consulta
$\{X_Q, \mathbf{E}\}$	conjunto de variáveis contendo X_Q e \mathbf{E}
$\mathbf{X} \setminus \{X_Q, \mathbf{E}\}$	conjunto de todas as variáveis contidas em \mathbf{X} , excluindo X_Q e \mathbf{E}
α, β	constantes de normalização durante propagação de mensagens
$\pi_A(B)$	mensagem enviada do nó pai B para o nó filho A
$\lambda_A(B)$	mensagem pontual enviada do nó filho A para o nó pai B
$\Lambda_A(B)$	mensagem intervalar enviada do nó filho A para o nó pai B
$\pi(X)$	suporte de previsão (probabilidade de X dadas as evidências provenientes dos nós não-descendentes)
$\lambda(X)$	suporte retrospectivo (probabilidade de X dadas as evidências provenientes dos nós descendentes)
$\Lambda(X)$	suporte retrospectivo de X no algoritmo 2U
λ_j	multiplicador de Lagrange

1 INTRODUÇÃO

A teoria de probabilidade é ferramenta básica em inteligência artificial hoje em dia. Ela permite manusear incerteza, muito comum em problemas reais (RUSSELL; NORVIG, 1995). Nela, a incerteza é quantificada na forma de valores de probabilidade.

Os *modelos probabilísticos baseados em grafos (Probabilistic Graphical Models)* (PEARL, 1988) têm-se demonstrado muito úteis para representar uma grande quantidade de variáveis aleatórias e visualizar suas dependências. Tais modelos são empregados com sucesso em diversas áreas, desde aplicações em engenharia (MADSEN *et al.*, 2004), em robótica (THRUN, 2000), em economia para análise de risco de portfólio (SHENOY, 1999), em estudos de estratégia de marketing (BAESENS *et al.*, 2004) e até utilizações em biologia (FRIEDMAN *et al.*, 2000). As *redes Bayesianas* (CHARNIAK, 1991) são um dos modelos mais populares (JENSEN, 1996) (COWELL *et al.*, 1999). Elas são grafos acíclicos direcionados associados a distribuições de probabilidade.

Existem situações em que medidas exatas de probabilidade não são disponíveis, seja por falta de informações mais detalhadas ou pela divergência de opinião de especialistas. Nestas situações, uma forma de representar imprecisão em probabilidade é através do uso de *intervalos de probabilidade*, ou, de forma mais geral, de *conjuntos de probabilidade* (WALLEY, 1991). *Rede credal* (COZMAN, 2000) (FAGIOLI; ZAFFALON, 1998) é o nome dado ao modelo baseado em grafos associado a conjuntos de probabilidade. Algumas aplicações utilizando redes credais têm sido desenvolvidas recentemente (KRIEGLER; HELD, 2003), (ANTONUCCI; SALVETTI; ZAFFALON, 2004). Para o avanço destas aplicações é necessário o desenvolvimento de eficientes algoritmos para inferência em redes credais.

Inferência em redes credais consiste no cálculo de valores inferiores e su-

periores de um intervalo de probabilidade. Este cálculo pode ser visto como um problema de otimização combinatória e possui uma alta complexidade computacional¹. Até para redes com estrutura mais simples como *poli-árvores*, a realização de *inferências exatas*² é um problema computacional *NP-completo* (ROCHA, 2004). Esta complexidade tem motivado inúmeras pesquisas (ROCHA, 2004) (CAMPOS, 2005), e algoritmos que realizam *inferências aproximadas*³ são vistos como naturais alternativas para contornar a complexidade (TESSEM, 1992) (CANO; MORAL, 1996) (CANO; MORAL, 2001) (CANO; MORAL, 2002).

Neste trabalho são propostos três novos algoritmos para inferência aproximada em redes credais associadas a variáveis binárias (*booleanas*) com estrutura multi-conectada. *Redes credais binárias*⁴ são particularmente promissoras para o desenvolvimento de *redes credais relacionais* (COZMAN *et al.*, 2004), que associam modelos probabilísticos imprecisos à *lógica probabilística* (NILSSON, 1986), estendendo trabalhos de probabilidade imprecisa em lógica (LUKASIEWICZ, 1998). Há inúmeros algoritmos para inferência aproximada em redes credais encontrados na literatura (TESSEM, 1992) (CANO; MORAL, 1996) (ROCHA; COZMAN; CAMPOS, 2003) (CAMPOS, 2005), mas nenhum deles enfoca especificamente o problema de inferência em redes credais *binárias*.

Os algoritmos propostos transformam o problema de inferência exata em redes credais binárias inicialmente *intratável* (problema computacional *NP-completo*) em um problema de inferência aproximada *tratável* (problema computacional com algoritmos executados em tempo polinomial). Essa estratégia é utilizada com sucesso em redes Bayesianas e outros modelos probabilísticos (SAUL; JORDAN, 1996), sendo estendida para redes credais neste trabalho. Mais precisamente, os algoritmos apresentados transformam o problema de inferência em redes credais binárias com *estrutura multi-conectada* em problema de inferência em redes credais binárias com *estrutura poli-árvore* -- para redes com esta estrutura existe um eficiente algoritmo, denominado 2U (FAGIUOLI; ZAFFALON, 1998), capaz de

¹A *alta complexidade* decorre do aumento exponencial de combinações possíveis a serem analisadas, conforme discutido na Seção 3.4.

²O termo *inferência exata* é empregado para se referir ao cálculo exato de valores de probabilidade.

³O termo *inferência aproximada* é empregado para se referir ao cálculo aproximado de valores de probabilidade.

⁴A expressão *redes credais binárias* é utilizada para se referir às redes credais associadas a variáveis binárias.

realizar inferências exatas em tempo polinomial.

Os métodos de aproximação desenvolvidos (para aproximar redes credais com estruturas multi-conectadas por poli-árvores) valem não só para redes credais binárias, mas também para redes com variáveis não-binárias e até contínuas. Porém, esta possível extensão de aplicação dos métodos de aproximação está condicionada ao desenvolvimento de algoritmos eficientes para inferência em redes credais com estrutura poli-árvore, e não está no escopo do presente trabalho.

Os algoritmos apresentados neste trabalho são:

1. Algoritmo L2U (*Loopy 2U*) (IDE; COZMAN, 2004b). Este algoritmo é extensão do popular algoritmo para redes Bayesianas denominado *Loopy Belief Propagation* (MURPHY; WEISS; JORDAN, 1999) e conhecido em transmissão de dados por *Turbo Decoding* (MCELIECE; MACKAY; CHENG, 1998).
2. Algoritmo IPE (*Iterated Partial Evaluation*) (IDE; COZMAN, 2004b). Este algoritmo é inspirado no método *Localized Partial Evaluation* (DRAPER; HANKS, 1994) empregado para inferência aproximada em redes Bayesianas.
3. Algoritmo SV2U (*Structured Variational 2U*) (IDE; COZMAN, 2005a) (IDE; COZMAN, 2005b). Este algoritmo é uma aplicação de técnicas variacionais em redes credais associadas a variáveis categóricas (discretas). Técnicas variacionais vêm sendo empregadas com êxito em diversos modelos probabilísticos, tais como redes Bayesianas com variáveis contínuas não gaussianas, *Boltzmann machines* e *Hidden Markov Models* (HMM) (JORDAN; GHARAMANI; JAAKKOLA, 1999). Os métodos variacionais são bastante flexíveis, podendo ser empregados para o manuseio de distribuições categóricas (discretas) e contínuas. Em particular, os métodos variacionais são promissores para o manuseio de variáveis contínuas como *Imprecise Dirichlet Model* (WALLEY, 1991) e de probabilidades imprecisas da família exponencial (QUAEGHEBEUR; COOMAN, 2005).

Esta tese contribui com soluções inovadores para inferência em redes credais. Os algoritmos foram implementados, testados e disponibilizados na internet. Assim, o trabalho avançou no estado-da-arte na manipulação de modelos probabilísticos baseados em grafos.

Este documento está organizado como segue. No Capítulo 2, redes Bayesianas e seus métodos de aproximação são descritos. No Capítulo 3, redes credais e o algoritmo exato de propagação de intervalos 2U são descritos. No Capítulo 4, apresentam-se as maiores contribuições desta tese — desenvolvimento e implementação de algoritmos para realização de inferência aproximada em redes credais binárias em tempo polinomial. No Capítulo 5, os experimentos conduzidos e os resultados obtidos são apresentados e discutidos. No Capítulo 6, as principais conclusões deste trabalho são apresentadas. Os algoritmos implementados estão disponíveis no programa desenvolvido 2UBayes (Apêndice I).

2 REDES BAYESIANAS E SEUS MÉTODOS DE APROXIMAÇÃO

Este capítulo apresenta uma revisão de tópicos necessários para o entendimento de capítulos posteriores. Definições relacionadas a grafos são apresentadas na Seção 2.1. Conceitos básicos sobre redes Bayesianas e seus principais algoritmos para inferência são revisados nas Seções 2.2 e 2.3, respectivamente. Algoritmos e métodos de aproximação, diretamente relacionados com os algoritmos desenvolvidos na tese, são detalhados nas posteriores seções: na Seção 2.4, o algoritmo de propagação de mensagens de Pearl, assim como a sua versão “loopy” são revisados; na Seção 2.5, o método de avaliação local é descrito; na Seção 2.6, apresenta-se uma breve descrição de técnicas variacionais.

2.1 Terminologia básica sobre grafos

Nesta seção é estabelecida uma terminologia de grafos que será empregada no restante do documento, assim como a definição de alguns conceitos. Maiores detalhes sobre teoria de grafos são encontrados em (CHARTRAND; OELLERMANN, 1993); (CASTILLO; GUTIERREZ; HADI, 1997), entre outros.

Um *grafo direcionado* é denotado por $\mathbf{G} = (\mathbf{V}, \mathbf{A})$, onde \mathbf{V} é um conjunto de nós e \mathbf{A} é um conjunto de arcos. Todo arco $a = (u, v)$ contido em \mathbf{A} é *direcionado*, ou seja, parte de um nó u e chega a outro nó v .

O *caminho* entre dois nós, $v[0]$ e $v[k]$, é uma seqüência de nós e de arcos $(v[0], a[1], v[1], a[2], \dots, a[k], v[k])$, onde $a[i] = (v[i-1], v[i])$ ou $a[i] = (v[i], v[i-1])$. Um *ciclo* é um caminho em que $v[0]$ e $v[k]$ são o mesmo nó. Um ciclo é *direcionado* se $a[i] = (v[i-1], v[i])$, para todo i . Um *grafo direcionado acíclico* (*Directed Acyclic Graph*) é um grafo que não contém nenhum ciclo direcionado. Um *grafo direcionado acíclico pode conter um ciclo*, desde que ele não seja direcionado.

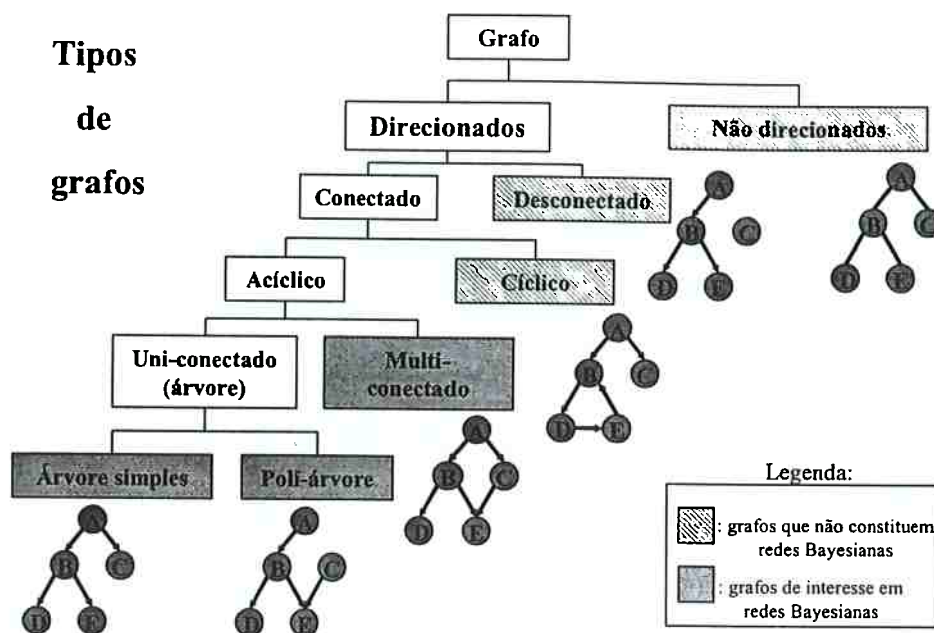


Figura 1: Tipos de grafos utilizados em redes Bayesianas.

Um grafo é *conectado* se existe caminho entre quaisquer dois nós pertencentes ao conjunto de vértices V . Um grafo é *uni-conectado* (também denominado *árvore*), se existe apenas um caminho entre quaisquer dois nós; caso contrário, o grafo é *multi-conectado*. Se todo nó da árvore possui apenas um arco chegando, ela será *árvore simples*; senão, ela será *poli-árvore*¹. Na Figura 1, apresenta-se uma classificação dos tipos de grafos, segundo características previamente descritas. As árvores simples, as poli-árvores e os grafos multi-conectados são grafos utilizados em redes Bayesianas.

As seguintes definições empregadas em redes Bayesianas (COWELL *et al.*, 1999) são utilizadas na tese:

- Em um arco $a = (u, v)$, u é o nó *pai* de v , e v é o nó *filho* de u . Diz-se que u e v são nós *vizinhos* ou *adjacentes*.
- Um nó é *raiz* se não possui nenhum nó pai; e um nó é uma *folha*, se não possui nenhum filho.
- Os *antecessores* (ou também *não-descendentes*) de um nó u são todos os

¹Como acontece na literatura (CASTILLO; GUTIERREZ; HADI, 1997), o termo *poli-árvore* será empregado no documento para se referir aos grafos uni-conectados em geral.

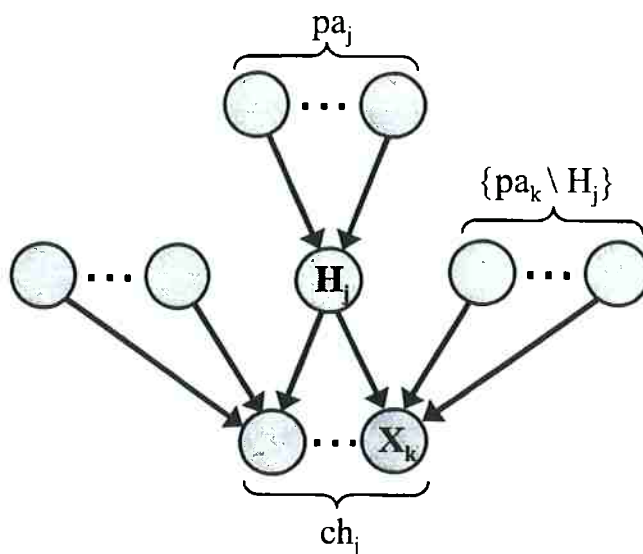


Figura 2: *Markov blanket* do nó H_j . O conjunto $\{pa_k \setminus H_j\}$ representa os pais de X_k , excluindo H_j .

nós que possuem um caminho direcionado até o nó u .

- O *Markov blanket* de um nó v é definido como sendo o conjunto de nós formado pelos pais e filhos de v e pelos pais dos filhos de v , excluindo o próprio nó v . Na Figura 2, é apresentado o *Markov blanket* do nó H_j .
- O *conjunto de condicionamento* (*cutset*) de um grafo multi-conectado $G = (V, A)$ é um conjunto de arcos A^* tal que, quando removido do grafo original, resulta um grafo poli-árvore $G^* = (V, \{A \setminus A^*\})$, onde $\{A \setminus A^*\}$ representa o conjunto inicial de arcos A menos o conjunto de condicionamento A^* .
- Seguindo a terminologia adotada por Wiegerinck (2000) e Winn (2003), o *cluster* de um nó v denota o conjunto de nós contendo o próprio nó v e seus pais.

Por exemplo, no grafo multi-conectado da Figura 1, o *cluster* do nó B é o conjunto $\{B, A\}$, o seu *Markov blanket* é o conjunto dos nós $\{A, D, E, C\}$ --- e um conjunto de condicionamento possível é constituído pelo arco (C, E) .

2.2 Redes Bayesianas

Nesta seção, apresentam-se os conceitos básicos sobre redes Bayesianas, assim como o seu problema de inferência.

Redes Bayesianas foram desenvolvidas na década de 80, como uma ferramenta para construção de sistemas especialistas (STEVENS, 1985). A característica principal de redes Bayesianas é a utilização de probabilidades, ao contrário de modelos que empregam *lógica difusa* (ZADEH; KACPRZYK, 1992), teoria de *Dempster-Shafer* (SHAFER, 1976), entre outros.

Uma *rede Bayesiana* (JENSEN, 1996) é definida como a tripla $(\mathbf{G}, \mathbf{X}, p(\mathbf{X}))$, onde \mathbf{G} é um grafo direcionado acíclico, \mathbf{X} é um conjunto de variáveis aleatórias e $p(\mathbf{X})$ é uma *distribuição de probabilidade conjunta* definida sobre \mathbf{X} . Cada variável $X_i \in \mathbf{X}$ é associada a um nó do grafo. A estrutura do grafo indica as relações de independência na rede: assume-se que *uma variável X_i é independente de seus não-descendentes, condicional em seus pais* (tal propriedade recebe o nome de *condição de Markov*). A condição de *Markov* permite que a distribuição conjunta $p(\mathbf{X})$ seja representada de forma compacta pelo produto de distribuições associadas a cada variável X_i :

$$p(\mathbf{X}) = \prod_i p(X_i | pa_i),$$

onde pa_i é o conjunto de pais de X_i e $p(X_i | pa_i) = p(X_i)$ quando X_i não tem pai.

Exemplo 1 . Para exemplificar a representação por redes Bayesianas, considere a Figura 3. Essa rede Bayesiana representa os hábitos de uma família. Cada nó do grafo representa uma variável, e cada arco representa uma dependência direta entre variáveis. Cada nó é também associado a uma distribuição de probabilidade. A variável F indica se a família está ou não em casa, e afeta diretamente a variável L (se a luz está acesa ou não). O nó D é associado à distribuição $p(D|F, B)$ — distribuição de D condicional às variáveis F e B , que “afetam” diretamente D . A distribuição conjunta da rede é dada por:

$$p(F, B, L, D, H) = p(F) \times p(B) \times p(L|F) \times p(D|F, B) \times p(H|D).$$

Neste trabalho, só se consideram distribuições de probabilidade associadas a

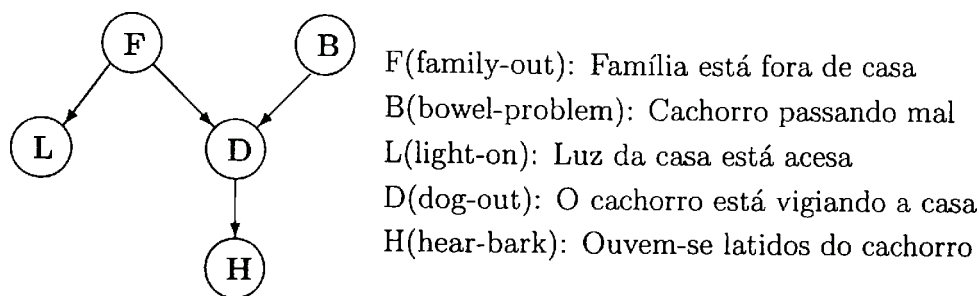


Figura 3: A rede *Dog-Problem* (CHARNIAK, 1991).

variáveis aleatórias discretas. Uma variável aleatória é *discreta* quando ela pode assumir *categorias* ou *estados* com dadas probabilidades. Se uma variável X_i assume um conjunto discreto de estados com s_i elementos $\{x_{i,1}, x_{i,2}, \dots, x_{i,s_i}\}$, com probabilidades $(p_{i,1}; p_{i,2}; \dots; p_{i,s_i})$, respectivamente; sendo $p_{i,1} + p_{i,2} + \dots + p_{i,s_i} = 1$, diz-se que está definida uma *distribuição de probabilidade discreta* $p(X_i)$ de X_i . $p(X_i)$ é uma função que assume os valores $(p_{i,1}; p_{i,2}; \dots; p_{i,s_i})$, para $X_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,s_i}\}$, respectivamente. Distribuições de probabilidade discreta são normalmente representadas na forma de *tabelas* de valores de probabilidade².

Exemplo 2 . Considere uma rede Bayesiana com 4 nós, com variáveis assumindo estados binários: 0 ou 1, onde cada variável está associada a uma distribuição de probabilidade representada na forma de tabela (Figura 4). A distribuição conjunta é expressa por:

$$p(A, B, C, D) = p(A) \times p(B) \times p(C|A, B) \times p(D|A, B). \quad (2.1)$$

Redes Bayesianas condensam conhecimento sobre determinado domínio, especificado na forma de distribuições condicionais de probabilidade. A partir de uma rede Bayesiana, inferências são realizadas. Uma *inferência* consiste no cálculo de valores de probabilidade, denotados por $P(\cdot)$, efetuado sobre a distribuição conjunta, dado uma coleção de eventos \mathbf{E} . *Eventos* são definidos sobre um conjunto de *variáveis observadas*. Uma variável X_i está *observada* quando ela assume um estado x_i dentre as possíveis categorias. A expressão geral para

²Distribuições condicionais de variáveis discretas são freqüentemente denominadas *pct* (*probability conditional table*) na literatura.

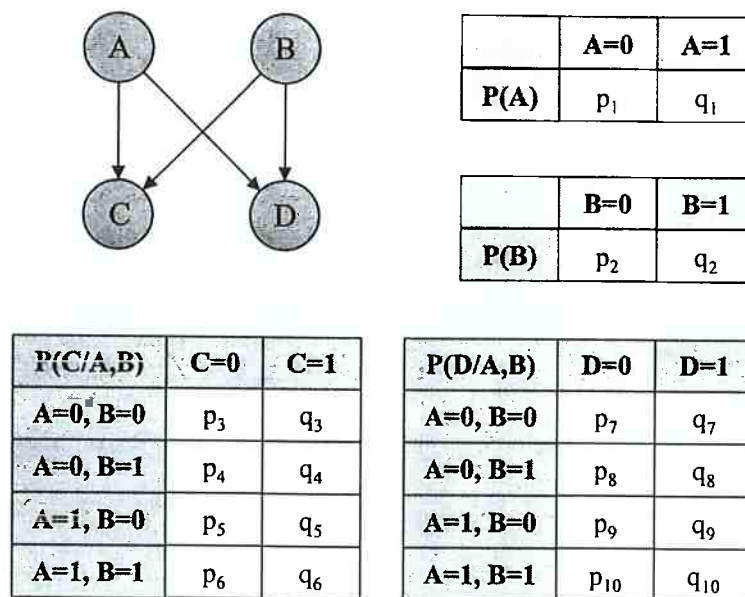


Figura 4: Rede Bayesiana com 4 nós e suas distribuições de probabilidade. Devemos ter $p_i + q_i = 1$ ($1 \leq i \leq 10$), por se tratarem de valores de probabilidade.

inferências em redes Bayesianas é:

$$p(X_Q|\mathbf{E}) = \frac{p(X_Q, \mathbf{E})}{p(\mathbf{E})} = \frac{\sum_{\mathbf{X} \setminus \{X_Q, \mathbf{E}\}} p(\mathbf{X})}{\sum_{\mathbf{X} \setminus \mathbf{E}} p(\mathbf{X})}, \quad (2.2)$$

onde X_Q é a variável de consulta (ou um conjunto de variáveis). A notação $\mathbf{X} \setminus \{X_Q, \mathbf{E}\}$ representa o conjunto de todas as variáveis contidas em \mathbf{X} , excluindo X_Q e \mathbf{E} . A notação $\sum_{\mathbf{X} \setminus \mathbf{E}} p(\mathbf{X})$ representa a somatória de produtos $\prod P(X_i = x_i | p_{a_i})$ efetuada sobre as variáveis $X_i \in \{\mathbf{X} \setminus \mathbf{E}\}$, para cada um dos estados x_i assumidos por X_i . As variáveis não somadas permanecem fixas durante a somatória. Uma vez obtido a distribuição $p(X_Q|\mathbf{E})$, a partir dela obtém-se os valores de probabilidade $P(X_Q = x_Q|\mathbf{E})$, onde x_Q é um estado de X_Q .

Exemplo 3 . Na rede Bayesiana da Figura 3, pode-se encontrar a probabilidade de “cachorro vigiando” dado a observação “luz acesa” $P(D = true|L = true)$, calculando-se a distribuição de probabilidade $p(D|L = true)$ ³ dada pela ex-

³Note que $P(\cdot)$ representa um valor de probabilidade, enquanto que $p(\cdot)$ representa uma distribuição de probabilidade.

pressão:

$$\begin{aligned}
 p(D|L = true) &= \frac{p(D, L = true)}{p(L = true)} = \frac{\sum_{F,B,H} p(F, B, L, D, H)}{\sum_{F,D,B,H} p(F, B, L, D, H)} \\
 &= \frac{\sum_{F,B,H} p(F) \times p(B) \times p(L|F) \times p(D|F, B) \times p(H|D)}{\sum_{F,D,B,H} p(F) \times p(B) \times p(L|F) \times p(D|F, B) \times p(H|D)}.
 \end{aligned}$$

2.3 Algoritmos para inferência em redes Bayesianas

Nesta seção, apresenta-se um breve resumo dos algoritmos existentes para inferência em redes Bayesianas. Os algoritmos podem ser divididos em dois tipos: os que realizam inferência exata e os que realizam inferência aproximada.

Os principais algoritmos para inferência exata são:

- Algoritmos para poli-árvores baseados em *propagação de crença* (*Belief Propagation* - BP), também conhecidos por *algoritmos de propagação de mensagens de Pearl* (PEARL, 1988).
- Algoritmos para redes multi-conectadas baseados em *agrupamento de variáveis* (*árvores de junção* (JENSEN, 1996) e *eliminação de variáveis* (DECHTER, 1996)).
- Algoritmos para redes multi-conectadas baseados em *condicionamento de variáveis* (SUERMONDT; COOPER, 1991).
- Algoritmos para redes multi-conectadas baseados na *reversão de arcos* (SHACHTER, 1986).
- Algoritmos para redes multi-conectadas baseados em *valores diferenciais* (DARWICHE, 2000).

Os principais algoritmos para inferência aproximada em redes multi-conectadas são:

- Algoritmos baseados em métodos de *Monte Carlo*: *Logic Sampling* (HENRION, 1988), *Importance Sampling* (FUNG; CHANG, 1989) e *Gibbs Sampling*

(GILKS; RICHARDSON; SPIEGELHALTER, 1996).

- Algoritmos baseados em *avaliações parciais* (DRAPER; HANKS, 1994) ou *redução de valores* (KJAERULFF, 1994).
- Algoritmos baseados na propagação de mensagens em ciclos (*Loopy Belief Propagation* - LBP) (MURPHY; WEISS; JORDAN, 1999) (YEDIDIA; FREEMAN; WEISS, 2000).
- Algoritmos baseados em métodos *variacionais* (JORDAN; GHAHRAMANI; JAAKKOLA, 1999) (JAAKKOLA, 2000).

2.4 Propagação de mensagens (algoritmo BP) e “Loopy Belief Propagation” (algoritmo LBP)

Um dos mais populares e pioneiros algoritmos em redes Bayesianas é o algoritmo de propagação de mensagens de Pearl (*Belief Propagation* - BP) (PEARL, 1988). O algoritmo BP fornece resultados exatos de inferência em redes Bayesianas com estrutura poli-árvore. Este algoritmo é fundamental para o entendimento dos algoritmos desenvolvidos neste trabalho.

Uma intuição sobre o algoritmo BP pode ser obtida pelo problema dos *soldados em linha*. Considere uma fileira de soldados. Suponha que o primeiro soldado deseja saber o número de soldados n_s que estão atrás dele. Uma maneira simples de descobrir n_s é a seguinte. O primeiro soldado começa a contagem “um!”, o seguinte soldado “dois!”, e assim por diante, até chegar no último soldado que contará, por exemplo, “cento e onze!”. Note que a contagem é feita de frente para trás (*forward-propagation*). Resta agora o primeiro soldado obter a informação $n_s = 111$. Isto é feito, transmitindo-se a mensagem n_s de trás para frente (*backward-propagation*). O último soldado transmite para o da frente o valor n_s , e assim recursivamente, até chegar ao soldado que faz cabeça. Deste modo, através da propagação de mensagens locais (de um soldado para outro), uma informação global n_s torna-se conhecida por todos os soldados.

No algoritmo BP, mensagens são propagadas de forma análoga ao exemplo dos soldados em linha. Só que agora, as mensagens transmitidas são distribuições de probabilidade. Cada soldado passa a ser um nó e as mensagens são transmitidas

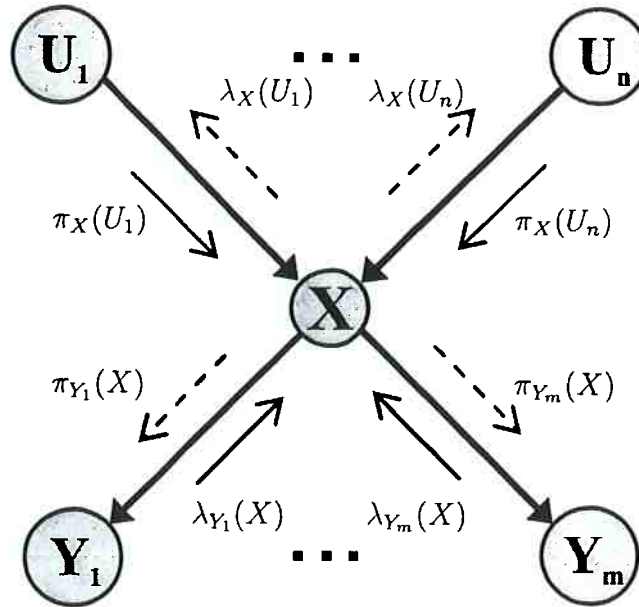


Figura 5: Esquema de propagação de mensagens (PEARL, 1988). $\pi_X(U_i)$ e $\lambda_{Y_i}(X)$ são mensagens que o nó X recebe (setas com linha cheia), para assim atualizar os valores $\pi(X)$ e $\lambda(X)$ (Equações (2.4) e (2.5)). $\pi_{Y_i}(X)$ e $\lambda_X(U_i)$ são mensagens que o nó X envia aos seus pais e filhos (setas com linha tracejada).

pelos arcos do grafo. A contagem de soldados passa a ser uma atualização de probabilidades das variáveis associadas ao grafo, onde cada nó do grafo atualiza sua *crença*⁴ em função da sua crença própria e dos nós adjacentes, e o envia aos nós adjacentes. A propagação de mensagens se dá dos nós raízes aos nós folhas (*forward-propagation*), e dos nós folhas aos nós raízes (*backward-propagation*).

