

中国科学技术大学 博士学位论文



贝叶斯网 不确定性推理研究

作者姓名：张什永
学科专业：计算机科学
导师姓名：王晓蒲 教授 彭云 教授
完成时间：二〇一〇年五月一日

University of Science and Technology of China
A dissertation for doctor's degree



Uncertain Reasoning in Bayesian Networks

Author's Name : Shenyong Zhang

Speciality : Computer Science

SupervisorAdvisers : Prof. Xiaopu Wang, Prof. Yun

Peng

书脊

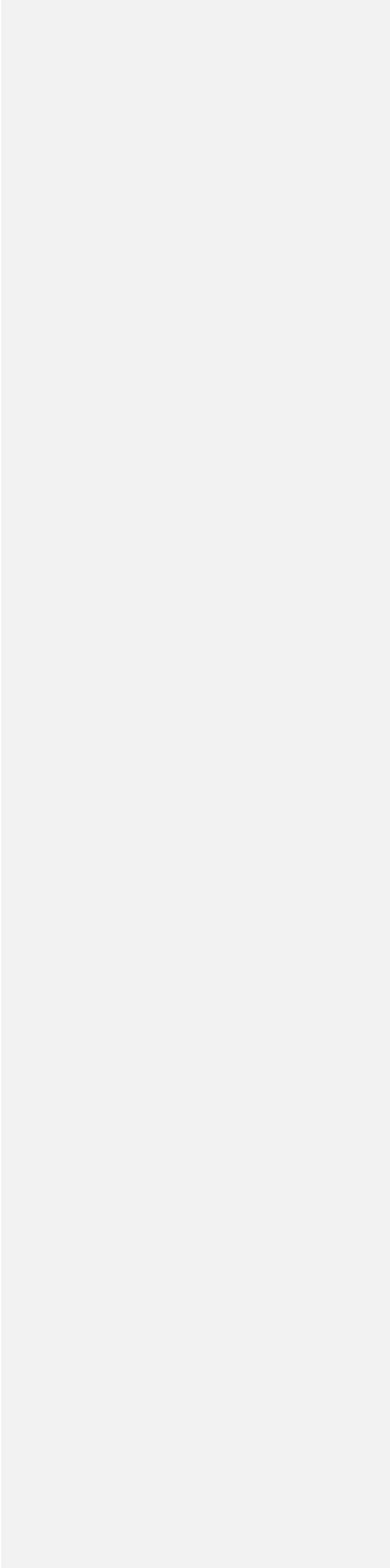


贝叶斯网络中的不确定性数据
贝叶斯网络中的不确定性推理
二十二系
张什永
中

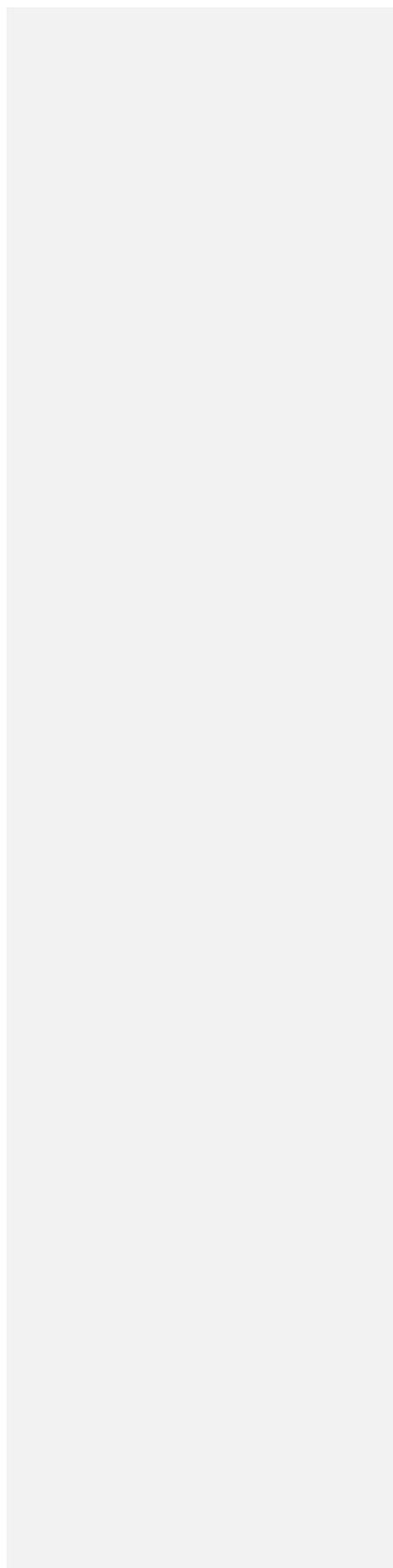


3cm 左右

3cm 左右



国科学技术大学



中国科学技术大学学位论文原创性声明

本人声明所呈交的学位论文,是本人在导师指导下进行研究工作所取得的成果。除已特别加以标注和致谢的地方外,论文中不包含任何他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的贡献均已在论文中作了明确的说明。

作者签名: _____

签字日期: _____

中国科学技术大学学位论文授权使用声明

作为申请学位的条件之一,学位论文著作权拥有者授权中国科学技术大学拥有学位论文的部分使用权,即:学校有权按有关规定向国家有关部门或机构送交论文的复印件和电子版,允许论文被查阅和借阅,可以将学位论文编入有关数据库进行检索,可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。本人提交的电子文档的内容和纸质论文的内容相一致。

保密的学位论文在解密后也遵守此规定。

公开 保密 (____年)

作者签名: _____

导师签名: _____

签字日期: _____

签字日期: _____

摘要

贝叶斯网（又称信度网或因果网）是一种基于概率推理的数学模型，它起源于人工智能领域的研究，是一种将概率统计应用于复杂领域、进行不确定性推理和数据分析的工具。随着对其研究的不断深入，贝叶斯网已经成为不确定性知识表达和推理领域最有效的模型之一，并因此在很多领域有着广泛的应用。尽管有着坚实的理论基础，关于贝叶斯网在给定不确定信息下的知识合成及推理机制的研究仍然存在很多有待进一步和探讨的问题。

本文首先较为系统地介绍了已有的知识合成算法，如 IPFP (iterative proportional fitting procedure) 方法，以及基于 IPFP 的 CIPFP、CC-IPFP、GEMA 等算法，比较了算法的性质及各自的特点，并且在此基础上提出了一种 SMOOTH 算法。该算法扩展了 IPFP，[在迭代中](#)通过双向修改约束集与联合概率分布，实现了对不一致概率知识的合成及算法的快速收敛。计算机实验结果表明，与原有算法相比，该算法计算复杂度低，[收敛过程稳定](#)，收敛速度快，并且可以根据需要通过调整平滑因子加快收敛速度。

基于 IPFP 的知识合成算法，由于以联合分布为目标知识库，并不能直接应用于贝叶斯网。本文在 Peng 和 Ding 工作的基础上，介绍了 E-IPFP 算法，并详细给出了算法在概率约束集一致时的收敛性证明。同时，为了解决约束集不一致时的收敛问题，通过引入 SMOOTH 算法，对 E-IPFP 进行了改进。实验结果表明，改进后的算法在概率约束集不一致时同样能够达到收敛，并能够处理约束条件与网络结构不一致的情况。

文中详细介绍了不确定性证据下的贝叶斯网信任更新。首先系统阐述了不确定性证据的分类及各自的特点，探讨了在单一证据下不同信任更新算法 (Jeffrey's rule, virtual evidence method 和 IPFP) 之间的关系。[在 Peng 和 Pan 工作的基础上](#)提出了多重证据下贝叶斯网的信任更新算法，BN-IPFP，并给出了算法在证据一致情况下的收敛性证明。同时对其进行了改进，使得算法不仅能够处理多重证据下的信任更新，而且能够解决[信任不一致的问题](#)，并在理论和实验上验证了改进后的算法的收敛性。

最后，论文讨论了贝叶斯网在语义网本体不确定性推理中的应用。通过扩展 BayesOWL 理论框架，使得其能够将更为一般的本体（如 OWL DL）转化为贝叶斯网。同时通过对 OWL 本体语言的扩展，使其能够描述任意以概率形式给出的关于本体的不确定性知识，并引入了贝叶斯网知识合成算法将其合成到转换后的贝叶斯网中，从而建立起关于本体的[概率](#)知识库。最后实现了该理论框架的原型系统，提供了相应的应用程序接口，为从事该领域研究工作的人提供了一个

Comment [Y1]: What is this? “信任不一致”还是“证据不一致”？

实用的工具。

关键词：贝叶斯网络 不确定性推理 知识合成 信任更新 语义网 本体推理

ABSTRACT

Bayesian network (BN, also ~~called known as~~ belief network or casual network) is a mathematical model based on probability ~~theorystic reasoning~~. ~~It comes from artificial intelligence research area~~. It is a tool ~~developed in Artificial Intelligence~~ for ~~using modeling~~ probabilistic ~~interdependencies statistics in in~~ complex ~~areaproblems and supporting~~, uncertainty reasoning and data analysis. BN ~~is has been recognized as~~ one of the most efficient theoretical models in representation and reasoning of uncertainty knowledge, and ~~is usedhas applied~~ in many areas. However, there are still many problems to be resolved for Bayesian networks in knowledge integration and reasoning, e.g. belief update, under uncertain information.

In this ~~dissertationthesis~~, we first discussed properties and features of different existing knowledge integration algorithms, such as IPFP (iterative proportional fitting procedure) and other IPFP based algorithms, e.g. CIPFP (conditional IPFP), CC-IPFP, GEMA etc. Then a novel algorithm, SMOOTH, was proposed. The new algorithm extends original IPFP by bi-directional modification of probabilistic constraints and target joint probability distribution ~~during the iterations~~, thus it converges for both consistent and inconsistent constraints. Experiment results show that, compared with existing algorithms, SMOOTH ~~has stable and fast convergences fast, is less complex~~, and can be accelerated by adjusting its smooth ratio.

~~Knowledge~~ integration algorithms based on IPFP ~~use work on~~ joint probability distributions ~~as the knowledge~~, thus it can not be used directly in Bayesian networks. For this problem, we introduced the E-IPFP algorithm proposed by Peng and Ding. A formal convergence proof of E-IPFP under consistent probabilistic constraints was given. ~~MeanwhileIn addition~~, an improved method was ~~given proposed~~ by integrating E-IPFP and SMOOTH ~~to deal with inconsistent constraints~~. Experiment results show that, the improved algorithm can handle not only inconsistency among probabilistic constraints but also inconsistency between constraints and the BN structure.

Algorithms for Bayesian network belief update under uncertain evidences were also discussed. Classification and features of different uncertain evidences are discussed thoroughly. Also we formally proved the equivalence of different belief update algorithms (Jeffrey's Rule, Virtual Evidence Method, and IPFP) for single

uncertain evidence. An algorithm, BN-IPFP, with its convergence proof was given for multiple evidences. With the combination of BN-IPFP and SMOOTH, the new algorithm can deal with not only multiple uncertain evidences but also inconsistent evidences. Both theoretical and experimental convergence proof of BN-IPFP are discussed thorough.

Finally, we discussed the application of Bayesian network ~~reasoning to modeling and reasoning with uncertainty~~ in Semantic Web ontologies ~~realies uncertainty reasoning~~. The BayesOWL framework was extended ~~to; thus it can support converting~~ general ontologies (e.g. OWL DL) ~~to BNs, and a~~ ~~Meanwhile, a conventional extension of~~ based on OWL language was ~~given proposed~~ to represent probabilistic knowledge ~~of for ontology classes and interclass relations~~ ontologies. Knowledge integration algorithms ~~we developed~~ are used to integrate probabilistic knowledge into ~~result converted~~ BN. ~~Also a~~ prototype system of ~~the extended~~ BayesOWL framework was implemented with its APIs available for researchers on similar research areas.

Key Words: Bayesian network, uncertainty reasoning, knowledge integration, belief update, semantic web, ontology reasoning

Comment [Y2]: You need to more careful on this section. Some of these works (e.g., equivalence of Jeffrey's Rule, Virtual Evidence Method, and IPFP and BN-IPFP were originally done by Pan Rong. Your works are
1.further analysis of these work,
2.combining it with smooth。
你总结 (Ch.6) 的第四条的写法比较好。

目 录

第 1 章 绪论	VII 4
1.1 研究背景及研究现状分析	1
1.2 本文研究的内容与方法	4
1.3 本文的组织结构	5
第 2 章 概率知识合成算法研究	6
2.1 问题描述	6
2.2 概率约束一致时的知识合成算法	7
2.2.1 IPFP 算法	7
2.2.2 C-IPFP 算法	9
2.3 概率约束不一致时的知识合成算法	10
2.3.1 CC-IPFP 算法	11
2.3.2 GEMA 算法	12
2.3.3 SMOOTH 算法	14
2.3.4 实验结果对比与分析	18
2.4 小结	22
第 3 章 贝叶斯网知识合成算法研究	23
3.1 问题描述	23
3.2 贝叶斯网知识合成算法	24
3.2.1 E-IPFP 算法	25
3.2.2 D-IPFP 算法 (original D-IPFP? ? ? not MB(Y))	30
3.2.3 约束集不一致时 E-IPFP 算法的改进	32
3.3 小结	34
第 4 章 不确定性证据下贝叶斯网信任更新	36
4.1 问题描述	36
4.2 不确定性证据下的信任更新算法	37
4.2.1 Jeffrey's Rule 和 IPFP	38 37
4.2.2 Virtual Evidence Method	39
4.2.3 多重证据时存在的问题	41
4.2.4 BN-IPFP 算法	42
4.2.5 证据不一致时的处理	45
4.3 小结	46
第 5 章 贝叶斯网在语义网本体推理中的应用	48
5.1 引言	48
5.2 研究现状	49

5.3 BayesOWL 框架.....	50
5.3.1 结构转换规则.....	50
5.3.2 BayesOWL 中的语义关系.....	55 54
5.3.3 BayesOWL 中概率知识的表示和合成.....	58 56
5.4 BayesOWL 理论框架的实现.....	62 60
5.5 小结.....	64 62
第 6 章 论文总结.....	6563
参考文献.....	1
附录 1 xxxxx.....	<u>Error! Bookmark not defined.</u> 6
致 谢.....	<u>67</u>
在读期间发表的学术论文与取得的研究成果.....	11