2.4.1 Propagação local

Nesta seção, as *mensagens* do algoritmo BP são definidas. Considere um nó qualquer X da poli-árvore. Denote por $U = \{U_1, U_2, \dots, U_n\}$ o conjunto de pais e por $Y = \{Y_1, Y_2, \dots, Y_m\}$ o conjunto de filhos da variável X , como ilustrado na Figura 5. Denote o conjunto de variáveis observadas por \mathbf{E} , também denominado *conjunto de evidências*. A evidência de nós antecessores de X é representada por e^+ e a evidência de nós descendentes de X é representada por e^- . As mensagens são propagadas do nó pai para nó filho, assim como do nó filho para nó pai — para

⁴O termo *crença* é empregado para se referir a valores de probabilidades.

o nó X , estas mensagens são representadas por $\pi_X(U_i)$ e $\lambda_{Y_j}(X)$, respectivamente (Figura 5).

Suponha que se deseja computar a crença $Bel(X)$ (*belief*) a respeito de X , dada a evidência \mathbf{E} . Esta crença é dada pela expressão (PEARL, 1988):

$$Bel(X) = p(X|\mathbf{E}) = \alpha\pi(X)\lambda(X), \quad (2.3)$$

onde, α é uma constante de normalização, $\pi(X) = p(X|e^+)$ (*suporte de previsão*) e $\lambda(X) = p(e^-|X)$ (*suporte retrospectivo*). Os termos *suporte de previsão* e *suporte retrospectivo* são empregados por Pearl (1988) para indicar as *crenças* de X em relação aos seus pais e filhos, respectivamente. Os suportes de previsão e retrospectivos da Equação (2.3) são computados pelas expressões:

$$\pi(X) = \sum_U p(X|U) \prod_{i=1}^n \pi_X(U_i), \quad (2.4)$$

$$\lambda(X) = \prod_{j=1}^m \lambda_{Y_j}(X). \quad (2.5)$$

Ou seja, dadas mensagens provenientes dos nós pais e filhos, $\pi_X(U_i)$ e $\lambda_{Y_j}(X)$, atualiza-se a crença a respeito de X , $Bel(X)$. Uma vez atualizado $Bel(X)$, resta ainda computar as mensagens a serem enviadas aos pais e filhos de X , para que estes por sua vez atualizem suas crenças:

$$\pi_{Y_j}(X) = \pi(X) \prod_{k \neq j} \lambda_{Y_k}(X), \quad (2.6)$$

$$\lambda_X(U_i) = \beta \sum_X \lambda(X) \sum_{\{U \setminus U_i\}} P(X|U) \prod_{k \neq i} \pi_X(U_k). \quad (2.7)$$

$\pi_{Y_j}(X)$ (Equação (2.6)) é a mensagem enviada ao filho Y_j , contendo as crenças dos pais, $\pi(X)$, e dos filhos de X , $\lambda_{Y_k}(X)$, excluindo Y_j . A mensagem enviada ao pai U_i é dada pela Equação (2.7), onde β é uma constante de normalização.

2.4.2 Propagação global

A propagação global das mensagens pela rede é realizada conforme as seguintes etapas:

1. Inicialização das mensagens da rede. Há três tipos de nós que possuem

mensagens fixas inicializadas no início da propagação. Para os nós raízes não observados, $\pi(X) = p(X)$. Para os nós folhas não observados, $\lambda(X)$ é um vetor unitário de dimensão s , onde s é o número de estados assumidos pela variável aleatória X . Enfim, se X é uma variável observada com evidência $\{X = x\}$, $\lambda(X)$ é um vetor, onde $\lambda(X = x) = 1$ e nulo para os demais estados de X .

2. Propagação das mensagens ao longo da rede. A propagação se dá em dois sentidos: dos nós raízes aos nós folhas (*forward-propagation*) e dos nós folhas aos nós raízes (*back-propagation*). Para cada nó atualizado, a propagação local é realizada conforme apresentada na Seção 2.4.1.
3. Computação das crenças em cada um dos nós, $Bel(X)$.

2.4.3 Propagação em redes com ciclo

Loopy Belief Propagation (LBP) (MURPHY; WEISS; JORDAN, 1999) é o nome popular para denotar a propagação de mensagens em estruturas com ciclos. O algoritmo LBP fornece resultados aproximados de inferência em redes Bayesianas. Apesar da falta de garantias teóricas sobre a qualidade das aproximações, o algoritmo LBP tem produzido resultados excelentes (MCELIECE; MACKAY; CHENG, 1998) (YEDIDIA; FREEMAN; WEISS, 2000).

No algoritmo LBP, as mensagens são propagadas de forma *iterativa* em ciclos. A propagação em ciclos do algoritmo LBP é ilustrada no exemplo de rede Bayesiana multi-conectada com 4 nós (Figura 6). Considere que a variável D está observada, $\{D = d\}$. A expressão da crença $Bel(A)$ a respeito da variável A na $(i+1)$ -ésima iteração é dada por:

$$Bel^{(i+1)}(A) = \alpha p(A) \cdot \lambda_B^{(i)}(A) \cdot \lambda_C^{(i)}(A).$$

As demais crenças são dadas pelas seguintes expressões:

$$\begin{aligned} Bel^{(i+1)}(B) &= \alpha \sum_A p(B|A) \cdot \pi_B^{(i)}(A) \cdot \lambda_D^{(i)}(B), \\ Bel^{(i+1)}(C) &= \alpha \sum_A p(C|A) \cdot \pi_C^{(i)}(A) \cdot \lambda_D^{(i)}(C). \end{aligned}$$

Seguem-se as inicializações: para a variável A raiz, $\pi(A) = p(A)$, e para a

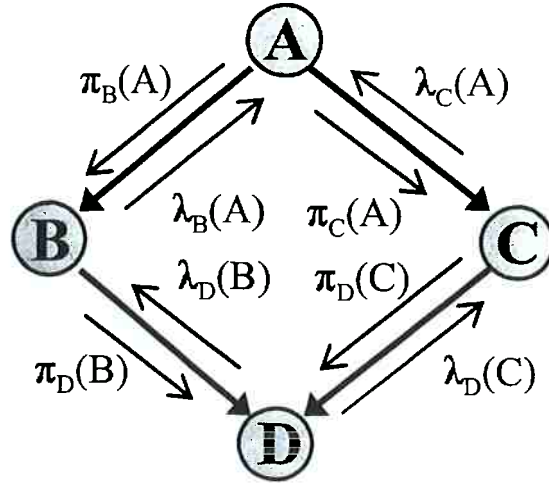


Figura 6: Algoritmo LBP em uma rede simples com 4 nós.

variável D observada, $\lambda(D = d) = 1$ e $\lambda(D \neq d) = 0$. Inicialmente, para $i = 0$, $Bel^{(1)}(A) = p(A)$, pois a evidência não foi ainda propagada; tomam-se $\lambda_B^{(0)}(A)$ e $\lambda_C^{(0)}(A)$ como vetores unitários. Para $i = 1$, mensagens são propagadas no sentido *forward* ($\pi_C(A) \rightarrow \pi_D(C)$ e $\pi_B(A) \rightarrow \pi_D(B)$) e *backward* ($\lambda_D(B) \rightarrow \lambda_B(A)$ e $\lambda_D(C) \rightarrow \lambda_C(A)$), obtendo-se um novo valor $Bel^{(2)}(A) = \alpha p(A) \cdot \lambda_B^{(1)}(A) \cdot \lambda_C^{(1)}(A)$. O procedimento é repetido iterativamente até se atingir a *convergência*⁵ da crença desejada. Note que a crença obtida, $Bel^{(i)}(A)$, não é necessariamente igual a $p(A|D = d)$, pois ela contabiliza duplamente a informação $p(D = d|A)$, através das mensagens $\lambda_B(A)$ e $\lambda_C(A)$.

Como descrito em (PEARL, 1988), quando a propagação de mensagens é realizada em uma estrutura com ciclos, as mensagens apresentadas na Seção 2.4.1 podem circular infinitamente através dos ciclos (*loops*) e o processo pode não convergir para um equilíbrio estável. Constata-se de modo empírico que em geral tais oscilações não ocorrem em redes Bayesianas - as mensagens convergem para um equilíbrio estável (MURPHY; WEISS; JORDAN, 1999). A crença $Bel(X)$ assim obtida acaba sendo uma aproximação de $p(X|\mathbf{E})$. Existem estudos (WEISS; FREEMAN, 1999) (IHLER; WILLSKY, 2005) analisando as condições de convergência do algoritmo LBP para alguns casos específicos. Mas, de forma geral, a convergência

⁵ Ao longo do documento, o termo *convergência* é empregado para significar a situação em que valores de probabilidade não variam entre uma iteração e outra.

e a qualidade das aproximações do algoritmo LBP são ainda questões em estudo.

2.5 Método de avaliação parcial local (algoritmo LPE)

O método de *avaliação parcial local* (*Localized Partial Evaluation - LPE*) é empregado para se gerar *intervalos* de uma desejada probabilidade da variável de consulta X_Q (DRAPER; HANKS, 1994). A idéia por trás do algoritmo LPE é transformar o problema de inferência na rede original em um problema de inferência em uma sub-rede denominada *active set* que contém parte da rede original. O algoritmo LPE pode ser aplicado tanto em poli-árvores como em redes multi-conectadas. Quando executado em poli-árvores, o algoritmo LPE reduz o custo computacional ⁶, pois ao invés de processar toda a rede, processa-se apenas parte dela. Quando aplicado em redes multi-conectadas, além da redução do custo computacional, o LPE reduz o problema a uma propagação de intervalos de probabilidade em poli-árvores.

O algoritmo LPE é baseado em técnicas tradicionais de propagação de mensagens, mas ao invés de propagar mensagens de valores pontuais (empregado pelo algoritmo BP), ele propaga *mensagens intervalares* (TESSEM, 1992), quando necessário. Por exemplo, ao invés de propagar o valor 0.25, ele propaga intervalos [0.2, 0.3].

O algoritmo LPE pode ser resumido nas seguintes etapas:

1. *Escolha do active set.* Não há regras fixas para a escolha do *active set*, mas de forma geral, ele é selecionado de modo a conter a variável de consulta X_Q e a simplificar o problema computacional -- o *active set* costuma ser uma *parte* da rede com estrutura poli-árvore.
2. *Inserção de mensagens vazias.* Toda "informação" fora do *active set* é aproximada através da inserção de *mensagens vazias* (*vacuous message*) compostas por intervalos [0, 1]. Os arcos que chegam e saem do *active set* são substituídos por arcos *ausentes* (*missing arcs*) que contém as mensagens

⁶Emprega-se o termo *custo computacional* para denominar o número de operações computacionais necessário para executar uma dada tarefa, e o espaço de memória necessário para realizar estas operações.

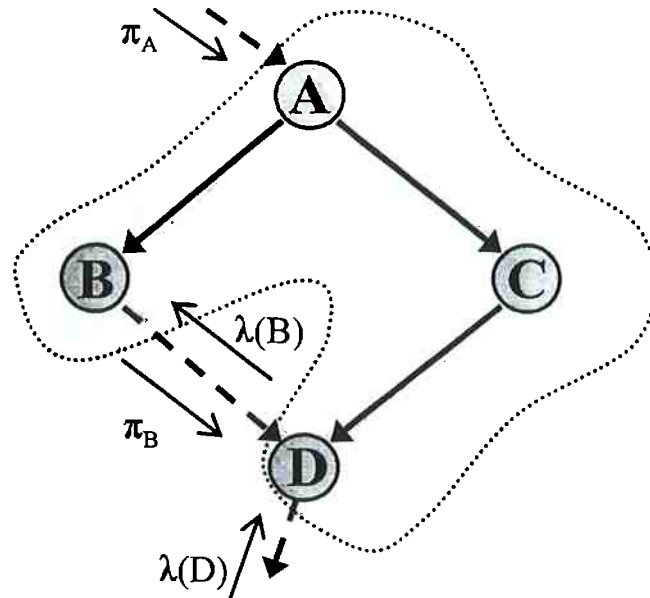


Figura 7: Ilustração do método LPE: escolha de um *active set* (área hachurada) e inserção de mensagens *vazias* (setas).

vazias. Por exemplo, na Figura 7, o *active set* escolhido é composto pelos nós: A, B, C e D e pelos arcos: (A, B) , (A, C) e (C, D) . Toda informação fora do *active set* é aproximada através da inserção de arcos ausentes com mensagens vazias $[0, 1]$ (no lugar dos arcos originais); isto é, as mensagens π_A , π_B , $\lambda(D)$ e $\lambda(B)$ são substituídas por intervalos $[0, 1]$. A idéia é, portanto, propagar mensagens, cujos valores variam entre 0 e 1.

3. *Propagação de intervalos de probabilidade.* O algoritmo LPE realiza propagação de intervalos através do algoritmo de *Aniquilação e Reforço (A/R)* (TESSEM, 1992) (ROCHA, 2004). O algoritmo *A/R* propaga de forma *aproximada* mensagens *intervalares*, ou seja, mensagens contendo valores inferiores e superiores: I_π e I_λ , no lugar das mensagens do algoritmo BP: π e λ . Propagando-se intervalos no *active set*, obtém-se ao final intervalos que necessariamente contém os valores exatos de probabilidade — a prova desta propriedade é apresentada em (DRAPER, 1995).

Para ilustrar o algoritmo LPE, considere o seguinte exemplo — uma rede Bayesiana com estrutura poli-árvore, exceto no ciclo formado pelos nós: A, B, C e D e os arcos: (A, B) , (A, C) , (B, C) e (C, D) (Figura 8). O *active set* selecionado é composto pela área hachurada mais o resto da rede. Seja Y o pai de A , e X

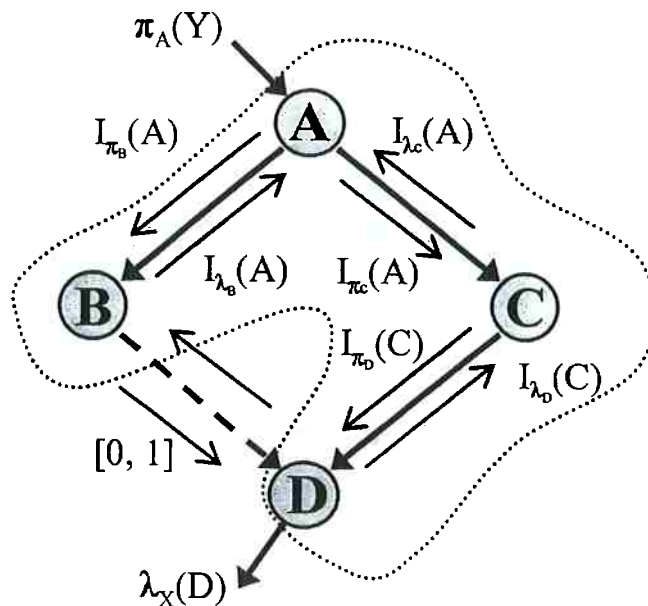


Figura 8: Representação das mensagens propagadas no algoritmo LPE. O *active set* é composto pela área hachurada mais o resto da rede. O arco pontilhado representa o arco ausente inserido no lugar do arco original.

o filho de D , as mensagens $\pi_A(Y)$ e $\lambda_X(D)$ são obtidas com a propagação do algoritmo BP, pois o resto da rede é poli-árvore. Suponha que se deseja calcular a probabilidade $P(C = c)$. Denote $P(C = c)$ por $P(c)$ para simplificar a notação. Vejamos como é que se obtém o intervalo $I_{Bel}(c)$ que contém $P(c)$ através do algoritmo LPE.

As mensagens propagadas pelo algoritmo LPE são ilustradas na Figura 8. As mensagens $\lambda_D(B)$ e $\pi_D(B)$ que chegam em B e D são substituídas por mensagens vazias: $I_{\lambda_D}(B) = I_{\pi_D}(B) = [0, 1]$. A mensagem $\lambda_B(A)$, que agora é composta por intervalos, é dada pela expressão:

$$I_{\lambda_B}(A) = \sum_B p(B|A) \cdot I_{\lambda_D}(B), \quad (2.8)$$

onde é necessário realizar a somatória de produtos $p(B|A) \cdot I_{\lambda_D}(B)$. Isto é feito através da operação A/R (TESSEM, 1992) entre a distribuição $p(B|A)$ e a distribuição composta de intervalos $[0, 1]$, $I_{\lambda_D}(B)$. As demais mensagens são dadas

pelas expressões:

$$I_{\pi_C}(A) = \pi(A) \cdot I_{\lambda_B}(A) = \sum_Y (p(A|Y) \cdot \pi_A(Y)) \cdot I_{\lambda_B}(A),$$

$$I_{\lambda_D}(c) = \sum_B \left(\sum_D (p(D|B, c) \cdot \lambda_X(D)) \cdot I_{\pi_D}(B) \right).$$

Note que as mensagens $\lambda_D(B)$ e $\pi_D(B)$ são substituídas por intervalos $[0, 1]$. Finalmente, obtém-se um intervalo para a crença $Bel(c)$, através da expressão:

$$I_{Bel}(c) = I_{\lambda_D}(c) \cdot \sum_A p(c|A) \cdot I_{\pi_C}(A)$$

$$= \sum_B \left(\left(\sum_{A,D,Y} p(A|Y) \cdot p(B|A) \cdot p(c|A) \cdot p(D|B, c) \cdot \pi_A(Y) \cdot \lambda_X(D) \right) \cdot I_{\pi_D}(B) \cdot I_{\lambda_D}(B) \right),$$

onde o intervalo $I_{Bel}(c)$ contém $P(c)$. A somatória \sum_B entre: a distribuição obtida por $\sum_{A,D,Y}$, e as distribuições $I_{\pi_D}(B)$ e $I_{\lambda_D}(B)$ formadas por intervalos, é realizada empregando a operação A/R .

2.6 Métodos variacionais em redes Bayesianas

Métodos variacionais são populares em diversas áreas, como em teoria de controle, otimização, estatística, economia e aprendizado. Recentemente, tais métodos vêm sendo utilizados com sucesso em modelos gráficos (JORDAN; GHAHRAMANI; JAAKKOLA, 1999). Várias das técnicas variacionais são empregadas para inferência e estimação estatística em modelos gráficos densamente conectados (SAUL; JAAKKOLA; JORDAN, 1996) (SAUL; JORDAN, 1996).

Uma breve revisão é feita nesta seção, baseando-se em materiais introdutórios ao assunto (JAAKKOLA, 2000) (WINN, 2003). Na Seção 2.6.1, descreve-se como a aproximação variacional é realizada. Na Seção 2.6.2, apresenta-se a aproximação variacional de campo médio. Na Seção 2.6.3, discute-se a vantagem computacional da aproximação de campo médio. Na Seção 2.6.4, apresenta-se a aproximação de campo médio estruturado. Na Seção 2.6.5, descreve-se um exemplo de aplicação da aproximação de campo médio estruturado em uma rede Bayesiana.

2.6.1 Aproximação variacional

Suponha um grafo direcionado associado a uma distribuição conjunta $p(\mathbf{X})$, onde \mathbf{X} representa todo o conjunto de variáveis. A idéia é aproximar uma distribuição *intratável* $p(\mathbf{H}|\mathbf{E})$ (obtida a partir de $p(\mathbf{X})$) por uma distribuição *tratável*⁷ $q(\mathbf{H})$, onde \mathbf{H} é o conjunto das variáveis não observadas e \mathbf{E} o das observadas — ou seja $\mathbf{X} = \{\mathbf{H}, \mathbf{E}\}$. Escolhe-se a entropia relativa *Kullback-Leibler* (WHITTAKER, 1990) como medida de “distância” entre as distribuições $p(\mathbf{H}|\mathbf{E})$ e $q(\mathbf{H})$:

$$\begin{aligned} KL(q||p) &= \sum_{\mathbf{H}} q(\mathbf{H}) \ln \frac{q(\mathbf{H})}{p(\mathbf{H}|\mathbf{E})} = \sum_{\mathbf{H}} q(\mathbf{H}) \ln \frac{q(\mathbf{H})}{p(\mathbf{H}, \mathbf{E})} + \ln p(\mathbf{E}) \\ &= \sum_{\mathbf{H}} q(\mathbf{H}) \ln q(\mathbf{H}) - \sum_{\mathbf{H}} q(\mathbf{H}) \ln p(\mathbf{H}, \mathbf{E}) + \ln p(\mathbf{E}). \end{aligned} \quad (2.9)$$

O primeiro termo da expressão é a entropia de Shannon (1948) da distribuição $q(\mathbf{H})$. O segundo termo negativo é a esperança de $\ln p(\mathbf{H}, \mathbf{E})$ em relação à distribuição $q(\mathbf{H})$. O último termo $\ln p(\mathbf{E})$ é uma constante em relação a $q(\mathbf{H})$. A entropia relativa $KL(q||p)$ é sempre positiva e ela será nula ($KL(q||p) = 0$) somente quando $q(\mathbf{H}) = p(\mathbf{H}|\mathbf{E})$. A meta é encontrar uma boa aproximação $q(\mathbf{H})$ da distribuição desejada $p(\mathbf{H}|\mathbf{E})$, minimizando a *distância* $KL(q||p)$. Deve-se ressaltar que a entropia relativa *Kullback-Leibler* não é uma verdadeira *medida de distância* no sentido que ela não é simétrica, ou seja, $KL(q||p) \neq KL(p||q)$ (WHITTAKER, 1990).

Na Figura 9, apresenta-se uma ilustração esquemática do método variacional: escolhe-se uma classe de família de distribuições *tratáveis*; e empregando a distância KL, tenta-se encontrar a distribuição $q(\mathbf{H})$ que mais se aproxima da distribuição desejada $p(\mathbf{H}|\mathbf{E})$.

2.6.2 Aproximação de campo médio

Um modelo aproximado utilizado com sucesso em métodos variacionais em diversas áreas é a distribuição *totalmente fatorada*. Esse método é freqüentemente chamado de aproximação de campo médio (*Mean Field*) em mecânica estatística (PARISI, 1988). A idéia em modelos probabilísticos é que a distribuição con-

⁷O termo distribuição *tratável* se refere a uma distribuição na qual inferências exatas podem ser realizadas com menor custo computacional.

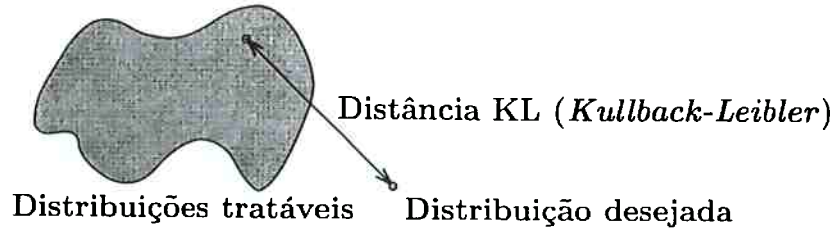


Figura 9: Ilustração esquemática de métodos variacionais. A entropia relativa KL (*Kullback-Leibler*) é utilizada para medir a “distância” entre a distribuição desejada e aproximada.

junta $p(\mathbf{H}|\mathbf{E})$ pode ser aproximada por uma distribuição $q(\mathbf{H})$ definida sobre um conjunto de variáveis independentes H_i (SAUL; JAAKKOLA; JORDAN, 1996). Utilizando a distribuição totalmente fatorada, pode-se minimizar a distância KL de forma iterativa e computacionalmente eficiente.

Considere uma distribuição totalmente fatorada:

$$q(\mathbf{H}) = \prod_i q_i(H_i).$$

Substituindo-a na Equação (2.9), obtém-se:

$$\begin{aligned} KL(q||p) &= \sum_{\mathbf{H}} \prod_i q_i(H_i) \ln \left(\prod_i q_i(H_i) \right) - \sum_{\mathbf{H}} \prod_i q_i(H_i) \ln p(\mathbf{H}, \mathbf{E}) + \ln p(\mathbf{E}) \\ &= \sum_i \sum_{H_i} q_i(H_i) \ln q(H_i) - \sum_{\mathbf{H}} \prod_i q_i(H_i) \ln p(\mathbf{H}, \mathbf{E}) + \ln p(\mathbf{E}) \\ &= -\sum_i \mathbb{H}(q_i) + \ln p(\mathbf{E}) - \sum_{\mathbf{H}} \prod_i q_i(H_i) \ln p(\mathbf{H}, \mathbf{E}), \end{aligned} \quad (2.10)$$

onde \mathbb{H} representa a entropia de q_i . A idéia é minimizar KL em relação a $q_j(h_j)$, onde h_j é um possível estado de H_j , assumindo como fixos os demais termos $q_i(H_i)$, $i \neq j$. Isto é feito reescrevendo a Equação (2.10) em termos de $q_j(H_j)$:

$$KL(q||p) = -\mathbb{H}(q_j) - \sum_{i \neq j} \mathbb{H}(q_i) - \sum_{H_j} q_j(H_j) \sum_{H_{i \neq j}} \prod_i q_i(H_i) \ln p(\mathbf{H}, \mathbf{E}) + \ln p(\mathbf{E}),$$

e diferenciando esta expressão em relação a $q_j(h_j)$. Note que a restrição de normalização $\sum_{h_j \in H_j} q(h_j) = 1$ deve ser levada em consideração. Esta operação é efetuada introduzindo-se um *multiplicador de Lagrange*⁸ λ_j e desenvolvendo a

⁸O multiplicador de Lagrange é empregado em problemas de otimização na presença de restrições. No caso, a restrição imposta é $\sum_{h_j \in H_j} q(h_j) - 1 = 0$, pois $q_j(H_j)$ é uma distribuição

expressão:

$$\frac{\partial}{\partial q_j(h_j)} [KL(q||p) - \lambda_j (\sum_{h_j \in H_j} q(h_j) - 1)] = 0. \quad (2.11)$$

Substituindo a expressão de $KL(q||p)$, anteriormente obtida, na Equação (2.11), obtém-se a expressão de $q_j^*(h_j)$ que parcialmente minimiza $KL(q||p)$:

$$\begin{aligned} (-\mathbb{H}(q_j))' - \left(\sum_{H_j} q_j(H_j) \sum_{H_{i \neq j}} \prod_i q_i(H_i) \ln p(\mathbf{H}, \mathbf{E}) \right)' - \left(\lambda_j \sum_{h_j \in H_j} q(h_j) \right)' &= 0, \\ q_j'(h_j) \ln q_j(h_j) + q_j(h_j) \frac{q_j'(h_j)}{q_j(h_j)} - q_j'(h_j) \sum_{H_{i \neq j}} \prod_i q_i(H_i) \ln p(\mathbf{H}, \mathbf{E}) - \lambda_j q_j'(h_j) &= 0, \\ q_j'(h_j) \left(\ln q_j(h_j) + 1 - \sum_{H_{i \neq j}} \prod_i q_i(H_i) \ln p(\mathbf{H}, \mathbf{E}) - \lambda_j \right) &= 0. \end{aligned}$$

Como $q_j'(h_j)$ é um valor finito, a equação anterior será satisfeita quando $q_j(h_j)$ assumir $q_j^*(h_j)$:

$$q_j^*(h_j) = k_{\lambda_j} \exp \left(\sum_{H_{i \neq j}} \prod_i q_i(H_i) \ln p(\mathbf{H}, \mathbf{E}) \right), \quad (2.12)$$

onde $k_{\lambda_j} = \exp(1 - \lambda_j)^{-1}$ é a constante de normalização. A constante k_{λ_j} é obtida através da expressão:

$$k_{\lambda_j} = \frac{1}{\sum_{h_j \in H_j} \exp \left(\sum_{H_{i \neq j}} \prod_i q_i(H_i) \ln p(\mathbf{H}, \mathbf{E}) \right)}.$$

A Equação (2.12) é solução única da Equação (2.11), e portanto, mínimo global de $KL(q||p)$ em relação a $q_j(h_j)$ e mínimo local em relação a $q(\mathbf{H})$. Significa que atualizando $q_j(h_j)$ de acordo com a Equação (2.12), para todos os estados h_j de H_j , o valor $KL(q||p)$ resultante necessariamente diminuirá. Uma vez que se atualiza a variável H_j , escolhe-se uma outra variável H_i para ser atualizada, e assim por diante, até percorrer todas as variáveis não observadas contidas em \mathbf{H} . Este é um processo iterativo que converge monotonicamente para um mínimo local da medida $KL(q||p)$, resultando em uma distribuição totalmente fatorada $q^*(\mathbf{H}) = \prod_i q_i^*(H_i)$ que aproxima $p(\mathbf{H}|\mathbf{E})$.

de probabilidade.

2.6.3 Importância da computação local

A Equação (2.12) anteriormente obtida pode ser simplificada no caso de redes Bayesianas. A distribuição conjunta $p(\mathbf{H}, \mathbf{E})$ passa a ser representada em termos de distribuições condicionais. Na Equação (2.12), substituindo $p(\mathbf{H}, \mathbf{E})$ por distribuições condicionais $p(X_k|pa_k)$, onde pa_k representa o conjunto dos nós pais de X_k , obtém-se:

$$\begin{aligned} q_j^*(h_j) &= k_\lambda \exp \left(\sum_{H_i \neq j} \prod_i q_i(H_i) \ln p(\mathbf{H}, \mathbf{E}) \right) \\ &= k_\lambda \exp \left(\sum_{H_i \neq j} \prod_i q_i(H_i) \sum_k \ln p(X_k|pa_k) \right). \end{aligned}$$

Toda distribuição condicional $p(X_k|pa_k)$ que não depende da variável H_j é anulada durante o cálculo da constante de normalização, e portanto, ela poder ser desconsiderada. Denote por ch_j o conjunto dos nós filhos de H_j — estes são os que dependem de H_j , e portanto, permanecem dentro da somatória \sum_k . A expressão anterior fica:

$$q_j^*(h_j) = k_\lambda \exp \left(\sum_{H_i \neq j} \prod_i q_i(H_i) [\ln p(h_j|pa_j) + \sum_{k \in ch_j} \ln p(X_k|pa_k)] \right).$$

Como todo termo $q_i(H_i)$ que não pertence a $\{pa_j\}$ e $\{X_k, pa_k\}$ pode ser retirado da somatória, a expressão anterior fica reduzida a:

$$\begin{aligned} q_j^*(h_j) &= k_\lambda \exp \left(\sum_{H_i \in \{pa_j\}} \prod_i q_i(H_i) \ln p(h_j|pa_j) \right. \\ &\quad \left. + \sum_{k \in ch_j} \sum_{H_i \neq j \in \{X_k, pa_k\}} \prod_i q_i(H_i) \ln p(X_k|pa_k) \right). \end{aligned}$$

Note que apenas computações “locais” são necessárias para atualizar a Equação (2.13), ou seja, apenas as distribuições condicionais das variáveis pertencentes ao *Markov blanket* (definido na Seção 2.1) de H_j são necessárias.