插图目录

<u>Fig. 2.1 K-L distance of $I(Q_k(X) Q_\theta(X))$</u>	Formatted: Distributed, First line: 0 ch
<u>Fig. 2.1 K-L distance of $I(Q_k(X) Q_\theta(X))$</u>9	Formatted [1]
<u>Fig. 2.2 Entry of $Q^*(X_1=1, X_2=1, X_3=1)$ for IPFP with inconsistent data</u>	Formatted [2]
<u>Fig. 2.2 Entry of $Q^*(X_1=1, X_2=1, X_3=1)$ for IPFP with inconsistent data</u>10	Formatted: Distributed, First line: 0 ch
<u>Fig. 2.3 K-L distance of $I(Q_k(X) Q_\theta(X))$ for IPFP and GEMA</u>	Formatted: Distributed, First line: 0 ch
<u>Fig. 2.3 K-L distance of $I(Q_k(X) Q_\theta(X))$ for IPFP and GEMA</u>13	Formatted [3]
<u>Fig. 2.4 K-L distance of $I(Q_k(X) Q_\theta(X))$ for GEMA</u>	Formatted [4]
<u>Fig. 2.4 K-L distance of $I(Q_k(X) Q_\theta(X))$ for GEMA</u>14	Formatted: Distributed, First line: 0 ch
<u>Fig. 2.5 First two iterations of SMOOTH phase 2</u>	Formatted: Distributed, First line: 0 ch
<u>Fig. 2.5 First two iterations of SMOOTH phase 2</u>16	Formatted [5]
<u>Fig. 2.6 K-L distance of $I(Q_k(X) Q_\theta(X))$ for SMOOTH (from 2nd phase)</u>	Formatted: Distributed, First line: 0 ch
<u>Fig. 2.6 K-L distance of $I(Q_k(X) Q_\theta(X))$ for SMOOTH (from 2nd phase)</u>17	Formatted [6]
<u>Fig. 2.7 Smooth ratio α vs iteration step</u>	Formatted: Distributed, First line: 0 ch
<u>Fig. 2.7 Smooth ratio α vs iteration step</u>18	Formatted [7]
<u>Fig. 2.8 K-L distance of $I(Q_k(X) Q_\theta(X))$ for SMOOTH (accelerated)</u>	Formatted [8]
<u>Fig. 2.8 K-L distance of $I(Q_k(X) Q_\theta(X))$ for SMOOTH (accelerated)</u>18	Formatted: Distributed
<u>Fig. 2.9 Comparison of I-Aggregate</u>	Formatted: Distributed
<u>Fig. 2.9 Comparison of I-Aggregate</u>20	Formatted [9]
<u>Fig. 2.10 Comparison of K-L distance</u>	Formatted: Distributed
<u>Fig. 2.10 Comparison of K-L distance</u>21	Formatted [10]
<u>Fig. 3.1 An simple BN example with three variable</u>	Formatted: Distributed, First line: 0 ch
<u>Fig. 3.1 An simple BN example with three variable</u>24	Formatted [11]
<u>Fig. 3.2 Successive JPDs from E-IPFP</u>	Formatted: Distributed, First line: 0 ch
<u>Fig. 3.2 Successive JPDs from E-IPFP</u>27	Formatted [12]
<u>Fig. 3.3 Initial Bayesian network</u>	Formatted: Distributed, First line: 0 ch
<u>Fig. 3.3 Initial Bayesian network</u>30	Formatted [13]
<u>Fig. 3.4 Result Bayesian network for consistent constraint</u>	Formatted: Distributed, First line: 0 ch, Line spacing: single
<u>Fig. 3.4 Result Bayesian network for consistent constraint</u>30	Formatted [14]
<u>Fig. 3.5 Markov Blanket</u>	Formatted: Font: 12 pt
<u>Fig. 3.5 Markov Blanket</u>31	
<u>Fig. 3.6 Constraint R and R'</u>	Formatted: Distributed
<u>Fig. 3.6 Constraint R and R'</u>33	Formatted [15]
<u>Fig. 3.7 Result Bayesian network</u>	Formatted: Distributed, Indent: Left: 0"
<u>Fig. 3.7 Result Bayesian network</u>34	Formatted: Font: Times New Roman, 12 pt
<u>Fig. 4.1 Virtual node U</u>	Formatted: Distributed, Indent: Left: 0"
<u>Fig. 4.1 Virtual node U</u>40	Formatted: Font: Times New Roman, 12 pt
<u>Fig. 4.2 A BN example with 4 nodes</u>	Formatted: Distributed, Indent: Left: 0"
<u>Fig. 4.2 A BN example with 4 nodes</u>42	Formatted [16]
<u>Fig. 4.3 Result BN of BN-IPFP-1 for example 4.3</u>	Formatted: Distributed, Indent: Left: 0"
<u>Fig. 4.3 Result BN of BN-IPFP-1 for example 4.3</u>44	Formatted [17]
<u>Fig. 4.4 Result BN of BN-IPFP-2 for example 4.4</u>	Formatted: Distributed, Indent: Left: 0"
<u>Fig. 4.4 Result BN of BN-IPFP-2 for example 4.4</u>44	Formatted [18]

Fig. 4.4 Result BN of BN IPFP 2 for example 4.4.....45
Fig. 5.1 Subnet of Subclass relation.....51
Fig. 5.2 Subnet of Equivalent relation.....52
Fig. 5.3 Translated BN from 'Nature' ontology.....54
Fig. 5.4 Result BN for 'Nature' ontology example (consistent).....61
Fig. 5.5 Result BN for 'Nature' ontology example (inconsistent).....62
Fig. 5.6 BayesOWL framework.....63
Fig. 5.7 BayesOWL system architecture.....64
Fig. 5.8 BayesOWL application programming interface.....64

Formatted: Font: Times New Roman, 12 pt
Formatted: Distributed
Formatted: Font: Times New Roman, 12 pt
Formatted: Distributed
Formatted: Font: Times New Roman, 12 pt
Formatted: Font: 12 pt
Formatted: Font: Times New Roman, 12 pt
Formatted: Distributed
Formatted: Font: Times New Roman, 12 pt
Formatted: Font: 12 pt
Formatted: Font: Times New Roman, 12 pt
Formatted: Distributed
Formatted: Font: Times New Roman, 12 pt
Formatted: Font: 12 pt
Formatted: Font: Times New Roman, 12 pt
Formatted: Font: 12 pt
Formatted: Distributed, First line: 0 ch, Line spacing: single
Formatted: Font: Times New Roman, 12 pt
Formatted: Font: 12 pt
Formatted: Font: Times New Roman, 12 pt
Formatted: Font: 12 pt
Formatted: Font: Times New Roman, 12 pt
Formatted: Font: Times New Roman
Formatted: Distributed, First line: 0 ch, Line spacing: single
Formatted: Font: Times New Roman, 12 pt
Formatted: Font: Times New Roman
Formatted: Distributed, First line: 0 ch, Line spacing: single
Formatted: Font: Times New Roman, 12 pt
Formatted: Font: Times New Roman
Formatted: Distributed, First line: 0 ch, Line spacing: single
Formatted: Font: Times New Roman, 12 pt
Formatted: Font: Times New Roman

表格目录

<u>Table 2.1</u> Constraint set R8	Formatted: Distributed, Indent: First line: 0" Formatted: Font: 12 pt
<u>Table 2.2</u> Result joint probability distribution.....8	Formatted [19] Formatted: Distributed, Indent: First line: 0"
<u>Table 2.3</u> Convergence result of GEMA and CC-IPFP.....14	Formatted: Distributed Formatted [20]
<u>Table 2.4</u> Convergence result of 4 algorithms for consistent constraints.....19	Formatted [21] Formatted: Distributed, Indent: First line: 0"
<u>Table 2.5</u> Two initial joint probability distributions.....19	Formatted: Distributed Formatted [22]
<u>Table 2.6</u> Comparison of convergence result for inconsistent constraint R_120	Formatted: Distributed Formatted [23]
<u>Table 2.7</u> Result of an joint probability distribution (8 variables).....21	Formatted: Distributed Formatted [24]
<u>Table 2.8</u> Result of an joint probability distribution (15 variables).....21	Formatted [25] Formatted: Distributed
<u>Table 3.1</u> Consistent constraint R29	Formatted: Distributed Formatted [26]
<u>Table 4.1</u> Belief update on BN of example 4.2.....42	Formatted: Distributed Formatted [27]
<u>Table 5.1</u> Concept constructor supported in BayesOWL.....50	Formatted: Distributed, Indent: First line: 0" Formatted [28]
<u>Table 5.2</u> Conditional probability table of node C in figure 5.1.....51	Formatted: Distributed Formatted [29]
<u>Table 5.3</u> Conditional probability table of L-Node in figure 5.2.....52	Formatted: Distributed Formatted [30]
<u>Table 5.4</u> Number of relations in Wine ontology.....54	Formatted: Distributed Formatted [31]
<u>Table 5.5</u> Interpretations for 3 embedded constructors used in BayesOWL.....55	Formatted: Distributed Formatted [32]
<u>Table 5.6</u> Conditions on interpretations used in BayesOWL.....55	Formatted [33] Formatted: Distributed

表格目录

~~Table 5.7 Conditions on interpretations used in BayesOWL~~
~~Table 5.7 Conditions on interpretations used in BayesOWL.....56~~

Formatted: Distributed

Formatted: Font: 12 pt

第1章 绪论

本章主要介绍论文研究背景、研究内容及论文的内容组织。

1.1 研究背景及研究现状分析

现实中人们所获得的信息和知识，往往含有大量的不准确、不完全、不一致的地方，即获得的信息和知识是不确定的。不确定性推理即指建立在这种不确定性知识和证据上的推理(张连文等 2006; 马少平等 2004)。对于一个智能系统来说，知识库是其核心。在这个知识库中，往往包含大量模糊性、随机性、不可靠性等不确定性因素的知识。为了解决这种条件下的推理计算问题，人们逐渐发展出了各种不确定性推理的方法。可以说，智能主要反映在求解不确定性问题的能力上。因此，不确定性推理是人工智能研究中的一个核心课题。一个智能系统，由于知识本身的不精确和不完全，采用标准逻辑意义下的推理方法难以达到解决问题的目的，因此常采用非标准意义下的不确定性推理方法。对于不确定性推理来讲，如何描述和传播不确定性是主要问题。

贝叶斯网起源于人工智能领域的研究，是一种将概率统计应用于复杂领域、进行不确定性推理和数据分析的工具，是概率论和图论相结合的产物(Pearl 1986, 1987)。它通过网络结构来定性地刻画问题变量之间的相互独立关系；通过网络节点的条件概率表来定量地描述各变量节点的独立关系(Pearl 1988)。同时，根据链式规则(Russell et al. 2003)，每个贝叶斯网确定了一个联合概率分布，该联合分布即为问题模型的一个知识库。通过对知识库的查询，按照概率论的相关知识可以回答关于问题领域的各种问题。这样，贝叶斯网利用图论的语言直观揭示问题的结构，同时按照概率论的原则对问题的结构加以利用，把复杂的联合概率分布分解成一系列相对简单的模块，从而大大降低了知识获取的难度和概率推理的复杂度，使概率论能够应用于大型问题的求解(Pearl 1988; 张连文等 2006)。由于贝叶斯网直观易懂，且具有坚实的理论基础，使得其在很多领域都有广泛的应用(Osterwalder et al. 2002; Costa et al. 2005; Laskey 2007; Yelland 1999; Yang et al. 2005; Ding et al. 2004)。

与人们认识客观世界的过程相对应，对于智能系统或知识库的建立也是一个循序渐进的过程，在这个过程中往往涉及到对原有知识的补充、更新和完善。因此，根据概率知识更新知识库，即概率知识合成，是不确定性推理研究中的一个重要问题(Cali et al. 2008; Jirousek et al. 1995; Vomlel 2003; Peng

et al. 2005; Zhang et al. 2008)。概率知识合成是指在已知概率知识时，如何将概率知识将其合成到已有的知识库（例如联合概率分布）中，即对一已知初始的知识库和一组给定的约束条件（概率知识），寻找满足给定约束的知识库的过程（Vomlel 1999）。

Kruithof 等人提出了一种迭代均衡满足算法（iterative proportional fitting procedure, IPFP）（Kruithof 1937; Deming et al. 1940; Fienberg 1970）。该算法通过迭代使用给定的约束条件，按照最小化 K-L 距离（Kullback et al. 1951）的原则，解决了概率约束的满足问题，并将其应用于联合概率分布的知识合成。Bock 等人对 IPFP 方法进行了扩展，使其能够处理概率约束为条件概率时的概率知识合成问题（Bock 1989; Cramer 2000）。自此以后，IPFP 的方法受到了研究人员广泛关注和越来越多的应用（Jirousek et al. 1995; Deming et al. 1940; Csiszar 1975; Vomlel 2003; Peng et al. 2005; Zhang et al. 2008）。

然而，IPFP 方法的局限性之一就是 IPFP 并不适用于约束集不一致时情况（Csiszar 1975; Vomlel 2004; Peng et al. 2005）。而在实际应用中由于知识获取的途径不同，对于事件的观察结果往往带有一定的随机性对于事件的观察结果往往带有一定的局限性和误差，这样来源不同的概率知识往往是不一致甚至是相互矛盾的（Vomlel 1999, 2004; Zhang et al. 2008）。为了使用这些不一致的知识来更新已有的知识库，Vomlel 研究了约束集不一致时的知识合成问题，并提出了一种新的算法（Vomlel 2003），GEMA，实现了约束集不一致时的知识合成。

IPFP 方法的另一局限之处在于，基于 IPFP 的方法只能应用于知识库为联合分布的情况，并不能直接应用于贝叶斯网（Peng et al. 2005）。为了解决这个问题，Peng 和 Ding 提出了一种基于 IPFP 的贝叶斯网知识合成算法，E-IPFP。该算法通过迭代修改网络参数，将给定概率知识编码到贝叶斯网的条件概率表中，实现了贝叶斯网知识合成的目的（Peng et al. 2005）。

现实生活中，人们往往希望根据对某些事件的观察，结合已知的经验知识，通过一定的手段来获得关于未知事件的信息。推理正是一个这样的手段，它是通过计算回答查询的过程。计算后验概率是贝叶斯网推理中的一类重要问题，它是指给定证据变量，计算查询变量的后验概率的过程（Jensen 1994; Pearl 1987, 1990; Valtorta et al. 2002; Chan et al. 2003）。自从 Pearl 在 1990 年首次提出了不确定性证据（*uncertain evidence*）的概念（Pearl 1990），并探讨了其推理机制之后，贝叶斯网在不确定性证据下的推理（信任更新）问题逐渐受到越来越多研究人员的关注。根据研究文献，不确定性证据可以分为两

类: *soft evidence* 指关于不确定性的证据 (*evidence of uncertainty*), 通常表示为关于一个或多个变量的概率分布 (Valtorta et al. 2002; Kim et al. 2004); *virtual evidence* 指证据的不确定性 (*evidence with uncertainty*), 通常用似然比率 (*likelihood ratio*) 表示 (Pearl 1990)。不同类别的证据在进行贝叶斯网信任更新时遵循不同的方法。Jeffrey 指出, 对于使用 *soft evidence* 进行信任更新后的联合分布, 需要满足两个原则: 首先更新后的联合分布要满足给定的 *evidence* 证据, 其次更新前后查询变量关于证据变量的条件概率保持不变。第二条原则即为概率动力学原则 (Wagner 2002), 它保证了进行信任更新更新前后, 对于联合分布的修改是最小化的。Jeffrey 根据这两条原则提出了 *Jeffrey's rule* 信任更新方法。然而 *Jeffrey's rule* 不能直接应用于贝叶斯网, 为此, Pearl 提出了一种 *virtual evidence method* (Pearl 1990), 该方法首先根据 *soft evidence* 计算似然比率, 并由此生成一个 *virtual* 节点添加到贝叶斯网中, 实现了单一 *soft evidence* 下的贝叶斯网信任更新。

随着研究的不断深入, 贝叶斯网正在越来越多的领域获得更加广泛的应用。例如, 近年来, 随着语义网及本体技术的兴起 (Osterwalder et al. 2002; Chaballier et al. 2007; Bottazzi et al. 2006; Batchelor et al. 2009; Schlicht et al. 2008), 为了解决本体推理中的不确定性, 越来越多的研究人员将目光投向了贝叶斯网 (Koller et al. 1997; Costa et al. 2005; Yang et al. 2005; Mitra et al. 2005; Costa et al. 2008; Laskey et al. 2007)。将贝叶斯网应用于本体不确定性推理主要有两个方向, 一是将贝叶斯方法与本体语言的底层逻辑 (例如描述逻辑等) 相结合进行不确定性推理, 例如 P-CLASSIC (Koller et al. 1997), PR-OWL (Costa et al. 2005), OntoBayes (Yang et al. 2005) 等; 另一个方向为将本体问题直接转化为贝叶斯网推理问题, 例如 Ding 和 Peng 等提出的 BayesOWL 理论框架 (Ding et al. 2004, 2006)。BayesOWL 首先将根据给定的本体按照结构转换规则生成一个贝叶斯网络结构, 然后将已知概率知识合成到贝叶斯网的 条件概率表 中, 并利用其来解决本体 不确定性 推理问题。

近几年来, 对知识合成和信任更新这两类贝叶斯网不确定推理中的重要问题, 虽然已经进行了大量研究, 但是仍然没有形成相应的实用方法。这是因为在实际实现中, 仍然存在许多有待进一步研究和探讨的问题, 例如:

知识合成中的不一致问题 所谓不一致, 是指关于获得的概率知识 (约束集), 往往是不准确、不完全甚至相互矛盾的。正如上文提到的, 基于 IPFP 的知识合成算法只在约束集一致时收敛, 而在约束集不一致时, 算法并不收敛, 而是进入振荡 (Vomlel 1999; Peng et al. 2005)。虽然 Vomlel 提出的 GEMA 算法能够

Formatted: Highlight

Comment [Y3]: I thought this was first proposed by Darwiche, not Pearl. Pearl proposes virtual evidence node only for virtual evidences. Please check the references.