A Figura 10 ilustra a estratégia de procura do mínimo local através de computações locais. Inicialmente, parte-se de uma distribuição qualquer. Ao compu-

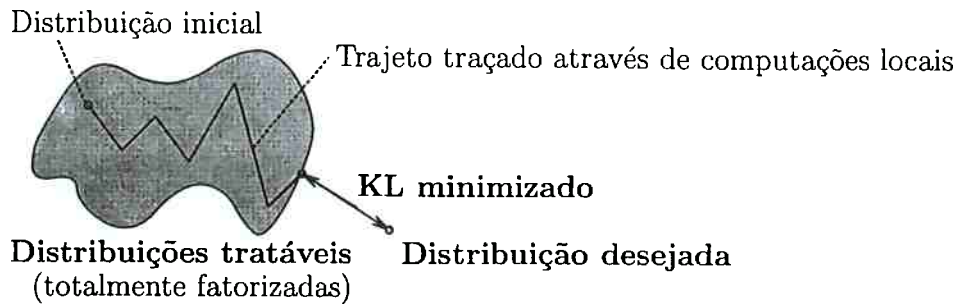


Figura 10: Ilustração de como a distância KL é minimizada, adotando-se distribuições aproximadas totalmente fatoradas, através de computações locais.

tar a Equação (2.13) para a variável H_j , encontra-se o mínimo local da medida KL, através de computação também local. Toma-se a seguir outra variável H_i e se realiza novamente uma minimização local. E assim por diante, até convergir para uma distribuição aproximada $q^*(\mathbf{H})$ de $p(\mathbf{H}, \mathbf{E})$. As variáveis são percorridas quantas vezes forem necessárias, durante este processo de convergência. Infelizmente, a medida $KL(q||p)$ não é convexa em relação a $q(\mathbf{H}) = \prod_i q_i(H_i)$, e portanto, a sua minimização é conduzida para mínimos locais, e não globais. A iteração converge para um mínimo local, em função dos valores de inicialização e de acordo com a ordem em que as variáveis são atualizadas (JAAKKOLA, 2000, página 14).

2.6.4 Aproximação de campo médio estruturado

A aproximação de campo médio anteriormente descrita é computacionalmente atrativa, mas em geral ela não conduz a resultados precisos, como discutido por Jaakkola (2000). Uma natural idéia de aperfeiçoamento da aproximação de campo médio é a escolha de uma distribuição aproximada $q(\mathbf{X}) = \prod_i q_i$ que leve em consideração algumas dependências entre as variáveis. Por exemplo, aproximando a distribuição original *intratável* por uma distribuição *tratável* com estrutura polívor. Esta estratégia é empregada pela aproximação de *campo médio estruturado* (*Structured Mean Field*) (SAUL; JORDAN, 1996) (JORDAN; GHARAMANI; JAAKKOLA, 1999) (WIEGERINCK, 2000) (JAAKKOLA, 2000) na qual cada fator q_i é definido sobre um *cluster* de variáveis $c_i = \{X_i, pa_i\}$, não mais sobre uma

única variável X_i , de modo a ter a seguinte distribuição conjunta aproximada:

$$q(\mathbf{X}) = \prod_i q_i(c_i).$$

A equação de atualização é modificada, pois na Equação (2.10), alguns termos de entropia não são mais constantes. Assim, a nova expressão de atualização para o *cluster* $c_j = \{X_j, pa_j\}$ é (WINN, 2003, página 104):

$$q_j^*(x_j|pa_j) = k_\lambda \exp \left(\sum_{k \in G_{X_j}} \sum_{l \in \{\mathbf{X} \setminus c_j\}} \prod_l q_l(X_l) \ln p(X_k|pa_k) - \sum_{i \in C_{X_j}} \sum_{l \in \{\mathbf{X} \setminus c_j\}} \prod_l q_l(X_l) \ln q_i(X_i|pa_i) \right), \quad (2.13)$$

onde x_j é um possível estado de X_j , $G_{X_j} = \{g_k\}$ representa o conjunto de *clusters* $g_k = \{X_k, pa_k\}$ que dependem de X_j na rede original (ou seja, G_{X_j} é o *Markov blanket* de X_j) e $C_{X_j} = \{c_i\}$ é o conjunto de *clusters* $c_i = \{X_i, pa_i\}$ que dependem de X_j na rede aproximada, excluindo c_j .

Os passos para emprego da aproximação de campo médio estruturado podem ser resumidos em:

1. Escolha de uma estrutura aproximada que seja *tratável* (isto é, estrutura na qual inferências exatas podem ser realizadas em tempo polinomial) e, ao mesmo tempo, consiga modelar as dependências da estrutura original de grafo.
2. Cálculo das distribuições da estrutura aproximada. Através do método variacional, as distribuições condicionais que melhor aproximam a distribuição original são computadas para cada uma das distribuições condicionais modificadas, através da Equação (2.13).
3. Repetição da etapa anterior até convergência dos valores das distribuições.
4. Realização de inferências desejadas na rede Bayesiana aproximada obtida.

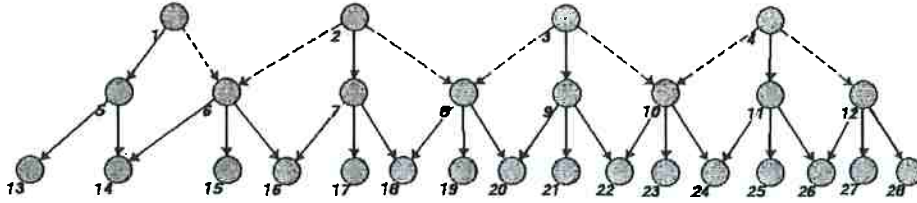


Figura 11: Exemplo de aplicação da aproximação de campo médio na rede Pyramid. Os arcos tracejados são os arcos removidos da rede original para obter uma poli-árvore.

2.6.5 Exemplo: aproximação de campo médio estruturado na rede Pyramid

Considere o seguinte exemplo para ilustrar o método de aproximação de campo médio estruturado. Seja a rede Bayesiana *Pyramid* (MURPHY; WEISS; JORDAN, 1999), ilustrada na Figura 11, com 28 nós, onde $\mathbf{X} = \{X_1, \dots, X_{28}\}$ são variáveis binárias (assumem estados x_i e $\neg x_i$) e $p(\mathbf{X}) = \prod_{i=1}^{28} p_i(X_i|pa_i)$. Uma possível estrutura de aproximação *tratável* é a poli-árvore obtida a partir da rede original removendo os arcos (1,6), (2,6), (2,8), (3,8), (3,10), (4,10) e (4,12). Com estas remoções, obtém-se uma nova distribuição conjunta aproximada $q(\mathbf{X}) = \sum_{i=1}^{28} q_i(X_i|pa_i)$ que difere nas distribuições condicionais das variáveis X_6 , X_8 , X_{10} e X_{12} . Considere a atualização da variável X_6 : temos $G_{X_6} = \{g_6, g_{14}, g_{15}, g_{16}\}$, $C_{X_j} = \{c_{14}, c_{15}, c_{16}\}$ e $c_6 = \{X_6\}$, onde cada um dos c_i e g_k são definidos conforme apresentado na Seção 2.6.4. Logo, temos a seguinte expressão de atualização da distribuição aproximada $q_6(x_6)$, da Equação (2.13):

$$\begin{aligned} q_6^*(x_6) &= k_{\lambda_6} \exp \left(\sum_{k \in G_{X_6}} \sum_{l \in \{\mathbf{X} \setminus X_6\}} \prod_l q_l(X_l) \ln p_k(X_k|pa_k) \right. \\ &\quad \left. - \sum_{i \in C_{X_6}} \sum_{l \in \{\mathbf{X} \setminus X_6\}} \prod_l q_l(X_l) \ln q_i(X_i|pa_i) \right), \\ &= k_{\lambda_6} \exp \left(\sum_{l \in \{\mathbf{X} \setminus X_6\}} \prod_l q_l(X_l) (\ln p_6(x_6|pa_6) + \ln p_{14}(X_{14}|pa_{14}) \right. \\ &\quad \left. + \ln p_{15}(X_{15}|pa_{15}) + \ln p_{16}(X_{16}|pa_{16}) - \ln q_{14}(X_{14}|pa_{14}) \right. \\ &\quad \left. - \ln q_{15}(X_{15}|pa_{15}) - \ln q_{16}(X_{16}|pa_{16}) \right). \end{aligned}$$

Considerando que na distribuição aproximada $q(\mathbf{X}) = \sum_{i=1}^{28} q_i(X_i|pa_i)$ nem todas as distribuições q_i são modificadas, e portanto, para as variáveis X_j com distri-

buições não modificadas, temos $q_j(X_j|pa_j) = p_j(X_j|pa_j)$; e fazendo simplificações como as da Equação (2.13), a expressão anterior fica:

$$\begin{aligned} q_6^*(x_6) &= k_{\lambda_6} \exp \left(\sum_{l \in \{X \setminus X_6\}} \prod_l q_l(X_l) \ln p_6(x_6|X_1, X_2) \right) \\ &= k_{\lambda_6} \exp \left(\sum_{X_1, X_2} q_1(X_1) \cdot q_2(X_2) \ln p_6(x_6|X_1, X_2) \right) \\ &= k_{\lambda_6} \exp \left(\sum_{X_1, X_2} p_1(X_1) \cdot p_2(X_2) \ln p_6(x_6|X_1, X_2) \right), \end{aligned}$$

onde a constante de normalização é dada por:

$$k_{\lambda_6} = \frac{1}{\exp(\sum_{X_1, X_2} p_1(X_1) \cdot p_2(X_2) \ln p_6(x_6|X_1, X_2)) + \exp(\sum_{X_1, X_2} p_1(X_1) \cdot p_2(X_2) \ln p_6(-x_6|X_1, X_2))}.$$

De forma análoga, as expressões para as demais distribuições aproximadas são:

$$\begin{aligned} q_8^*(x_8) &= k_{\lambda_8} \exp \left(\sum_{X_2, X_3} p_2(X_2) \cdot p_3(X_3) \ln p_8(x_8|X_2, X_3) \right), \\ q_{10}^*(x_{10}) &= k_{\lambda_{10}} \exp \left(\sum_{X_3, X_4} p_3(X_3) \cdot p_4(X_4) \ln p_{10}(x_{10}|X_3, X_4) \right), \\ q_{12}^*(x_{12}) &= k_{\lambda_{12}} \exp \left(\sum_{X_4} p_4(X_4) \ln p_{12}(x_{12}|X_4) \right). \end{aligned}$$

Neste exemplo apresentado, como as distribuições aproximadas $q_6(X_6)$, $q_8(X_8)$, $q_{10}(X_{10})$ e $q_{12}(X_{12})$ são independentes entre si, não há necessidade de realizar uma computação iterativa. Isto significa que as distribuições aproximadas são obtidas na primeira iteração.

3 REDES CREDAIS

Uma das dificuldades na construção de redes Bayesianas em aplicações reais é a obtenção de valores para todas as probabilidades das distribuições condicionais $p(X_i|pa_i)$. Por isso, é interessante desenvolver modelos que permitam representar eventuais imprecisões nos valores de probabilidades. Uma maneira bastante comum para representar imprecisão em probabilidade é o uso de *intervalos* de probabilidade, ou, de forma mais geral, de *conjuntos* de probabilidade (WALLEY, 1991). Intervalos e conjuntos de probabilidade podem se originar da falta de conhecimentos sobre determinado aspecto do sistema modelado, ou da falta de dados experimentais, ou do desejo de investigar a robustez da rede a variações de parâmetros (BERGER, 1985). Estas representações têm sido objeto de intenso estudo nas últimas décadas, tendo várias denominações: teoria de *probabilidades imprecisas* (WALLEY, 1991); teoria Bayesiana *conveza* ou teoria de *conjuntos credais* (LEVI, 1980); teoria *Quasi-Bayesiana* (GIRON; RIOS, 1980). A teoria de conjuntos credais tem encontrado aplicações em diversas áreas, como engenharia e física (FINE, 1988) e administração (LUO *et al.*, 1996).

As *redes credais* são modelos probabilísticos que estendem os conceitos aplicados a redes Bayesianas para conjuntos credais (TESSEM, 1992) (CANO; DELGADO; MORAL, 1993) (FAGIUOLI; ZAFFALON, 1998) (COZMAN, 2000).

Este capítulo apresenta os principais conceitos sobre conjuntos e redes credais. Uma breve revisão sobre conjuntos credais é apresentada na Seção 3.1. As redes credais e seus métodos de inferências são descritos nas Seções 3.2 e 3.3, respectivamente. O problema de inferência em redes credais binárias é pormenorizado na Seção 3.4. Na Seção 3.5, o algoritmo exato de propagação de intervalos (2U) é descrito em detalhes. Este algoritmo é peça fundamental dos algoritmos novos desenvolvidos neste trabalho.

3.1 Conjuntos credais

O intuito desta seção e da Seção 3.2 é apresentar somente os aspectos essenciais da teoria de conjuntos credais. Uma explicação mais extensa é encontrada nas referências (COZMAN, 2000) (ROCHA, 2004).

A teoria de conjuntos credais propõe um formalismo que permite representar probabilidades imprecisas ou parcialmente especificadas (LEVI, 1980) (GIRON; RIOS, 1980) (WALLEY, 1996). Um conjunto credal é um *conjunto de distribuições de probabilidade* que, supostamente, contém a distribuição correta das variáveis aleatórias de interesse.

Um conjunto credal é definido como se segue. Seja $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$ um conjunto de variáveis aleatórias discretas. O símbolo $K(\mathbf{X})$ denota um *conjunto credal* definido sobre \mathbf{X} , ou seja, $K(\mathbf{X})$ representa um conjunto cujos elementos são distribuições de probabilidade definidas sobre as variáveis em \mathbf{X} .

Se $K(\mathbf{X})$ definir uma *região poliedral convexa fechada (politopo)*, então não é necessário especificar todos os seus elementos para representá-lo. Basta aplicar a *operação de casco convexo*¹ sobre os vértices do conjunto (EDELSBRUNNER, 1987). Denote por \mathbf{V}_K o conjunto dos vértices de um conjunto credal K e seja $cc(\cdot)$ a notação para a operação de casco convexo — se $K(\mathbf{X})$ é um politopo cujos vértices são dados pelo conjunto $\mathbf{V}_{K(\mathbf{X})} = \{p_1(\mathbf{X}), p_2(\mathbf{X}), \dots, p_k(\mathbf{X})\}$ então, $K(\mathbf{X})$ é igual a $cc(\mathbf{V}_{K(\mathbf{X})})$.

Seguem-se dois exemplos de conjuntos credais para elucidar as definições apresentadas.

Exemplo 4 (Cfr. (ROCHA, 2004)). Seja X_1 uma variável binária assumindo estados “true” ou “false”. Então $K(X_1) = cc(\mathbf{V}_{K(X_1)})$, com $\mathbf{V}_{K(X_1)} = \{p_1(X_1), p_2(X_1)\}$, $p_1(X_1) = (0.5; 0.5)$ e $p_2(X_1) = (0.3; 0.7)$, denota um conjunto credal sobre X_1 (Figura 12).

Exemplo 5 Seja X_1 uma variável ternária assumindo estados “baixo”, “médio” ou “alto”. Temos que $K(X_1) = cc(\mathbf{V}_{K(X_1)})$, com $\mathbf{V}_{K(X_1)} = \{p_1(X_1), p_2(X_1), p_3(X_1)\}$.

¹O casco convexo de um conjunto de pontos no \mathfrak{R}^n é o menor conjunto convexo que contém estes pontos. Este conjunto também é chamado de *envoltória convexa* (CAMPELLO; MACULAN, 1994).

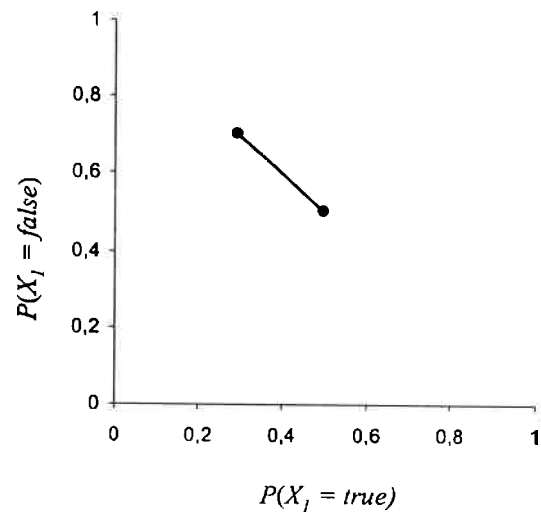


Figura 12: Representação gráfica do conjunto credal do Exemplo 4.

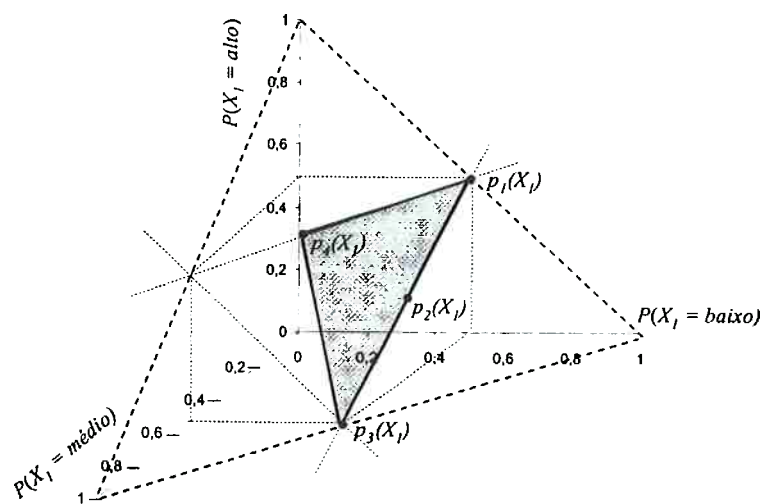


Figura 13: Representação gráfica do conjunto credal do Exemplo 5.

$p_3(X_1), p_4(X_1)\}$, onde $p_1(X_1) = (0.5; 0; 0.5)$, $p_2(X_1) = (0.5; 0.25; 0.25)$, $p_3(X_1) = (0.5; 0.5; 0)$ e $p_4(X_1) = (0.25; 0.25; 0.5)$. As distribuições $p_1(X_1)$, $p_3(X_1)$ e $p_4(X_1)$ formam um politopo que define o conjunto credal sobre X_1 (Figura 13). Note que $p_2(X_1)$ é um vértice desnecessário para representar o politopo.

Seja x_i um estado da variável X_i . Um *conjunto credal condicional*, $K(X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n | X_i = x_i)$, também denotado como $K(\mathbf{X} \setminus \{X_i\} | X_i = x_i)$, é composto por distribuições condicionais da forma $p(X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n | X_i = x_i)$ (GIRON; RIOS, 1980). O conjunto $K(\mathbf{X} \setminus \{X_i\} | X_i = x_i)$ pode ser determinado a partir de $K(\mathbf{X})$, calculando-se a distribuição condicional $p(X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n | X_i = x_i)$ em cada vértice de $K(\mathbf{X})$ e em seguida tomando o casco convexo destes pontos.

Sejam $x_{i,1}, \dots, x_{i,s_i}$ os estados assumidos pela variável X_i , onde s_i é o número de estados de X_i . O conjunto $\mathbf{K}(\mathbf{X} \setminus \{X_i\} | X_i) = \{K(X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n | X_i = x_{i,j}) : j = 1 \dots s_i\}$ representa uma *coleção de conjuntos credais especificados em separado*. Os conjuntos credais que compõem esta coleção são obtidos com o condicionamento em separado das variáveis contidas em $\mathbf{X} \setminus \{X_i\}$ pela variável X_i . Isto é, a coleção $\mathbf{K}(\mathbf{X} \setminus \{X_i\} | X_i)$ tem um conjunto credal para cada estado da variável condicionante. Por exemplo, o conjunto credal $K(X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n | X_i = x_{i,1})$ corresponde à variável X_i assumindo o estado $x_{i,1}$.

Por fim, apresenta-se a definição de *independência forte*. Sejam \mathbf{X} , \mathbf{Y} e \mathbf{Z} três conjuntos disjuntos cujos elementos são variáveis aleatórias; \mathbf{X} e \mathbf{Y} são *fortemente independentes* dado \mathbf{Z} se cada vértice do conjunto credal de distribuições conjuntas definido sobre \mathbf{X} , \mathbf{Y} e \mathbf{Z} satisfaz a equação de independência condicional:

$$P(X = x | Y = y, Z = z) = P(X = x | Z = z),$$

para todas os estados x , y e z de X , Y e Z , respectivamente.

3.2 Redes credais com extensão forte

Neste trabalho, consideram-se *redes credais* associadas a *conjuntos credais especificados em separado* com *independência forte*:

Definição 1 *Uma rede credal com conjuntos credais especificados em separado é um par $C = (G, K)$ onde:*

1. $G = (X, A)$ é um grafo direcionado acíclico, tal que:
 - (a) cada nó $X_i \in X$ simboliza uma variável aleatória;
 - (b) $A \subseteq X \times X$ é o conjunto de arcos de G ;
2. K é um conjunto cujos elementos são coleções de conjuntos credais especificados em separado, tal que para cada nó $X_i \in X$ existe uma e somente uma coleção $K(X_i|pa_i)$ em K .

Definição 2 *Uma rede credal especificada em separado implementa relações de independência forte se, cada nó $X_i \in X$ é fortemente independente dos demais nós da rede, exceto seus descendentes, quando o estado de seus nós pais é conhecido.*

Uma vez especificada a rede credal, conjuntos credais são formados a partir dela, denominados *extensões* da rede. A *extensão forte* é o maior conjunto credal de distribuições conjuntas que pode ser associado a uma rede credal com independência forte:

Definição 3 *A extensão forte $K(X_1, \dots, X_n)$ de uma rede credal C é dada pela expressão:*

$$K(X_1, \dots, X_n) = cc(\prod_i p(X_i|pa_i)),$$

$$\begin{aligned} \text{onde} \quad & p(X_i|pa_i) \in K(X_i|pa_i) \\ & K(X_i|pa_i) \in \mathbf{K}(X_i|pa_i) \\ & \mathbf{K}(X_i|pa_i) \in \mathbf{K} \\ & \forall X_i \in X. \end{aligned} \tag{3.1}$$

Assumindo que as variáveis são binárias (estados “true” ou “false”), considere os seguintes exemplos para ilustrar as definições anteriores:

Exemplo 6 (Cfr. (ROCHA, 2004)). A Figura 14 mostra uma rede credal com duas variáveis binárias e conjuntos credais especificados em separado. A Tabela 1 mostra os vértices da extensão forte desta rede.

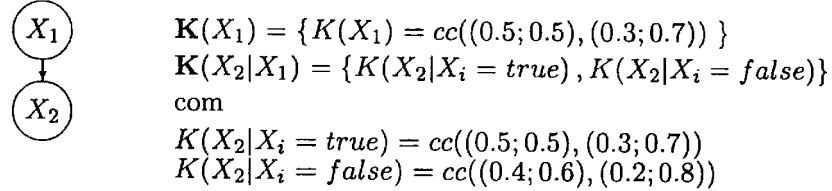


Figura 14: Um exemplo de rede credal com 2 variáveis binárias, associadas a conjuntos credais especificados em separado.

Tabela 1: Vértices da extensão forte da rede credal da Figura 14.

Vértice	$p(X_1 X_2)$	Vértice	$p(X_1 X_2)$
p_1	0.25;0.25;0.20;0.30	p_2	0.25;0.25;0.10;0.40
p_3	0.15;0.35;0.10;0.40	p_4	0.15;0.35;0.20;0.30
p_5	0.15;0.15;0.28;0.42	p_6	0.15;0.15;0.14;0.56
p_7	0.09;0.21;0.28;0.42	p_8	0.09;0.21;0.14;0.56

Exemplo 7 Tomando o grafo da rede Bayesiana *Dog-Problem* apresentado na Figura 3 e associando cada um dos nós a conjuntos credais especificados em separado, obtém-se a rede credal representada na Figura 15. Para redes credais com variáveis binárias, como neste caso, a operação de caso convexo é desnecessário, bastando atribuir intervalos de probabilidades para especificar cada conjunto credal (maiores detalhes são apresentados na Seção 3.4).

Cada vértice, gerado através da extensão forte a partir da rede credal especificada em separado, corresponde a uma rede Bayesiana. No Exemplo 6, o vértice $p_6 = (0.15; 0.15; 0.14; 0.56)$ da Tabela 1 corresponde a uma rede Bayesiana com variáveis X_1 e X_2 , onde $p(X_1) = (0.3; 0.7)$, $p(X_1|X_2 = true) = (0.5; 0.5)$ e $p(X_1|X_2 = false) = (0.2; 0.8)$. No Exemplo 7, como há 10 conjuntos credais especificados em separado (Figura 15), é possível gerar 2^{10} vértices através da extensão forte. Cada vértice corresponde a uma rede Bayesiana.

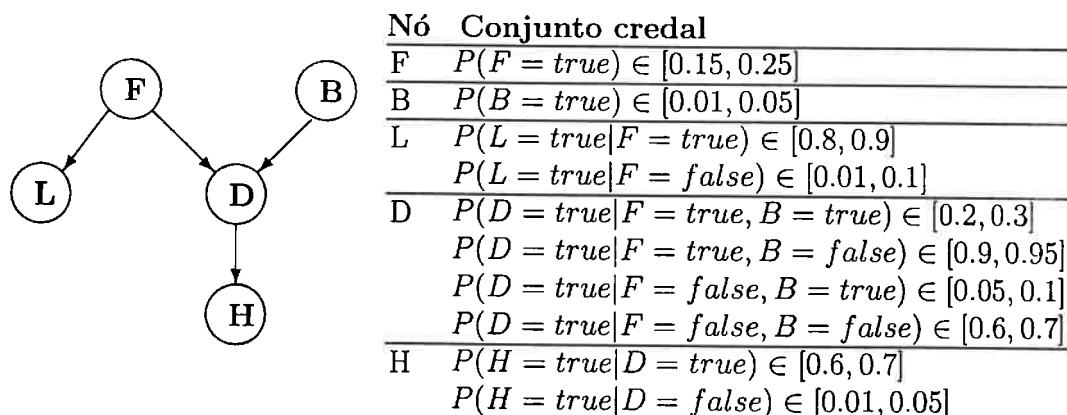


Figura 15: Exemplo de uma rede credal: rede *Dog-Problem* associada a variáveis binárias com os respectivos conjuntos credais locais.

3.3 Inferência em redes credais

Nesta seção, apresenta-se o problema de inferência em uma rede credal, que fundamentalmente consiste no cálculo de um intervalo para a probabilidade de um evento. Para estruturas mais simples de grafo como *poli-árvores*, prova-se que o problema de inferência exata é *NP-completo* (ROCHA; COZMAN, 2002). Para redes credais multi-conectadas, o problema é NP-difícil (COZMAN *et al.*, 2004).

Definição 4 Considerando uma rede credal $C = (G, K)$ com conjuntos credais especificados separadamente com extensão forte, uma inferência (COZMAN, 2000) (FAGIUOLI; ZAFFALON, 1998) consiste no cálculo das probabilidades inferiores e superiores de um evento $\{X_q = x_q\}$ dada a evidência \mathbf{E} , onde \mathbf{E} indica o conjunto de eventos $\{X_e = x_e\}$.

As variáveis que não pertencem a \mathbf{E} são chamadas variáveis *ocultas* (*hidden variables*) e neste trabalho elas são representadas por $\mathbf{H} = \{\mathbf{X} \setminus \mathbf{E}\}$. Defina o intervalo de probabilidade $I(X_q = x_q|\mathbf{E}) = [\underline{P}(X_q = x_q|\mathbf{E}), \overline{P}(X_q = x_q|\mathbf{E})]$, onde a probabilidade inferior deste intervalo é obtida resolvendo o seguinte problema de minimização:

$$\underline{P}(X_q = x_q | \mathbf{E}) = \min \frac{\sum_{(\mathbf{H} \setminus X_q), X_q = x_q} \prod_i p(X_i | pa_i)}{\sum_{\mathbf{H}} \prod_i p(X_i | pa_i)},$$

onde : $p(X_i | pa_i) \in K(X_i | pa_i),$
 $\forall X_i \in \mathbf{X}.$

(3.2)

A probabilidade superior é obtida resolvendo o seguinte problema de maximização:

$$\overline{P}(X_q = x_q | \mathbf{E}) = \max \frac{\sum_{(\mathbf{H} \setminus X_q), X_q = x_q} \prod_i p(X_i | pa_i)}{\sum_{\mathbf{H}} \prod_i p(X_i | pa_i)},$$

onde : $p(X_i | pa_i) \in K(X_i | pa_i),$
 $\forall X_i \in \mathbf{X}.$

(3.3)

Retomando o problema da Figura 15, se a variável H for observada como verdadeira, é possível inferir que o intervalo de probabilidade da família estar em casa $\{F = true\}$: $P(F = true | H = true) \in [0.18, 0.35]$.

O cálculo dos valores exatos dos extremos do intervalo $I(X_q = x_q | \mathbf{E})$ é denominado *inferência de intervalos exatos*, ou simplesmente *inferência exata*. Já a *inferência de intervalos aproximados* gera um intervalo que contém (*intervalo externo*) ou está contido (*intervalo interno*) no intervalo exato. Os três tipos de intervalos são ilustrados na Figura 16.

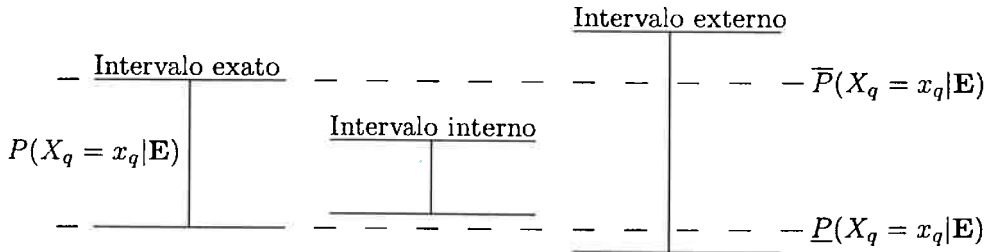


Figura 16: Intervalos exatos, aproximações internas e aproximações externas para a probabilidade de um evento.

No momento, os algoritmos mais eficientes para computação de intervalos exatos em redes credais genéricas são os algoritmos baseados em *programação multi-linear* (ROCHA; COZMAN; CAMPOS, 2003) (CAMPOS; COZMAN, 2004) (CAMPOS, 2005). Mas, mesmo tais algoritmos são incapazes de realizar inferências exatas em

redes de *porte médio* (redes com 30 nós com razoável número de conexões). Neste panorama, os algoritmos aproximados parecem ser os únicos caminhos viáveis para realização de inferência em redes credais bem conectadas.

Existem alguns métodos para inferência aproximada em redes credais (CANO; MORAL, 1996) (TESSEM, 1992) (CANO; MORAL, 1999), mas mesmo estes são incapazes de realizar inferências em tempo polinomial. Existe porém uma exceção para redes credais com estrutura *poli-árvore associadas a variáveis binárias*. Para estas redes credais, o *algoritmo 2U* (FAGIUOLI; ZAFFALON, 1998) produz intervalos exatos em tempo polinomial. Este algoritmo é descrito em detalhes na Seção 3.5, pois é a base dos algoritmos desenvolvidos e implementados nesta tese.

3.4 Inferência em redes credais binárias

Considere uma rede credal \mathbf{C} com conjuntos credais especificados em separado (Definição 1). Seja $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$ o conjunto de variáveis aleatórias binárias associado à rede \mathbf{C} . Cada variável $X_i \in \mathbf{X}$ está associada a uma coleção de conjuntos credais $\mathbf{K}(X_i|pa_i) = \{K(X_i|pa_i = pa_{i,j})\}$, onde $pa_{i,j}$ representa um possível estado assumido pelos pais pa_i de X_i e $j = \{1, \dots, s\}$ para s possíveis combinações de estados dos pais. Por exemplo, se a variável X_i possuir três pais, então $s = 2^3 = 8$, pois as variáveis são binárias – e significa que a variável será associada a 8 diferentes conjuntos credais, um para cada diferente estado assumido pelos pais.

Para especificar um conjunto credal $K(X_i|pa_i = pa_{i,j})$ associado a uma variável binária basta atribuir dois vértices, pois o politopo associado é simplesmente um intervalo de linha reta, como ilustrado no Exemplo 4 da Figura 12. Portanto, ao invés de representar o conjunto credal através da operação de casco convexo $cc(\cdot)$ sobre vértices, como no Exemplo 6 da Figura 14, o conjunto credal de uma variável binária pode ser representado simplesmente por um intervalo de probabilidade $[\underline{P}, \overline{P}]$, como no Exemplo 7 da Figura 15. Denote por x_i e $\neg x_i$ os estados assumidos pela variável X_i . Como $P(X_i = \neg x_i) = 1 - P(X_i = x_i)$, para representar o conjunto credal $K(X_i|pa_i)$, basta especificar $K(X_i = x_i|pa_i) = [\underline{P}, \overline{P}]$, pois $K(X_i = \neg x_i|pa_i = pa_{i,j}) = [1 - \overline{P}, 1 - \underline{P}]$.

Assumindo que a rede credal \mathbf{C} possui relações de independência forte (De-

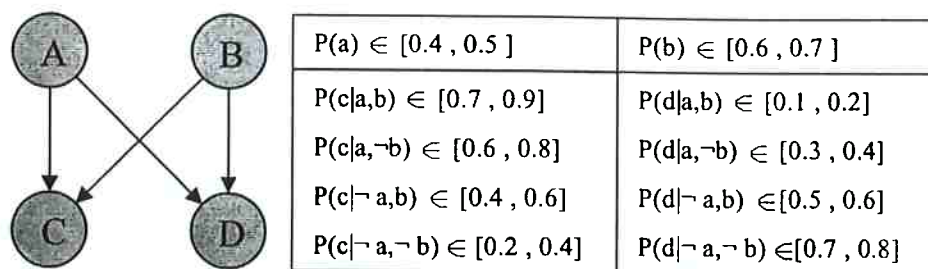


Figura 17: Exemplo de rede credal com conjuntos credais especificados em separado. As variáveis são binárias.

finição 2), a partir de seus conjuntos credais $K(x_i|pa_{i,j})$ especificados separadamente, é possível definir a sua extensão forte $K(X_1, \dots, X_n)$ (Definição 3). O número de *potenciais vértices*² n_V do conjunto credal obtido a partir da extensão forte de **C** pode ser quantificado da seguinte forma. Denote por n_{pa_i} o número de pais da variável X_i . Denote por n_K o número de conjuntos credais, no caso, de intervalos de probabilidade necessários para especificar a rede credal. Como todas as variáveis são binárias, temos que $n_K = \sum_{i=1}^n 2^{n_{pa_i}}$. Como cada intervalo é composto por dois valores extremos, logo o número total de potenciais vértices será dado pela expressão:

$$n_V = 2^{n_K} = 2^{\sum_{i=1}^n 2^{n_{pa_i}}} \quad (3.4)$$

Exemplo 8 . A rede credal representada na Figura 17 possui 4 variáveis binárias. Cada variável está associada a um conjunto credal representada na forma de intervalos de probabilidade. O número de conjuntos credais é dado por $n_k = 2^0 + 2^0 + 2^2 + 2^2 = 10$. Portanto, a partir destes 10 conjuntos credais especificados separadamente é possível gerar $n_V = 2^{10} = 1024$ diferentes vértices, através da extensão forte. Ou seja, esta simples rede credal apresentada representa um conjunto de 1024 diferentes redes Bayesianas.

Um método exaustivo para computar o intervalo $I(X_q = x_q|\mathbf{E})$ consiste no cálculo da probabilidade $P(X_q = x_q|\mathbf{E})$ para cada um dos vértices (cada vértice é uma rede Bayesiana) do conjunto credal gerado pela extensão forte e obter seus valores mínimo e máximo. No Exemplo 6, um procedimento para se computar $I(A = a|D = d)$ seria calcular $P(A = a|D = d)$ em cada uma das 1024 diferentes

²Nem toda extensão de **C** é necessariamente um vértice, pois ela pode estar na região interna do politopo formado, como ilustrada no Exemplo 5 da Figura 13.

redes Bayesianas geradas, mantendo-se os seus valores extremos. Este método de inferência é denominado *método de enumeração exaustiva* (ROCHA, 2004). Este método é demasiadamente ineficiente, tornando-se inviável para redes credais de porte médio (cerca de 30 nós) e bem conectadas. Por exemplo, no grafo da Figura 11, associando conjuntos credais para cada uma das variáveis, teremos um total de 2^{70} vértices³.

3.5 Algoritmo 2U: propagação de intervalos exatos para poli-árvores binárias

Tessem (1992) aplicou e adaptou a propagação de mensagens de Pearl (descrito na Seção 2.4) em redes credais, denominando o processo de *propagação de intervalos*. Um algoritmo exato de propagação de intervalos denominado 2U foi proposto por Fagioli e Zaffalon (1998) para redes credais com estrutura poli-árvore e associadas a variáveis binárias, e ele é capaz de realizar inferência em tempo polinomial⁴. O algoritmo 2U é peça fundamental dos algoritmos propostos neste trabalho.

3.5.1 Propagação local de intervalos

A propagação local de intervalos do algoritmo 2U ocorre de modo semelhante ao algoritmo BP. Como visto na Seção 2.4, no algoritmo BP, para cada nó X calcula-se o valor da crença $Bel(X) = p(X|\mathbf{E}) = \alpha\pi(X)\lambda(X)$, combinando mensagens provenientes dos filhos Y_j de X (mensagens $\lambda_{Y_j}(X)$) e pais U_i de X (mensagens $\pi_X(U_i)$), onde \mathbf{E} é o conjunto de variáveis observadas, e α é uma constante de normalização.

No algoritmo BP, onde as mensagens são pontuais, as constantes α e β só precisam ser calculadas no momento em que $p(X|\mathbf{E}) = \frac{p(X, \mathbf{E})}{p(\mathbf{E})}$ é computado. Já para conjuntos credais, como $p(X, \mathbf{E})$ e $p(\mathbf{E})$ são dependentes, seus valores máximos e mínimos precisam ser computados separadamente. Razões detalhadas desta primordial alteração são dadas pelos autores do algoritmo (FAGIUOLI; ZAF-

³Este valor é obtido através da Equação (3.4), contando o número de nós raízes, de nós com 1 pai e de nós com 2 pais: $n_V = 2^{2^0 \times 4 + 2^1 \times 15 + 2^2 \times 9} = 2^{70}$.

⁴O algoritmo 2U foi implementado em arquitetura análoga ao de propagação de mensagens de Pearl, e disponibilizado no software *2UBayes* (ver Apêndice I) desenvolvido neste trabalho.

FALON, 1998). Para estender o algoritmo BP para intervalos de probabilidade é necessário absorver as constantes α (Equação (2.4)) e β (Equação (2.6)) durante a propagação de mensagens, para que a computação das probabilidades $p(X|\mathbf{E})$ e $\bar{p}(X|\mathbf{E})$ continue sendo função de mensagens independentes. Assim, novas definições e expressões são necessárias.

Antes de dar continuidade à explicação, algumas observações são importantes. Considere que a variável binária X assume os estados x e $\neg x$ — ao invés de computar distribuições $p(X)$, basta computar o valor $P(X = x)$, pois necessariamente $P(X = \neg x) = 1 - P(X = x)$. O mesmo ocorre para a mensagem π — basta computar o valor $\pi(x)$. Para simplificar a notação, $\pi(x)$ é usado no lugar de $\pi(X = x)$. Idem para $p(x|U)$, $\pi_{Y_i}(x)$ e $\rho(x|U_i)$.

A primeira modificação feita é a reparametrização da mensagem λ por $\Lambda^X = \frac{\lambda(x)}{\lambda(\neg x)}$. Isto elimina as constantes de normalização α e β . Substituindo Λ^X nas Equações (2.3)-(2.7), obtém-se novas expressões:

$$P(X = x|\mathbf{E}) = \left(1 + \left(\frac{1}{\pi(x)} - 1 \right) \frac{1}{\Lambda^X} \right)^{-1}, \quad (3.5)$$

$$\pi(x) = \sum_U p(x|U) \prod_{i=1}^n \pi_X(U_i), \quad (3.6)$$

$$\Lambda^X = \prod_{j=1}^m \Lambda_{Y_j}^X, \quad (3.7)$$

$$\pi_{Y_j}(x) = \left(1 + \left(\frac{1}{\pi(x)} - 1 \right) \frac{1}{\prod_{k \neq j} \Lambda_{Y_k}^X} \right)^{-1}, \quad (3.8)$$

$$\Lambda_X^{U_i} = \frac{\rho(x|u_i) + (\Lambda^X - 1)^{-1}}{\rho(x|\neg u_i) + (\Lambda^X - 1)^{-1}}, \quad (3.9)$$

onde:

$$\rho(x|U_i) = \sum_{U_k \neq U_i} p(x|U) \prod_{k \neq i} \pi_X(U_k). \quad (3.10)$$

As mensagens propagadas pelo algoritmo 2U são representadas na Figura 18, onde:

- O valor $\pi(x)$ é computado através das mensagens $\pi_X(U_i)$ provenientes dos pais (Equação (3.6)), onde n é o número de pais.
- O valor Λ^X é computado através das mensagens $\Lambda_{Y_j}^X$ provenientes dos filhos

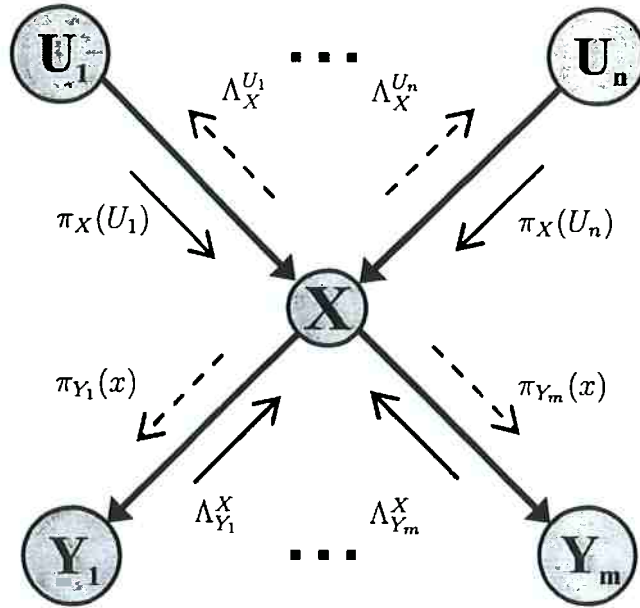


Figura 18: Esquema de propagação de mensagens do algoritmo 2U (FAGIUOLI; ZAFFALON, 1998). $\pi_X(U_i)$ e $\Lambda_{Y_i}^X$ são mensagens que o nó X recebe de seus pais e filhos (setas com linha cheia), para assim atualizar os valores $\pi(X)$ e Λ^X (Equações (3.6) e (3.7)). $\pi_{Y_i}(x)$ e $\Lambda_X^{U_i}$ são mensagens que o nó X envia aos seus pais e filhos (setas com linha tracejada).

(Equação (3.7)), onde m é o número de filhos.

- A mensagem $\pi_{Y_j}(x)$ enviada ao filho Y_j é dada em função de $\pi(x)$ e mensagens $\Lambda_{Y_k}^X$ ($k \neq j$) provenientes de outros filhos de X (Equação (3.8)).
- A mensagem $\Lambda_X^{U_i}$ enviada ao pai U_i é função de Λ^X e ρ (Equação (3.9)), onde $\neg u_i$ representa o complemento binário de $\{U_i = u_i\}$.
- O valor ρ é função das mensagens $\pi_X(U_k)$ ($k \neq i$) provenientes de outros pais de X (Equação (3.10)).

A partir das Equações (3.5)-(3.10), novas expressões são desenvolvidas para se obter mensagens em forma de intervalos:

$$P(X = x|\mathbf{E}) = \left(1 + \left(\frac{1}{\pi(x)} - 1\right) \frac{1}{\Lambda^X}\right)^{-1}, \quad (3.11)$$

$$\bar{P}(X = x|\mathbf{E}) = \left(1 + \left(\frac{1}{\bar{\pi}(x)} - 1\right) \frac{1}{\bar{\Lambda}^X}\right)^{-1}, \quad (3.12)$$

onde:

$$\pi(x) = \min_{\substack{j \in \{1, \dots, n\} \\ \pi_X(u_j) \in \{\underline{\pi}_X(u_j), \bar{\pi}_X(u_j)\}}} \sum_U \underline{p}(x|U) \prod_{i=1}^n \pi_X(U_i), \quad (3.13)$$

$$\bar{\pi}(x) = \max_{\substack{j \in \{1, \dots, n\} \\ \pi_X(u_j) \in \{\underline{\pi}_X(u_j), \bar{\pi}_X(u_j)\}}} \sum_U \bar{p}(x|U) \prod_{i=1}^n \pi_X(U_i), \quad (3.14)$$

$$\underline{\Delta}^X = \prod_{j=1}^m \underline{\Delta}_{Y_j}^X, \quad (3.15)$$

$$\bar{\Lambda}^X = \prod_{j=1}^m \bar{\Lambda}_{Y_j}^X, \quad (3.16)$$

$$\pi_{Y_j}(x) = \left(1 + \left(\frac{1}{\pi(x)} - 1 \right) \frac{1}{\prod_{k \neq j} \underline{\Delta}_{Y_k}^X} \right)^{-1}, \quad (3.17)$$

$$\bar{\pi}_{Y_j}(x) = \left(1 + \left(\frac{1}{\bar{\pi}(x)} - 1 \right) \frac{1}{\prod_{k \neq j} \bar{\Lambda}_{Y_k}^X} \right)^{-1}, \quad (3.18)$$

onde nas Equações (3.13) e (3.14), temos um problema de minimização e maximização, variando-se $\pi_X(u_j)$ entre valores máximos e mínimos para cada $j \in \{1, \dots, n\}$.

As mensagens enviadas aos pais U_i de X são dadas por:

$$\underline{\Delta}_X^{U_i} = \min_{\substack{j \in \{1, \dots, n\}, j \neq i \\ \pi_X(u_j) \in \{\underline{\pi}_X(u_j), \bar{\pi}_X(u_j)\}}} \left(\min_{\Lambda^X \in \{\underline{\Delta}^X, \bar{\Lambda}^X\}} \widehat{\Delta}_X^{U_i}(\Lambda^X) \right), \quad (3.19)$$

$$\bar{\Lambda}_X^{U_i} = \max_{\substack{j \in \{1, \dots, n\}, j \neq i \\ \pi_X(u_j) \in \{\underline{\pi}_X(u_j), \bar{\pi}_X(u_j)\}}} \left(\max_{\Lambda^X \in \{\underline{\Lambda}^X, \bar{\Lambda}^X\}} \bar{\Lambda}_X^{U_i}(\Lambda^X) \right), \quad (3.20)$$

onde nas Equações (3.19) e (3.20), temos um problema de minimização e maximização, variando-se $\pi_X(u_j)$ entre valores mínimos e máximos para cada $j \in \{1, \dots, n\}, j \neq i$. Temos também outro problema de minimização e maximização, variando-se Λ^X entre seus valores mínimos e máximos; e as expressões $\hat{\Lambda}_X^{U_i}(\Lambda^X)$ e $\bar{\Lambda}_X^{U_i}(\Lambda^X)$ são computadas de acordo com a Tabela 2, onde diferentes fórmulas são dadas em função do valor Λ^X , na qual $\hat{\rho}(x|u_i)$ e $\bar{\rho}(x|u_i)$ são computadas de acordo com a Equação (3.10).

Tabela 2: Fórmulas para obter os valores $\hat{\Lambda}_X^{U_i}(\Lambda^X)$ das Equações (3.19) e (3.20).

Diferentes fórmulas em função do valor Λ^X		
	$\hat{\Lambda}_X^{U_i}(\Lambda^X)$	$\bar{\Lambda}_X^{U_i}(\Lambda^X)$
$\Lambda^X < 1$	$\frac{\hat{\rho}(x u_i) + (\Lambda^X - 1)^{-1}}{\bar{\rho}(x \neg u_i) + (\Lambda^X - 1)^{-1}}$	$\frac{\bar{\rho}(x u_i) + (\Lambda^X - 1)^{-1}}{\hat{\rho}(x \neg u_i) + (\Lambda^X - 1)^{-1}}$
$\Lambda^X = 1$	1	1
$\Lambda^X > 1$	$\frac{\bar{\rho}(x u_i) + (\Lambda^X - 1)^{-1}}{\hat{\rho}(x \neg u_i) + (\Lambda^X - 1)^{-1}}$	$\frac{\hat{\rho}(x u_i) + (\Lambda^X - 1)^{-1}}{\bar{\rho}(x \neg u_i) + (\Lambda^X - 1)^{-1}}$

3.5.2 Propagação global de intervalos

A propagação global das mensagens intervalares pela rede é realizada conforme as seguintes etapas:

1. Inicialização das mensagens da rede. Há três tipos de nós que possuem mensagens fixas inicializadas no início da propagação. Para os nós raízes não observados, $\underline{\pi}(x) = \underline{\pi}_{V_i}(x) = \underline{P}(x)$ e $\bar{\pi}(x) = \bar{\pi}_{V_i}(x) = \bar{P}(x)$. Para os nós folhas não observados, $\underline{\Lambda}^X = \bar{\Lambda}^X = \underline{\Lambda}_{U_i}^X = \bar{\Lambda}_{U_i}^X = 1$. Enfim, para os nós observados com estado $\{X = x\}$, assumem-se $\underline{\Lambda}^X = \bar{\Lambda}^X = \frac{1}{0} = \infty$, e para os nós observados com estado $\{X = \neg x\}$, $\underline{\Lambda}^X = \bar{\Lambda}^X = \frac{0}{1} = 0$.

2. Propagação das mensagens intervalares ao longo da rede. Para cada nó, através das Equações (3.13)-(3.20), mensagens com extremos “inferiores” (valores mínimos) e “superiores” (valores máximos) são atualizadas.
3. Computação das probabilidades inferiores $\underline{P}(X = x|\mathbf{E})$ e superiores $\overline{P}(X = x|\mathbf{E})$ dos nós X desejados (Equações 3.11 e 3.12).

Assim, todo nó requer apenas uma computação local e a computação global é obtida pela atualização das mensagens em cada um dos nós da rede. A ordem em que os nós são atualizados dependerá da estrutura do grafo.

Para ilustrar o algoritmo 2U descrito, considere o exemplo de rede credal apresentada anteriormente na Figura 15. Suponha que se deseja calcular os extremos do intervalo da probabilidade de $\{D = true\}$ dado que $\{H = false\}$. Para simplificar a notação, denote por $\{D = true = d\}$, $\{H = false = \neg h\}$ e de modo análogo para as demais variáveis. Logo, deseja-se calcular os valores mínimos e máximos de $P(d|\neg h)$. A propagação de mensagens intervalares se dá do seguinte modo⁵:

- *Inicialização das mensagens da rede.* Para o nó raiz B , $\underline{\pi}_D(b) = \underline{\pi}(b) = \underline{P}(b) = 0.01$ e $\overline{\pi}_D(b) = \overline{\pi}(b) = \overline{P}(b) = 0.05$. Para o nó raiz F , $\underline{\pi}_D(f) = \underline{\pi}(f) = \underline{P}(f) = 0.15$ e $\overline{\pi}_D(f) = \overline{\pi}(f) = \overline{P}(f) = 0.25$. Para o nó observado $\{H = \neg h\}$, $\underline{\Lambda}^H = \overline{\Lambda}^H = 0$.
- *Atualização do nó H .* O nó H envia a mensagem Λ_H^D para D . Como H não possui outro pai além de D , tem-se que $\rho(h|d) = P(h|d)$ e $\rho(h|\neg d) = P(h|\neg d)$ (Equação (3.10)). Como H é um nó observado, $\Lambda^H = 0$, e substituindo nas Equações (3.19) e (3.20), temos $\underline{\Lambda}_H^D = \hat{\Lambda}_H^D(0)$ e $\overline{\Lambda}_H^D = \overline{\Lambda}_H^D(0)$, desenvolvidas a seguir:

$$\begin{aligned} \underline{\Lambda}_H^D = \hat{\Lambda}_H^D(0) &= \frac{\overline{P}(h|d) + (0 - 1)^{-1}}{\underline{P}(h|\neg d) + (0 - 1)^{-1}} = \frac{0.7 - 1}{0.01 - 1} = 0.3030, \\ \overline{\Lambda}_H^D = \overline{\Lambda}_H^D(0) &= \frac{\underline{P}(h|d) + (0 - 1)^{-1}}{\overline{P}(h|\neg d) + (0 - 1)^{-1}} = \frac{0.6 - 1}{0.05 - 1} = 0.4211. \end{aligned}$$

- *Atualização do nó D .* O nó D recebe mensagens dos nós pais F e B . Assim,

⁵Apenas as mensagens estritamente necessárias para o cálculo de $P(d|\neg h)$ são apresentadas.

das Equações (3.13) e (3.14), obtém-se as seguintes expressões:

$$\begin{aligned}\underline{\pi}(d) &= \underline{P}(d|f, b)\pi_D(f)\pi_D(b) + \underline{P}(d|f, \neg b)\pi_D(f)\pi_D(\neg b) + \\ &\quad \underline{P}(d|\neg f, b)\pi_D(\neg f)\pi_D(b) + \underline{P}(d|\neg f, \neg b)\pi_D(\neg f)\pi_D(\neg b), \\ \bar{\pi}(d) &= \bar{P}(d|f, b)\pi_D(f)\pi_D(b) + \bar{P}(d|f, \neg b)\pi_D(f)\pi_D(\neg b) + \\ &\quad \bar{P}(d|\neg f, b)\pi_D(\neg f)\pi_D(b) + \bar{P}(d|\neg f, \neg b)\pi_D(\neg f)\pi_D(\neg b).\end{aligned}$$

Para as expressões anteriores, existem quatro casos possíveis:

Caso 1: $\pi_D(f) = \underline{\pi}_D(f)$ e $\pi_B(f) = \underline{\pi}_B(f)$.

$$\begin{aligned}\underline{\pi}(d) &= 0.2 \times 0.15 \times 0.01 + 0.9 \times 0.15 \times 0.99 + \\ &\quad 0.05 \times 0.85 \times 0.01 + 0.6 \times 0.85 \times 0.99 = 0.6393, \\ \bar{\pi}(d) &= 0.3 \times 0.15 \times 0.01 + 0.95 \times 0.15 \times 0.99 + \\ &\quad 0.1 \times 0.85 \times 0.01 + 0.7 \times 0.85 \times 0.99 = 0.7314.\end{aligned}$$

Caso 2: $\pi_D(f) = \underline{\pi}_D(f)$ e $\pi_B(f) = \bar{\pi}_B(f)$.

$$\begin{aligned}\underline{\pi}(d) &= 0.2 \times 0.15 \times 0.05 + 0.9 \times 0.15 \times 0.95 + \\ &\quad 0.05 \times 0.85 \times 0.05 + 0.6 \times 0.85 \times 0.95 = 0.6164, \\ \bar{\pi}(d) &= 0.3 \times 0.15 \times 0.05 + 0.95 \times 0.15 \times 0.95 + \\ &\quad 0.1 \times 0.85 \times 0.05 + 0.7 \times 0.85 \times 0.95 = 0.7071.\end{aligned}$$

Caso 3: $\pi_D(f) = \bar{\pi}_D(f)$ e $\pi_B(f) = \underline{\pi}_B(f)$.

$$\begin{aligned}\underline{\pi}(d) &= 0.2 \times 0.25 \times 0.01 + 0.9 \times 0.25 \times 0.99 + \\ &\quad 0.05 \times 0.75 \times 0.01 + 0.6 \times 0.75 \times 0.99 = 0.6691, \\ \bar{\pi}(d) &= 0.3 \times 0.25 \times 0.01 + 0.95 \times 0.25 \times 0.99 + \\ &\quad 0.1 \times 0.75 \times 0.01 + 0.7 \times 0.75 \times 0.99 = 0.7564.\end{aligned}$$

Caso 4: $\pi_D(f) = \bar{\pi}_D(f)$ e $\pi_B(f) = \bar{\pi}_B(f)$.

$$\begin{aligned}\underline{\pi}(d) &= 0.2 \times 0.25 \times 0.05 + 0.9 \times 0.25 \times 0.95 + \\ &\quad 0.05 \times 0.75 \times 0.05 + 0.6 \times 0.75 \times 0.95 = 0.6456, \\ \bar{\pi}(d) &= 0.3 \times 0.25 \times 0.05 + 0.95 \times 0.25 \times 0.95 + \\ &\quad 0.1 \times 0.75 \times 0.05 + 0.7 \times 0.75 \times 0.95 = 0.7319.\end{aligned}$$

Tomando-se os valores mínimos e máximos, obtém-se o intervalo $\pi(d) \in [0.6164, 0.7564]$.

O nó D recebe a mensagem Λ_H^D de H . Pelas Equações (3.15) e (3.16), obtém-se o intervalo $\Lambda^D \in [0.3030, 0.4211]$. Substituindo os valores anteriormente obtidos nas Equações (3.11) e (3.12), obtém-se:

$$\begin{aligned} \underline{P}(d|\neg h) &= \left(1 + \left(\frac{1}{\underline{\pi}(d)} - 1\right) \frac{1}{\underline{\Lambda}^D}\right)^{-1}, \\ &= \left(1 + \left(\frac{1}{0.6164} - 1\right) \frac{1}{0.3030}\right)^{-1} = 0.3275, \\ \overline{P}(d|\neg h) &= \left(1 + \left(\frac{1}{\overline{\pi}(d)} - 1\right) \frac{1}{\overline{\Lambda}^D}\right)^{-1}, \\ &= \left(1 + \left(\frac{1}{0.7564} - 1\right) \frac{1}{0.4211}\right)^{-1} = 0.5666. \end{aligned}$$

Logo, o resultado da inferência obtido é $P(d|\neg h) \in [0.3275, 0.5666]$. Este exemplo numérico apresenta um caso simples. Um exemplo numérico mais completo é apresentado em Zaffalon (1997).

Como descrito por Fagioli e Zaffalon (1998, página 93), a complexidade do algoritmo 2U é $O(2^{2n_{pa_{max}}})$ no pior caso, onde $n_{pa_{max}}$ é o número máximo de pais que um nó pode possuir. A complexidade do algoritmo BP é $O(s^{n_{pa_{max}}})$, onde s é o número de estados da variável com $n_{pa_{max}}$ número de pais. Reescrevendo o termo $O(2^{2n_{pa_{max}}})$ por $O(4^{n_{pa_{max}}})$, conclui-se que a complexidade do algoritmo 2U é equivalente ao do algoritmo BP com variáveis assumindo 4 estados.

4 NOVOS ALGORITMOS PARA INFERÊNCIA APROXIMADA EM REDES CREDAIS

Existem muitas semelhanças entre redes Bayesianas e redes credais, pois ambas são modelos probabilísticos associados a grafos. A diferença é que redes Bayesianas são associadas a distribuições de probabilidade, enquanto que as redes credais são associadas a *conjuntos* de distribuições de probabilidade, ou seja, conjuntos credais. Isso sugere que algoritmos para redes Bayesianas sejam adaptados em redes credais. Esta é a principal estratégia adotada neste trabalho.

Neste capítulo, três novas propostas de inferência aproximada em redes credais são descritas. O ponto de partida são os métodos conhecidos e utilizados em redes Bayesianas. A novidade deste trabalho está na aplicação de tais métodos para redes credais. Os algoritmos apresentados são:

1. Algoritmo L2U (*Loopy 2U*) (IDE; COZMAN, 2004b). Este algoritmo aplica a idéia de *propagação em ciclos*¹ ao algoritmo 2U descrito na Seção 3.5.
2. Algoritmo IPE (*Iterated Partial Evaluation*) (IDE; COZMAN, 2004b). Este algoritmo emprega *mensagens vazias*² para aproximar a rede multi-conectada por uma poli-árvore, e realiza um processo iterativo de refinamento dos intervalos de inferência obtidos.
3. Algoritmo SV2U (*Structured Variational 2U*) (IDE; COZMAN, 2005a) (IDE; COZMAN, 2005b). Este algoritmo aplica *técnicas variacionais* (Seção 2.6) para aproximar uma rede credal multi-conectada por uma poli-árvore, e assim executa o algoritmo 2U nesta rede aproximada.

Fundamentalmente, os novos algoritmos transformam o problema de inferência em redes multi-conectadas para o problema de inferência em poli-árvores. Os

¹Idéia explorada pelo algoritmo LBP descrito na Seção 2.4.

²Recurso utilizado pelo algoritmo LPE descrito na Seção 2.5.

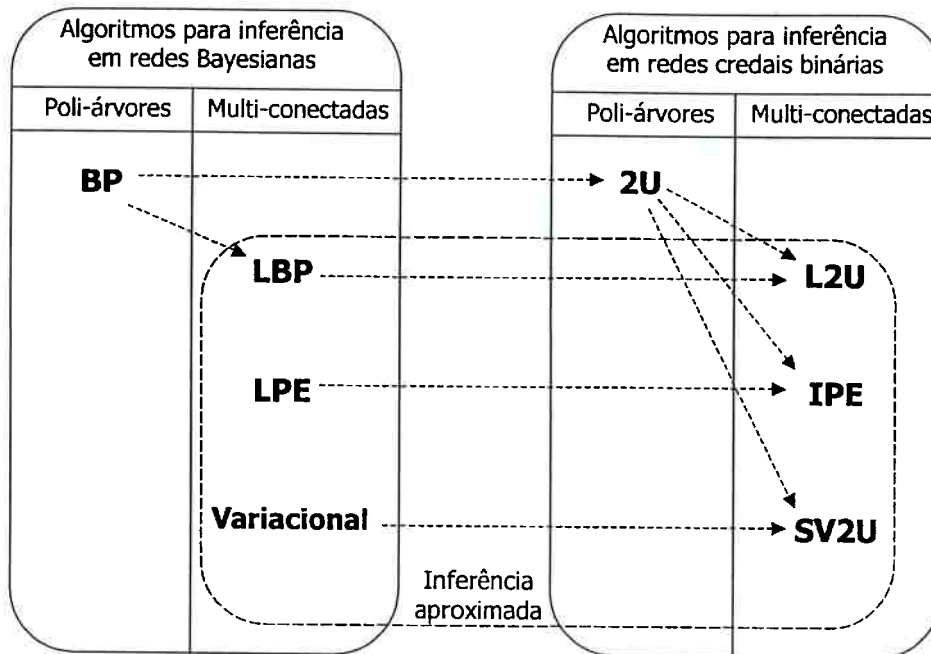


Figura 19: Resumo esquemático dos algoritmos desenvolvidos.

algoritmos apresentados realizam inferência aproximada somente em redes credais *binárias* (problema descrito na Seção 3.4), pois dependem do algoritmo 2U. Os métodos de aproximação da rede podem ser aplicados a redes credais com variáveis não binárias, e até contínuas, desde que se possua um eficiente algoritmo para inferência em poli-árvores.