在约束不一致时实现收敛，但是实验表明其计算复杂度很高，且收敛过程受初始条件影响很大(Zhang et al. 2008)。那么就需要寻找更加有效的方法，来解决这个问题。同样在贝叶斯网知识合成中，也存在类似问题。

信任更新中的多重证据问题 无论 Jeffrey's rule 或 virtual evidence 方法，都不能处理多重证据时的信任更新问题。由于在现实应用中，多重证据问题的普遍存在，且观察证据之间仍然会存在相互不一致的地方，那么就需要寻找一个有效的方法来解决。

贝叶斯网推理在语义网本体中的应用问题 将贝叶斯网作为本体语言底层逻辑扩展的方法，虽然保留了描述逻辑在问题描述方面的能力，但是它改变了本体描述语言的底层逻辑。这使得对于描述逻辑的不同扩展，需要提供不同的逻辑语义规则及相应的推理机制，所以不利于实际应用。而原始的 BayesOWL 的方法同样存在如下限制：只能应用于简单的分类学意义上的本体；知识合成时只能处理简单的概率知识（例如 $P(A)$ ， $P(A|B)$ 的简单情况）。所以需要对其进行扩展，以便能够处理更为一般的情况。

1.2 本文研究的内容与方法

贝叶斯网不确定性推理及其方法仍有很多有待改进和进一步研究的问题。本文针对其中的知识合成和信任更新中存在的问题，主要从事了以下几个方面的研究：

- 知识合成算法及其性质的探讨。基于 IPFP 及其扩展 CIPFP 的知识合成算法的研究已经比较深入，然而遗憾的是该方法只在约束集一致时收敛。而现实应用中，由于海量数据的来源及获取手段的不同，作为概率知识的约束集相互之间往往存在相互矛盾之处。为此，Vomlel 提出了约束集不一致时的知识合成算法 CC-IPFP、GEMA 算法。而实验结果表明，CC-IPFP 和 GEMA 算法在计算复杂度、收敛性能等方面的表现不尽人意：算法计算复杂度高，且收敛速度受初始值的影响很大，使得其在应用中性能不可预测。针对该种情况，本文在研究现有算法的基础上，提出了一种新的基于 IPFP 的合成算法 SMOOTH，并在理论和实验上讨论了新算法的收敛性能。
- 贝叶斯网知识合成算法的研究。如何通过修改网络参数，使贝叶斯网满足给定概率知识，即概率约束满足，是贝叶斯网不确定性推理研究中的一个重要问题。Peng 和 Ding 在知识合成算法的基础上提出了一种贝叶斯网知识合成算法 E-IPFP，通过迭代修改网络参数，实现了贝叶斯网的约束满足。本文详细讨论了该算法的性质，并给出了算法的收

敛性证明。同时，通过引入不一致时的知识合成算法，对 E-IPFP 进行了改进，解决了算法在约束集不一致时的收敛问题。实验结果表明，改进后的算法在约束集相互不一致，或与网络结构不一致时均能达到收敛。

- 不确定性证据下贝叶斯网信任更新算法研究。根据给定观察证据，如何计算查询变量的后验概率，是贝叶斯网信任更新中的一个重要问题。Pearl 首次提出了不确定性证据的概念，并提出了一种 virtual evidence 方法，用以处理给定单一 soft evidence 下贝叶斯网信任更新。[然而该方法并适用于多重不确定性证据的情况然而该方法并适用于多个 soft evidence 的情况](#)。本文首先分析了各种证据的性质与特点；回顾了几种不确定性证据下贝叶斯网信任更新方法：virtual evidence method, Jeffrey's Rule 和基于 IPFP 的方法；分析了各种方法之间的关系；介绍了一种 BN-IPFP 的新算法，证明了其在证据一致时的收敛性；对 BN-IPFP 算法进行了改进，使其能够解决证据不一致时的情况，并在理论和实验上对算法进行了分析。
- 本体不确定性推理问题的研究。本文研究了如何将贝叶斯网不确定性推理方法应用于本体不确定性推理问题中。着重介绍并扩展了 Ding 和 Peng 提出的 BayesOWL 框架。该方法通过将本体转化为贝叶斯网，同时将关于本体的不确定性知识通过知识集成的方法合成到转化后的贝叶斯网中，实现了本体推理到贝叶斯网推理的转化。我们通过采用本体推理工具，扩展了该方法对更一般本体的支持；分析了转化后贝叶斯网中的语义；通过引入改进后的贝叶斯网知识合成算法，解决了对不一致概率知识的合成问题；最后，实现了该方法的原型系统，此系统为从事该领域研究工作的人员提供了一个实用工具。

1.3 本文的组织结构

本文第二章介绍了知识合成及基于 K-L 距离最小化原则下的知识合成；第三章研究了贝叶斯网知识合成算法及其改进；第四章讨论了不确定性证据下贝叶斯网信任更新问题，及 BN-IPFP 算法；第五章探讨贝叶斯网不确定性推理在语义网本体推理中的应用；第六章对全文工作进行了总结。

第2章 概率知识合成算法研究

本章讨论概率知识合成算法。首先简要介绍了概率约束一致时的知识合成算法 IPFP、CIPFP，及约束不一致时算法 CC-IPFP、GEMA；探讨了约束不一致时合成算法的性质，[提出了一种简单的基于提出了一种简便而有效的基于 IPFP 的新合成算法](#)，并在理论和实验上讨论了新算法的性能。

2.1 问题描述

根据概率论框架下的定义，我们将领域问题用一组随机变量来表示 $\mathbf{X}=(X_1, X_2, \dots, X_n)$ ；将关于问题的知识表示为一个联合概率分布 $P(\mathbf{X})$ 。在进行进一步的讨论之前，我们首先给出如下定义：

定义 2.1 (约束集) 关于问题模型的新的概率知识集合。这些概率知识通常为定义在 \mathbf{X} 子空间的低维概率分布，记为：

$$R = \{R(Y^1), R(Y^2), \dots, R(Y^m)\}$$

其中 $Y^j \subseteq X$ ， m 为约束条件的个数。

定义 2.2 (约束集的一致性) 如果存在至少一个联合概率分布 $P(\mathbf{X})$ ，使得 $P(\mathbf{X})$ 满足概率约束集 R ，即对任意 $1 \leq j \leq m$ ， $P(Y^j) = R(Y^j)$ 成立，则称约束集 R 是一致的；否则称 R 为不一致的。

定义 2.3 (K-L 距离) 又称[交叉熵](#)，反映了两个联合概率分布之间的差异，其值定义为：

$$I(P \parallel Q) = \begin{cases} \sum_{P(x)>0} P(x) \log \frac{P(x)}{Q(x)} & \text{if } P \ll Q \\ +\infty & \text{else} \end{cases}$$

其中 $P \ll Q$ 表示 $\{x | P(x) > 0\} \subseteq \{x' | Q(x') > 0\}$ 。规定 $0 \cdot \log \frac{0}{q} = 0$ ，且

$$p \cdot \log \frac{p}{0} = \infty。$$

K-L 距离是信息论衡量两个离散分布之间距离中广泛使用的一种方法。根据其定义可知 K-L 距离是非负值，即 $I(P \parallel Q) \geq 0$ ，且仅当 $P=Q$ 时等号成立。与一般欧几里得空间距离不同，K-L 距离是非对称的，即在一般情况下有

Comment [Y4]: It should be relative entropy **【相对熵】**, not cross entropy. You can check the Wikipedia for their definitions. These two entropy are closely related to each other, though.

$I(P \parallel Q) \neq I(Q \parallel P)$ 。

定义 2.4 (知识合成) 寻找满足给定约束集的联合概率分布的过程。

由此我们的问题模型可以描述为：

已知联合概率分布 $P(X)$ 和概率约束集 $R = \{R(Y^1), R(Y^2), \dots, R(Y^m)\}$ ，构造新的联合概率分布 $Q(X)$ ，使其满足约束集 R ，并且与初始概率分布 $P(X)$ 具有最小的 K-L 距离。

2.2 概率约束一致时的知识合成算法

2.2.1 IPFP 算法

IPFP (iterative proportional fitting procedure) 算法由 Kruithof 提出。其核心为 I-Projection 计算：

定义 2.5 (I-Projection) 给定一个联合概率分布集合 \mathcal{Q} 和 $P(X)$ ，若 $Q(X) \in \mathcal{Q}$ ，且满足

$$I(Q \parallel P) = \min_{Q \in \mathcal{Q}} I(Q \parallel P)$$

则称 $Q(X)$ 为 $P(X)$ 在 \mathcal{Q} 上的 I-Projection。

根据 (Vomlel 2003)， $P(X)$ 在满足约束条件 $R(Y)$ 的联合概率分布集合上的 I-Projection 可以由下式求解得到：

$$Q(X) = \begin{cases} 0 & \text{if } P(Y) = 0 \\ P(X) \cdot \frac{R(Y)}{P(Y)} & \text{else} \end{cases} \quad (2.2.1)$$

其中 $R(Y)$ 为定义在 X 子空间上的低维概率分布，即 $Y \subseteq X$ ， $P(Y)$ 为 $P(X)$ 在 Y 空间上的边缘概率分布。

它是一个迭代过程，算法每一步使用约束集中的一个约束条件，通过 (2.1.1) 式计算，寻找一个满足当前约束条件的联合概率分布。通过循环使用约束集中的所有约束条件，直至算法收敛。

Comment [Y5]: Should it be 2.2.1?

IPFP 算法：

1. 初始值： $Q_0(X)$ ， $R = \{R(Y^1), R(Y^2), \dots, R(Y^m)\}$

2. for $k = 1$ ，重复以下迭代过程，直至收敛：

2.1 $i = k \bmod m$ ；

$$2.2 \quad Q_k(X) = \begin{cases} 0 & \text{if } Q_{k-1}(Y^i) = 0 \\ Q_{k-1}(X) \cdot \frac{R(Y^i)}{Q_{k-1}(Y^i)} & \text{else} \end{cases} ;$$

2.3 $k = k + 1$;

定理 2.1 IPFP 算法在约束集 R 一致时算法收敛。若记收敛结果为 $Q^*(X)$ ，记所有满足 R 的联合概率分布集合为 S_R ，则 $Q^*(X)$ 满足 R ，且在所有满足 R 的联合概率分布中， $Q^*(X)$ 具有与初始 $Q_0(X)$ 最小的 K-L 距离，即 $Q^*(X)$ 为 $Q_0(X)$ 在 S_R 上的 I-Projection。

定理证明见 (Csiszar 1975; Vomlel 2003)，这里不再赘述。

例 2.1 设一联合概率分布定义在 $X=(X_1, X_2, X_3)$ 上，且各随机变量均为二元的，即取值为 T 或 F。已知约束集 $R = \{R(X_1, X_2), R(X_2, X_3), R(X_3, X_1)\}$ ，如表 2.1 所示。则当 $\varepsilon = 4/20$ 时， R 为一致的。

Table2.1 Constraint set R

$R_j, j = 1, 2$	$X_{j+1} = T$	$X_{j+1} = F$
$X_j = T$	$1/2 - \varepsilon$	ε
$X_j = F$	ε	$1/2 - \varepsilon$
R_3	$X_3 = T$	$X_3 = F$
$X_1 = T$	ε	$1/2 - \varepsilon$
$X_1 = F$	$1/2 - \varepsilon$	ε

若分布 $Q(X_1, X_2, X_3)$ 的取值均相等如表 2.2 所示，那么运行 IPFP 算法，在迭代 84 步后，算法收敛到表 2.2 所示的联合概率分布 $Q^*(X_1, X_2, X_3)$ 。

Table2.2 Result joint probability distribution

X_1	X_2	X_3	$Q(X_1, X_2, X_3)$	$Q^*(X_1, X_2, X_3)$
T	T	T	0.125	0.15
T	T	F	0.125	0.15
T	F	T	0.125	0.05
T	F	F	0.125	0.15
T	T	T	0.125	0.15
T	T	F	0.125	0.05
T	F	T	0.125	0.15
T	F	F	0.125	0.15

图 2.1 给出了每一步迭代求得联合概率 $Q_k(X)$ 与 $Q_0(X)$ 的 K-L 距离。其中 Q^* 与 Q_0 的 K-L 距离 $I(Q^* || Q_0) = 0.10453816$ 。

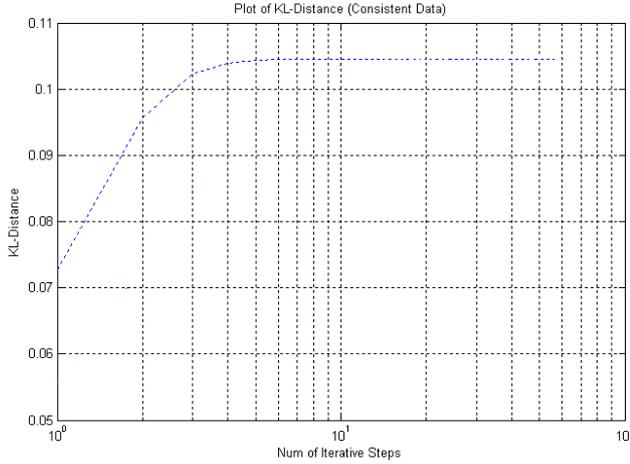


Fig. 2.1 K-L distance of $I(Q_k(X)||Q_0(X))$

2.2.2 C-IPFP 算法

Bock (Bock et al. 1989), Cramer (Cramer 2000) 等研究了 IPFP 算法，并将其进行了扩展，使其能够处理约束条件为条件概率分布的情况，即给定约束条件 $R(Y|Z)$ ，式 2.2.1 中对 I-Projection 的计算扩展为：

$$Q(X) = \begin{cases} 0 & \text{if } P(Y|Z) = 0 \\ P(X) \cdot \frac{R(Y|Z)}{P(Y|Z)} & \text{else} \end{cases} \quad (2.2.2)$$

由式 (2.2.2) 可见 IPFP 只是 C-IPFP 的一种特殊情况，即 $Z = \phi$ 。则扩展后的 C-IPFP 算法如下：

C-IPFP 算法：

1. 初始值： $Q_0(X)$, $R = \{R_1, R_2, \dots, R_m\}$
2. for $k = 1$, 重复以下迭代过程，直至收敛：
 - 2.1 $i = k \bmod m$;
 - 2.2 若 R_i 为联合概率，则使用式 (2.2.1) 计算 I-Projection，若 R_i 为条件概率，则使用式 (2.2.2) 计算 I-Projection;
 - 2.3 $k = k + 1$;

定理 2.2 若约束集 R 一致，那么算法 C-IPFP 收敛。记收敛结果为 $Q^*(X)$ ，记所有满足 R 的联合概率分布集合为 S_R ，则 $Q^*(X)$ 满足 R ，且在所有满足 R 的联合概率分布中， $Q^*(X)$ 具有与初始 $Q_0(X)$ 最小的 K-L 距离，即 $Q^*(X)$ 为 $Q_0(X)$

在 S_R 上的 I-Projection。

定理证明参见 (Cramer 2000, 定理 5.2)。

2.3 概率约束不一致时的知识合成算法

实验证明 (Vomlel 2003; Ding et al. 2005), IPFP 在约束集不一致时并不收敛, 而是在多个联合概率分布之间振荡。例如在例 2.1 中, 当 $\varepsilon = 3/20$ 时, 并不存在一个满足约束集的联合概率分布。则此时运行 IPFP 算法, 每一步迭代计算获得的联合概率分布中 $Q^*(X_1=1, X_2=1, X_3=1)$ 的值如图 2.2 所示:

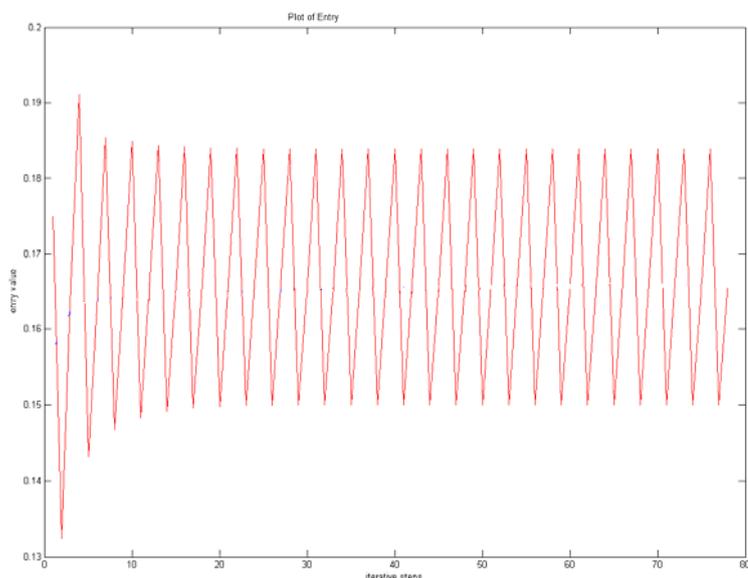


Fig. 2.2 Entry of $Q^*(X_1=1, X_2=1, X_3=1)$ for IPFP with inconsistent data

由图 2.2 可见, 在约束集不一致时, 目标分布中 $Q^*(X_1=1, X_2=1, X_3=1)$ 的值并不收敛, 即进行 IPFP 迭代获得的联合分布在固定的几个状态间振荡。为了实现约束集不一致时的知识合成, 需要寻找新的算法。且新的算法应该满足如下要求:

- I. 如果约束集一致, 则与 IPFP 收敛结果相同;
- II. 约束集不一致时, 算法同样收敛;
- III. 收敛结果不受约束条件顺序的影响;
- IV. 收敛结果的边缘概率应尽量靠近约束条件。