Um resumo esquemático dos algoritmos desenvolvidos é apresentado na Figura 19. Os algoritmos para inferência aproximada em redes Bayesianas (algoritmo LBP, algoritmo LPE e algoritmos baseados em métodos variacionais) são adaptados para realização de inferência aproximada em redes credais binárias, e novos algoritmos são obtidos: algoritmos L2U, IPE e SV2U. Todos os novos algoritmos utilizam o algoritmo 2U, que fornece resultados de inferência exata em redes credais binárias com estrutura poli-árvore.

Os algoritmos desenvolvidos L2U, IPE e SV2U são apresentados nas Seções 4.1, 4.2 e 4.3, respectivamente.

4.1 “Loopy Belief Propagation” em redes credais: algoritmo L2U

Nesta seção descreve-se o primeiro dos métodos desenvolvidos nesse trabalho, o algoritmo L2U. O algoritmo L2U é uma versão “loopy” do algoritmo 2U (Seção 3.5), inspirado no popular algoritmo *Loopy Belief Propagation* (LBP) (MURPHY; WEISS; JORDAN, 1999) descrito na Seção 2.4. O algoritmo L2U realiza inferência aproximada em redes credais multi-conectadas com variáveis binárias (problema apresentado na Seção 3.4).

Assim como os métodos baseados em LBP, o algoritmo L2U não fornece garantias teóricas de convergência dos resultados que gera. Entretanto, ele produz os resultados mais precisos entre os algoritmos desse trabalho, com menor custo computacional, como demonstrado pelos resultados proporcionados no Capítulo 5.

Na Seção 4.1.1, o algoritmo L2U é descrito em detalhes e na Seção 4.1.2, um exemplo numérico aplicando o método é apresentado.

4.1.1 Descrição do algoritmo L2U

Numa rede credal multi-conectada C , considere que uma seqüência de nós S seja aleatoriamente escolhida, de modo que todos os nós sejam percorridos. A inicialização das variáveis e mensagens segue o esquema padrão empregado no algoritmo 2U. Assim, os nós são repetidamente atualizados seguindo a seqüência S . As iterações são indexadas por t , começando em 0, e atualizações são repetidas até observar a convergência das probabilidades ou até atingir um limite máximo de iterações especificado pelo usuário.

O esquema de propagação global do algoritmo L2U segue as mesmas etapas percorridas pelo algoritmo LBP, mas ao invés de propagar as mensagens do algoritmo BP (valores “pontuais” de probabilidade), o algoritmo transmite mensagens intervalares do algoritmo 2U. O algoritmo L2U é apresentado na Figura 20. As etapas seguidas são as seguintes:

1. Como no algoritmo BP, nós raízes, filhos e os nós observados são inicializados (Linha 02) (Cfr. primeiro parágrafo da Seção 3.5.2). Se o nó X é

- L2U: algoritmo Loopy 2U**
 Entrada: rede credal multi-conectada \mathbf{C} .
 Saída: aproximações das probabilidades inferior/superior da variável X_Q dadas a evidência \mathbf{E} .
01. Carregar a rede \mathbf{C} multi-conectada;
 02. Inicializar e manter fixas as mensagens dos nós raízes e folhas, assim como as mensagens dos nós que estão observados (como descrito na Seção 3.5.2). Considerar esta etapa de inicialização como $t = 0$.
 03. Escolher aleatoriamente uma seqüência S de nós a serem percorridos na rede.
 04. Repetir até a convergência ou até atingir um limite máximo de iterações (inicializar com $t = 0$):
 05. para cada nó X da seqüência S , computar:
 06. a) $\underline{\pi}^{(t+1)}(x)$ e $\overline{\pi}^{(t+1)}(x)$ (Equações (3.13) e (3.14)), dadas as mensagens $\underline{\pi}_X^{(t)}(U_i)$ e $\overline{\pi}_X^{(t)}(U_i)$;
 07. b) $\underline{\Lambda}^{(t+1)X}$ e $\overline{\Lambda}^{(t+1)X}$ (Equações (3.15) e (3.16)), dadas as mensagens $\underline{\Lambda}_{Y_i}^{(t)X}$ e $\overline{\Lambda}_{Y_i}^{(t)X}$;
 08. c) mensagens a serem enviadas aos filhos $\underline{\pi}_{Y_j}^{(t+1)}(x)$ e $\overline{\pi}_{Y_j}^{(t+1)}(x)$ (Equações (3.17) e (3.18)), dadas as mensagens $\underline{\pi}_X^{(t)}(x)$ e $\overline{\pi}_X^{(t)}(x)$; e $\underline{\Lambda}_{Y_k}^{(t)X}$ e $\overline{\Lambda}_{Y_k}^{(t)X}$;
 09. d) mensagens a serem enviadas aos pais $\underline{\Lambda}_X^{(t+1)U_i}$ e $\overline{\Lambda}_X^{(t+1)U_i}$ (Equações (3.19) e (3.20)), dadas $\underline{\Lambda}^{(t)X}$, $\overline{\Lambda}^{(t)X}$, $\underline{\pi}_X^{(t)}(U_k)$ e $\overline{\pi}_X^{(t)}(U_k)$.
 10. Obter o intervalo de probabilidades dos nós X_Q desejados $\underline{P}^*(X_Q = x_Q|\mathbf{E})$ e $\overline{P}^*(X_Q = x_Q|\mathbf{E})$ (Equações (3.11) e (3.12)), dadas as mensagens ao final da iteração: $\underline{\pi}(x_Q)$, $\overline{\pi}(x_Q)$, $\underline{\Lambda}^{X_Q}$, $\overline{\Lambda}^{X_Q}$.

Figura 20: Algoritmo L2U.

observado no estado x , então $\Lambda^X = \frac{\lambda(x)}{\lambda(\neg x)} = \frac{1}{0} = \infty$; e se observado $\neg x$, então $\Lambda^X = \frac{\lambda(x)}{\lambda(\neg x)} = \frac{0}{1} = 0$. Para os demais nós, mensagens são inicializadas convenientemente para serem atualizadas posteriormente – mensagens $\Lambda_X^{U_i}$ e $\pi_{Y_i}(x)$ recebem o valor “1”, onde U_i e Y_i denotam os nós pais e filhos de X , respectivamente. O significado probabilístico de $\Lambda_X^{U_i} = 1$ e $\pi_{Y_i}(x) = 1$ é “não consta nenhuma observação até o momento” (Cfr. segundo parágrafo da Seção 2.4).

2. Uma seqüência S aleatória de nós é escolhida para as atualizações (Linha 03). Na propagação com ciclos, a seqüência S pode ser qualquer uma, desde que todos os nós sejam percorridos. Como alguns estudos indicam (YEDIDIA; FREEMAN; WEISS, 2000), esta seqüência está relacionada com a convergência do algoritmo, sendo um tópico interessante de pesquisa para

o algoritmo L2U.

3. Nas Linhas 06 e 07, mensagens $\pi(x)$ (*suporte de previsão*) e Λ^X (*suporte retrospectivo*³) são atualizadas de acordo com as mensagens provenientes⁴ dos nós pais e filhos de X (Equações (3.13)-(3.16)). Uma observação deve ser feita com relação ao índice t . Ao atualizar o nó X na iteração $(t+1)$, caso alguns de seus pais e filhos já tenham sido atualizados na mesma iteração, empregam-se as mensagens $\pi_X^{(t+1)}(U_i)$ e $\Lambda_{Y_i}^{(t+1)X}$ já atualizadas. Note que isto acelera o processo de convergência.
4. Nas Linhas 08 e 09 mensagens a serem enviadas aos pais e filhos de X (Equações (3.17)-(3.20)) são atualizadas.
5. Na Linha 10, uma vez que a propagação converge ou o número máximo de iterações é atingido, os valores $\underline{P}^*(X_Q = x_Q|\mathbf{E})$ e $\overline{P}^*(X_Q = x_Q|\mathbf{E})$ são calculados (Equações (3.11) e (3.12)).

O cálculo das Equações (3.13) e (3.14) é o que exige maior esforço computacional. Para cada equação, o nó X_i exige a busca dos valores extremos entre $2^{n_{pa_i}}$ possíveis combinações das mensagens provenientes dos pais, onde n_{pa_i} é o número de pais de X_i ; em seguida, a somatória se dá entre $2^{n_{pa_i}}$ possíveis estados dos pais. Logo, para atualizar X_i , é preciso realizar uma busca da ordem de $2 \times 2^{n_{pa_i}} \times 2 \times 2^{n_{pa_i}} = 4 \times 2^{2n_{pa_i}}$. Deste modo, a complexidade geral do algoritmo L2U é $O(4k \times 2^{2n_{pa_{max}}})$ no pior caso, onde k é o número máximo de iterações e $n_{pa_{max}}$ é o número máximo de pais que um nó pode possuir. Nos experimentos conduzidos no Capítulo 5 para redes credais com 10 a 37 nós, verifica-se empiricamente que o número k é pequeno, sendo da ordem de dezenas para produzir resultados com erros até cerca de 2%.

A convergência do algoritmo L2U é um problema em aberto, assim como o é para o algoritmo LBP (Seção 2.4.3). Mas, os experimentos conduzidos no Capítulo 5 mostram que os intervalos produzidos pelo algoritmo L2U convergem em todas as redes testadas. Para que a convergência seja atingida, existem duas possibilidades:

³Ver as definições apresentadas na Seção 2.4.1.

⁴Note que as mensagens provenientes dos nós pais e filhos na iteração $(t+1)$ correspondem às mensagens enviadas por eles na iteração anterior (t) . Por exemplo, se A é o nó pai e B o nó filho, então $\Lambda_B^{(t)A}$ é a mensagem recebida por A na iteração $(t+1)$ para atualizar $\pi^{(t+1)}(a)$.

i) Atingem-se os vértices que fornecem os valores inferiores e superiores do intervalo. Estes vértices não variam mais após a convergência.

ii) Os intervalos fornecidos pelo algoritmo convergem, mas os vértices individualmente não convergem.

No primeiro caso, quando o algoritmo L2U converge, ou seja, os intervalos fornecidos pelo algoritmo convergem para valores fixos, recai-se no mesmo problema de convergência do algoritmo LBP. Pois, como todo vértice corresponde a uma rede Bayesiana, executar o algoritmo L2U significa executar o algoritmo LBP em duas redes Bayesianas distintas, uma que fornece os valores inferiores e outra que fornece os valores superiores das mensagens.

4.1.2 Exemplo numérico de aplicação do algoritmo L2U

Nesta seção é apresentado um exemplo simples de aplicação do algoritmo L2U em uma rede credal com 4 nós. Como ilustrado no exemplo numérico da Seção 3.5.2, a análise dos casos é demasiadamente extensa — razão pela qual apresentam-se apenas as expressões das mensagens e seus respectivos intervalos.

Considere a rede credal da Figura 17. Suponha que o nó D esteja observado e que se deseja computar o intervalo $I(a|d) = [\underline{P}(a|d), \overline{P}(a|d)]$. A sequência de atualização adotada aleatoriamente é $B \rightarrow D \rightarrow C \rightarrow A$. As mensagens dos nós são inicializadas em $t=0$ como descrito na Seção 4.1.1 e as mensagens, atualizadas em cada iteração t , são apresentadas na Tabela 3. Como D está observado em d , $\Lambda^D = \infty$ e como C é nó folha, $\Lambda^C = \Lambda_C^B = \Lambda_C^A = 1$. As expressões para atualização das mensagens são as seguintes⁵:

Nó B

• Como B é raiz, a mensagem $\pi(b)$ inferior (Cfr. Equação (3.13)) é fixa e dada pela expressão:

$$\underline{\pi}(b) = \underline{p}(b).$$

⁵Para não tomar muito espaço, apresentam-se somente as expressões para os valores inferiores das mensagens.

- Mensagem Λ^B inferior (Equação (3.15)) é dada pela expressão:

$$\underline{\Lambda}^B = \underline{\Lambda}_C^B \times \underline{\Lambda}_D^B = \underline{\Lambda}_D^B.$$

- Mensagem $\pi_C(b)$ inferior (Equação (3.17)) enviada ao nó B é dada pela expressão:

$$\pi_C(b) = \left(1 + \left(\frac{1}{\pi(b)} - 1 \right) \frac{1}{\underline{\Lambda}_D^B} \right)^{-1}.$$

- Mensagem $\pi_D(b)$ inferior (Equação (3.17)) enviada ao nó D é dada pela expressão:

$$\pi_D(b) = \left(1 + \left(\frac{1}{\pi(b)} - 1 \right) \frac{1}{\underline{\Lambda}_C^B} \right)^{-1} = \pi(b) = \underline{p}(b).$$

Nó D

- A mensagem Λ_D^A inferior (Equação (3.19) e Tabela 2) enviada ao nó A é dada pela expressão:

$$\underline{\Lambda}_D^A = \min_{\pi_D(b) \in \{\underline{\pi}_D(b), \bar{\pi}_D(b)\}} \left(\frac{\bar{p}(d|a)}{\underline{p}(d|\neg a)} \right) = \min_{\pi_D(b) \in \{\underline{\pi}_D(b), \bar{\pi}_D(b)\}} \left(\frac{\sum_B \bar{p}(d|a, B) \times \pi_D(b)}{\sum_B \underline{p}(d|\neg a, B) \times \pi_D(b)} \right).$$

- A mensagem Λ_D^B inferior (Equação (3.19) e Tabela 2) enviada ao nó B é dada pela expressão:

$$\underline{\Lambda}_D^B = \min_{\pi_D(a) \in \{\underline{\pi}_D(a), \bar{\pi}_D(a)\}} \left(\frac{\bar{p}(d|b)}{\underline{p}(d|\neg b)} \right) = \min_{\pi_D(a) \in \{\underline{\pi}_D(a), \bar{\pi}_D(a)\}} \left(\frac{\sum_A \bar{p}(d|A, b) \times \pi_D(b)}{\sum_A \underline{p}(d|A, \neg b) \times \pi_D(a)} \right).$$

Nó C

- A mensagem $\pi(c)$ inferior (Equação (3.13)) é dada pela expressão:

$$\pi(c) = \min_{\substack{\pi_C(a) \in \{\underline{\pi}_C(a), \bar{\pi}_C(a)\} \\ \pi_C(b) \in \{\underline{\pi}_C(b), \bar{\pi}_C(b)\}}} \sum_{A,B} \underline{p}(c|A, B) \times \pi_C(A) \times \pi_C(B).$$

Nó A

- Como A é raiz, a mensagem $\pi(a)$ inferior (Equação (3.13)) é fixa e dada pela expressão:

$$\pi(a) = \underline{p}(a).$$

- Mensagem Λ^A inferior (Equação (3.15)) é dada pela expressão:

$$\underline{\Lambda}^A = \underline{\Lambda}_C^A \times \underline{\Lambda}_D^A = \underline{\Lambda}_D^A.$$

- Mensagem $\pi_C(a)$ inferior (Equação (3.17)) enviada ao nó C é dada pela expressão:

$$\pi_C(a) = \left(1 + \left(\frac{1}{\pi(a)} - 1 \right) \frac{1}{\underline{\Lambda}_D^A} \right)^{-1}.$$

- Mensagem $\pi_D(a)$ inferior (Equação (3.17)) enviada ao nó D é dada pela expressão:

$$\pi_D(a) = \left(1 + \left(\frac{1}{\pi(a)} - 1 \right) \frac{1}{\underline{\Lambda}_C^A} \right)^{-1} = \pi(a) = \underline{p}(a).$$

Utilizando o software desenvolvido 2UBayes, o esquema iterativo apresentado anteriormente converge⁶ em 15 iterações. O resultado obtido ao final é o intervalo aproximado $I(a|d) = [0.1432, 0.3368]$. O intervalo exato é $I(a|d) = [0.1391, 0.3256]$. A partir dos dados da iteração $t=2$, obtém-se aplicando as

⁶Adotou-se um critério “rigoroso” de convergência: a invariância de todas as mensagens entre uma iteração e outra, com precisão na 16^a casa decimal.

Tabela 3: Tabela contendo mensagens intervalares do algoritmo L2U para a rede credal da Figura 17, para $t=0, 1$ e 2 .

Mensagens	$t = 0$	$t = 1$	$t = 2$
Λ^B	[1.0,1.0]	[1.0,1.0]	[0.5000,0.8000]
$\pi_C(b)$	[1.0,1.0]	[0.6000,0.7000]	[0.4286,0.6512]
Λ_D^A	[1.0,1.0]	[0.2424,0.4828]	[0.2535,0.5116]
Λ_D^B	[1.0,1.0]	[0.5000,0.8000]	[0.5521,0.8448]
$\pi(c)$	[1.0,1.0]	[0.4900,0.7050]	[0.3354,0.6393]
Λ^A	[1.0,1.0]	[0.2424,0.4828]	[0.2535,0.5116]
$\pi_C(a)$	[1.0,1.0]	[0.1391,0.3256]	[0.1446,0.3385]

Equações (3.11) e (3.12):

$$\underline{P}(a|d) = \left(1 + \left(\frac{1}{0.4} - 1\right) \frac{1}{0.2535}\right)^{-1} = 0.1446, \quad (4.1)$$

$$\bar{P}(a|d) = \left(1 + \left(\frac{1}{0.5} - 1\right) \frac{1}{0.5116}\right)^{-1} = 0.3384. \quad (4.2)$$

4.2 “Localized Partial Evaluation” em redes credais: algoritmo IPE

Nesta seção, descreve-se o algoritmo de *avaliação parcial iterada* (*Iterated Partial Evaluation* - IPE). O algoritmo IPE é inspirado no método LPE descrito na Seção 2.5 e realiza inferências em redes credais multi-conectadas com variáveis binárias (problema apresentado na Seção 3.4). O algoritmo desenvolvido é capaz de gerar *intervalos externos* de probabilidade, com garantias teóricas de convergência. Entretanto, os resultados que o algoritmo IPE fornece não são tão precisos, se comparados aos dos demais algoritmos desenvolvidos (Cfr. Capítulo 5).

Na Seção 4.2.1, o algoritmo IPE é descrito em detalhes e na Seção 4.2.2, um exemplo aplicando o método é apresentado.

4.2.1 Descrição do algoritmo IPE

O algoritmo IPE utiliza a mesma estratégia empregada por Draper e Hanks (1994): inserção de arcos ausentes (*missing arcs*) com mensagens vazias (*vacuous*

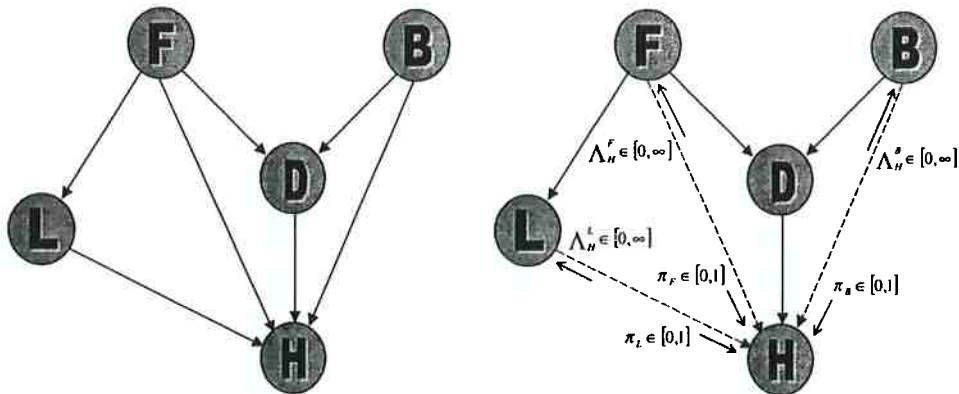


Figura 21: Ilustração de como os *arcos ausentes* (arcos com linha pontilhada) são inseridos. Esquerda: a rede multi-conectada original; direita: rede poli-árvore com *arcos ausentes* e suas mensagens vazias.

messages); e escolha de uma sub-rede (*active-set*). No algoritmo IPE, escolhe-se como *active-set* uma poli-árvore obtida a partir da rede multi-conectada original, pela remoção de arcos contidos no *conjunto de condicionamento*. Para se obter uma poli-árvore a partir de uma estrutura multi-conectada é necessário encontrar um *conjunto de condicionamento (loopy cutset)*, como descrito por Pearl (1988). Uma vez obtido o conjunto de condicionamento, basta remover os arcos que estão neste conjunto. Por exemplo, se o grafo multi-conectado possui n nós e n_a arcos, é necessário remover $(n_a - n + 1)$ arcos, mantendo o grafo conectado.

O método pode ser visto como um processo de aproximação da rede credal multi-conectada por uma rede credal com estrutura poli-árvore, através da inserção de mensagens vazias. O processo é ilustrado na Figura 21. Uma vez obtida a poli-árvore, o algoritmo 2U (Seção 3.5) é executado, obtendo-se ao final um intervalo externo de probabilidade.

Além disso, o algoritmo IPE executa um processo iterativo para refinar os intervalos obtidos. O *processo iterativo* consiste na seleção de não só uma, mas de várias poli-árvores que podem ser obtidas a partir da mesma rede multi-conectada. Isto significa que vários intervalos externos são obtidos. Denote por I_i^* o intervalo externo obtido na iteração i . O intervalo refinado I_i^* é obtido, mantendo-se o maior valor \underline{P}_{max} dos extremos inferiores de I_i^* e o menor valor \overline{P}_{min} dos extremos superiores de I_i^* . Geometricamente, este intervalo refinado consiste na interseção dos intervalos externos I_i^* . Garantias teóricas de con-

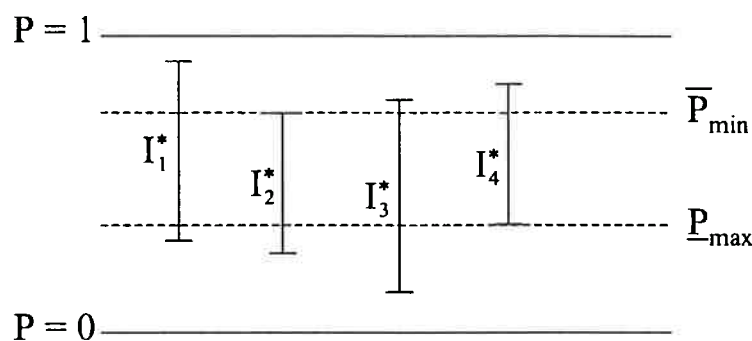


Figura 22: Ilustração do processo de refinamento dos intervalos externos obtidos. Em termos gráficos, o intervalo externo final consiste na interseção dos intervalos.

vergência do intervalo $I_r^* = [\underline{P}_{max}, \overline{P}_{min}]$ são dadas pelo Teorema 1 apresentado mais adiante. O processo iterativo de refinamento é ilustrado na Figura 22.

O algoritmo IPE é descrito na Figura 23. Seja \mathbf{C} a rede credal multi-conectada original e \mathbf{C}_p^* a poli-árvore obtida pela inserção de *arcos ausentes* (dado um conjunto de condicionamento). O algoritmo possui as seguintes principais etapas:

1. Construção da poli-árvore $\mathbf{C}_{p_i}^*$ na iteração i (Linha 02). As mensagens dos *arcos ausentes* são inicializadas. As mensagens a serem enviadas aos nós filhos são inicializadas como no algoritmo da Draper e Hanks (1994): $\underline{\pi}_{Y_j}(x) = 0$ e $\overline{\pi}_{Y_j}(x) = 1$. Já as mensagens enviadas aos pais precisam ser adaptadas. Como $\lambda(x) \in [0, 1]$, as mensagens a serem enviadas aos pais serão $\underline{\Lambda}_{Y_k}^X = \frac{\lambda(x)}{\lambda(\neg x)} = \frac{0}{1} = 0$ e $\overline{\Lambda}_{Y_k}^X = \frac{1}{0} = \infty$ (veja a Figura 21). As mensagens dos *arcos ausentes* não são atualizadas ao longo da iteração.
2. Execução do algoritmo 2U na poli-árvore $\mathbf{C}_{p_i}^*$ e cômputo do intervalo I_i^* (Linha 03). As mensagens são inicializadas e propagadas como no algoritmo 2U (Seção 3.5.2). Os valores extremos do intervalo I_i^* são computados após a propagação e guardados, a cada iteração i .
3. Computação do intervalo refinado I_r^* (Linha 04). I_r^* é computado encontrando o conjunto interseção dos intervalos I_i^* , onde $\{i = 1, \dots, i_{max}\}$; e computando ao final o menor intervalo de probabilidades $[\underline{P}_{max}, \overline{P}_{min}] =$

<p>Algoritmo IPE</p> <p>Entrada: rede credal multi-conectada C.</p> <p>Saída: aproximações das probabilidades inferior/superior da variável X_Q dada a evidência E.</p> <p>01. Denote por i o índice da iteração, começando por 1. Repetir até atingir o número máximo de iterações i_{max} especificado pelo usuário;</p> <p>02. a) Selecionar um conjunto de condicionamento e construir uma poli-árvore $C_{p_i}^*$, substituindo os arcos contidos no conjunto condicionamento por <i>arcos ausentes</i> e inserindo as mensagens vazias. $C_{p_i}^*$ representa a poli-árvore na iteração i;</p> <p>03. b) Para cada poli-árvore $C_{p_i}^*$, executar a propagação 2U, com suas inicializações como no algoritmo L2U; e computar os valores extremos do intervalo $I_i^* = [\underline{P}_{C_{p_i}^*}(X_Q = x_Q E), \overline{P}_{C_{p_i}^*}(X_Q = x_Q E)]$;</p> <p>04. Computar o intervalo refinado $I_r^* = [\underline{P}_{max}, \overline{P}_{min}]$ para a variável deseja X_Q.</p>

Figura 23: Descrição do algoritmo *Iterated Partial Evaluation* (IPE).

$$[\max \underline{P}_{C_p^*}(X_Q = x_Q|E), \min \overline{P}_{C_p^*}(X_Q = x_Q|E)].$$

Resta agora, provar que o intervalo aproximado obtido I_r^* contém o intervalo exato. Considere as probabilidades extremas dos intervalos, tem-se que:

Teorema 1 *O intervalo de probabilidade $I_r^* = [\max \underline{P}_{C_p^*}(X_Q = x_Q|E), \min \overline{P}_{C_p^*}(X_Q = x_Q|E)]$, produzido pelo algoritmo IPE, converge e contém o intervalo exato $[\underline{P}_C(X_Q = x_Q|E), \overline{P}_C(X_Q = x_Q|E)]$ requisitado no problema de inferência.*

Prova. Em primeiro lugar, o intervalo I_r^* converge, pois, a cada iteração i , o algoritmo IPE fornece um intervalo I_r^* atualizado que é sempre menor ou igual ao intervalo anterior. Em segundo lugar, os valores $\underline{P}_{C_{p_i}^*}(X_Q = x_Q|E)$ e $\overline{P}_{C_{p_i}^*}(X_Q = x_Q|E)$ obtidos em cada iteração i resultam de alguns dos possíveis vértices da extensão forte da rede $C_{p_i}^*$ (Cfr. problema de inferência descrito na Seção 3.4). Ou seja, há um vértice que resulta no valor mínimo e outro no valor máximo. Todo vértice é uma rede Bayesiana na qual, ao se aplicar o esquema de propagação do algoritmo LPE, sabe-se que para um dado conjunto de *arcos ausentes* os intervalos de probabilidade obtidos contém a probabilidade pontual (DRAPER, 1995). Deste modo, fazendo a interseção de todos os intervalos de valores extremos $\underline{P}_{C_{p_i}^*}(X_Q = x_Q|E)$ e $\overline{P}_{C_{p_i}^*}(X_Q = x_Q|E)$ — obtém-se o menor intervalo $[\underline{P}_{C_p^*}(X_Q = x_Q|E), \overline{P}_{C_p^*}(X_Q = x_Q|E)]$ que necessariamente conterá o intervalo exato desejado $[\underline{P}_C(X_Q = x_Q|E), \overline{P}_C(X_Q = x_Q|E)]$. QED

A complexidade do algoritmo IPE é da mesma ordem de grandeza do algoritmo 2U. Como o algoritmo roda i_{max} vezes o algoritmo 2U, a sua complexidade será $O(i_{max} \times 2^{2n_{pa_{max}}})$, onde $n_{pa_{max}}$ é o número máximo de pais, no pior caso. Nos experimentos conduzidos no Capítulo 5, para redes credais com 10 a 37 nós, verifica-se empiricamente que o número i_{max} é da ordem de centenas para produzir resultados com erros até cerca de 7%.

4.2.2 Exemplo de aplicação do algoritmo IPE

Nesta seção é apresentado um exemplo simples de aplicação do algoritmo IPE. As expressões das mensagens empregadas são fornecidas.

Considere o mesmo problema de inferência apresentado na Seção 4.1.2. A rede credal tem quatro nós (Figura 17). O nó D está observado para d e se deseja computar o intervalo $I(a|d) = [\underline{P}(a|d), \bar{P}(a|d)]$. Para aplicar o algoritmo IPE, deve-se selecionar uma poli-árvore a partir da rede original. Adote por exemplo a poli-árvore que se obtém retirando-se o arco (B, C) . Um *missing arc* é colocado no lugar deste arco. Este novo arco possui *vacuous messages* da seguinte forma⁷: $\Lambda_C^{B*} \in [0, \infty]$ e $\pi_C^*(b) \in [0, 1]$.

As mensagens dos nós da nova rede poli-árvore devem ser inicializadas. Como D está observado para d , $\Lambda^D = \infty$. Como C é nó folha, $\Lambda^C = \Lambda_C^A = 1$. Recorde que todas estas mensagens inicializadas até o momento são fixas, ou seja, não devem ser atualizadas. As expressões de atualização das demais mensagens são:

Nó B

- Mensagem Λ^B inferior (Equação (3.15)) é dada pela expressão:

$$\underline{\Lambda}^B = \underline{\Lambda}_C^B \times \underline{\Lambda}_D^B.$$

- Mensagem $\pi_D(b)$ inferior (Equação (3.17)) enviada ao nó D é dada pela expressão:

$$\pi_D(b) = \left(1 + \left(\frac{1}{\pi(b)} - 1 \right) \frac{1}{\underline{\Lambda}_C^{B*}} \right)^{-1}.$$

⁷Ou seja, $\underline{\Lambda}_C^{B*} = 0$, $\bar{\Lambda}_C^{B*} = \infty$, $\pi_C^*(b) = 0$ e $\bar{\pi}_C^*(b) = 1$.