本节主要介绍两种处理约束集不一致的算法: CC-IPFP 和 GEMA 方法。

2.3.1 CC-IPFP 算法

定义 2.6 (全变差) 给定两个定义在 X 上的联合概率分布 $P(X)$ 和 $Q(X)$, 则其全变差 (total variation) 定义为:

$$\delta(P, Q) = \sum_{x \in X} |P(x) - Q(x)|$$

为解决约束不一致时知识合成的收敛问题, Vomlel 提出了一种 CC-IPFP 算法 (Vomlel 1999, 2003)。该算法描述如下:

CC-IPFP 算法:

1. 初始值: $Q_0(X)$, $R = \{R_1, R_2, \dots, R_m\}$
2. for $k = 1$, 重复以下迭代过程, 直至收敛:
 - 2.1 $i = k \bmod m$;

$$2.2 \quad Q_k(X) = \begin{cases} 0 & \text{if } Q_{k-1}(Y^i) = 0 \\ (1 - \alpha_k)Q_{k-1}(X) + \alpha_k Q_{k-1}(X) \cdot \frac{R(Y^i)}{Q_{k-1}(Y^i)} & \text{else} \end{cases};$$

- 2.3 $k = k + 1$;

其中 α_k 随迭代步骤 k 的增加而减小, 例如 $\frac{1}{1+k}$ 。下面我们给出 CC-IPFP 的收敛性证明。

定理 2.3 记初始概率分布为 $Q_0(X)$, 约束集为 $R = \{R_1, R_2, \dots, R_m\}$, $\alpha_k = \frac{1}{1+k}$, 则 CC-IPFP 算法收敛。

证明:

根据算法第 2.2 步, 对于 CC-IPFP 迭代前后的联合分布 $Q_{k-1}(X)$ 和 $Q_k(X)$, 有:

$$\begin{aligned}
 \delta(Q_k, Q_{k-1}) &= \sum_{x \in \mathbf{X}} |Q_k(x) - Q_{k-1}(x)| \\
 &= \sum_{x \in \mathbf{X}} |(1 - \alpha_k)Q_{k-1}(X) + \alpha_k Q_{k-1}(X) \cdot \frac{R(Y^i)}{Q_{k-1}(Y^i)} - Q_{k-1}(x)| \\
 &= \sum_{x \in \mathbf{X}} \alpha_k Q_k(x) \left| 1 - \frac{R(Y^i)}{Q_{k-1}(Y^i)} \right| \\
 &= \sum_{z \in \mathbf{X} \setminus \mathbf{Y}^i} \sum_{y^i \in \mathbf{Y}^i} \alpha_k Q_k(y^i, z) \left| 1 - \frac{R(Y^i)}{Q_{k-1}(Y^i)} \right| \\
 &= \sum_{y^i \in \mathbf{Y}^i} \alpha_k |Q_{k-1}(Y^i) - R(Y^i)| \\
 &= \alpha_k \delta(Q_{k-1}(Y^i), R(Y^i)) \\
 &\leq \alpha_k \cdot \sum_{y^i \in \mathbf{Y}^i} (Q_{k-1}(Y^i) + R(Y^i)) \\
 &= 2 \cdot \alpha_k
 \end{aligned}$$

即 CC-IPFP 每一步迭代前后的联合分布之间的全变差的上限为 $2 \cdot \alpha_k$ 。由于 $\alpha_k = \frac{1}{1+k}$ ，在迭代步骤 $k \rightarrow \infty$ 时， $\alpha_k \rightarrow 0$ ，即 $Q_k \xrightarrow{k \rightarrow \infty} Q_{k-1}$ ，意味着算法 CC-IPFP 收敛。

文献(1)指出 CC-IPFP 算法收敛结果 $Q^*(X)$ 满足 (Conjecture 2):

$$\sum_{j=1}^m \delta(Q^*(Y^j), R(Y^j)) = \min_{\tilde{Q}} \sum_{j=1}^m \delta(\tilde{Q}(Y^j), R(Y^j))$$

即 $Q^*(X)$ 与约束条件的全变差之和最小化。与最小化 K-L 距离不同，最小化全变差之和的联合概率分布并不是唯一的，而是一个联合概率分布的凸集 (Vomlel 1999, 表 8)。

2.3.2 GEMA 算法

Vomlel 提出了另外一种算法 (Vomlel 2003)，GEMA。算法描述如下：

GEMA 算法：

1. 初始值： $Q_0(X)$, $R = \{R_1, R_2, \dots, R_m\}$
2. for $k=1$ ，重复以下迭代过程，直至收敛：

2.1 $i = k \bmod m$ do

$$2.2 \quad Q'_k(X) = \sum_{j=1}^m \omega_j Q_{k-1}(X) \cdot \frac{R(Y^i)}{Q_{k-1}(Y^i)};$$

2.3 for $j = 1$ to m

2.3.1 由 $Q'_k(X)$ 生成新的约束集 $R' = \{R'_1, R'_2, \dots, R'_m\}$;

2.4 for $j = 1$ to m

$$2.4.1 \quad Q_k(X) = \begin{cases} 0 & \text{if } Q_{k-1}(Y^i) = 0 \\ Q_{k-1}(X) \cdot \frac{R'(Y^i)}{Q_{k-1}(Y^i)} & \text{else} \end{cases};$$

2.4.2 $k = k + 1$;

其中 $0 < \omega_j < 1$, 且 $\sum_{j=1}^m \omega_j = 1$ 。

定理 2.4 GEMA 算法在约束集 R 一致不一致时均收敛, 记收敛结果为 $Q^*(X)$, 则 $Q^*(X)$ 最小化与约束条件的 K-L 距离和 (I-Aggregate), 即

$$\sum_{j=1}^m \omega_j \cdot I(Q^*(Y^j) \| R(Y^j)).$$

定理证明参见 (Vomlel 1999)。

例 2.2 根据例 2.1, 约束集一致时 ($\varepsilon = 4/20$), GEMA 在迭代 194 步后, 收敛于和 IPFP 相同结果¹。图 2.3 给出了每一步迭代结果与初始联合概率分布的 K-L 距离, 其中蓝线为 IPFP 结果, 红线为 GEMA 结果。

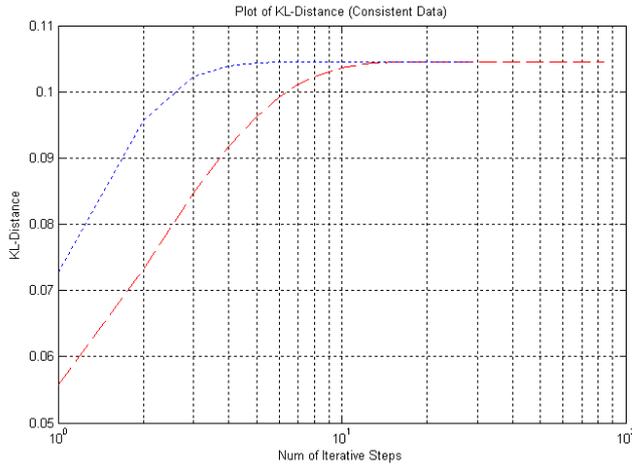


Fig. 2.3 K-L distance of $I(Q_k(X) \| Q_0(X))$ for IPFP and GEMA

¹ 根据 GEMA 算法, 每一步 GEMA 迭代包含 $2m$ 次 I-Projection 计算, 本例中 194 次 GEMA 迭代包含了 1164 步 I-Projection 计算。

例 2.3 根据例 2.1，约束集不一致时 ($\varepsilon = 3/20$)，收敛条件为每一步 GEMA 迭代前后联合概率分布的全变差小于 10^{-12} ； $\omega_j = 1/m$ （本例中为 $1/3$ ），收敛步数为 1290741（同样，这里的每一步包含了 $2 \times m$ 次 I-Projection 计算）图 2.4 给出了每一步迭代结果与初始联合概率分布的 K-L 距离。

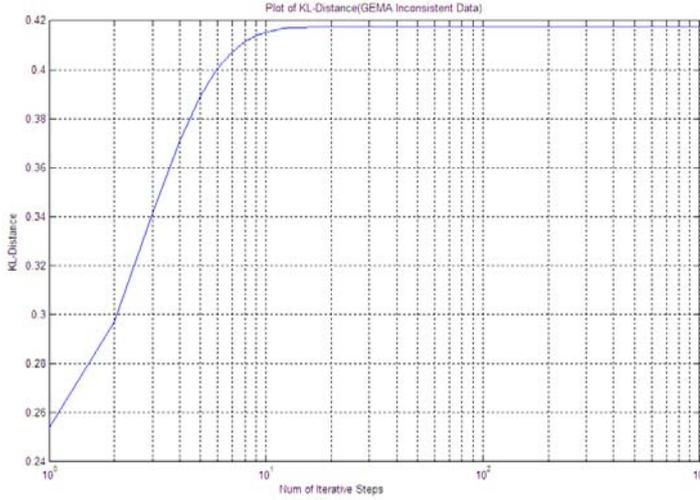


Fig. 2.4 K-L distance of $I(Q_k(X)||Q_0(X))$ for GEMA

2.3.3 SMOOTH 算法

实验证明 CC-IPFP 算法和 GEMA 算法计算复杂度高，且算法收敛速度慢。表 2.3 给出了例 2.1 中约束集不一致时两种算法的收敛步数。

算法	收敛步数	K-L 距离	I-Aggregate
GEMA	1290741	0.41502431	0.00367169
CC-IPFP	>10,000,000	0.37048603	0.00461839

其中对 GEMA 算法， $\omega_j = 1/m$ ；对 CC-IPFP 算法， $\alpha_k = 1/(k+1)$ ，且在算法迭代步数达到 $1.0E7$ 时仍未收敛。且根据文献 (Zhang et al. 2008)，GEMA 算法收敛速度受初始联合概率分布影响很大。

CC-IPFP 和 GEMA 算法，在迭代过程中只是根据约束集来修改目标联合概率分布。然而由于约束集之间的不一致性，使得迭代过程最后收敛非常慢(图 2.4)。其原因为：在迭代开始时，目标联合概率分布中包含约束集中的信息部分越来

越多，随着迭代过程的继续，包含在约束条件中一致性的部分完全合成到联合概率分布中，而此时 K-L 距离受约束条件之间不一致部分的影响则逐渐增大。为此，我们提出了一个新的算法，其核心思想为：在迭代过程中，双向修改约束集和联合概率分布，在迭代过程中同时实现对约束集的合成—和降低约束条件之间的不一致度。算法描述如下：

SMOOTH 算法：

1. 初始值： $Q_0(X)$, $R = \{R_1, R_2, \dots, R_m\}$
2. 做标准 IPFP 迭代，若收敛，[则推出算法则退出算法](#)；否则进入步骤 3，记当前联合概率分布为 $Q_0(X)$ ；
3. for $k = 1$ ，重复以下迭代过程，直至收敛：
 - 3.1 $i = k \bmod m$ do
 - 3.2 $R_i(Y^i) = (1 - \alpha)R(Y^i) + \alpha Q_{k-1}(Y^i)$ ；
 - 3.3 $Q_k(X) = \begin{cases} 0 & \text{if } Q_{k-1}(Y^i) = 0 \\ Q_{k-1}(X) \cdot \frac{R(Y^i)}{Q_{k-1}(Y^i)} & \text{else} \end{cases}$ ；
 - 3.4 $k = k + 1$ ；

根据算法描述可知，SMOOTH 算法分两个阶段，第一阶段为标准 IPFP 迭代，若收敛则退出算法，否则进入第二阶段直至收敛。下面我们给出只有两个约束条件时，SMOOTH 算法的收敛证明。

定理 2.5 给定初始分布 $P(X)$ ，不一致约束条件 $R(Y^1)$ 和 $R(Y^2)$ ， $\alpha \in (0,1)$ ，则 SMOOTH 收敛。

证明：

由于 IPFP 在约束集不一致时进入振荡，则不失一般性，记 SMOOTH 算法进入第二步时的初始分布为 $Q_0(X)$ ， $Q_0(X)$ 满足 $R(Y^2)$ ，且第二步迭代的初始约束条件为 $R_0(Y^1) = R(Y^1)$ 和 $R_0(Y^2) = R(Y^2)$ 。令 $Q_1(x)$ 为 $Q_0(X)$ 在 $\mathbf{P}_{R_0(Y^2)}$ 上的 I-Projection， $Q_2'(X)$ 为 $Q_1(x)$ 在 $\mathbf{P}_{R_0(Y^2)}$ 上的 I-Projection， $Q_2(X)$ 为 $Q_1(x)$ 在 $\mathbf{P}_{R_1(Y^1)}$ 上的 I-Projection。则根据算法第 3.2 步，有

$$R_1(y^2) = \alpha R_0(y^2) + (1 - \alpha)Q_1(y^2)$$

，如图 2.5 所示。

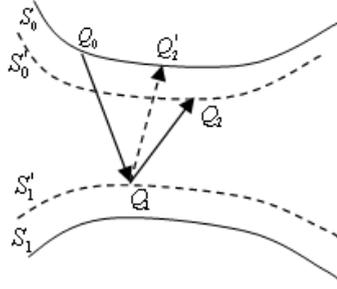


Fig. 2.5 First two iterations of SMOOTH phase 2

因为 $Q_2'(X)$ 是 $Q_1(x)$ 在 $\mathbf{P}_{R_0(y^2)}$ 上的 I-Projection, 则

$$\begin{aligned} \delta(Q_1, Q_2') &= \sum_{x \in \mathbf{X}} \left| Q_1(x) - Q_1(x) \frac{R(y)}{Q_1(y)} \right| \\ &= \sum_{x \in \mathbf{X}} Q_1(x) \left| 1 - \frac{R(y)}{Q_1(y)} \right| = \sum_{y \in \mathbf{Y}} \sum_{z \in \mathbf{X} \setminus \mathbf{Z}} Q_1(y, z) \left| 1 - \frac{R(y)}{Q_1(y)} \right| \\ &= \sum_{y \in \mathbf{Y}} Q_1(y) \left| 1 - \frac{R(y)}{Q_1(y)} \right| = \sum_{y \in \mathbf{Y}} |Q_1(y) - R(y)| \end{aligned}$$

而

$$\begin{aligned} \delta(Q_0, Q_1) &= \sum_{x \in \mathbf{X}} |Q_0(x) - Q_1(x)| \\ &= \sum_{y \in \mathbf{Y}} \sum_{z \in \mathbf{X} \setminus \mathbf{Y}} |Q_0(x) - Q_1(x)| \geq \sum_{y \in \mathbf{Y}} \left| \sum_{z \in \mathbf{X} \setminus \mathbf{Y}} Q_0(x) - \sum_{z \in \mathbf{X} \setminus \mathbf{Y}} Q_1(x) \right| \\ &= \sum_{y \in \mathbf{Y}} |Q_0(y) - Q_1(y)| = \sum_{y \in \mathbf{Y}} |Q_0(y) - Q_1(y)| \end{aligned}$$

所以有 $\delta(Q_1, Q_2') \leq \delta(Q_0, Q_1)$ 。由 $Q_2'(x) = Q_1(x) \frac{R_0(y^2)}{Q_0(y^2)}$, 和

$$\begin{aligned} Q_2(x) &= Q_1(x) \frac{R_1(y^2)}{Q_0(y^2)} \\ &= \alpha Q_1(x) \frac{R_0(y^2)}{Q_1(y^2)} + (1-\alpha) Q_1(x) \\ &= \alpha Q_2'(x) + (1-\alpha) Q_1(x), \end{aligned}$$

得

$$\begin{aligned} \delta(Q_1, Q_2) &= \sum_{x \in \mathbf{X}} |Q_1(x) - Q_2(x)| \\ &= \sum_{x \in \mathbf{X}} |Q_1(x) - (\alpha Q_2'(x) + (1-\alpha) Q_1(x))| \\ &= \sum_{x \in \mathbf{X}} \alpha |Q_1(x) - Q_2'(x)| \end{aligned}$$

由于 $0 < \alpha < 1$, 所以 $\delta(Q_1, Q_2) < \delta(Q_0, Q_1)$, 即 $\delta(Q_{k-1}, Q_k)$ 递减, 所以 SMOOTH 算法收敛。

例 2.4 根据例 2.1 已知条件，当 R 为一致时，SMOOTH 退化为标准 IPFP；当 R 不一致时，设置平滑因子 $\alpha = 0.01$ ，收敛步数为 4002 (177+3825，IPFP 在第 177 后进入振荡，即从第 178 步迭代开始进入 SMOOTH 算法的第二阶段)。图 2.6 给出了每一步迭代结果与初始联合概率分布的 K-L 距离。

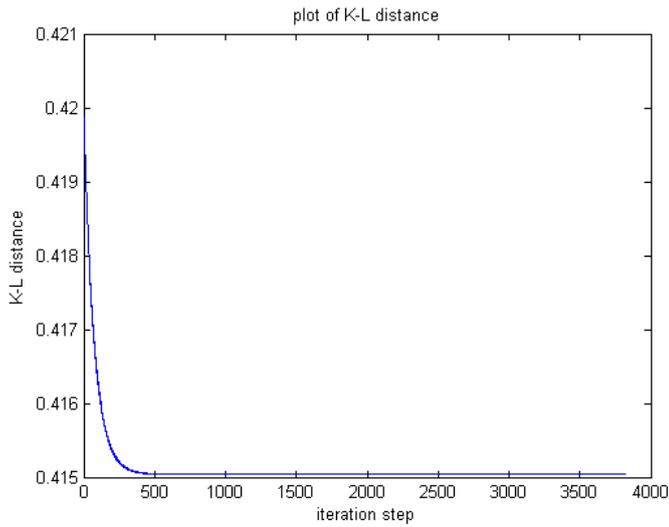


Fig. 2.6 K-L distance of $I(Q_k(X)||Q_0(X))$ for SMOOTH (from 2nd phase)

根据图 2.6，SMOOTH 随着迭代过程的继续，算法收敛过程变慢。由于此时约束集中的信息大部分已经被合成到目标联合概率分布中，所以可以增大平滑因子，来实现算法的加速收敛。例如可以设置平滑因子为 sigmoid 函数：

$$\alpha = \frac{1}{1 + \exp(A - k / B)} \quad (2.3.1)$$

其中 k 为迭代步数。由 (2.3.1) 所示的平滑因子随迭代步数的变化如图 2.7 所示。

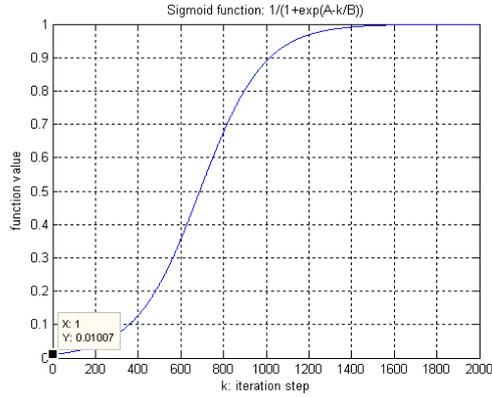


Fig. 2.7 Smooth ratio α vs iteration step

图 2.7 中, α 初始值设置为 0.01, $A = 4.595$, $B = 150$ 。在开始和结束部分变化很慢, 中间部分变化很快, 而且随着迭代步数增加, α 趋近于 1。其中参数 A 越大, α 前半部分的增加速度越慢; B 越小, 中间部分增加速度越快。使用上述 sigmoid 函数对 SMOOTH 加速, 根据例 2.1 中已知条件, 当 R 不一致时, 算法收敛进入振荡后只需 375 步即可收敛, 仅为不加速的十分之一。其与初始全概率的中的 K-L 距离如图 2.8 所示:

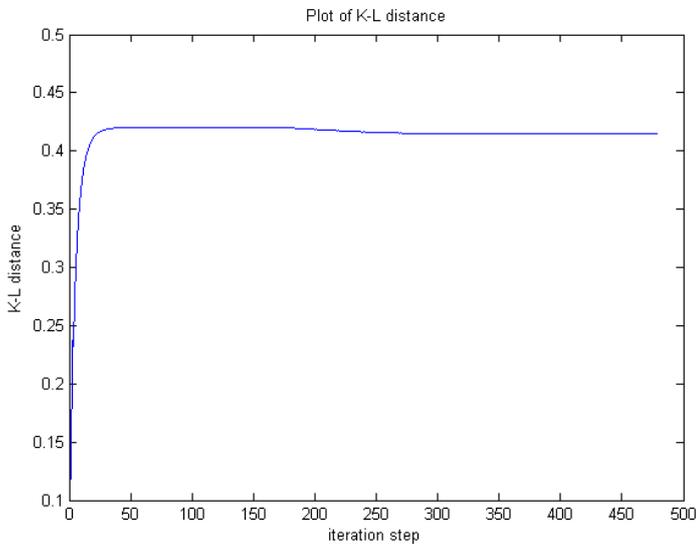


Fig. 2.8 K-L distance of $I(Q_t(X)||Q_0(X))$ for SMOOTH (accelerated)

2.3.4 实验结果对比与分析

本节我们通过实验比较几种算法的收敛性能。主要对比本章介绍的几种算法：IPFP, CC-IPFP, GEMA, SMOOTH 和使用 sigmoid 函数加速收敛的 SMOOTH (简记为 A-SMOOTH)。以下实验中, 对于 CC-IPFP, 我们取 $\alpha_k = 1/(1+k)$; 对算法 GEMA, 取 $\omega_j = 1/m$; 对算法 SMOOTH, 取 $\alpha = 0.01$; 对 A-SMOOTH, α 初始值设置为 0.01, $A = 4.595$, $B = 150$ 。

根据例 2.1 所给已知条件, 当 R 一致时, 各算法收敛情况如表 2.4 所示¹。由此可见, 在约束一致时, 各算法收敛于相同结果。其中 SMOOTH 和 A-SMOOTH 收敛速度较快, CC-IPFP 收敛速度最慢。根据算法的描述, 此时 SMOOTH 和 A-SMOOTH 退化为标准 IPFP 算法。

Table2.4 Convergence result of 4 algorithms for consistent constraints

Algorithm	GEMA	CC-IPFP	SMOOTH	A-SMOOTH
Fitting steps	1164	3507	84	84
KL Distance	0.10453816	0.10453816	0.10453816	0.10453816

当 R 不一致时, GEMA 算法和 CC-IPFP 算法收敛速度非常慢 (表 2.3), 且 GEMA 算法收敛受初始联合概率分布影响很大。为此, 我们设计如下实验, 取两组初始联合概率分布 (如表 2.5 所示), 一组不一致约束集 (例 2.1 中取 $\varepsilon = 3/20$, 记为 R_1)。

Table2.5 Two initial joint probability distributions

X_1	X_2	X_3	$Q_1(X_1, X_2, X_3)$	$Q_2(X_1, X_2, X_3)$
T	T	T	0.125	0.25
T	T	F	0.125	0.0
T	F	T	0.125	0.23
T	F	F	0.125	0.08
T	T	T	0.125	0.11
T	T	F	0.125	0.13
T	F	T	0.125	0.06
T	F	F	0.125	0.14

分别运行四组算法, 收敛情况如下表所示。根据表 2.6 结果, 算法 CC-IPFP 收敛速度最慢, GEMA 次之, A-SMOOTH 收敛最快。其中, GEMA 收敛结果具有最小的 I-Aggregate; CC-IPFP 算法收敛结果具有最小的 K-L 距离, 但是具有最大的 I-Aggregate, 原因为 $\alpha_k = 1/(1+k)$ 在开始时减小太快, 使得迭代过程中较多地丢失了约束条件的信息。比较 GEMA、SMOOTH 和 A-SMOOTH 的收敛结果, 可以

¹ 本章以下内容中收敛步数均以单步 I-Projection 计算为单位, 例如本例中 GEMA 算法的每一步迭代包含了 6 (2×3) 次 I-Projection 计算。

近似认为三种算法收敛到相同结果，很显然在收敛速度方面，SMOOTH 和 A-SMOOTH 比 GEMA 收敛速度快得多。

Table 2.6 Comparison of convergence result for inconsistent constraint R_I

	Fitting- step I-projections	KL Distance	I-aggregate
GEMA			
JPD-1	7,744,446	0.41502431	0.00367169
JPD-2	9,064,080	0.71979040	0.05727919
CC- IPFP			
JPD-1	>10,000,000	0.37048603	0.00461839
JPD-2	>10,000,000	0.70127029	0.05742945
SMOOTH			
JPD-1	177+3825	0.41503774	0.00367172
JPD-2	129+4899	0.71306584	0.05729201
A-SMOOTH			
JPD-1	177+375	0.41503891	0.00367227
JPD-2	129+402	0.71439294	0.05729532

图 2.9 和 2.10 分别给出了从 SMOOTH 算法进入第二步时，三种算法的 I-Aggregate 和 K-L 距离对比。

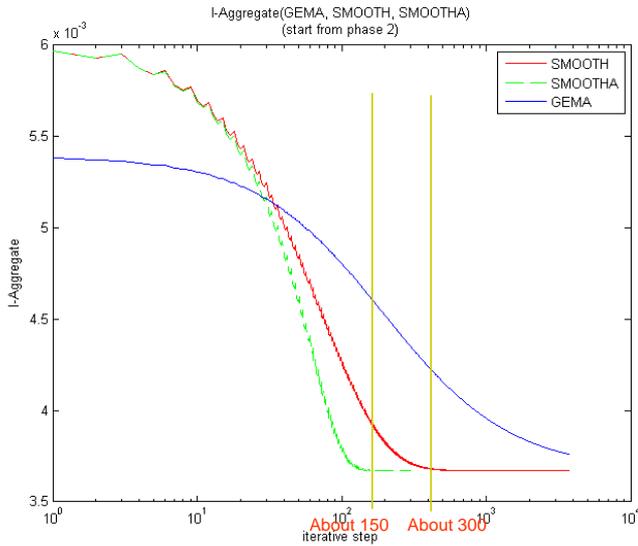


Fig. 2.9 Comparison of I-Aggregate

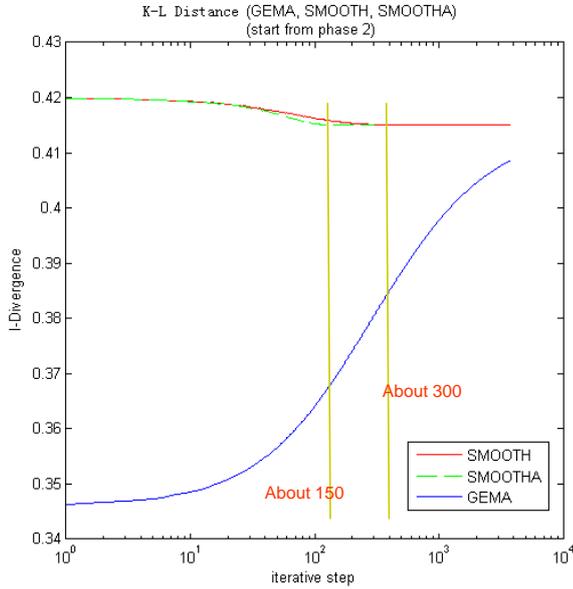


Fig. 2.10 Comparison of K-L distance

我们设计了分别具有 8 个和 15 个随机变量规模的初始联合概率分布，并得到了同样的实验结果（表 2.7 和 2.8）。

Table2.7 Result of an joint probability distribution (8 variables)

	<u>Fitting</u> <u>step</u> <u>I-projections</u>	KL Distance	I-aggregate
GEMA	912	0.93720845	0.02345286
SMOOTH	48+5000	0.94365473	0.02347122
A-SMOOTH	48+568	0.94366720	0.02347214

Table2.8 Result of an joint probability distribution (15 variables)

	<u>I-projections</u> <u>Fitting</u> <u>step</u>	KL Distance	I-aggregate
GEMA	617	0.45972134	0.03407491
SMOOTH	1736+5460	0.45978419	0.03408528
A-SMOOTH	1736+584	0.45989650	0.03408416

综合以上实验结果，算法 SMOOTH 具有以下特点：

- I. 在约束集一致时，退化为标准的 IPFP 算法。其收敛结果满足给定约束集，且最小化与初始分布的 K-L 距离。与已有算法（CC-IPFP，GEMA）相比，SMOOTH 具有更快的收敛速度。见表 2.4。
- II. 当约束集不一致时，SMOOTH 算法同样收敛。其收敛结果具有与初始分布较小的 K-L 距离和 I-Aggregate。且可是通过调整平滑因子实现算法的快速收敛。

见表 2.6, 2.7, 2.8 和图 2.9, 2.10。

2.4 小结

本章主要研究给定约束条件下的概率知识合成算法。首先详细介绍了概率约束一致时的 IPFP 及 CIPFP 算法。IPFP 算法通过迭代使用约束集中的约束条件, 每一步迭代计算出满足当前约束条件的联合概率分布, 并最终收敛到满足所有约束条件的联合分布。该联合分布满足所有约束条件, 并且在所有满足约束条件的联合分布中, IPFP 算法的收敛结果具有与初始分布最小化的 K-L 距离。

当约束集不一致时, IPFP 并不收敛而是在固定的联合分布之间振荡。为此, 我们介绍了 Vomlel 提出的 CC-IPFP 及 GEMA 算法, 并给出了 CC-IPFP 的收敛性证明。实验结果表明, CC-IPFP 算法收敛速度非常慢; GEMA 收敛速度不稳定, 受初始条件影响较大。为此我们提出了一个新的 SMOOTH 算法。算法核心思想为双向修改目标分布与约束条件, 每一步迭代利用目标联合分布对约束条件进行平滑操作, 从而降低约束条件之间的不一致度, 实现算法的收敛。实验结果表明, SMOOTH 算法在约束集一致时, 退化为标准 IPFP 算法; 在约束集不一致时, SMOOTH 算法与 GEMA 算法具有相同的收敛结果。且 SMOOTH 不受初始值的影响, 算法收敛速度稳定, 与 GEMA 相比收敛速度快。同时可以调整平滑因子实现算法的加速收敛。

第3章 贝叶斯网知识合成算法研究

本章首先介绍子现有的贝叶斯网知识合成算法 E-IPFP, D-IPFP, 讨论子算法的性质, 并重新给出子算法的收敛性证明; 然后同时通过引入不一致时的知识合成算法 SMOOTH, 对 E-IPFP 进行子改进, 以解决子算法在约束集不一致时的收敛问题; 最后并对改进后的算法给出子理论和实验上的分析。

3.1 问题描述

在贝叶斯网络问题的研究中, 一个经常碰到的问题为, 在已知概率知识的前提下, 如何将给定概率知识合成到已有的贝叶斯网络中。此问题在设计贝叶斯网络, 将若干小的贝叶斯网络合并为一个大的网络, 或者根据新的更可靠的概率知识更新已有贝叶斯网络时尤为常见。

给定一个贝叶斯网络 G , G 包含了 n 个随机变量节点 $X=(X_1, X_2, \dots, X_n)$ 。记随机变量 x_i 的父节点集合为 π_i , 则网络结构可以表示为 $G_s = \{(x_i, \pi_i)\}$; 记随机变量节点的条件概率表集合为 $G_p = \{P(x_i | \pi_i)\}$, 则贝叶斯网 G 可以表示为:

$$G = (G_s, G_p) \quad (3.1.1)$$

根据链式规则, 贝叶斯网络 G 确定了一个联合概率分布:

$$P(x) = \prod_{i=1}^n P(x_i | \pi_i) \quad (3.1.2)$$

蕴含于 G 的联合概率分布可以描述给定的概率知识库, 因此可以将贝叶斯网络作为问题领域的知识模型。例如在医疗诊断中, 医生往往需要根据患者表现出的症状对病人所患疾病做出诊断。那么该模型就可以用一个贝叶斯网络 $G=(G_s, G_p)$ 表示, 其中:

$$G_s = \{(症状 1, 疾病), \dots, (症状 n, 疾病)\};$$

$$G_p = \{P(症状 1 | 疾病), \dots, P(症状 n | 疾病)\}.$$

根据 G 所确定的模型, 即可通过贝叶斯网络推理求解在观察患者症状情况下所患疾病的概率。在该模型中, $P(症状 i | 疾病)$ 为给定患者已患该疾病的情况下具有症状 i 的条件概率, 其值一般由专家给出。而在实际应用中, 根据临床观察或者统计等其他途径会获得新的关于症状与疾病之间的概率知识, 而这些概率知识通常只涉及到整个模型的一部分, 如只涉及到症状 3 和 4 等。那么就需要将这些新的概率知识合成到原有的贝叶斯网络中, 使更新后的贝叶斯网络满足新的概率知识。

Formatted: Font: Italic

Formatted: Font: Italic

相应地，根据 2.1 节对知识合成的定义，我们给出贝叶斯网知识合成的定义：

定义 3.1 (贝叶斯网知识合成) 通过修改贝叶斯网络节点参数，使其满足给定概率约束集的过程¹。

Comment [Y6]: 前面没有给出节点参数的定义 (e.g., CPT)

由此我们的问题模型可以描述为：

已知贝叶斯网络 $G = (G_s, G_p)$ 和概率约束集 $R = \{R(Y^1), R(Y^2), \dots, R(Y^m)\}$ ，构造新的贝叶斯网络 $G' = (G_s, G'_p)$ ，使得由 G' 确定的联合概率分布 $P'(x)$ 满足给定的约束集，且 $P'(x)$ 与 $P(x)$ 之间的距离最小。

3.2 贝叶斯网知识合成算法

根据式 (3.1.2) 所示链式规则，每一个贝叶斯网确定了一个联合概率分布。在给定约束的情况下，如何利用知识合成算法将概率知识合成到贝叶斯网中，仍然是一个待解决的问题。我们可以用 IPFP 生成一个满足 R 的联合概率，但是如果简单地根据此联合概率通过边缘化来生成贝叶斯网随机变量节点的条件概率表，那么通过贝叶斯网链式规则由这些条件概率表得到的联合概率分布与给定联合概率分布一般是不同的。

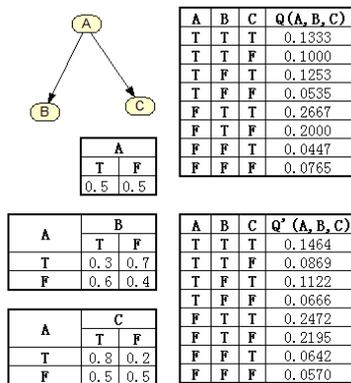


Fig. 3.1 An simple BN example with three variable

例 3.1 如图 3.1，给出了一个有三个随机变量的贝叶斯网的例子。其中左边为贝叶斯网络结构和初始条件概率表，右上 $Q(A,B,C)$ 为使用约束条件 $P(B, C) = (0.4, 0.3, 0.17, 0.13)$ 之后获得的联合概率分布。如果根据 $Q(A,B,C)$ 按照网

¹本章不考虑网络结构的变化，即本章讨论的知识合成算法是在网络结构给定的情况下进行的（假设网络结构能够反映问题模型）。

络结构生成新的条件概率表,那么根据链式规则生成的联合概率分布为图 3.1 中的 $Q'(A,B,C)$ 。比较可见, $Q'(A,B,C)$ 并不满足约束条件 $P(B,C)$ 。这是因为 IPFP 知识合成方法并未考虑由 BN 网络结构所确定的随机变量间的条件独立关系。

Comment [Y7]: This sentence does not read well, you probably need to rewrite it.