Nó D

• A mensagem Λ_D^A inferior (Equação (3.19) e Tabela 2) enviada ao nó A é dada pela expressão:

$$\underline{\Lambda}_D^A = \min_{\pi_D(b) \in \{\underline{\pi}_D(b), \bar{\pi}_D(b)\}} \left(\frac{\bar{p}(d|a)}{\underline{p}(d|\neg a)} \right) = \min_{\pi_D(b) \in \{\underline{\pi}_D(b), \bar{\pi}_D(b)\}} \left(\frac{\sum_B \bar{p}(d|a, B) \times \pi_D(b)}{\sum_B \underline{p}(d|\neg a, B) \times \pi_D(b)} \right).$$

• A mensagem Λ_D^B inferior (Equação (3.19) e Tabela 2) enviada ao nó B é dada pela expressão:

$$\underline{\Lambda}_D^B = \min_{\pi_D(a) \in \{\underline{\pi}_D(a), \bar{\pi}_D(a)\}} \left(\frac{\bar{p}(d|b)}{\underline{p}(d|\neg b)} \right) = \min_{\pi_D(a) \in \{\underline{\pi}_D(a), \bar{\pi}_D(a)\}} \left(\frac{\sum_A \bar{p}(d|A, b) \times \pi_D(b)}{\sum_A \underline{p}(d|A, \neg b) \times \pi_D(a)} \right).$$

Nó C

• A mensagem $\pi(c)$ inferior (Equação (3.13)) é dada pela expressão:

$$\underline{\pi}(c) = \min_{\substack{\pi_C(a) \in \{\underline{\pi}_C(a), \bar{\pi}_C(a)\} \\ \pi_C(b) \in \{\underline{\pi}_C(b), \bar{\pi}_C(b)\}}} \sum_{A, B} \underline{p}(c|A, B) \times \pi_C(A) \times \pi_C^*(B).$$

Nó A

• Mensagem Λ^A inferior (Equação (3.15)) é dada pela expressão:

$$\underline{\Lambda}^A = \underline{\Lambda}_C^A \times \underline{\Lambda}_D^A = \underline{\Lambda}_D^A.$$

• Mensagem $\pi_C(a)$ inferior (Equação (3.17)) enviada ao nó C é dada pela expressão:

$$\underline{\pi}_C(a) = \left(1 + \left(\frac{1}{\underline{\pi}(a)} - 1 \right) \frac{1}{\underline{\Lambda}_D^A} \right)^{-1}.$$

• Mensagem $\pi_D(a)$ inferior (Equação (3.17)) enviada ao nó D é dada pela

expressão:

$$\pi_D(a) = \left(1 + \left(\frac{1}{\pi(a)} - 1\right) \frac{1}{\frac{\Delta A}{\Delta C}}\right)^{-1} = \pi(a) = \underline{p}(a).$$

Comparando as expressões anteriores com as da aplicação do algoritmo L2U (Seção 4.1.2), percebem-se algumas modificações nas expressões das mensagens do arco removido (B, C) : Λ_C^{B*} e $\pi_C^*(b)$.

O algoritmo 2U é rodado na poli-árvore selecionada e um intervalo externo I_1^* é computado. O procedimento descrito até agora deve ser repetido para uma diferente poli-árvore, obtendo-se ao final outro intervalo externo I_2^* . No caso da rede em questão (Figura 17), o número de diferentes poli-árvores que podem ser geradas a partir dela é apenas 4. Ou seja, pode-se obter mais duas poli-árvores, mais dois intervalos externos: I_3^* e I_4^* . Por fim, pela interseção destes intervalos externos, obtém-se o menor intervalo externo I_r^* que aproxima o intervalo exato desejado $I(a|d) = [\underline{P}(a|d), \overline{P}(a|d)]$.

4.3 Métodos variacionais em redes credais

Nesta seção é apresentada uma abordagem baseada em métodos variacionais para a resolução do problema de inferência exposto na Seção 3.4. Primeiramente, apresenta-se a aproximação de um conjunto credal por outro, através de técnicas variacionais de *campo médio*. Em segundo lugar, apresenta-se o algoritmo SV2U que aplica técnicas de aproximação de *campo médio estruturado* para realização de inferências em redes credais. O algoritmo SV2U gera intervalos aproximados, cuja precisão se equipara com a do algoritmo L2U em alguns casos, como demonstrado pelos resultados descritos no Capítulo 5.

Na Seção 4.3.1, a abordagem de técnicas variacionais em conjuntos credais é apresentada e um exemplo ilustrativo é proporcionado na Seção 4.3.2. Na Seção 4.3.3, o algoritmo SV2U é apresentado e um exemplo de sua aplicação é oferecido na Seção 4.3.4.

4.3.1 Abordagem baseada em conjuntos

O objetivo desejado é empregar métodos variacionais para realização de inferências em redes credais. Uma abordagem direta é a aproximação *baseada em vértices*: aplicar o método *de campo médio* para cada um dos vértices da coletânea de distribuições conjuntas; ou seja, aproximar individualmente cada um dos vértices obtidos pela extensão forte de conjuntos credais especificadas separadamente, ou simplesmente, conjuntos credais *locais* $K_P(H_j|pa_j)$. Porém, a aproximação baseada em vértices não soluciona a principal dificuldade da extensão forte, que é o crescimento exponencial do número de potenciais vértices.

Uma abordagem alternativa é focar em uma aproximação *baseada em conjuntos*. A meta é aproximar o conjunto credal original da distribuição conjunta por uma coletânea de intervalos de probabilidade $I_Q(H_j = h_j) = [\underline{Q}(H_j = h_j), \overline{Q}(H_j = h_j)]$ para cada um dos estados da variável H_j . A vantagem de se trabalhar com estes intervalos é que eles estão diretamente relacionados com as probabilidades inferiores e superiores $\underline{P}(H_j = h_j|E)$ e $\overline{P}(H_j = h_j|E)$. A aproximação baseada em conjuntos emprega as computações locais dos métodos de campo médio padrões, mas ao invés de computar valores pontuais, obtém valores intervalares. O resultado é um processo pelo qual iterativamente se calculam intervalos de probabilidade para todos os estados de todas as variáveis H_j .

Considere a Equação (2.13). No contexto de conjuntos credais, pode-se expressar os limites inferior/superior de $Q(H_j = h_j)$: $\underline{Q}(H_j = h_j)$ e $\overline{Q}(H_j = h_j)$, respectivamente por:

$$\begin{aligned} \underline{Q}_j^*(H_j = h_j) = \min & k_\lambda \exp\left(\sum_{H_i \in \{pa_j\}} \prod_i q_i(H_i) \ln p(H_j|pa_j)\right) \\ & + \sum_{k \in ch_j} \sum_{H_i \neq j \in \{X_k, pa_k\}} \prod_i q_i(H_i) \ln p(X_k|pa_k), \quad (4.3) \end{aligned}$$

$$\begin{aligned} \overline{Q}_j^*(H_j = h_j) = \max & k_\lambda \exp\left(\sum_{H_i \in \{pa_j\}} \prod_i q_i(H_i) \ln p(H_j|pa_j)\right) \\ & + \sum_{k \in ch_j} \sum_{H_i \neq j \in \{X_k, pa_k\}} \prod_i q_i(H_i) \ln p(X_k|pa_k), \quad (4.4) \end{aligned}$$

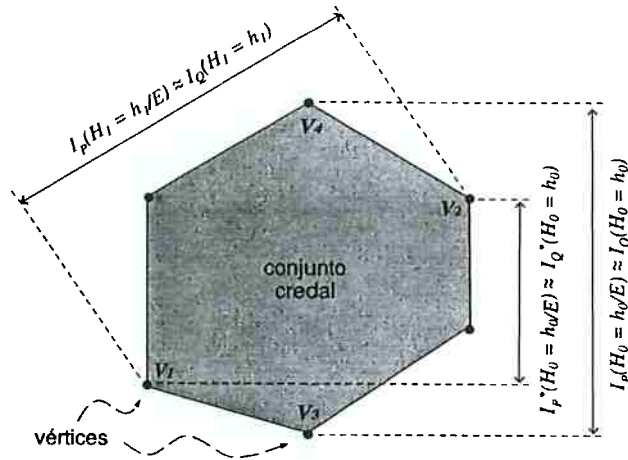


Figura 24: Aproximação de *limite externo*. O intervalo $I_Q^*(H_0 = h_0)$ é aproximado por uma borda externa $I_Q(H_0 = h_0)$.

onde:

$$\begin{aligned} q_i(H_i) &\in I_Q(H_i), \\ p(H_j|pa_j) &\in K_p(H_j|pa_j), \\ p(X_k|pa_k) &\in K_p(X_k|pa_k). \end{aligned}$$

Infelizmente, a computação exata do intervalo $I_Q(H_j = h_j)$ requer uma busca combinatorial global, pois os intervalos são todos dependentes entre si. Os vértices que fornecem os valores máximos e mínimos de $Q(H_j = h_j)$ não são necessariamente os mesmos que fornecem os máximos e mínimos de $Q(H_i = h_i)$ para $i \neq j$. Isso obriga o armazenamento de todas as combinações, cujo número cresce exponencialmente.

Propõe-se agora uma segunda aproximação, além da variacional, que reduz a complexidade combinatorial anteriormente discutida. A idéia é tomar os intervalos $I_Q(H_j = h_j)$ independentes um dos outros, reduzindo o problema à computação de intervalos aproximados. Aproximação que é conduzida de modo a manter a propriedade computacional de fato "local" desejada e fornecida pela aproximação variacional. Para tanto, os intervalos $I_Q(H_j = h_j)$ das Equações (4.3) e (4.4) são atualizados empregando intervalos externos de $I_Q(H_i = h_i)$ para $i \neq j$. Ou seja, na atualização da variável H_j , são utilizados os valores aproxi-

<p>Aproximação de campo médio baseado em conjuntos</p> <p>Entrada: rede credal C.</p> <p>Saída: aproximações das probabilidades inferior/superior da variável H_j.</p> <ol style="list-style-type: none"> 01. Inicializar os intervalos $I_Q(H = h) = [\underline{Q}^*(\mathbf{H}), \overline{Q}^*(\mathbf{H})]$, para todos os estados das variáveis ocultas \mathbf{H}. 02. Para cada variável oculta H_i, repetir até convergência: 03. Para cada um dos possíveis vértices V_i das Equações (4.3) e (4.4): 04. a) Computar $Q^*(H_i)$ para todos os estados de H_i e normalizar $Q^*(H_i = h_i)$; 05. b) Manter os valores máximos e mínimos de $Q^*(H_i = h_i)$ e atualizar o intervalo $I_Q(H_i = h_i)$. 06. Retornar $\underline{Q}(H_i = h_i)$ e $\overline{Q}(H_i = h_i)$ como aproximações de $\underline{P}(H_i = h_i E)$ e $\overline{P}(H_i = h_i E)$.

Figura 25: *Campo médio baseado em conjuntos*: uma aplicação do método variacional de campo médio para inferência em redes credais.

mados de $I_Q(H_i = h_i)$, computados para aqueles vértices em que $Q(H_j = h_j)$ é máximo e mínimo, ao invés dos vértices em que $Q(H_i = h_i)$ é máximo ou mínimo. Esta aproximação foi denominada *Limite Externo*.

A aproximação de *limite externo* é ilustrada na Figura 24. Suponha que se deseja aproximar o intervalo $I_P(H_0, H_1)$ por $I_Q(H_0, H_1)$, onde H_0 e H_1 são variáveis ocultas, usando métodos variacionais. Para se atualizar o intervalo $I_Q(H_1 = h_1)$ ⁸, é necessário o intervalo $I_Q^*(H_0 = h_0)$ dos vértices V_1 e V_2 relacionados aos valores extremos de $I_Q(H_1 = h_1)$. Entretanto, o intervalo que se possui é apenas daquele obtido pelos vértices V_3 e V_4 que conduzem para máximos e mínimos de $I_Q(H_0 = h_0)$ — toma-se tal intervalo como borda externa de $I_Q^*(H_0 = h_0)$. Note que tal sequência de aproximações ocorre em cada uma das iterações do método variacional baseado em conjuntos.

Assim, o método de campo médio baseado em conjuntos pode ser resumido em 2 etapas:

- **Etapa 1 - Aproximação de campo médio.** Conjunto credal original $\mathbf{K}_P(\mathbf{X})$, obtido pela extensão forte de conjuntos credais locais $K(H_i|pa_i)$, é aproximado por outro conjunto credal $\mathbf{K}_Q(\mathbf{X})$, obtido pela extensão forte de intervalos de probabilidade $I_Q(H_j = h_j) = [\underline{Q}(H_j = h_j), \overline{Q}(H_j = h_j)]$,

⁸Note que o intervalo $I_Q(H_1 = h_1)$ é uma aproximação de $I_P(H_1 = h_1|E)$, e assim por diante.

especificados para cada um dos estados da variável H_j . No caso, $\underline{Q}(H_j = h_j)$ aproxima o valor mínimo de $P(H_j = h_j|\mathbf{E})$ e $\overline{Q}(H_j = h_j)$, o valor máximo. Esta etapa é responsável pela computação local desejada — os vértices analisados estão restritos à extensão forte das variáveis pertencentes ao *Markov blanket* de H_j , desde que se considere a independência entre os intervalos; ponto crucial das Equações (4.3) e (4.4).

- **Etapa 2 - Aproximação de limite externo.** Para se atualizar $I_Q(H_j = h_j)$, pelas Equações (4.3) e (4.4), empregam-se intervalos aproximados externos $I_Q(H_i = h_i)$, onde $i \neq j$ — de modo a tornar os intervalos independentes entre si. Os intervalos externos são obtidos em iterações anteriores, em que outros vértices foram levados em conta. Assim se preserva a propriedade de computação local.

Resumindo, o conjunto credal original $\mathbf{K}_P(\mathbf{X})$ é aproximado por outro conjunto credal $\mathbf{K}_Q(\mathbf{X})$, especificado por intervalos locais $I_Q(H_j = h_j)$. Os valores extremos dos intervalos obtidos são exatamente aproximações dos valores buscados na inferência. O problema de inferência na rede credal original, que exige uma análise combinatória em escala global, fica reduzido a um problema combinatório local, pois: (1) A aproximação variacional de campo médio é baseado em um mecanismo de atualização “local” (Equação (2.13)), restrito ao *Markov blanket*; (2) A aproximação de *limite externo* torna cada etapa de atualização independente combinatorialmente das demais, assim garantindo a computação local.

A Figura 25 apresenta um algoritmo que implementa aproximações variacionais de campo médio para inferência em redes credais. Note que na Linha 03, os vértices V_i são obtidos através da extensão forte, envolvendo apenas variáveis pertencentes ao *Markov blanket* de H_i .

4.3.2 Exemplo: Campo médio baseado em conjuntos em rede com 2 nós

Suponha uma rede credal com 2 variáveis binárias ocultas A e B (Figura 26). Considere que os conjuntos credais associados são definidos por intervalos: $P(b) \in [0.4, 0.75]$, $P(a|b) \in [0.1, 0.2]$ e $P(a|\neg b) \in [0.3, 0.4]$. Deseja-se obter os valores mínimo e máximo da probabilidade $P(a)$. Das Equações (4.3) e (4.4),

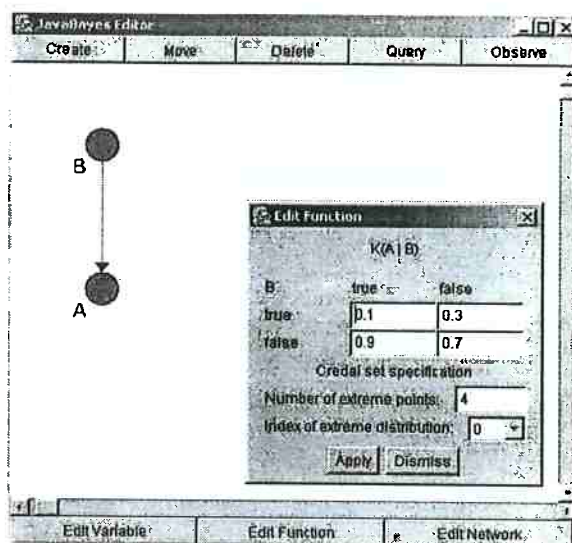


Figura 26: Rede credal com 2 nós associados a variáveis binárias (visualizada no sistema JavaBayes, pacote que permite representar conjuntos credais locais e extensão forte).

obtem-se as seguintes expressões de atualização para a variável A :

$$\underline{Q}^*(a) = \min k_\lambda \cdot \exp(Q^*(b) \cdot \ln[P(a|b)] + Q^*(\neg b) \cdot \ln[P(a|\neg b)]),$$

$$\overline{Q}^*(a) = \max k_\lambda \cdot \exp(Q^*(b) \cdot \ln[P(a|b)] + Q^*(\neg b) \cdot \ln[P(a|\neg b)]),$$

onde $Q^*(b) \in I(b) = [\underline{Q}^*(b), \overline{Q}^*(b)]$. A constante de normalização k_λ é obtida calculando-se para o mesmo vértice a razão:

$$\frac{1}{\exp(Q^*(b) \cdot \ln[P(a|b)] + Q^*(\neg b) \cdot \ln[P(a|\neg b)]) + \exp(Q^*(b) \cdot \ln[P(\neg a|b)] + Q^*(\neg b) \cdot \ln[P(\neg a|\neg b)])}$$

De modo análogo, para a variável B obtém-se:

$$\underline{Q}^*(b) = \min k_\lambda \cdot \exp(Q^*(a) \cdot \ln[P(a|b) \cdot P(b)] + Q^*(\neg a) \cdot \ln[P(\neg a|b) \cdot P(b)]),$$

$$\overline{Q}^*(b) = \max k_\lambda \cdot \exp(Q^*(a) \cdot \ln[P(a|b) \cdot P(b)] + Q^*(\neg a) \cdot \ln[P(\neg a|b) \cdot P(b)]),$$

onde: $Q^*(a) \in I(a) = [\underline{Q}^*(a), \overline{Q}^*(a)]$. Estas expressões são iterativamente atualizadas até convergirem. Note que no início, o intervalo $I(b)$ é inicializado com valores arbitrários, como por exemplo $[0.5, 0.5]$. Com esta inicialização, obtém-se ao final da iteração o intervalo $Q(a) \in [0.13, 0.31]$. Para termos de comparação, o intervalo exato é $P(a) \in [0.15, 0.32]$.

4.3.3 Algoritmo SV2U

Como descrita na Seção 2.6.4, a aproximação de campo médio é aperfeiçoada em sua forma *estruturada*. Apresenta-se aqui um algoritmo que implementa métodos de campo médio estruturado combinado com a abordagem baseada em conjuntos (Seção 4.3.1), o algoritmo *Structured Variational-2U* (SV2U). O SV2U usa a abordagem baseada em conjuntos para aproximar a rede credal binária multi-conectada por uma com estrutura poli-árvore, e executa o algoritmo 2U de propagação exata de intervalos (FAGIUOLI; ZAFFALON, 1998). Note que para variáveis binárias, o intervalo $I_Q = [Q, \bar{Q}]$, obtido pelas Equações (4.3) e (4.4)), corresponde ao conjunto credal local K_Q .

A idéia é desenvolver o campo médio estruturado descrito na Seção 2.6.4 para redes credais. Aplicando a abordagem baseado em conjunto para a Equação (2.13), obtém-se expressões análogas às apresentadas nas Equações (4.3) e (4.4):

$$\begin{aligned} \underline{Q}_j^*(X_j = x_j | pa_j) = \min & k_\lambda \exp\left(\sum_{k \in G_{X_j}} \sum_{l \in \{X \setminus c_j\}} \prod_l q_l(X_l | pa_l) \ln p(X_k | pa_k)\right. \\ & \left. - \sum_{i \in C_{X_j}} \sum_{l \in \{X \setminus c_j\}} \prod_l q_l(X_l | pa_l) \ln q_i(X_i | pa_i)\right), \quad (4.5) \end{aligned}$$

$$\begin{aligned} \bar{Q}_j^*(X_j = x_j | pa_j) = \max & k_\lambda \exp\left(\sum_{k \in G_{X_j}} \sum_{l \in \{X \setminus c_j\}} \prod_l q_l(X_l | pa_l) \ln p(X_k | pa_k)\right. \\ & \left. - \sum_{i \in C_{X_j}} \sum_{l \in \{X \setminus c_j\}} \prod_l q_l(X_l | pa_l) \ln q_i(X_i | pa_i)\right), \quad (4.6) \end{aligned}$$

onde:

$$\begin{aligned} q_l(X_l | pa_l) & \in K_Q(X_l | pa_l), \\ q_i(X_i | pa_i) & \in K_Q(X_i | pa_i), \\ p(X_k | pa_k) & \in K_P(X_k | pa_k). \end{aligned}$$

O algoritmo SV2U é apresentado na Figura 27. Suponha uma rede credal multi-conectada com variáveis binárias, \mathbf{C} , associada a uma coleção de distribuições conjuntas do tipo $p(X) = \prod_k p_k(g_k)$. Deseja-se aproximar \mathbf{C} por uma rede com estrutura poli-árvore \mathbf{C}_p associada a uma coleção de distribuições conjuntas

Algoritmo SV2U

Entrada: rede credal binária multi-conectada \mathbf{C} .

Saída: aproximações das probabilidades condicionais inferior/superior da variável X_j .

01. Selecionar um conjunto de condicionamento da rede original \mathbf{C} , com distribuição conjunta $p(X) = \prod_k p_k(g_k)$, para obter uma rede com estrutura aproximada poli-árvore \mathbf{C}_p .
02. Encontrar o conjunto de *clusters* $G = \{g_k\}$ da rede \mathbf{C} e $C = \{c_i\}$ da rede \mathbf{C}_p .
03. Inicializar para os conjuntos c_j que não pertencem a G : $K_Q(X_j = x_j|pa_j) = [0.5, 0.5]$ para todos os estados de pa_j .
04. Para os conjuntos c_j , repetir até convergência:
 05. Para todos os possíveis estados dos pais do conjunto c_j :
 06. a) Computar $\underline{Q}^*(X_j = x_j|pa_j)$ e $\overline{Q}^*(X_j = x_j|pa_j)$ para cada estado de X_j e normalizar, pelas Equações (4.5) e (4.6);
 07. b) Manter os valores mínimos e máximos de $Q^*(X_j = x_j|pa_j)$ para cada estado dos pais e atualizar $K_Q(X_j = x_j|pa_j)$.
08. Executar o algoritmo 2U na poli-árvore associada à distribuição conjunta $q(X) = \prod_k p_k(g_k) \times \prod_j q_j(c_j)$, onde k é o índice dos conjuntos que pertencem a ambas coleções G e C ; e $Q_j(c_j) \in K_Q(X_j = x_j|pa_j)$. Manter os valores mínimos e máximos das probabilidades das distribuições condicionais de X_j .

Figura 27: Descrição de implementação do algoritmo SV2U.

do tipo $q(X) = \prod_i q_i(c_i)$, selecionando um conjunto de condicionamento (Linha 01). A idéia é encontrar uma boa aproximação $q_i(c_i)$ das distribuições condicionais modificadas, empregando técnicas variacionais de campo médio. Para isso, em primeiro encontram-se o conjunto de $G = \{g_k\}$ de todos os *clusters* da rede \mathbf{C} e o conjunto $C = \{c_i\}$ de todos os *clusters* da rede \mathbf{C}_p . Os *clusters* c_j do conjunto C que não pertencem ao conjunto G são os que possuem suas distribuições modificadas (Linha 02). Os intervalos de $q(c_j)$ são atualizados conforme as Equações (4.5) e (4.6). Recorde que, conforme definido na Seção 2.6.4, $c_j = \{X_j, pa_j\}$, G_{X_j} é o conjunto dos *clusters* que dependem de X_j na rede \mathbf{C} e C_{X_j} é o conjunto dos *clusters* que dependem de X_j na rede \mathbf{C}_p . A atualização é feita de forma iterativa, como descrito na Seção 4.3.1, até se obter a convergência dos intervalos (Linhas 04-07). Durante estas iterações, os valores de $K_Q(X_j = x_j|pa_j) \in [\underline{Q}^*(X_j = x_j|pa_j), \overline{Q}^*(X_j|pa_j)]$ são computados. Ao final do processo iterativo, chega-se a uma poli-árvore associada a um conjunto de distribuições do tipo $q(X) = \prod_k p_k(g_k) \times \prod_j q_j(c_j)$, onde k é o índice dos *clusters* que pertencem a ambas coleções G e C ; e $q_j(c_j) = q_j(X_j|pa_j) \in K_Q(X_j|pa_j)$. Com esta

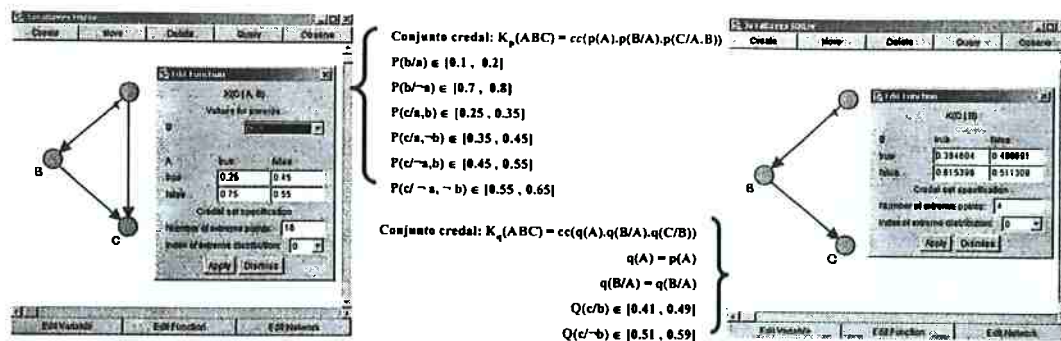


Figura 28: Esquerda: rede credal multi-conectada com 3 variáveis binárias. Direita: uma estrutura poli-árvore aproximada, diferindo no conjunto credal local associado à variável C .

estrutura aproximada poli-árvore, associada a um conjunto credal aproximado, computam-se os valores extremos do intervalo desejado, executando o algoritmo 2U (Linha 08).

Sobre a complexidade do algoritmo SV2U, pode-se afirmar que é dada fundamentalmente em função do número de variáveis n_{bk} contidas no *Markov blanket* G_{X_j} de X_j . Considerando as Equações (4.5) e (4.6), no pior caso (sem nenhuma simplificação das expressões), a combinação entre valores mínimos e máximos de $q_l(X_l|pa_l) \in K_Q(X_l|pa_l)$ exige uma busca entre $2^{n_{bk}}$ possibilidades, a somatória se dá sobre $2^{n_{bk}}$ diferentes estados das variáveis contidas em G_{X_j} e a combinação entre valores mínimos e máximos de $p(X_k|pa_k) \in K_P(X_k|pa_k)$ exige outra busca entre $2^{n_{bk}}$ possibilidades. Assim, a complexidade do algoritmo SV2U é da ordem de $O(2^{8n_{bk}})$. Dependendo de como a estrutura original é aproximada, n_{bk} pode levar a uma complexidade muito alta.

Sobre a convergência do algoritmo SV2U, infelizmente não se pode garantir nada, pois com a *aproximação de limite externo* (Seção 4.3.1), a distância KL não é mais minimizada monotonicamente.

Apesar da falta de garantias teóricas de convergência, resultados empíricos obtidos nos experimentos do Capítulo 5 mostram que o algoritmo SV2U converge em todos os casos testados.

4.3.4 Exemplo: SV2U em rede com 3 nós

Considere o enfoque de campo médio estruturado (Seção 4.3.3) em uma rede credal com 3 variáveis binárias A , B e C . A rede é mostrada na Figura 28. Para cada vértice da rede credal, tem-se a distribuição conjunta $p(A, B, C) = p(A).p(B|C).p(C|A, B)$ que é aproximada pela distribuição $q(A, B, C) = p(A).p(B|C).q(C|B)$ — a rede multi-conectada original é aproximada por uma poli-árvore, para a qual inferências exatas e eficientes são disponíveis através do algoritmo 2U (FAGIUOLI; ZAFFALON, 1998).

Neste exemplo, apenas a variável C precisa ser atualizada. Desenvolvendo a Equação (4.5), onde $G_{X_j} = \{g_B, g_C\} = \{C, A, B\}$ e $C_{X_j} = \{c_B\} = \{B, A\}$, obtém-se:

$$\begin{aligned} \underline{Q}^*(C = c|B) &\propto \min \sum_A p(A) \left[\sum_{k \in G_{X_j}} \ln p(X_k | pa_k) - \sum_{i \in C_{X_j}} \ln q_i(X_i | pa_i) \right] \\ &= \sum_A p(A) [\ln p(A) + \ln p(B|A) + \ln p(C = c|A, B) - \ln p(B|A)] \\ &= \sum_A p(A) \ln p(A) + \sum_A p(A) \ln p(C = c|A, B) \\ &= \sum_A p(A) \ln p(C = c|A, B), \end{aligned}$$

e analogamente:

$$\begin{aligned} \overline{Q}^*(C = c|B) &\propto \max \sum_A p(A) \left[\sum_{k \in G_{X_j}} \ln p(X_k | pa_k) - \sum_{i \in C_{X_j}} \ln q_i(X_i | pa_i) \right] \\ &= \sum_A p(A) \ln p(C = c|A, B). \end{aligned}$$

Destas expressões, dois intervalos de distribuições condicionais aproximadas $Q(C = c|b) \in [0.41, 0.49]$ e $Q(C = c|-b) \in [0.51, 0.59]$ são obtidos. Empregando este conjunto credal com os demais anteriores, pode-se inferir que $Q(C = c) \in [0.4262, 0.5513]$, enquanto que o resultado exato para esta inferência é $P(C = c) \in [0.428, 0.552]$.

5 EXPERIMENTOS E ANÁLISE COMPARATIVA ENTRE OS ALGORITMOS

Neste capítulo apresentam-se os diversos experimentos conduzidos com os algoritmos desenvolvidos no capítulo anterior. Primeiramente, na Seção 5.1, discute-se a dificuldade para se avaliar a precisão dos algoritmos aproximados em redes credais, e a metodologia adotada nos experimentos conduzidos. Os experimentos foram divididos em dois conjuntos: experimentos com redes credais geradas aleatoriamente (Seção 5.2) e com redes com estrutura conhecida (Seção 5.3). Na Seção 5.4, discutem-se comparativamente os resultados obtidos nos experimentos.

5.1 Metodologia dos experimentos conduzidos

Nesta seção, apresenta-se a metodologia seguida nos experimentos conduzidos nas seções posteriores. Os algoritmos desenvolvidos L2U, IPE e SV2U fornecem intervalos *aproximados* de inferência em redes credais. Deseja-se, portanto, medir a qualidade, ou seja, a precisão destes algoritmos. Três fatores devem ser considerados na discussão:

- Devido à complexidade computacional, os algoritmos exatos para inferência em redes credais existentes estão restritos ao processamento de redes com cerca de 40 nós e não muito conectadas (ROCHA, 2004) (CAMPOS, 2005).
- Não existe até o momento sistemas computacionais com algoritmos aproximados para redes credais, o que seria desejável para se comparar com os algoritmos desenvolvidos.