3.2.1 E-IPFP 算法

为解决约束条件下 BN 的知识合成问题, Peng 和 Ding 将 IPFP 引入贝叶斯网络中,提出了一种进行贝叶斯网络知识合成的算法, E-IPFP 算法 (Peng et al. 2005)。该算法将贝叶斯网络的结构 G_s (随机变量间的条件独立关系) 作为新的约束条件,在进行 IPFP 算法中,同时使用约束集中的约束条件和网络结构约束条件 (记为 R_s),通过迭代使收敛结果同时满足约束集和网络结构。

在 IPFP 每一轮迭代结束时,根据当前联合概率分布 $Q_k(x)$,按照网络结构 G_s 计算贝叶斯网络中所有节点的条件概率表 G_P ,

$$P(x_i | \pi_i) = \frac{Q_k(x_i, \pi_i)}{Q_k(\pi_i)} \quad (3.2.1)$$

其中 $Q_k(x_i, \pi_i)$ 和 $Q_k(\pi_i)$ 可以由 $Q_k(x)$ 通过计算边缘概率分布获得。通过 (3.2.1) 式计算更新后贝叶斯网络,根据链式规则 (3.1.2),决定了一个新的联合概率分布,该联合概率分布为下一轮迭代的初值。算法描述如下:

E-IPFP 算法:

1. 初始值: $Q_0(x) = \prod_{i=1}^n Q_0(x_i | \pi_i)$, $R = \{R_1, R_2, \dots, R_m\}$;
2. for $k = 1$, 重复以下迭代过程,直至收敛;
 - 2.1 $i = ((k-1) \bmod (m+1)) + 1$;
 - 2.2 if $i < m+1$, $Q_k(x) = Q_{k-1}(x) \cdot \frac{R(y^i)}{Q_{k-1}(y^i)}$;
 - 2.3 else
由 $Q_{k-1}(x)$ 根据网络结构生成 $Q_{k-1}(x_i | \pi_i)$;
 $Q_k(x) = \prod_{i=1}^n Q_{k-1}(x_i | \pi_i)$;
 - 2.4 $k = k + 1$;

文献 (Peng et al. 2005) 中对于 E-IPFP 的收敛性说明是不完善的¹: 由于算法的第 2.3 步按照网络结构对当前联合概率分布的分解,并不是一个

¹ Y. Peng et al. 2005. Modifying Bayesian Networks by Probability Constraints, page 4, Section 4.1.

I-Projection 计算(例 3.1), 所以对于 E-IPFP 算法的循环计算¹, 并不满足 IPFP 的收敛属性 (Vomlel 1999, 2003)。下面我们给出 E-IPFP 的证明。

引理 3.1 若已知初始联合概率分布 $Q_0(x)$ 和一致约束集 $R = \{R(Y^1), R(Y^2), \dots, R(Y^m)\}$, $Q^*(x)$ 为 IPFP 算法以 $Q_0(x)$ 为初值, R 为约束集的收敛结果。记 $Y = Y^1 U Y^2 U \dots U Y^m$, $Q^*(y)$ 为 IPFP 算法以 $Q_0(y)$ 为初值, R 为约束集的收敛结果。则下式成立:

$$Q^*(x) = Q_0(x) \frac{Q^*(y)}{Q_0(y)}.$$

[证明]

根据 2.2.1 节 IPFP 算法描述, 对于每一步 I-Projection 计算, 有

$$\begin{aligned} Q_k(x) &= Q_{k-1}(x) \cdot \frac{R(y^j)}{Q_{k-1}(y^j)} \\ &= Q_{k-1}(x|y) \cdot Q_{k-1}(y) \frac{R(y^j)}{Q_{k-1}(y^j)}. \\ &= Q_{k-1}(x|y) \cdot Q_k(y) \end{aligned}$$

注意到每一步计算中保持 $Q_{k-1}(x|y)$ 不变, 即 $Q_{k-1}(x|y) = Q_0(x|y)$ 。根据定义 2.5 及式 (2.2.1), $Q_k(y)$ 为 $Q_{k-1}(y)$ 在 $R(y^j)$ 上的一个 I-Projection。由于 R 在 Y 上收敛于 $Q^*(y)$, 即

$$Q^*(x) = Q_0(x|y) \cdot Q^*(y) = Q_0(x) \frac{Q^*(y)}{Q_0(y)}.$$

根据引理 3.1, 在约束集 R 一致时, IPFP 算法等效为求 $Q_0(x)$ 在 $Q^*(y)$ 上的 I-Projection, 即约束集 R 等效为单个约束条件 $Q^*(y)$, 记该约束条件为 $R(y)$ 。

定理 3.1 给定贝叶斯网 G 和约束集 $R = \{R(Y^1), R(Y^2), \dots, R(Y^m)\}$ 。若约束集 R 一致, 且与网络结构一致, 则 E-IPFP 算法收敛。

[证明]

由引理 3.1, 不失一般性, E-IPFP 等效为只有两个约束条件 $R(y)$ 和结构约束条件 $\prod_{i=1}^n Q_k(x_i|\pi_i)$ 的迭代计算。记 $Q_0(x) = \prod_{i=1}^n Q_0(x_i|\pi_i)$, $Q_1(x) = Q_0(x) \frac{R(y)}{Q_0(y)}$, $Q_2(x) = \prod_{x_i \in x} Q_1(x_i|\pi_i)$, $Q_3(x) = Q_2(x) \frac{R(y)}{Q_2(y)}$ 。则 E-IPFP 算法可图示为:

¹ 这里的一个循环是指所有约束条件 (包括约束集中的约束条件和网络结构约束条件) 使用一次。

Formatted: Highlight

Formatted: Highlight

Formatted: Highlight

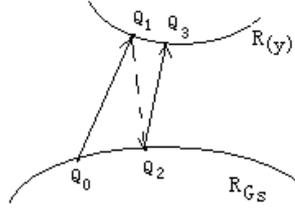


Fig. 3.2 Successive JPDs from E-IPFP

要证明 E-IPFP 收敛, 只需证明

$$I(Q_1 \parallel Q_0) \geq I(Q_3 \parallel Q_2) \quad (3.2.2)$$

即 $Q_{2k+1}(x)$ 与 $Q_{2k}(x)$ 的 K-L 距离在 E-IPFP 迭代过程中递减即可。由于 $Q_1(x)$ 和 $Q_3(x)$ 均满足 $R(y)$, 且 $Q_3(x)$ 为 $Q_3(x)$ 在 $R(y)$ 上的 I-Projection, 根据 I-Projection 的性质 (Csiszar 1975; Vomlel 2003), 有 $I(Q_1 \parallel Q_2) \geq I(Q_3 \parallel Q_2)$ 。所以, 要证明 (3.2.2), 只需证明

$$\Delta(x) = I(Q_1 \parallel Q_0) - I(Q_1 \parallel Q_2) \geq 0 \quad (3.2.3)$$

即可。下面我们用归纳法证明 E-IPFP 的收敛性。

记 $|X|$ 为给定贝叶斯网 G 的随机变量节点个数。

当 $|X|=1$ 时:

$X = \{X_1\}$, 约束为 $R(x_1)$, 显然 $Q_2(x_1) = Q_1(x_1) = R(x_1)$ 。根据 (3.2.3) 有

$$\Delta(x) = \sum R(x_1) \log \frac{R(x_1)}{Q_0(x_1)} = I(R(x_1) \parallel Q_0(x_1))。$$

根据 K-L 距离的非负性 (Kullback et al. 1951), 有

$$\Delta(x) \geq 0。$$

归纳假设: 对任意 $n \geq 1$, 有 $\Delta(x_1, x_2, \dots, x_n) \geq 0$ 。

归纳证明:

不失一般性, 假设 X_0 为贝叶斯网根节点, 记 $x = (x_1, x_2, \dots, x_n)$ 。则由 (3.2.3) 式,

$$\begin{aligned} \Delta(x_0, x) &= \sum_{x_0, x} Q_1(x_0, x) \log \frac{Q_2(x_0, x)}{Q_0(x_0, x)} \\ &= \sum_{x_0, x} Q_1(x_0, x) \log \frac{Q_1(x_0) Q_2(x | x_0)}{Q_0(x_0) Q_0(x | x_0)} \quad (3.2.4) \\ &= \sum_{x_0, x} Q_1(x_0, x) \log \frac{Q_1(x_0)}{Q_0(x_0)} + \sum_{x_0, x} Q_1(x_0, x) \log \frac{Q_2(x | x_0)}{Q_0(x | x_0)} \\ &= \Delta_1(x_0, x) + \Delta_2(x_0, x) \end{aligned}$$

其中,

$$\begin{aligned}
 & \Delta_1(x_0, x) \\
 &= \sum_{x_0, x} Q_1(x_0, x) \log \frac{Q_1(x_0)}{Q_0(x_0)} \\
 &= \sum_{x_0} \left(\sum_x Q_1(x_0, x) \right) \log \frac{Q_1(x_0)}{Q_0(x_0)} \quad (3.2.5) \\
 &= \sum_{x_0} Q_1(x_0) \log \frac{Q_1(x_0)}{Q_0(x_0)} = I(Q_1(x_0) \| Q_0(x_0)) \geq 0
 \end{aligned}$$

下面分情况讨论 $\Delta_2(x_0, x)$ 。

I. $x_0 \in y$

记 $y' = y \setminus \{x_0\}$ ，则 $R(y) = R(x_0, y')$ 。由于 Q_1 为 Q_0 在 $R(x_0, y')$ 上的 I-Projection，则 $Q_1(x_0) = R(x_0)$ 。根据

$$\begin{aligned}
 Q_1(x_0, x) &= Q_1(x_0) \cdot Q_1(x | x_0) \text{ 和} \\
 Q_1(x_0, x) &= Q_0(x_0, x) \cdot \frac{R(x_0, y')}{Q_0(x_0, y')} \\
 &= Q_0(x_0) \frac{R(x_0)}{Q_0(x_0)} Q_0(x | x_0) \frac{R(y' | x_0)}{Q_0(y' | x_0)},
 \end{aligned}$$

我们有

$$Q_1(x | x_0) = Q_0(x | x_0) \frac{R(y' | x_0)}{Q_0(y' | x_0)}.$$

注意到对于变量 x_0 的任意状态 x_0^* ， $Q_0(x | x_0^*) = \prod_{x_i \in x} Q_0^*(x_i | \pi_i)$ 表示了一个关于 X 的贝叶斯网，其中

$$Q_0^*(x_i | \pi_i) = \begin{cases} Q_0(x_i | \pi_i, x_0 = x_0^*) & \text{if } x_i \text{ is a child of } x_0^*; \\ Q_0(x_i | \pi_i) & \text{otherwise.} \end{cases} \quad (3.2.6)$$

因此， $Q_1(x | x_0^*)$ 为 $Q_0(x | x_0^*)$ 在 $R(y' | x_0)$ 上的 I-Projection，且 $Q_2(x | x_0^*)$ 上的条件概率表由 $Q_1(x | x_0^*)$ 按照贝叶斯网结构获得。所以根据归纳假设，有

$$\sum_x Q_1(x | x_0^*) \log \frac{Q_2(x | x_0^*)}{Q_0(x | x_0^*)} = \Delta(x | x_0^*) \geq 0,$$

并且

$$\begin{aligned}
 \Delta_2(x_0, x) &= \sum_{x_0, x} Q_1(x_0, x) \log \frac{Q_2(x | x_0)}{Q_0(x | x_0)} \quad (3.2.7) \\
 &= \sum_{x_0} Q_1(x_0) \sum_x Q_1(x_0, x) \log \frac{Q_2(x | x_0)}{Q_0(x | x_0)} \geq 0
 \end{aligned}$$

II. $x_0 \notin y$

根据 Q_1 的定义有，

$$Q_1(x|x_0) = Q_0(x|x_0) \frac{R(y)/Q_1(x_0)}{Q_0(y)/Q_0(x_0)}.$$

由于 $Q_0(y)/Q_0(x_0) = Q_0(y|x_0)/Q_0(x_0|y)$, 所以

$$Q_1(x|x_0) = Q_0(x|x_0) \frac{R^*(y)}{Q_0(y|x_0)} \quad (3.2.8)$$

其中, $R^*(y) = R(y) \frac{Q_0(x_0|y)}{Q_1(x_0)}$.

记 $x' = x \setminus y$, 则

$$\begin{aligned} Q(x_0|y) \cdot R(y) &= Q_0(x_0, y) \frac{R(y)}{Q_0(y)} \\ &= \sum_{x'} Q_0(x_0, y, x') \frac{R(y)}{Q_0(y)} = \sum_{x'} Q_1(x_0, y, x') = Q_1(x_0, y) \end{aligned}$$

所以 $R^*(y) = \frac{Q_0(x_0, y)}{Q_1(x_0)} = Q(y|x_0)$ 是一个概率分布。

所以对任意 x_0^* , 根据 (3.2.8), $Q_1(x|x_0^*)$ 为 $Q_0(x|x_0^*)$ 在 $R^*(y)$ 上的 I-Projection。由归纳假设, 有

$$\Delta_2(x_0, x) = \sum_{x_0, x} Q_1(x_0, x) \log \frac{Q_2(x|x_0)}{Q_0(x|x_0)} \geq 0 \quad (3.2.9)$$

综上 (3.2.5), (3.2.7) 和 (3.2.9), 有 $\Delta(x_0, x) \geq 0$ 。

例 3.2 已知初始贝叶斯网如图 3.3 所示。其中所有节点为二元随机变量, 取值为 T 或 F 。给定约束集 R 为

Table3.1 Consistent constraint R

A	B	C	$R_1(A,B,C)$	A	D	$R_2(A,D)$
T	T	T	0.126	T	T	0.260
T	T	F	0.014	T	F	0.140
T	F	T	0.234	F	T	0.300
T	F	F	0.026	F	F	0.300
T	T	T	0.288	$R_3(B,E)$		
T	T	F	0.192	T	T	0.227
T	F	T	0.072	T	F	0.393
T	F	F	0.048	F	T	0.129
				F	F	0.251

Comment [Y8]: Better to separate the three constraints in three tables.

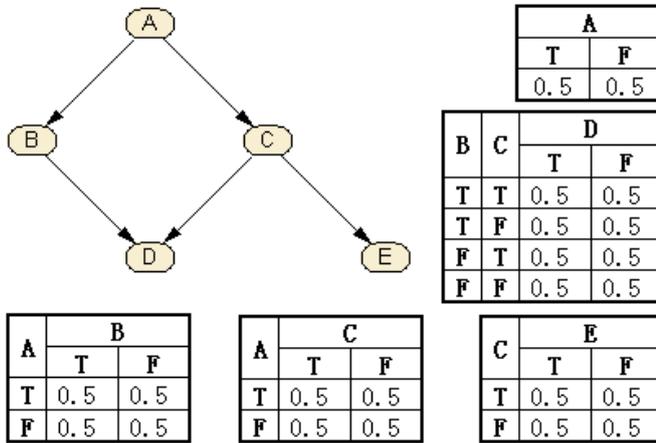


Fig. 3.3 Initial Bayesian network

E-IPFP 算法在迭代 347 步后收敛于图 3.4 所示贝叶斯网。

Comment [Y9]: Say that this BN satisfies all three constraints while keeping the structure unchanged.

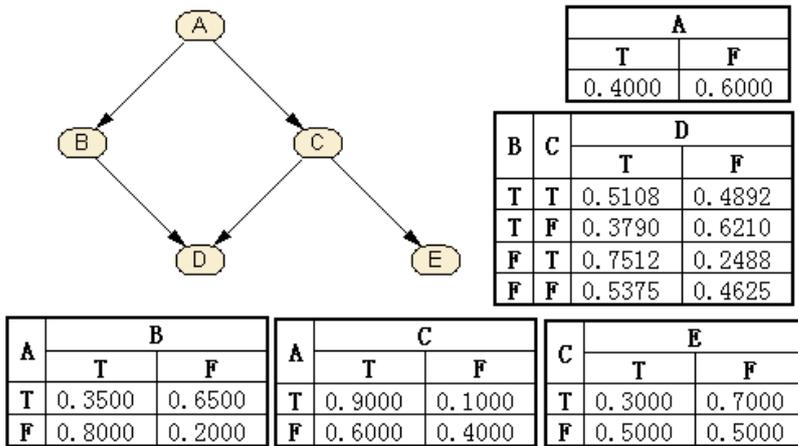


Fig. 3.4 Result Bayesian network for consistent constraint

3.2.2 D-IPFP 算法

注意到算法 IPFP 和 E-IPFP 每一步迭代均为在整个联合概率分布上进行，当网络规模很大时，计算复杂度呈指数增加，这样很不利于实际应用。考虑到每一个约束条件的随机变量集合 Y^i 均为 X 的子集，那么我们可以将对每一个约束条件的迭代计算在 X 的子集上进行，这样将对于 X 的全局计算分解为对于 Y^i 的局部计算，从而达到降低计算复杂度的目的。

根据马尔科夫覆盖的定义，变量集合 Y^i 的马尔科夫覆盖定义为：

定义 3.2 随机变量 A 的马尔科夫覆盖 $MB(A)$ 定义为所有 A 的父节点、子节点和其子节点的其他父节点的集合。

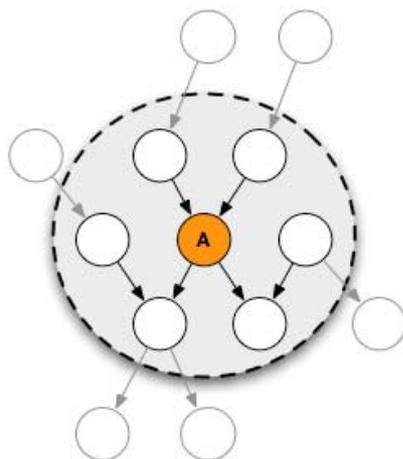


Fig. 3.5 Markov Blanket

图 3.5 给出了一个马尔科夫覆盖的例子¹。

马尔科夫覆盖定义了一种条件独立关系，即在给定一个随机变量的马尔科夫覆盖时，该随机变量与网络中其他节点是相互独立的。利用这种条件独立关系，在实现 E-IPFP 算法时，将 E-IPFP 中的每一步迭代计算从在整个随机变量集合 X 上分解为在 y^i 和其马尔科夫覆盖 $MB(y^i)$ 上进行。算法描述如下：

D-IPFP 算法：

1. 初始值： $Q_0(x) = \prod_{i=1}^n Q_0(x_i | \pi_i)$ ， $R = \{R_1, R_2, \dots, R_m\}$ ；
2. for $k = 1$ ，重复以下迭代过程，直至收敛
 - 2.1 $i = ((k-1) \bmod (m+1)) + 1$ ；
 - 2.2 在 $\{y^i, MB(y^i)\}$ 上使用 $R(y^i)$ 进行 E-IPFP 迭代：

$$Q_k(y^i, MB(y^i)) = Q_{k-1}(y^i, MB(y^i)) \cdot \frac{R(y^i)}{Q_{k-1}(y^i | MB(y^i))};$$

- 2.3 根据 $Q_k(y^i, MB(y^i))$ ，为 y^i 中的随机变量生成条件概率表，更新贝叶斯网；
- 2.4 $k = k + 1$ ；

定理 3.2 给定贝叶斯网 G 和约束集 $R = \{R(Y^1), R(Y^2), \dots, R(Y^m)\}$ 。若约束集 R 一致，且与网络结构一致，则 D-IPFP 算法收敛。

证明：（参考定理 3.1）

¹ http://en.wikipedia.org/wiki/Markov_blanket

例 3.3 假设所有随机变量为二元的，那么对 E-IPFP 算法，其计算复杂度为 $O(2^{|\mathbf{X}|+|Y^k|})$ ，其中 $|Y^k| = \max_{i=1}^m \{|Y^i|\}$ ；而对 D-IPFP 算法，其复杂度为 $O(2^{\max_{i=1}^m \{|MB(Y^i)|+|Y^i|\}})$ 。注意到 $|MB(Y^i)|+|Y^i|$ 不大于 $|\mathbf{X}|$ ，从而通过分解全局计算为局部计算降低了算法的复杂度。

3.2.3 约束集不一致时 E-IPFP 算法的改进

在实际应用中，由于对问题领域的信息来源及获取手段不同，会导致概率知识间的相互不一致，甚至与网络结构不一致¹。为了将不一致的概率知识合成到贝叶斯网，我们对 E-IPFP 算法进行了改进，改进后的算法描述如下：

E-IPFP-SMOOTH 算法：

1. 初始值： $Q_0(x) = \prod_{i=1}^n Q_0(x_i | \pi_i)$ ， $R = \{R_1, R_2, \dots, R_m\}$ ；
2. for $k = 1$ ，重复以下迭代过程，直至收敛
 - 2.1 $i = ((k-1) \bmod m) + 1$ ；
 - 2.2 if $i < m+1$

$$R(Y^j) = (1-\alpha)R(Y^j) + \alpha Q_k(Y^j);$$

$$Q_{k+1}(x) = \begin{cases} 0 & \text{if } Q_k(y^j) = 0 \\ Q_k(x) \cdot \frac{R(y^j)}{Q_k(y^j)} & \text{else} \end{cases};$$

- 2.3 else

由 $Q_{k-1}(x)$ 根据网络结构生成 $Q_{k-1}(x_i | \pi_i)$ ；

$$Q_k(x) = \prod_{i=1}^n Q_{k-1}(x_i | \pi_i);$$

- 2.4 $k = k + 1$ ；
-

例 3.4 根据例 3.2 所示初始条件，我们使用图 3.6 所示两组约束集 $R = \{R_1, R_2\}$ 和 $R' = \{R_3, R_4\}$ 。由图 3.3 贝叶斯网络结构可知，节点 A 为 B 、 C 的父节点，即 B 、 C 在给定 A 的前提下，为条件独立关系。同样 D 、 E 在给定 C 的前提下相互条件独立。而根据图 3.6 所示约束条件，其中第一组约束集中两个约束条件 $R_1(A, B, C)$ 和 $R_2(C, D, E)$ 满足网络结构中的条件独立关系，即 $R_1(B, C | A) = R_1(B | A) \cdot R_1(C | A)$ 和 $R_2(D, E | C) = R_2(D | C) \cdot R_2(E | C)$ 。但是 R_1 和

¹ 这里假设网络结构能够描述问题领域的模型。即这里提到的与网络结构不一致，是由于获得的概率知识不准确而引起的，下文不再赘述。

R_2 相互不一致, 即 $R_1(C) \neq R_2(C)$ 。第二组约束集中两个约束条件相互一致, 即 $R_3(C) = R_4(C)$, 但约束条件 R_3 与网络结构不一致, 即 $R_3(B, C|A) \neq R_3(B|A) \cdot R_3(C|A)$ 。

A	B	C	R1(A, B, C)	C	D	E	R2(C, D, E)
T	T	T	0.126	T	T	T	0.282
T	T	F	0.014	T	T	F	0.089
T	F	T	0.234	T	F	T	0.334
T	F	F	0.026	T	F	F	0.105
F	T	T	0.288	F	T	T	0.026
F	T	F	0.192	F	T	F	0.061
F	F	T	0.072	F	F	T	0.031
F	F	F	0.048	F	F	F	0.072

(a) Constraint R

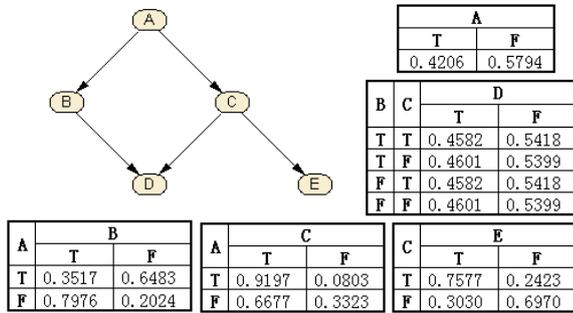
A	B	C	R3(A, B, C)	C	D	R4(C, D)
T	T	T	0.228	T	T	0.182
T	T	F	0.372	T	T	0.222
T	F	T	0.040	T	F	0.308
T	F	F	0.040	T	F	0.288
F	T	T	0.076			
F	T	F	0.124			
F	F	T	0.060			
F	F	F	0.060			

(b) Constraint R'

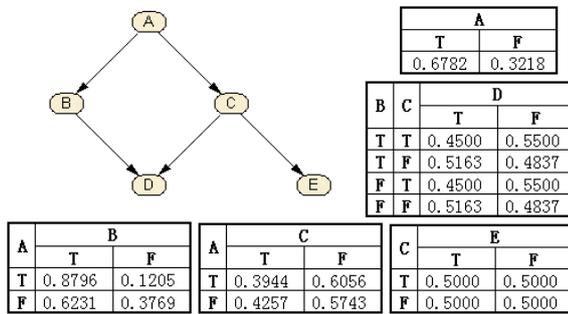
Fig. 3.6 Constraint R and R'

运行 E-IPFP-SMOOTH 算法后, 在第一组约束集下, 算法迭代 76 步后收敛于图 3.7 (a) 所示贝叶斯网; 在第二组约束集下, 算法迭代 151 步后收敛于图 3.7 (b) 所示贝叶斯网。本例中每一步迭代包含了两步 IPFP 计算和一次贝叶斯网络更新的计算)。实验中, 平滑因子 α 取值为 0.01; 我们称算法的每一步迭代为使用约束集中所有约束条件各一次; 若每一步迭代前后两个贝叶斯网络所有节点的信任值差别的绝对值之和 $\sum_{i=1}^n |Q_{k+1}(x_i) - Q_k(x_i)|$ 小于 $1.0E-8$, 则称算法收敛, 即收敛的阈值设置为 $1.0E-8$ 。与 2.3.3 节讨论相一致, 随着平滑因子取值的增大, 算法收敛速度会加快。在本例中, 当平滑因子取值为 0.1 时, 第一组约束集在迭代 23 步时收敛, 第二组约束集在迭代 35 步后收敛, 两组收敛结果与 $\alpha = 0.01$ 时有微小差异。

Comment [Y10]: It would be good if you give the detail of what you mean by "small difference" here.



(a) Result BN for R



(b) Result BN for R'

Fig. 3.7 Result Bayesian network

例 3.4 实验结果表明, E-IPFP-SMOOTH 算法不仅能够处理约束集不一致的情况, 同样能够处理约束条件与网络结构之间的不一致, 从而实现算法的收敛。

在本章的讨论中, 进行知识合成的过程中我们保持了网络结构的不变性, 即默认网络结构能够反映问题的模型。但是, 当约束条件来源于非常可靠的数据, 且与网络结构表现出很大的不一致时, 这表明给定的贝叶斯网络不能精确地描述问题模型, 此时需要对网络结构进行修改, 例如增加网络节点、改变网络节点的条件独立关系等。

3.3 小结

本章讨论了以贝叶斯网的形式表示的知识库中的知识合成。介绍了一种基于 IPFP 方法的贝叶斯网知识合成算法, E-IPFP。该算法通过将贝叶斯网的网络结构作为新的附加约束条件, 对 IPFP 方法进行了扩展: 首先根据链式规则获得迭代初始的联合分布, 在每一轮 IPFP 迭代结束, 使用网络结构约束对当前联合分布进行分解获得网络节点参数 (条件概率表), 根据生成的条件概率表按照贝叶斯网链式规则, 获得新的联合分布, 并将其作为下一轮迭代的初值。文中重

新给出子文中首次给出了 E-IPFP 算法的收敛性完备的详细证明与及实验验证。同时本章介绍了 D-IPFP 算法，即将 E-IPFP 在整个贝叶斯网上的全局计算，分解为对各个约束条件对应的马尔科夫覆盖所决定的子网上的局部计算，降低了计算复杂度，节省了计算空间。

另外，针对 E-IPFP 在约束集不一致时不收敛的情况，本章通过引入约束集不一致时的知识合成算法，对 E-IPFP 算法进行了改进。实验结果表明，改进后的算法不仅能够实现约束条件之间不一致时的收敛，同时能够实现约束条件与网络结构不一致时的收敛。

然而文中讨论的算法并不适合网络结构需要改变的情况，即本章中的算法并不适用于当网络结构不能准确地描述问题模型时的情况。此时需要贝叶斯网结构学习的相关算法进行合成，这也一个进一步研究的方向。

第4章 不确定性证据下贝叶斯网信任更新

本章讨论不确定性证据下贝叶斯网推理问题。首先介绍了几种不确定性证据下贝叶斯网信任更新方法: virtual evidence method, Jeffrey's Rule 和基于 IPFP 的方法; 分析了几种方法之间的关系; 介绍了一种 BN-IPFP 的新算法, 证明了其在证据一致时的收敛性; 对 BN-IPFP 算法进行了改进, 使其能够解决证据不一致时的情况, 并在理论和实验上对算法进行了分析。

4.1 问题描述

推理是指通过计算回答查询的过程。贝叶斯网中的推理是指已知贝叶斯网中某些变量的取值, 计算另外一些变量的后验概率分布的问题。例如在临床诊断中, 若已知患者具有某症状 B , 这时自然会想知道患者患有某种疾病 A 的概率, 即计算“给定患者具有症状 b 时, 患者患有疾病 a ”的概率, 即 $P(\text{疾病}=a | \text{症状}=b)$ 。在这种情况下, 我们称已知的变量为**证据变量**, 称需要计算后验概率分布的变量为**查询变量**。

证据为对于事件的观察。事件由一系列随机变量表示, 记为 $\mathbf{X}=(X_1, X_2, \dots, X_n)$ 。其中每个随机变量可以有多个不同状态。例如, 疾病诊断可以描述为这样一个事件: 即 (疾病, 症状)。其中疾病可以是感冒, 症状可以是发烧、流鼻涕等。在进一步讨论之前, 我们首先给出几个定义:

定义 4.1 (hard evidence) hard evidence 指这样一个观察, 即事件中的某些随机变量具有确定的状态。

例如“患者的症状为发烧”即为一个 hard evidence。这里“发烧”症状具有一个确定的状态“发烧真”, 即 $P(\text{症状}=\text{发烧真})=1$ 。然而在实际应用中, 还有这样一种观察, 即对于给定的随机变量, 往往很难获得其具体的状态, 而只能给出其处于不同状态的分布; 或者观察本身是不完全可靠的, 即对于观察给出的证据是具有不确定性的。这时就需要引入不确定性证据 (uncertain evidence) 的概念。本章讨论如下两种不确定性证据。

定义 4.2 (soft evidence) evidence of uncertainty, 即关于不确定性的证据, 通常表示为对某些随机变量所处状态的概率分布。(Valtorta et al. 2002)

例如在某些情况下, 人们很难观察到某随机变量具体的状态, 但是却能够获得其处于不同状态的概率分布, 而且有时关于随机变量概率分布的观察比知道其所处具体状态更为重要。对于两个不同的贝叶斯网 (或其他概率、统计模

Formatted: Font: Italic

Formatted: Font: Italic

Formatted: Font: Italic

型), 当他们互相交互时, 其相互交换的信息通常表现为关于共享变量的概率分布。

定义 4.3 (virtual evidence) evidence with uncertainty, 即对于证据的不确定性, 通常表示为似然率(likelihood ratio)。