Nestas circunstâncias, adotaram-se os seguintes procedimentos para avaliar os algoritmos desenvolvidos:

- Os intervalos aproximados obtidos são comparados com intervalos exatos, quando os algoritmos exatos existentes são capazes de produzir intervalos exatos. Redes credais com até 37 nós são empregadas e intervalos exatos são obtidos através de métodos baseados em programação multilinear (CAMPOS, 2005). O tempo médio de processamento destes algoritmos varia entre 5 a 30 minutos.
- A qualidade das aproximações é medida pelo cálculo do erro médio entre o valor aproximado e o exato de inferência, como efetuado por exemplo em (ROCHA; COZMAN; CAMPOS, 2003). O *erro quadrático médio (MSE)* é empregado para se medir o erro médio, como proposto por Cheng e Druzdzel (2000) para situações em que a análise teórica dos erros é difícil. Seria desejável calcular a entropia relativa KL dos conjuntos credais originais e aproximados, mas isto é algo tão complexo quanto a realização de inferência, não sendo algo viável. Assim, emprega-se o *MSE* adaptado para redes credais com variáveis binárias, dado pela expressão:

$$\sqrt{\frac{1}{2n_{\mathbf{H}}} \sum_{X_i \in \mathbf{H}} \left((\underline{P}^*(x_i|\mathbf{E}) - \underline{P}(x_i|\mathbf{E}))^2 + (\overline{P}^*(x_i|\mathbf{E}) - \overline{P}(x_i|\mathbf{E}))^2 \right)},$$

onde, \mathbf{X} representa o conjunto de todas as variáveis da rede credal com variáveis binárias, \mathbf{E} representa o conjunto de todas as variáveis observadas, $\mathbf{H} = \{\mathbf{X} \setminus \mathbf{E}\}$ é o conjunto das variáveis não observadas, $n_{\mathbf{H}}$ é o número de elementos de \mathbf{H} e x_i denota o estado assumido pela variável X_i . Por fim, $\underline{P}^*(x_i|\mathbf{E})$ e $\overline{P}^*(x_i|\mathbf{E})$ são os valores extremos dos intervalos aproximados; $\underline{P}(x_i|\mathbf{E})$ e $\overline{P}(x_i|\mathbf{E})$ são os extremos dos intervalos exatos.

- A correlação entre os resultados aproximados e exatos são mostrados *graficamente*, como utilizado por Murphy, Weiss e Jordan (1999) para avaliação empírica do algoritmo LBP. Este procedimento serve para verificar visualmente a qualidade dos resultados aproximados.
- A complexidade da rede credal processada é quantificada pelo número de nós n e de arcos n_a . A razão $\frac{n_a}{n}$ é empregada para se quantificar a *densidade de arcos* na rede, como proposto por Draper (1995).
- Os algoritmos desenvolvidos são testados: a) Em redes geradas aleatoriamente (Seção 5.2), com o objetivo de medir o desempenho dos algoritmos

Tabela 4: Tabela contendo parâmetros de entrada do BNGenerator para geração de amostras de redes credais. $nNodes$ denota o número de nós das $nBNs$ redes geradas. $maxArcs$ é o número arcos da rede. $nval = 2$ denota variável binária e $structure = multi$ denota a estrutura multi-conectada das redes geradas.

Amostra	$nNodes$	$maxArcs$	$nBNs$	$nval$	$structure$
A_1	10	12	10	2	multi
A_2	10	14	10	2	multi
A_3	10	16	10	2	multi

na medida em que se aumenta a densidade da rede, e b) Em redes com estrutura conhecida (Seções 5.3), com o intuito de ilustrar o desempenho dos algoritmos em modelos utilizados em problemas reais e permitir que, posteriormente, outros autores possam realizar testes nestas mesmas redes;

- Adotou-se como critério de convergência: a invariância dos valores extremos computados em todas as variáveis, entre uma iteração e outra. A ordem de comparação entre os valores vai até a 16^a casa decimal, pois empregam-se variáveis da classe *double*.

5.2 Experimentos em redes credais geradas aleatoriamente

Nesta seção, apresentam-se os experimentos conduzidos com redes credais geradas aleatoriamente. Para tanto, empregou-se uma versão do programa *BN-Generator*¹ (IDE; COZMAN, 2004a) capaz de gerar redes credais binárias. As redes foram geradas conforme os parâmetros apresentados na Tabela 4. A amostra A_1 é constituída por dez redes credais com densidade de arcos $\frac{12}{10} = 1.2$. A amostra A_2 é constituída por dez redes credais com densidade 1.4. A amostra A_3 contém dez redes credais com densidade 1.6. O objetivo é observar o desempenho dos algoritmos na medida em que se aumenta a densidade das redes testadas.

Para cada rede credal da amostra, valores extremos do intervalo são calculados empregando os algoritmos L2U, IPE e SV2U. Denotando as variáveis de uma rede qualquer por $\{X_1, \dots, X_{10}\}$, os valores extremos computados são as probabilidades

¹Disponível pelo endereço: www.pmr.poli.usp.br/ltd/Software/BNGenerator.

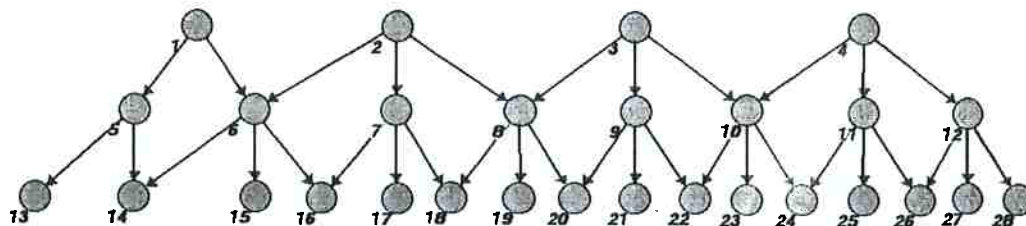


Figura 29: A rede *Pyramid*, grafo multi-camadas com 28 variáveis binárias, 34 arcos e densidade de arcos 1.21.

inferiores e superiores de $P(x_i)$, para $i = 1, \dots, 10$, ou seja, são as probabilidades em cada uma das variáveis, assumindo que não há nenhuma variável observada na rede. Assim, para cada rede e algoritmo, o erro quadrático médio é calculado pela expressão:

$$\sqrt{\frac{1}{2 \times 10} \sum_{i=1}^{10} \left((P^*(x_i) - P(x_i))^2 + (\bar{P}^*(x_i) - \bar{P}(x_i))^2 \right)}$$

Os erros quadráticos médios e os respectivos tempos de computação em cada uma das redes da amostra são apresentados nas Tabelas 5-7.

Testes também foram conduzidos com o algoritmo L2U em redes credais bastante largas e bem conectadas, obtendo-se resultados em tempos surpreendentes. Para uma rede credal aleatória com 50 nós, 150 arcos, variáveis binárias e com *induced-width* 10 (IDE; COZMAN, 2004a); obteve-se convergência em 13 iterações e com tempo de processamento de 8 minutos.²

5.3 Experimentos em redes credais com estrutura conhecida

Experimentos foram conduzidos nas mesmas estruturas de rede utilizadas no estudo empírico do algoritmo LBP (MURPHY; WEISS; JORDAN, 1999): as redes *Pyramid* e *Alarm*. A rede *Pyramid* (Figura 30) é um grafo com estrutura “multi-camada”, empregada em processamento de imagens, associado a variáveis binárias e com conexões locais entre camadas. A rede *Alarm* é um modelo proba-

²Até o momento, não há como medir a precisão do resultado, pois não há algoritmos exatos que possam processar redes com tal número de nós e densidade.

Erros quadráticos médios (MSE)*

Rede credal	L2U	IPE	SV2U
1	0,7131%	8,2574%	1,9507%
2	0,0054%	0,7918%	2,7330%
3	0,0406%	1,6784%	5,4049%
4	0,1198%	10,2149%	6,7856%
5	0,6121%	11,0043%	3,7832%
6	2,8565%	9,5533%	5,2619%
7	0,4878%	2,2474%	3,2498%
8	1,8167%	10,3117%	4,2084%
9	1,1173%	7,6517%	4,1408%
10	1,2776%	5,8203%	10,5682%

* Os MSE acima computam os erros quadráticos médios das inferências dos extremos inferiores e superiores de todas as variáveis, entre o intervalo exato e aproximado.

Tempos de processamento em milisegundos**

Rede credal	L2U	IPE	SV2U
1	172	8000	125
2	156	7875	110
3	156	7860	141
4	156	7828	985
5	203	8203	766
6	250	8156	1172
7	78	8234	1843
8	172	8172	78
9	93	8266	625
10	172	8235	1500

** Dados do processador
Intel®Pentium®M
1.70GHz, 480MB de RAM

Tabela 5: Resultados obtidos pelos algoritmos desenvolvidos, executando-os em redes credais binárias com 10 nós e 12 arcos (densidade de arcos 1.2).

Erros quadráticos médios (MSE)*

Rede credal	L2U	IPE	SV2U
1	2,0225%	14,9727%	9,8884%
2	2,0829%	19,5164%	3,8024%
3	0,3330%	24,3015%	8,2405%
4	1,0566%	12,0761%	4,0378%
5	0,5521%	19,6027%	12,1903%
6	1,3264%	19,5075%	2,6671%
7	1,2562%	13,4954%	15,7858%
8	1,5126%	19,5411%	15,2773%
9	0,3039%	16,4888%	3,8848%
10	1,4284%	19,9589%	1,0476%

* Os MSE acima computam os erros quadráticos médios das inferências dos extremos inferiores e superiores de todas as variáveis, entre o intervalo exato e aproximado.

Tempos de processamento em milisegundos**

Rede credal	L2U	IPE	SV2U
1	297	8390	171500
2	250	8203	922
3	172	8203	203
4	94	8375	94
5	172	8313	1938
6	172	8344	5078
7	203	8250	64391
8	172	8250	4672
9	188	8203	187
10	172	8219	203

** Dados do processador
Intel®Pentium®M
1.70GHz, 480MB de RAM

Tabela 6: Resultados obtidos pelos algoritmos desenvolvidos, executando-os em redes credais binárias com 10 nós e 14 arcos (densidade de arcos 1.4).

Erros quadráticos médios (MSE)*

Rede credal	L2U	IPE	SV2U
1	1,7854%	22,3723%	4,4399%
2	0,6300%	20,8745%	22,0326%
3	0,4125%	10,9228%	6,6156%
4	2,3425%	14,9065%	5,1510%
5	1,6495%	16,2039%	10,2713%
6	0,5526%	15,0865%	13,7073%
7	0,2232%	22,5234%	2,4251%
8	1,4155%	11,2333%	12,1138%
9	0,3502%	14,6524%	5,4794%
10	1,1407%	12,3767%	4,2201%

* Os MSE acima computam os erros quadráticos médios das inferências dos extremos inferiores e superiores de todas as variáveis, entre o intervalo exato e aproximado.

Tempos de processamento em milisegundos**

Rede credal	L2U	IPE	SV2U
1	94	8297	172
2	94	8375	17359
3	157	8219	828
4	203	8203	2953
5	109	8360	812360
6	188	8406	1281
7	188	8468	219
8	172	8437	1812
9	172	8328	1281
10	172	8406	250

** Dados do processador
Intel®Pentium®M
1.70GHz, 480MB de RAM

Tabela 7: Resultados obtidos pelos algoritmos desenvolvidos, executando-os em redes credais binárias com 10 nós e 16 arcos (densidade de arcos 1.6).

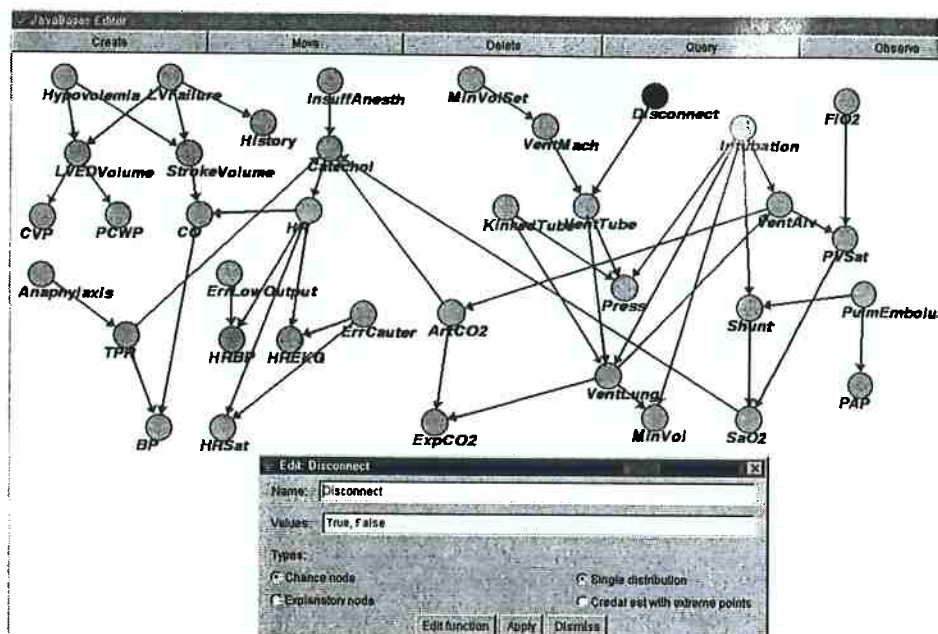


Figura 30: A rede *Alarm* possui 37 variáveis binárias, 46 arcos e densidade de arcos 1.24. O console mostra a variável “Disconnect” sendo editada.

bilístico clássico utilizado em diagnóstico médico. Para conduzir os experimentos nesta rede, algumas variáveis não-binárias da rede *Alarm* foram transformadas em variáveis binárias.

Para cada rede credal, valores extremos do intervalo foram computados empregando os algoritmos L2U, IPE e SV2U. Assumindo-se que não há variáveis observadas, as probabilidades inferiores e superiores de $P(x_i)$ foram computadas e erros quadráticos médios (MSE) foram calculados entre os valores aproximados e exatos.

Nas Seções 5.3.1, 5.3.2 e 5.3.3, apresentam-se os resultados obtidos empregando os algoritmos L2U, IPE e SV2U, respectivamente.

5.3.1 Testes com o algoritmo L2U

Esta seção apresenta os resultados obtidos com o algoritmo L2U. O algoritmo L2U converge após 4 iterações na rede *Pyramid*, e após 9 iterações na rede *Alarm*. O erro quadrático médio (MSE), entre os resultados aproximados e exa-

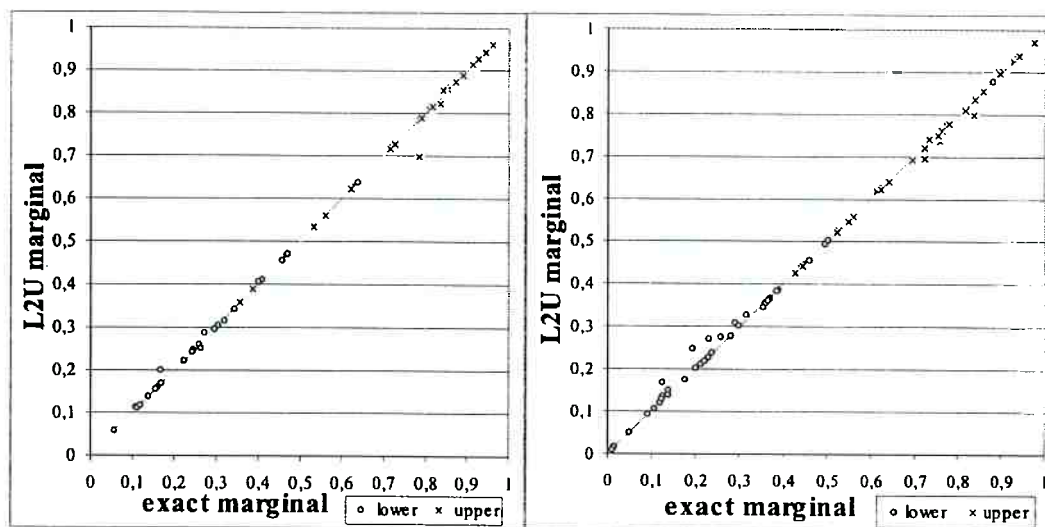


Figura 31: Correlação entre os valores extremos aproximados (obtidos pelo algoritmo L2U) e exatos, em dois exemplos: (a) rede *Pyramid*, com 28 variáveis binárias, sem evidências, e convergência em 4 iterações; (b) a rede *Alarm* “binarizada”, com 37 variáveis, sem evidências, e convergência em 9 iterações.

tos, das inferências em cada uma das variáveis da rede foi de apenas 1,3% para ambas as redes. Estes resultados podem ser visualizados na Figura 31. Deve-se ressaltar que o algoritmo L2U produz resultados aproximados muito rapidamente: inferências na rede *Alarm* (em todas as variáveis) foram produzidas em milissegundos com processador Pentium M 1.7GHz. Tal velocidade é consequência da propriedade polinomial do algoritmo 2U e do fato dele atingir a convergência em poucas iterações.

5.3.2 Testes com o algoritmo IPE

Esta seção apresenta os resultados obtidos com o algoritmo IPE. Valores extremos de inferência foram obtidos com o algoritmo IPE. A qualidade dos resultados pode ser visualizada na Figura 32; a maioria dos resultados são precisos, com a média de erro quadrático médio (MSE) de 5% para a rede *Pyramid* e 7.2% para rede *Alarm*. Estas redes são processadas em menos de 10 segundos com processador Pentium M 1.7 GHz. Note que para cada iteração, uma poli-árvore diferente é selecionada a partir da rede multi-conectada original, e diferentes intervalos externos são obtidos; e o resultado final é o intervalo obtido pela interseção dos intervalos obtidos anteriormente. Quanto maior o número de

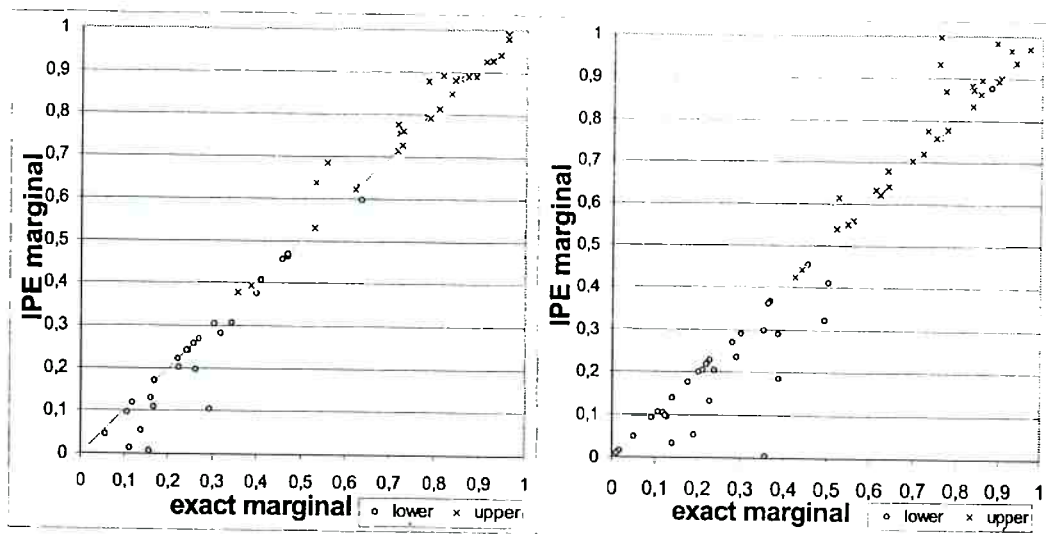


Figura 32: Correlação entre os valores extremos aproximados (obtidos pelo algoritmo IPE) e exatos, em dois exemplos: (a) rede *Pyramid*, com 28 variáveis binárias, sem evidências, 100 iterações e $MSE = 5.1\%$; (b) rede *Alarm* “binarizada”, com 37 variáveis, sem evidências, 100 iterações e $MSE=7.2\%$.

iteraões, melhor são os resultados.

Levanta-se aqui a questão: qual o número ideal de iterações para executar o algoritmo IPE? É uma questão que fica em aberto. O único cuidado que deve ser tomado é que o número máximo de iterações i_{max} não seja muito maior que o número de diferentes poli-árvores n_{dif} pois, neste caso, haveria uma *repetição* dos diferentes intervalos computados. Teoricamente, o número máximo de iterações i_{max} que pode ser utilizado depende do número de diferentes poli-árvores n_{dif} que podem ser obtidas a partir da rede multi-conectada original. O valor n_{dif} é função de cada estrutura específica, do número n de nós e n_a de arcos; valor da ordem³ de $\prod_{i=0}^{(n_a-n)} (n_a - i)$. Por exemplo, na rede *Alarm* que tem $n = 37$ nós e $n_a = 46$ arcos; $n_a - n + 1 = 10$ arcos precisam ser removidos para se obter uma poli-árvore. Deste modo, poderia ser obtido um número da ordem de $n_{dif} \cong \prod_{i=0}^9 (46 - i) = 1,48 \times 10^{16}$ diferentes poli-árvores. Na prática, como n_{dif} é um valor muito alto, emprega-se um valor i_{max} bem menor que retorna um resultado considerado satisfatório.

³Esta expressão é obtida analisando o número de diferentes modos em que $(n_a - n + 1)$ arcos podem ser removidos a partir do grafo com n nós, para se obter uma poli-árvore.

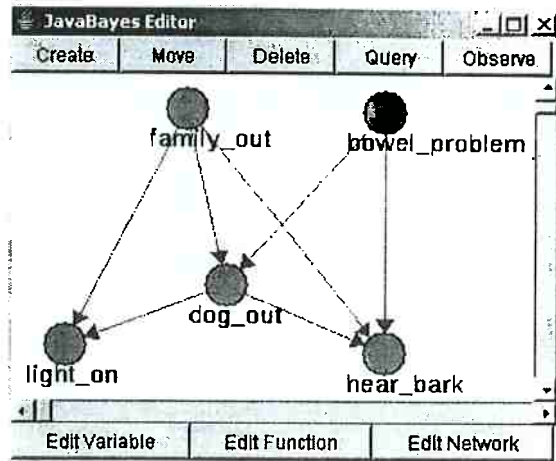


Figura 33: Rede *Multidog* (uma versão modificada da *Dog-Problem*), uma rede credal com 5 nós binários.

5.3.3 Testes com o algoritmo SV2U

Esta seção apresenta os resultados obtidos com o algoritmo SV2U. Além disso, um detalhamento maior de como se aplica o algoritmo na rede *Pyramid* (Figura 30) é apresentado. Mas antes, é interessante saber qual é o desempenho do algoritmo de campo médio *não estruturado*. O algoritmo apresentado na Figura 25 foi implementado no programa 2UBayes e alguns testes foram conduzidos. A precisão deste algoritmo é razoável, se levarmos em consideração que ele é um método *bruto*, no sentido em que ignora quaisquer dependências entre as variáveis (Seção 4.3.1). Por exemplo, na rede *Multidog* (Figura 33), comparando os resultados do campo médio com exatos, obtém-se um MSE de 7%; enquanto que o algoritmo SV2U (versão estruturada do campo médio) gera um MSE de 2.4%. Entretanto, para redes um pouco maiores, a sua execução se torna inviável, além de gerar resultados muito pouco precisos.

Testes com o algoritmo SV2U apontam para um desempenho muito melhor da abordagem de campo médio estruturado. Considere o algoritmo SV2U sendo aplicado na rede *Pyramid*. Note que para cada variável um conjunto credal binário é associado, formando um casco convexo do conjunto contendo todas as distribuições conjuntas que fatoram-se como $\prod_i p(X_i|pa_i)$, $i = \{1, \dots, 28\}$,

onde cada distribuição condicional $p(X_i|pa_i)$ é selecionada do conjunto credal local $K_P(x_i|pa_i) = [\underline{P}, \overline{P}]$, para cada estado dos pais pa_i . Um possível conjunto de condicionamento é formado pelos arcos (1,6), (2,6), (2,8), (3,8), (3,10), (4,10) e (4,12). Removendo estes arcos, obtém-se a poli-árvore que difere da rede original pelos conjuntos credais locais associados às variáveis X_6 , X_8 , X_{10} e X_{12} . Isto significa que os conjuntos credais locais $K_p(X_6|X_1, X_2)$, $K_p(X_8|X_2, X_3)$, $K_p(X_{10}|X_3, X_4)$ e $K_p(X_{12}|X_4)$ são aproximados por $K_q(X_j)$, $j = \{6, 8, 10, 12\}$. Assumindo que não há variáveis observadas, as expressões de atualização das Equações (4.5) e (4.6) são:

$$\begin{aligned}\ln q^*(X_6) &\propto \sum_{X_1, X_2} p(X_1)p(X_2) \ln p(X_6|X_1, X_2), \\ \ln q^*(X_8) &\propto \sum_{X_2, X_3} p(X_2)p(X_3) \ln p(X_8|X_2, X_3), \\ \ln q^*(X_{10}) &\propto \sum_{X_3, X_4} p(X_3)p(X_4) \ln p(X_{10}|X_3, X_4), \\ \ln q^*(X_{12}) &\propto \sum_{X_4} p(X_4) \ln p(X_{12}|X_4).\end{aligned}$$

Destas equações, obtém-se valores mínimos e máximos dos conjuntos credais locais: $K_Q(x_6) = [0.099, 0.346]$, $K_Q(X_8 = x_8) = [0.203, 0.664]$, $K_Q(X_{10} = x_{10}) = [0.278, 0.753]$ e $K_Q(X_{12} = x_{12}) = [0.532, 0.810]$. Executando o algoritmo 2U na poli-árvore associada a este novo conjunto modificado de conjuntos credais, obtém-se probabilidades inferior/superior aproximadas; e neste caso o erro quadrático médio (MSE), entre os resultados fornecidos pelo SV2U e os exatos, é de 2%. O resultado pode ser visualizado na Figura 34.

Na rede *Alarm*, obteve-se um MSE de 2,9%. Os resultados obtidos podem ser visualizados na Figura 35.

5.4 Discussão dos resultados

Nesta seção, discutem-se os resultados obtidos nos experimentos. Como mencionado inicialmente, o objetivo destes experimentos é mostrar comparativamente o desempenho dos algoritmos desenvolvidos. Os fatores analisados são: a precisão

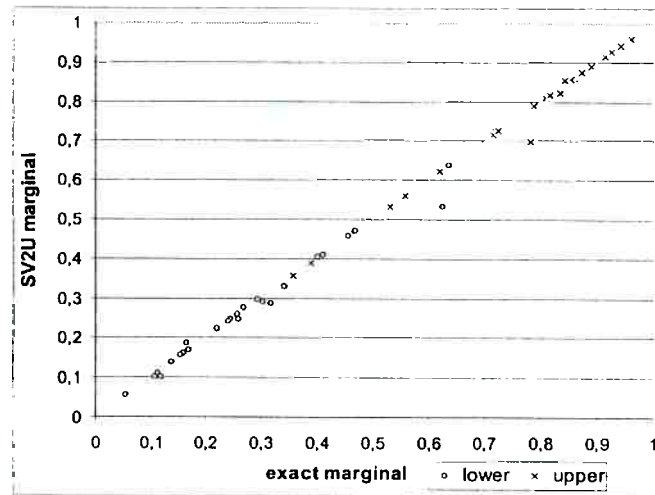


Figura 34: Correlação entre os valores extremos aproximados (obtidos pelo algoritmo SV2U) e exatos na rede *Pyramid*. As probabilidades inferior/superior de todas as variáveis são computadas de modo exato e aproximado, obtendo-se MSE=2%, em 0.078 segundos.

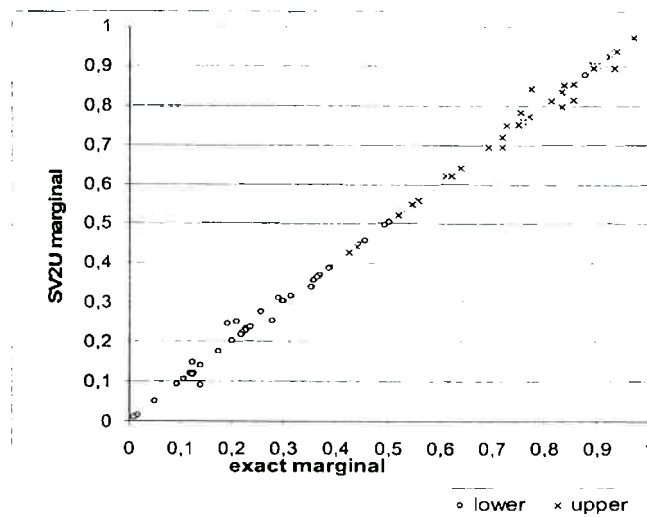


Figura 35: Correlação entre os valores extremos aproximados (obtidos pelo algoritmo SV2U) e exatos na rede *Alarm* “binarizada”. As probabilidades inferior/superior de todas as variáveis são computadas de modo exato e aproximado, obtendo-se MSE = 2.9%, em 0.422 segundos.

Tabela 8: MSE e tempo médio de processamento (em segundos) obtidos pelos algoritmos no conjunto de redes geradas aleatoriamente. O símbolo d denota a densidade das redes.

Algoritmo	L2U	IPE	SV2U
redes com $d = 1.2$ (tempo médio)	0.9% (0.2 seg)	6.8% (8.0 seg)	4.8% (0.7 seg)
redes com $d = 1.4$ (tempo médio)	1.2% (0.2 seg)	18.9% (8.3 seg)	7.7% (25 seg)
redes com $d = 1.6$ (tempo médio)	1.1% (0.2 seg)	16.1% (8.3 seg)	8.6% (83 seg)

Tabela 9: Média dos MSE e tempos de processamento obtidos pelos algoritmos em redes com estrutura conhecida.

Algoritmo	L2U	IPE	SV2U
rede Pyramid (tempo médio)	1,3% (0.13 seg)	5,0% (5.6 seg)	2,0% (0.08 seg)
rede Alarm (tempo médio)	1,3% (0.16 seg)	7,2% (7.2 seg)	2,9% (0.42 seg)

(medida pelo MSE) e o tempo de processamento⁴ exigido. Também é possível chegar às conclusões em termos de convergência.

As médias dos MSE e dos tempos de processamento obtidas nas redes credais geradas aleatoriamente são apresentadas na Tabela 8. O algoritmo L2U é definitivamente o que apresenta os melhores resultados: menores erros e tempos de processamento. Entretanto, neste algoritmo não há garantias teóricas de que os seus resultados convergem para a resposta certa. Verifica-se também que o seu desempenho é invariante em relação à densidade das redes processadas. O algoritmo IPE é o que apresenta os piores resultados. Entretanto, este algoritmo proporciona garantia teórica de que seus resultados sempre contém os intervalos corretos (**Teorema 1** da Seção 4.2). O algoritmo SV2U apresenta um desempenho intermediário e verifica-se que sua precisão e o seu tempo de processamento incrementam consideravelmente com o aumento da densidade.

Como observado nas Tabelas 5-7, há uma grande variação dos MSE dos algoritmos IPE e SV2U, ou seja, a qualidade de suas aproximações depende muito da estrutura da rede processada. Para redes com a mesma densidade, por exemplo $d = 1.2$ (Tabela 5), os algoritmos IPE e SV2U geram resultados com MSE variando entre 1% e 11% — fato intrigante que sugere posteriores investigações.

Na Tabela 9 é sumarizada os MSE obtidos pelos algoritmos nas redes credais com estrutura conhecida: *Pyramid* (densidade 1.21) e *Alarm* (densidade 1.24). Observa-se que o desempenho dos algoritmos nestas redes está de acordo com a média dos MSE obtidos nas redes geradas aleatoriamente com densidade 1.2. O que corrobora com a hipótese de que a densidade de arcos é uma característica da rede que influencia no desempenho dos algoritmos IPE e SV2U.

Com relação ao tempo de processamento, algumas observações devem ser feitas. Apesar do algoritmo L2U e IPE apresentarem complexidades semelhantes, da ordem $O(2^{2n_{pa}})$, como discutido no final das Seções 4.1.1 e 4.2.1; o algoritmo L2U é muito mais rápido, pois este algoritmo converge em menos de 5 iterações nos casos testados, enquanto que o algoritmo IPE é executado com 100 iterações para se obter resultados mais precisos. O algoritmo SV2U possui tempos de processamento com grande oscilação. Isto se explica pelo fato de que sua comple-

⁴Todos os experimentos são obtidos em computador com processador Pentium M 1.7 GHz com 480 MB de RAM.

xidade de $O(2^{8n_{bk}})$ (discutido na Seção 4.3.3) varia exponencialmente em função do número de variáveis n_{bk} na *Markov blanket*⁵ que, por sua vez, depende muito da estrutura aproximada escolhida. Nos experimentos conduzidos, observa-se que o algoritmo SV2U converge, em todos os casos, em menos de 30 iterações; mesmo que para que isso ocorra, ele exige um considerável tempo de processamento.

Finalmente, toda a análise comparativa, entre os algoritmos L2U, IPE e SV2U, apresentada até o momento é sumarizada na Tabela 10. As seguintes observações, por coluna da tabela, devem ser feitas:

- **Observação 1.** As qualificações de precisão: “alta”, “média” e “baixa” foram empregadas simplesmente para *comparar* a precisão dos algoritmos entre eles.
- **Observação 2.** Os algoritmos L2U e SV2U não fornecem garantias teóricas de convergência de seus resultados. O algoritmo IPE garante que seus intervalos contém o intervalo exato.
- **Observação 3.** As constantes: k_{L2U} , k_{IPE} e k_{SV2U} denotam os números de iterações exigidas. Verifica-se empiricamente que: k_{L2U} é bem menor que k_{IPE} e k_{SV2U} é próximo de k_{L2U} . O símbolo n_{pa} denota o número de pais e n_{bk} o número de nós contidos no *Markov Blanket* de um nó. Em grafos multi-conectados em geral, n_{bk} é muito maior que n_{pa} .
- **Observação 4.** Por *velocidade* entende-se *menor tempo de processamento* e por *precisão* entende-se *menor MSE*. A abordagem introduzida pelo SV2U — a aplicação de métodos variacionais em conjuntos credais — proporciona a possibilidade de manusear variáveis contínuas em redes credais.
- **Observação 5.** A aproximação variacional de conjuntos credais é a componente do algoritmo SV2U que requer maior esforço computacional — o valor $2^{8n_{bk}}$ pode ser muito alto, dependendo da estrutura da rede aproximada.

⁵Em geral, n_{bk} é muito maior que o número de pais de uma variável, pois o *Markov blanket* contém não só os pais, mas também os filhos e os pais dos filhos.

Algoritmo	Precisão	Garantia teórica de convergência	Complexidade	Principal vantagem	Principal desvantagem
L2U	Alta	Não há	$O(k_{L2U} \cdot 2^{2n_{pa}})$	Velocidade e precisão	Sem garantia teórica de convergência
IPE	Baixa	Aproximação externa (Teorema 1)	$O(k_{IPE} \cdot 2^{2n_{pa}})$	Garantia teórica de aproximação externa	Imprecisão dos resultados
SV2U	Média	Não há	$O(k_{SV2U} \cdot 2^{8n_{bk}})$	Possibilidade em estender o método para variáveis contínuas	Alta complexidade na aproximação variacional de conjuntos credais
	Observação 1.	Observação 2.	Observação 3.	Observação 4.	Observação 5.

Tabela 10: Análise comparativa entre os algoritmos L2U, IPE e SV2U. A notação foi simplificada na tabela, por questões de espaço. Ver as respectivas observações para maiores detalhes.

6 CONSIDERAÇÕES FINAIS

Neste último capítulo são apresentadas as considerações finais no que se refere ao que foi realizado e ao que pode ser realizado em trabalhos futuros. Espera-se que este trabalho sirva como ponto de partida de um próspero caminho de pesquisa e desenvolvimento em redes credais.

Considerações com relação ao que foi realizado nesta tese são resumidas em:

- **Desenvolvimento de três novos métodos para inferência aproximada em redes credais.** Três algoritmos foram desenvolvidos: L2U, IPE e SV2U — aproveitando-se da existência de um eficiente algoritmo exato de propagação de intervalos (2U) em redes credais binárias (Capítulo 4). Os algoritmos processam apenas redes credais binárias — entretanto, as idéias centrais dos métodos de aproximação empregados nos algoritmos L2U, IPE e SV2U não estão limitadas a apenas redes binárias, e podem ser estendidas para redes mais genéricas, condicional ao desenvolvimento de eficientes algoritmos de propagação de mensagens em poli-árvores.
- **Realização de experimentos empíricos.** Inúmeros experimentos foram realizados seguindo uma metodologia elaborada no trabalho (Capítulo 5). Reconhecendo o “ponto fraco” dos algoritmos apresentados: *a falta de garantidas teóricas sólidas de convergência* (exceto para o algoritmo IPE), é inegável a eficiência destes algoritmos capazes de realizar inferências em redes credais binárias em questão de milissegundos, com baixas taxas de erro em muitos casos. Outros algoritmos mais recentes existentes funcionam com tempos de processamento na ordem de minutos (CAMPOS, 2005).
- **Implementação dos algoritmos desenvolvidos.** Os algoritmos foram implementados em linguagem Java e um programa, denominado 2UBayes (Apêndice I), contendo os algoritmos está disponível no site do *Laboratório*

de Tomada de Decisões da Escola Politécnica da USP pelo endereço: <http://www.pmr.poli.usp.br/ltd/>. Também está disponível o conjunto de redes utilizado nos experimentos.

Com relação a cada um dos algoritmos desenvolvidos, deve-se ressaltar que:

- O algoritmo L2U é particularmente atrativo para aplicações práticas, em redes credais de maior complexidade, mesmo que até o presente momento não disponha de uma análise sólida de convergência — pela sua velocidade de processamento e precisão de suas respostas, como averiguado empiricamente nos experimentos. Redes credais binárias encontram promissora aplicação em redes credais relacionais (COZMAN *et al.*, 2004).
- O algoritmo IPE, apesar de sua limitação de desempenho, pode ser sempre usado como aquele que fornece resultados intervalares que garantidamente contém o intervalo exato.
- O algoritmo SV2U é resultado de um pioneiro desenvolvimento realizado, aplicando explicitamente métodos variacionais para inferência aproximada em redes credais. Aproximações adicionais, além do variacional, foram propostas para se garantir a continuidade da propriedade computacional local. Devido à generalidade e flexibilidade dos métodos variacionais, o enfoque baseado em conjunto apresentado neste trabalho é promissor para redes credais com variáveis contínuas. Modelos que contém variáveis contínuas, como por exemplo *Imprecise Dirichlet Model* (WALLEY, 1991), cujos conjuntos credais são definidos por um número bem maior de parâmetros, são intratáveis para os algoritmos existentes — e a princípio poderiam ser tratados por métodos variacionais.

As considerações em relação aos trabalhos futuros¹ são as seguintes:

¹Quando se mencionam os *trabalhos futuros*, na verdade, estamos falando de tudo aquilo que não foi feito no trabalho. Assim, aparentemente é melhor não falar deles. Mas, na realidade, só é possível fazer estas considerações de trabalhos futuros, quando algo já foi conquistado. É como um excursionista que, ao chegar no topo de um pico de 3000 metros, após um longo esforço, descobre que há outro pico ainda mais alto! Portanto, falar de trabalhos futuros é algo meritório, é falar que o pico de 3000 metros foi atingido. Dar um ponto final ao trabalho é algo importante que faz parte do próprio trabalho — como em uma boa peça de teatro que termina com genialidade. Não há genialidade nesta tese apresentada, mas um ponto final oportuno foi colocado para que, depois venham futuros parágrafos que irão compor toda a obra literária.

- **Estudo aprofundado de convergência do algoritmo L2U.** Resultados muito positivos obtidos pelo algoritmo L2U, demonstrados empiricamente no Capítulo 5, sugerem uma intrigante investigação das razões de sua convergência. No caso de redes Bayesianas binárias, o algoritmo LBP converge para respostas certas? No caso de redes credais binárias, o algoritmo L2U converge?
- **Aplicação do algoritmo L2U para inferência em redes credais não-binárias.** Um trabalho de transformação de redes credais não-binárias para binárias, para posterior aplicação do algoritmo L2U, está em andamento junto ao pesquisador do *Instituto Dalle Molle di Studi sull'Intelligenza Artificiale* (IDSIA), Alessandro Antonucci, da equipe do Prof. Marco Zaffalon. A idéia é aproximar os vértices da rede credal não-binária por vértices gerados a partir de redes credais binárias, e depois executar o algoritmo L2U nesta rede credal binária aproximada.
- **Extensão do estudo de técnicas variacionais em conjuntos credais.** A abordagem variacional feita em conjuntos credais, apresentada na Seção 4.3, é somente um ponto inicial para futuras pesquisas. Uma extensão da abordagem variacional em conjuntos credais pode ser feita com outras medidas de distância entre conjuntos convexos de probabilidade como apresentado por Abellan (2003).
- **Estudo das estruturas de aproximação do algoritmo SV2U.** Verifique-se que o algoritmo SV2U gera resultados precisos em certos casos, como por exemplo na rede credal 10 da Tabela 6, onde $MSE=1.05\%$. Estes resultados sugerem um estudo aprofundado da forma estruturada ótima de aproximação.

REFERÊNCIAS

- ABELLAN, J. *Medidas de Entropía y Distancia en Conjuntos Convexos de Probabilidad: Definiciones y Aplicaciones*. Tese (Doutorado) — Universidad de Granada, Granada, Espanha, 2003.
- ANTONUCCI, A.; SALVETTI, A.; ZAFFALON, M. Hazard assessment of debris flows by credal networks. In: LÓPEZ-DÍAZAND, M. *et al.* (Ed.). *Proceedings of the Second International Conference on Soft Methods in Probability and Statistics*, 2004. p. 125–132.
- BAESENS, B. *et al.* Bayesian network classifiers for identifying the slope of the customers lifecycle of long-life customers. In: *European Journal of Operational Research*, 2004. v. 156, n. 2, p. 508–523.
- BERGER, J. O. *Statistical Decision Theory and Bayesian Analysis*. Segunda ed. New York: Springer-Verlag, 1985.
- CAMPELLO, R. E.; MACULAN, N. *Algoritmos e Heurísticas*. Rio de Janeiro: EDUFF, 1994.
- CAMPOS, C. P. de. *Redes Credais e Qualitativas: Complexidade e Algoritmos*. Tese (Doutorado) — Departamento de Engenharia Mecatrônica, Escola Politécnica, Universidade de São Paulo, São Paulo, SP, Brasil, 2005.
- CAMPOS, C. P. de; COZMAN, F. G. Inference in credal networks using multilinear programming. In: *Proc. of the Starting Artificial Intelligence Researchers Symposium, STAIRS 2004, Valencia - Spain*. Amsterdam - Holanda: IOS Press, 2004. p. 50–61.
- CANO, A.; MORAL, S. A genetic algorithm to approximate convex sets of probabilities. *Proceedings of the Six International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU-96)*, v. 2, p. 847–852, 1996.
- CANO, A.; MORAL, S. A review of propagation algorithms for imprecise probabilities. In: *Proc. of 1st International Symposium on Imprecise Probabilities and their Applications*, 1999.
- CANO, A.; MORAL, S. Computing intervals of probabilities with simulated annealing and probability trees. *Lecture Notes in Computer Science*, v. 2143, p. 278–289, 2001. ISSN 0302-9743.
- CANO, A.; MORAL, S. Using probability trees to compute marginals with imprecise probabilities. *International Journal of Approximate Reasoning*, v. 29, p. 1–46, 2002.

- CANO, J.; DELGADO, M.; MORAL, S. An axiomatic framework for propagating uncertainty in directed acyclic networks. In: *International Journal of Approximate Reasoning*, 1993. v. 8, p. 253–280.
- CASTILLO, E.; GUTIERREZ, J. M.; HADI, A. S. *Expert Systems and Probabilistic Network Models*. 1. ed. New York: Springer, 1997. ISBN 0-387-94858-9.
- CHARNIAK, E. Bayesian networks without tears. In: *AI Magazine*, 1991. v. 12, p. 5–63.
- CHARTRAND, G.; OELLERMANN, O. R. *Applied and Algorithmic Graph Theory*. New York: McGraw-Hill, 1993.
- CHENG, J.; DRUZDZEL, M. Computational investigation of low-discrepancy sequences in simulation algorithms for Bayesian networks. In: BOUTILIER, C.; GOLDSZMIDT, M. (Ed.). *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence UAI-00*. San Francisco: Morgan Kaufmann Publishers, 2000. p. 72–81.
- COWELL, R. G. *et al.* *Probabilistic Networks and Expert Systems*. New York: Springer-Verlag, 1999. 256 p. (Statistics for Engineering and Information Science).
- COZMAN, F. G. Credal networks. *Artificial Intelligence*, v. 120, n. 2, p. 199–233, 2000.
- COZMAN, F. G. *et al.* Propositional and relational Bayesian networks associated with imprecise and qualitative probabilistic assessments. In: *20th Annual Conference on Uncertainty in Artificial Intelligence*. San Francisco: Morgan Kaufmann, 2004. p. 104–111.
- DARWICHE, A. A differential approach to inference in Bayesian networks. In: BOUTILIER, C.; GOLDSZMIDT, M. (Ed.). *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI-00)*. San Francisco: Morgan Kaufmann Publishers, 2000. p. 123–132.
- DECHTER, R. Bucket elimination: An unifying framework for probabilistic inference. In: HORVITZ, E.; JENSEN, F. (Ed.). *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence UAI-96*. San Francisco: Morgan Kaufmann Publishers, 1996. p. 211–219. ISBN 1-55860-412-X.
- DRAPER, D. L. *Localized Partial Evaluation of Belief Networks*. Tese (Doutorado) — Dept. of Computer Science, University of Washington, Washington, WA, 1995.
- DRAPER, D. L.; HANKS, S. Localized partial evaluation of belief networks. In: *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*. Seattle, WA: Morgan Kaufmann, 1994.
- EDELSBRUNNER, H. *Algorithms in Computational Geometry*. Berlin: Springer-Verlag, 1987.

- FAGIUOLI, E.; ZAFFALON, M. 2U: An exact interval propagation algorithms for polytrees with binary variables. *Artificial Intelligence*, v. 106, p. 77-107, 1998.
- FINE, T. L. Lower probability models for uncertainty and nondeterministic processes. *Journal of Statistical Planning and Inference*, v. 20, p. 389-411, 1988.
- FRIEDMAN, N. *et al.* Using Bayesian networks to analyze expression data. In: *Proc. 4th Int. Conf. on Computational Molec. Biol.*, 2000. p. 127-135.
- FUNG, R.; CHANG, K. C. Weighting and integrating evidence for stochastic simulation in Bayesian networks. In: *Proceedings of the Fifth Conference on Uncertainty in Artificial Intelligence UAI-89*. Windsor, Ontario: Morgan Kaufmann, 1989.
- GILKS, W. R.; RICHARDSON, S.; SPIEGELHALTER, D. J. *Markov Chain Monte Carlo in Practice.*, Chapman&Hall, 1996.
- GIRON, F. J.; RIOS, S. Quasi-Bayesian behaviour: a more realistic approach to decision making? In: BERNARDO, J. M. *et al.* (Ed.). *Bayesian Statistics*. Valencia: Valencia University Press, 1980. v. 1, p. 17-38 e 49-66.
- HENRION, M. Propagation of uncertainty in Bayesian networks by probabilistic logic sampling. In: LEMMER, J. F.; KANAL, L. N. (Ed.). *Uncertainty in Artificial Intelligence 2*. Amsterdam, London, New York: Elsevier/North-Holland, 1988. p. 149-163.
- IDE, J. S.; COZMAN, F. G. Generating random Bayesian networks with constraints on induced width. In: *Proc. the 16th European Conference on Artificial Intelligence (ECAI-04), Valencia - Spain*. Amsterdam - Holanda: IOS Press, 2004. p. 323-327.
- IDE, J. S.; COZMAN, F. G. IPE and L2U: Approximate algorithms for credal networks. In: *Proc. of the Starting Artificial Intelligence Researchers Symposium, STAIRS 2004, Valencia - Spain*. Amsterdam - Holanda: IOS Press, 2004. p. 118-127.
- IDE, J. S.; COZMAN, F. G. Approximate inference in credal networks by variational mean field methods. In: *4th International Symposium on Imprecise Probabilities and Their Applications*. Pittsburgh, Pennsylvania: Brightdocs, 2005. p. 203-212.
- IDE, J. S.; COZMAN, F. G. Set-based variational methods in credal networks: the SV2U algorithm. In: *Proc. of XXV Congresso da Sociedade Brasileira de Computação*. São Leopoldo, Rio Grande do Sul, Brasil: Ed. Ana Cristina Garcia and Fernando Osório, 2005. V Encontro Nacional de Inteligência Artificial, p. 872-881.

- IHLER, J. F. A.; WILLSKY, A. Message errors in belief propagation. *Advances in Neural Information Processing Systems*, MIT Press, v. 17, 2005.
- JAAKKOLA, T. S. Tutorial on variational approximation methods. *Advanced mean field methods: theory and practice*, MIT Press, 2000.
- JENSEN, F. V. *An Introduction to Bayesian Networks*. New York: Springer-Verlag, 1996.
- JORDAN, M. I.; GHAHRAMANI, Z.; JAAKKOLA, T. S. An introduction to variational methods for graphical models. In: JORDAN, M. I. (Ed.). *Learning Graphical Models*, Kluwer Academic Publishers, 1999.
- KJAERULFF, U. Reduction of computational complexity in Bayesian networks through removal of weak dependences. In: *Proceedings of the Tenth Conference on Uncertainty in AI*, 1994. p. 374–382.
- KRIEGLER, E.; HELD, H. Climate projections for the 21st century using random sets. In: *ISIPTA*, 2003. p. 345–359.
- LEVI, I. *The Enterprise of Knowledge*. Cambridge, Massachusetts: MIT Press, 1980.
- LUKASIEWICZ, T. Probabilistic logic programming. In: *European Conference on Artificial Intelligence*, 1998. p. 388–392.
- LUO, C. *et al.* Computation of best bounds of probabilities from uncertain data. *Computational Intelligence*, v. 12, n. 4, p. 541–566, 1996. ISSN 0824-7935.
- MADSEN, A. *et al.* Applications of probabilistic graphical models to diagnosis and control of autonomous vehicles. In: *The Second Bayesian Modeling Applications Workshop (durante UAI-2004)*, 2004.
- MCELIECE, R. J.; MACKAY, D. J. C.; CHENG, J. F. Turbo decoding as an instance of Pearl's 'belief propagation' algorithm. *IEEE Journal on Selected Areas in Communication*, v. 162, n. CSD-99-1046, p. 140–152, 1998.
- MURPHY, K. P.; WEISS, Y.; JORDAN, M. I. Loopy belief propagation for approximate inference: An empirical study. In: *In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, 1999. p. 467–475.
- NILSSON, N. J. Probabilistic logic. In: *Artificial Intelligence*, 1986. v. 28, p. 71–87.
- PARISI, G. *Statistical Field Theory*. MA: Addison-Wesley, 1988.
- PEARL, J. *Probabilistic Reasoning in Intelligent Systems*, Morgan-Kaufman, 1988.

- QUAEGHEBEUR, E.; COOMAN, G. de. Imprecise probability models for inference in exponential families. In: *4th International Symposium on Imprecise Probabilities and Their Applications*. Pittsburgh, Pennsylvania: Brightdocs, 2005.
- ROCHA, J. C. da. *Algoritmos para Inferência em Redes Credais*. Tese (Doutorado) — Departamento de Engenharia Mecatrônica, Escola Politécnica, Universidade de São Paulo, São Paulo, SP, Brasil, 2004.
- ROCHA, J. C. da; COZMAN, F. G. Inference with separately specified sets of probabilities in credal networks. In: *Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA: Morgan Kaufmann, 2002. p. 430–437.
- ROCHA, J. C. da; COZMAN, F. G.; CAMPOS, C. P. de. Inference in polytrees with sets of probabilities. In: *Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA: Morgan Kaufmann, 2003. p. 217–224.
- RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence. A Modern Approach*. Englewood Cliffs: Prentice-Hall, 1995.
- SAUL, L. K.; JAAKKOLA, T. S.; JORDAN, M. I. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research*, v. 4, p. 61–76, 1996. ISSN 1076-9757.
- SAUL, L. K.; JORDAN, M. I. Exploiting tractable substructures in intractable networks. In: TOURETZKY, D. S.; MOZER, M. C.; HASSELMO, M. E. (Ed.). *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1996. v. 8, p. 486–492.
- SHACHTER, R. D. Intelligent probabilistic inference. In: KANAL, L. N.; LEMMER, J. F. (Ed.). *Uncertainty in Artificial Intelligence*. Amsterdam: Elsevier Science Publishers, 1986. p. 371–382.
- SHAFER, G. *A Mathematical Theory of Evidence*. Princeton, NJ: Princeton University Press, 1976.
- SHANNON, C. E. A mathematical theory of communication. In: *Bell System Technical Journal*, 1948. v. 27, p. 379–423, 623–656.
- SHENOY, P. S. C. Bayesian network models of portfolio risk and return. In: ABU-MOSTAFA B. LEBARON, A. W. L. Y. S.; WEIGAND, A. S. (Ed.). *Computational Finance*. The MIT Press, Cambridge, MA, 1999. v. 156, n. 2, p. 87–106.
- STEVENS, L. *Artificial Intelligence : The Search for the Perfect Machine*. Hayden Book Company, 1985.
- SUERMONDT, H. J.; COOPER, G. F. Initialization for the method of conditioning in Bayesian belief networks. *Artificial Intelligence*, v. 50, n. 1, p. 83–94, 1991.

- TESSEM, B. Interval probability propagation. *International Journal of Approximate Reasoning*, v. 7, p. 95–120, 1992.
- THRUN, S. Probabilistic algorithms in robotics. In: *AI Magazine*, 2000. v. 21, n. 4, p. 93–109.
- WALLEY, P. *Statistical Reasoning with Imprecise Probabilities*. London: Chapman and Hall, 1991.
- WALLEY, P. Measures of uncertainty in expert systems. *Artificial Intelligence*, v. 83, n. 1, p. 1–58, 1996.
- WEISS, Y.; FREEMAN, W. T. *Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology*, 1999.
- WHITTAKER, J. *Graphical Models in Applied Multivariate Statistics*. John Wiley, 1990.
- WIEGERINCK, W. Variational approximations between mean field and the junction tree algorithms. In: BOUTILIER, C.; GOLDSZMIDT, M. (Ed.). *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI-00)*. San Francisco: Morgan Kaufmann Publishers, 2000.
- WINN, J. *Variational Message Passing and its Applications*. Tese (Doutorado) — Department of Physics, University of Cambridge, Cambridge, UK, 2003.
- YEDIDIA, J. S.; FREEMAN, W. T.; WEISS, Y. Generalized belief propagation. In: *NIPS*, 2000. p. 689–695.
- ZADEH, L. A.; KACPRZYK, J. (Ed.). *Fuzzy Logic for the Management of Uncertainty*. 1. ed. New York: John Wiley & Sons, 1992. ISBN 0-471-54799-9.
- ZAFFALON, M. *Inferenze e Decisioni in Condizioni di Incertezza con Modelli Grafici Orientati*. Tese (Doutorado) — Universidade de Milão, Milão, Itália, 1997. Em italiano.

APÊNDICE I – SOFTWARE 2UBAYES

O programa 2UBayes foi implementado em linguagem Java. Ele implementa os algoritmos 2U, L2U, IPE e SV2U — e é capaz de realizar inferência em redes credais binárias, ou seja, fornecer intervalos aproximados de probabilidades inferiores e superiores para uma variável, dadas as observações (a especificação do problema de inferência é descrita na Seção 3.4). O algoritmo 2U fornece resultados exatos para redes credais com estrutura poli-árvore.

2UBayes emprega a estrutura de dados fornecida pelo programa EmBayes, um programa para processamento de redes Bayesianas em sistemas embarcados (portanto, possui uma arquitetura compacta de dados). Além disso, ele utiliza o pacote BNJ para carregar redes com extensão “*.xml”. O programa 2UBayes, assim como seus componentes necessários e conjuntos de testes empregados no Capítulo 5 estão disponíveis no site do *Laboratório de Tomada de Decisões* da Escola Politécnica da USP pelo endereço: <http://www.pmr.poli.usp.br/ltd/>.

Para executar o programa 2UBayes, basta entrar com a seguinte linha de comando (um console com as possíveis opções será apresentado como na Figura 36):

```
java -classpath ../2UBayes.jar embayes.EmBayes
```

Após inicialização, o programa deve carregar a rede credal a ser processada. A linha de comando é como a que segue:

```
>> a dog-problem,
```

onde “a” é o comando para se carregar a rede, cujo nome é “dog-problem”. O programa 2UBayes carrega duas redes Bayesianas equivalentes a uma rede credal binária. Uma rede contém os valores inferiores dos intervalos da probabilidade $P(X_i = x_i|pa_i)$, e outra rede, os valores superiores de $P(X_i = x_i|pa_i)$ ¹. As redes

¹Note que redes credais binárias podem ser especificadas por intervalos de probabilidade.

```

C:\j2soft\2UBayes\dist>C:\j2sdk1.4.2_04\bin\java -classpath C:\j2sdk1.4.2_04\lib\*.jar;
C:\j2soft\2UBayes\dist\2UBayes.jar embayes.Embayes

Embayes Console (for Bayesian networks)
Commands (brackets indicate several values):
h -> This help message.
l "name" -> Load a network.
o "variable" -> Set variable as not observed.
o "variable" "value" -> Observe variable.
r -> Generate a random number of observations.
t { "variable" } -> Set variables as explanatory.
u { "variable" } -> Set variables as non-explanatory.
i { "variable" } -> Posterior marginal for variables.
e "variable" -> Expected value for variable.
m -> Maximum a posteriori for explanatory variables.
p filename -> Parameter learning (updating) using data in filename
s -> Crash test for computation of marginal.
d { "variables" } -> Posterior marginal using Loopy BP.
.....
2UBayes console (for credal networks)
a -> Load a binary credalnetwork (you need the Lowe.xml and Upper.xml).
    Windows (networks must be in the same directory of Embayes.class)
    Linux (you must set the path in the code at Line 213 and 218)
c "queriedvariable" "label" -> L2U(multiconnected)/2U(polytree) posterior marginals.
    you have to enter with a label(name) for the simulation.
    (2 files will be generated with the results - L2U simulation stop when it converge)
b "variable" -> Set variable as not observed.
b "variable" "value" -> Observe variable.
v "label" "option" -> sv2U posterior marginal. You have to enter with a label(name) for the simulation.
    You have to set (random or order) polytree generation method.
    (File containing the results is generated)
f "queriedvariable" "niteration" -> IPE posterior marginals.
    you have to enter with number of iteration for the simulation.
    (File containing the results is generated with network name and niteration)
g "name" -> Save the credal network loaded in JavaBayes BIF Format.
q -> Quit.
Insert command character (l|o|t|u|i|m|e|x|a|q) or (b|c|i|d|i|f|i|q):

```

Figura 36: Ilustração do console do programa 2UBayes.

Bayesianas devem estar no formato “xml”. Por exemplo, na rede Dog-problem da Figura 15, a rede com valores inferiores “dog-problemLower.xml” terá como distribuições: $P(f) = 0.15$, $P(b) = 0.01$, $P(l|f) = 0.8$, $P(l|b) = 0.8$, $P(l|\neg f) = 0.01$, etc. Logo, para se carregar uma rede credal binária no programa 2UBayes são necessários dois arquivos (duas redes Bayesianas): “(nome)Lower.xml” e “(nome)Upper.xml”. Este procedimento adotado possibilita que a rede credal binária possa ser construída em qualquer programa que manuseie redes Bayesianas. A Figura 37 ilustra as redes sendo carregadas no programa 2UBayes.

Uma vez que a rede credal foi carregada, o usuário pode entrar com as opções de inferências desejadas: executar o algoritmo L2U, IPE, SV2U ou realizar algumas observações antes, etc. A execução do algoritmo L2U na rede carregada *dog-problem* é ilustrada na Figura 38. Um arquivo contendo os resultados é gerado pelo programa.

```

Insert command character (|o|t|u|i|i|m|e|x|a|q) or (b|c|d|f|g|q):
>>a dog-problem
    Parsed command: a
load XMLBIF .xml file ...
load variables ...
---nodename : light_on
---node position : 75.0 ,223.0
---nodename : bowel_problem
---node position : 438.0 ,82.0
---nodename : dog_out
---node position : 305.0 ,172.0
---nodename : hear_bark
---node position : 323.0 ,308.0
---nodename : family_out
---node position : 163.0 ,73.0
load ProbabilityDistributions ...
load XMLBIF .xml file ...
load variables ...
---nodename : light_on
---node position : 75.0 ,223.0
---nodename : bowel_problem
---node position : 438.0 ,82.0
---nodename : dog_out
---node position : 305.0 ,172.0
---nodename : hear_bark
---node position : 323.0 ,308.0
---nodename : family_out
---node position : 163.0 ,73.0
load ProbabilityDistributions ...

```

Figura 37: Ilustração da rede credal sendo carregada no programa 2UBayes.

```

Insert command character (|o|t|u|i|i|m|e|x|a|q) or (b|c|d|f|g|q):
>>c dog-out teste
    Parsed command: c
Invalid variable names!
Insert command character (|o|t|u|i|i|m|e|x|a|q) or (b|c|d|f|g|q):
>>c dog_out teste
    Parsed command: c

***** Minimum and maximum BEL values by L2U propagation *****

P(light_on)=[0.405 0.5]
P(bowel_problem)=[0.01 0.05]
P(dog_out)=[0.616375 0.756375]
P(hear_bark)=[0.305 0.375]
P(family_out)=[0.15 0.25]
**End of Iteration:0

***** Minimum and maximum BEL values by L2U propagation *****

P(light_on)=[0.1285 0.30000000000000004]
P(bowel_problem)=[0.01 0.05]
P(dog_out)=[0.616375 0.756375]
P(hear_bark)=[0.37366125 0.54164375]
P(family_out)=[0.15 0.25]
**End of Iteration:1

***** Minimum and maximum BEL values by L2U propagation *****

P(light_on)=[0.1285 0.30000000000000004]
P(bowel_problem)=[0.01 0.05]
P(dog_out)=[0.616375 0.756375]
P(hear_bark)=[0.37366125 0.54164375]
P(family_out)=[0.15 0.25]
**End of Iteration:2
Process time: 265 milliseconds.
Results saved as result_L2U_dog-problem_teste.txt
Insert command character (|o|t|u|i|i|m|e|x|a|q) or (b|c|d|f|g|q):
>>

```

Figura 38: Ilustração da execução do algoritmo L2U no programa 2UBayes.