例 4.1 张三和李四为邻居。他们居住小区治安不太好, 所以张三和李四相互约定, 若听到对方家里警铃响则互相电话通知。而李四听力有障碍。某天, 张三收到李四的电话, 说他听到张三家里警铃响, 那么由于李四具有听力障碍, 对于“李四听到张三家里警铃响”这件事则具有不确定性。若用 A 表示警铃, 其具有两个状态记为 $A = (a, \bar{a})$, 李四对于警铃响的观察证据记为 $ob(A)$, 则“李四听到张三家里警铃响”的观察, 可以用似然率记为 $L(A) = P(ob(a) | A = a)$ 。若 $L(A) = 0.9$, 则表示“李四听到张三家里警铃响, 而警铃确实响的概率为 0.9”。

Pearl 首次指出了不确定性证据的问题, 并提出了 virtual evidence 的方法(Pearl 1990), 但是该方法并不适用于给定多重 soft evidence 的情况。Chan 和 Darwiche 在联合概率分布和贝叶斯网中, 详细地分析了 virtual evidence 和 Jeffrey's Rule 两种方法的联系(Chan et al. 2003)。他们指出 soft evidence 可以转化为 virtual evidence, 因此对于单个 soft evidence 下的信任更新可以通过 virtual evidence 的方法来实现。Vomlel 指出, 多重不确定性证据, 即使相互之间不一致, 依然是允许的, 他提出了算法 GEMA 并将其应用于联合概率分布(Vomlel 2003)。Valtorta 等提出了基于 IPFP 的一种扩展方法, 将其用于解决多重一致性 soft evidence 下的贝叶斯网信任更新(Valtorta et al. 2002)。

本章我们首先回顾了各种不确定性证据下贝叶斯网信任更新算法, 分析了他们之间的联系。在现有研究的基础上提出了一种新的算法, BN-IPFP, 并对其进行了证明和验证。

4.2 不确定性证据下的信任更新算法

在本章中, 我们使用大写字母 $X = (X_1, X_2, \dots, X_n)$ 表示所有随机变量, 其中 X_i 表示单个随机变量; 使用小写字母 x 和 x_i 分别表示相应随机变量的实例化; 使用黑体大写字母 \mathbf{X} , \mathbf{X}_i 表示所有可能的实例化。 $Y, Y^1, Y^2, L \subseteq X$ 表示 X 的子集; 类似的 y^j 和 \mathbf{Y}^j 表示对应的实例化。大写字母 P, Q, R, S, T 用来表示概率分布; $P(X)$ 表示联合分布; $Q(Y)$ 表示 $Q(X)$ 在 Y 上的边缘分布。大写黑体字母 $\mathbf{P}, \mathbf{Q}, \mathbf{R}, \mathbf{S}, \mathbf{T}$ 表示概率分布的集合。 $\mathbf{P}_{R(Y)} = \{P(X) | P(Y) = R(Y)\}$ 表示定义在 X 上, 且关于 Y 的边缘分布满足 $R(Y)$ 的概率分布的集合。

Comment [Y11]: Say that this is a variation of Pearl's alarm problem (and provide the reference)

4.2.1 Jeffrey's Rule 和 IPFP

考虑使用一个给定的低维概率分布 $R(Y)$, $Y \subseteq X$, 来更新概率分布 $P(X)$ 。由于 $R(Y)$ 本身不是一个事件, 所以 Bayes' rule 并不能直接应用。Jeffrey [reference] 提出, 更新过的后验分布 $Q(X)$ 应该符合如下两个条件: 1) 满足 $R(Y)$ (即 $Q(Y) = R(Y)$); 2) 更新后的分布应该满足概率动力学 (probability kinematics), 即 $Q(X|Y) = P(X|Y)$ 。其中, 第二个条件保证了更新过程中对概率分布的改变最小化。

根据上述讨论, 若记 $Z \subseteq X \setminus Y$, 则可以计算出关于 Z 的后验概率:

$$Q(z) = \sum_{y \in Y} P(z|y)R(y) = \sum_{y \in Y} \frac{P(z,y)}{P(y)}R(y) \quad (4.2.1)$$

其中 $y \in Y$ 表示求和是在 Y 的所有可能的实例化上, 式 (4.2.1) 称为 Jeffrey's Rule (或 J-conditioning)。根据式 (4.2.1), 对于 X 的一个实例化 x , 则有

$$Q(x) = P(x) \frac{R(y)}{P(y)} \quad (4.2.2)$$

对比 (4.2.2) 与式 (2.2.1), 容易得出以下结论:

定理 4.1 已知分布 $P(X)$ 和 $R(Y)$, $Y \subseteq X$, 则根据 Jeffrey's Rule 更新获得的分布 $Q(X)$ 为 $P(X)$ 在 $\mathbf{P}_{R(Y)}$ 上的 I-Projection。

根据第 2.2 节讨论, I-Projection 最小化目标概率分布与初始分布的 K-L 距离。下面我们证明 I-Projection 同样最小化目标概率分布与初始分布的全变差。

定理 4.2 记 $Q(X)$ 为 $P(X)$ 在 $\mathbf{P}_{R(Y)}$ 上的 I-Projection, 则有

$$\delta(P, Q) = \min_{Q' \in \mathbf{P}_{R(Y)}} \delta(P, Q')$$

证明:

根据全变差的定义, 有

$$\delta(P, Q) = \sum_{x \in X} |P(x) - Q(x)|$$

由 I-Projection 定义, 有

$$\begin{aligned}
 \delta(P, Q) &= \sum_{x \in X} \left| P(x) - P(x) \frac{R(y)}{P(y)} \right| \\
 &= \sum_{x \in X} P(x) \left| 1 - \frac{R(y)}{P(y)} \right| \\
 &= \sum_{y \in Y} \sum_{z \in Z} P(y, z) \left| 1 - \frac{R(y)}{P(y)} \right| \\
 &= \sum_{y \in Y} P(y) \left| 1 - \frac{R(y)}{P(y)} \right| = \sum_{y \in Y} |P(y) - R(y)|
 \end{aligned} \tag{4.2.3}$$

注意到 $\mathcal{Q}(Y) = R(Y)$ ，由于 $\mathcal{Q}(X) \in \mathbf{P}_{R(Y)}$ ，则有：

$$\begin{aligned}
 \delta(P, \mathcal{Q}) &= \sum_{x \in X} |P(x) - \mathcal{Q}(x)| \\
 &= \sum_{y \in Y} \sum_{z \in Z} |P(x) - \mathcal{Q}(x)| \\
 &\geq \sum_{y \in Y} \left| \sum_{z \in Z} P(x) - \sum_{z \in Z} \mathcal{Q}(x) \right| \\
 &= \sum_{y \in Y} |P(y) - \mathcal{Q}(y)| = \sum_{y \in Y} |P(y) - R(y)|
 \end{aligned} \tag{4.2.4}$$

比较 (4.2.3) 和 (4.2.4)，对于任意 $\mathcal{Q}(X) \in \mathbf{P}_{R(Y)}$ ，我们有 $\delta(P, Q) \leq \delta(P, \mathcal{Q})$ ，即 I-Projection 最小化全变差。

I-Projection 的方法解决了单一证据下的信任更新。根据 2.2 节讨论，当给定一个约束集 $\mathbf{R} = (R(Y^1), \dots, R(Y^m))$ 时，循环使用 \mathbf{R} 中的约束条件计算 I-Projection 的过程称为 IPFP。在 \mathbf{R} 一致的情况下，IPFP 收敛，并且收敛结果满足 \mathbf{R} ，同时最小化 $\mathbf{P}_{\mathbf{R}}$ 中的分布与初始分布的 K-L 距离。IPFP 的方法解决了多重约束的问题。下面我们证明 IPFP 同样满足最小化 $\mathbf{P}_{\mathbf{R}}$ 中的分布与初始分布的全变差。

定理 4.3 已知初始概率分布 $Q_0(X)$ 和一致约束集 $\mathbf{R} = (R(Y^1), \dots, R(Y^m))$ ，记 $Q^*(X)$ 为 IPFP 收敛结果，则 $\delta(P, Q^*) = \min_{\mathcal{Q} \in \mathbf{P}_{\mathbf{R}}} \delta(P, \mathcal{Q})$ 。

根据引理 3.1，IPFP 可以看作 $Q_0(X)$ 在 $Q^*(Y)$ 上的单步 I-Projection 计算；由定理 4.2， $\delta(P, Q^*) = \min_{\mathcal{Q} \in \mathbf{P}_{\mathbf{R}}} \delta(P, \mathcal{Q})$ 成立。

4.2.2 Virtual Evidence Method

Pearl (1990) 提出通过在贝叶斯网中添加 virtual 节点，可以实现贝叶斯网在 virtual evidence 下的推理。例如对 4.1 例，若 A 为贝叶斯网中的一个节点，则可以在网络中添加一个记为 U 的节点，其状态 u 表示事件 $A = a$ 发生。那么

在贝叶斯网中添加的 virtual 节点 U 为：唯一父节点为 A ；条件概率表满足 $P(u|A=a):P(u|A \neq a)=0.9:0.1$ 。如图 4.1 所示。

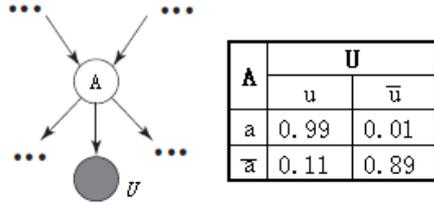


Fig. 4.1 Virtual node U

根据图 4.1，在设置 virtual 节点的条件概率表时，其具体值只要满足给定的似然率即可。在给定观察和对于观察的不确定性（以似然率的形式给出）知识下，对贝叶斯网进行信任更新，只需要在添加 virtual 节点 U 的贝叶斯网中，将 U 例示为 u 即可（例如将 u 作为 hard evidence）。

Chan 和 Darwiche 对 virtual evidence 方法进行了扩展 (Chan et al. 2003)。扩展后在 $Y \subseteq X$ 上的似然率表示为：

$$L(Y) = P(ob(y_1)|y_1) : P(ob(y_2)|y_2) : L : P(ob(y_s)|y_s)。$$

其中 $y_1, y_2, \dots, y_s \in \mathbf{Y}$ 为 Y 的所有实例化； $ob(y_{(i)})$ 表示观察到 $Y = y_{(i)}$ 为真的事件； $P(ob(y_{(i)})|y_{(i)})$ 表示当观察到 $Y = y_{(i)}$ 时， Y 确实处于状态 $y_{(i)}$ 的概率。此时 virtual 节点 U 的父节点为 Y 中包含的所有节点。而当给定 virtual evidence 时，关于 Y 的后验概率，可以通过下式计算：

$$P(Y|u) = c \cdot P(Y) \cdot L(Y) = c \cdot (P(y_1)L(y_1), \dots, P(y_s)L(y_s)) \quad (4.2.6)$$

其中 $c = 1 / \sum_i P(y_i)L(y_i)$ 为归一化因子。由于 virtual 节点的父节点为 Y ，则根据贝叶斯网的 D-分离特性， U 在给定 Y 时条件独立于贝叶斯网其他节点，所以有：

$$P(X \setminus Y | u) = \sum_{y_i} P(X \setminus Y | y_i) P(y_i | u) \quad (4.2.7)$$

类似式 (4.2.2)，根据 (4.2.7) 式可得：

$$\begin{aligned} P(X|u) &= P(X) \frac{P(Y|u)}{P(Y)} \\ &= P(X) \frac{c \cdot P(Y) \cdot L(Y)}{P(Y)} = c \cdot P(X) \cdot L(Y) \end{aligned} \quad (4.2.8)$$

虽然 Jeffrey's Rule 可以用来在给定 soft evidence 时更新联合分布，但是却不能解决联合分布以贝叶斯网表示的情况。Chan 和 Darwiche 指出，可以通过将 soft evidence 转化为 virtual evidence 的方法来实现贝叶斯网信任更新。例如对于 soft evidence $R(Y)$ ， $Y \subseteq X$ ，和一个给定的联合分布 $P(X)$ ， Y

的所有可能的实例化 $y_1, y_2, \dots, y_s \in \mathbf{Y}$ 组成一个互斥的完备事件集。那么 $R(Y)$ 可以转化为一个 virtual evidence, 其似然率为:

$$L(y) = \frac{R(y_1)}{P(y_1)} : \frac{R(y_2)}{P(y_2)} : \dots : \frac{R(y_s)}{P(y_s)} \quad (4.2.9)$$

结合 (4.2.8) 和 (4.2.9), 有

$$P(X|u) = c \cdot P(X) \cdot \frac{R(Y)}{P(Y)} \quad (4.2.10)$$

即更新后的分布与使用 Jeffrey's rule 方法获得的结果相同。结合定理 4.1 可知, 对于单个 soft evidence, 三种信任更新方法 (Jeffrey's rule, I-Projection 和 virtual evidence method) 是等价的。

4.2.3 多重证据时存在的问题

在实际应用中, 往往会同时获得多个观察事件, 即 evidence 是多重的情况。在这种情况下, 若多重证据为 hard evidences, 由于贝叶斯网推理算法对于概率值为 0 的情况有专门的处理, 例如 Junction-Tree 算法, 若某概率值为 0, 则每一步操作后该概率值仍然为 0, 所以上述信任更新算法依然适用。若多重证据为 virtual evidences, 由于只要在更新前后保证似然率得到满足即可, 而根据 4.2.2 节讨论的 virtual evidence 方法, 由于似然率是编码在 virtual 节点的条件概率表中, 而这些条件概率表在推理过程中不会改变, 所以信任更新算法依然适用。

当多重证据为 soft evidences 时, 例如已知 $se1=R(Y^1)$ 和 $se2=R(Y^2)$ 为 soft evidences, 为了满足 $se1$ 和 $se2$, 则更新后的分布 $Q(X)$ 需满足 $Q(Y^1) = R(Y^1)$ 和 $Q(Y^2) = R(Y^2)$ 。若首先将 $se1$ 和 $se2$ 转化为 virtual 节点, 然后使用 virtual evidence 方法进行信任更新, 则结果不一定满足 $Q(Y^1) = R(Y^1)$ 和 $Q(Y^2) = R(Y^2)$ 。原因为当使用 $se1$ 之后, 再使用 $se2$, 那么 $Q(X)$ 将不再满足 $Q(Y^1) = R(Y^1)$ 。同样若 soft evidences 的顺序不同, 更新结果也不相同。这种现象称为迭代修改的交换性 (commutativity of iterated revisions)。文献 (Chan et al. 2003; Wagner 2002) 指出了 Jeffrey's rule 方法存在这个问题。

例 4.2 已知两个 soft evidences 为 $se1: R(B) = (0.7, 0.3)$ 和 $se2: R(C) = (0.3, 0.7)$, 图 4.2 给出了一个具有四个节点的贝叶斯网, 其中 $P(B) = (0.44, 0.56)$, $P(C) = (0.45, 0.55)$ 。按照 virtual evidence 方法 (式 4.2.9), 其似然率分别为 $L(B) = 0.7/0.44:0.3/0.56 = 1.5909:0.5357$ 和 $L(C) = 0.6667:1.2727$ 。

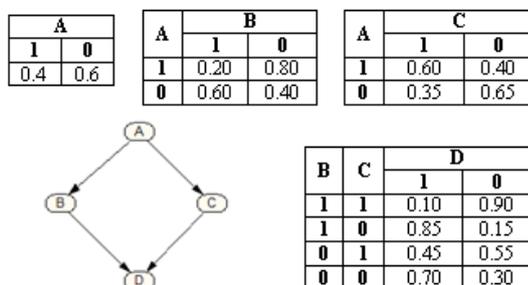


Fig. 4.2 A BN example with 4 nodes

表 4.1 给出了按照不同顺序，对贝叶斯网进行信任更新的结果。由第 2 和 3 行可以看出，当分别使用两个 *se1* 和 *se2* 时，对于 B 和 C 的信任分别满足两个给定的 soft evidences。第 4 和 5 行表明，当同时使用 *se1* 和 *se2* 和顺序使用 *se1* 和 *se2* 时，信任更新结果相同。虽然 $P(u1|B=1, u2):P(u1|B=0, u2) = L(B)$ and $P(u2|C=1, u1): P(u2|C=0, u1) = L(C)$ ，即在给定另外一个证据时似然率并未改变，但是两个 soft evidences 却不再被满足。第 6 行给出的是，在使用 *se2* 前，首先使用第 2 行的结果为 *se2* 计算新的似然率 $L'(C): L'(C) = 0.3/0.425:0.7/0.575 = 0.7:1.2174$ ，然后按照 virtual evidence 方法进行信任更新。然而从计算结果可以看出，使用 *se2* 更新后的分布，并不满足 *se1* ($R(B=1) = 0.7$)。

Table 4.1 Belief update on BN of example 4.2

evidences	Belief on B=1	belief on C=1
1. original	0.440	0.450
2. using $L(B)$	0.700	0.425
3. using $L(C)$	0.455	0.300
4. $L(B)$ and $L(C)$	0.712	0.279
5. $L(B)$ then $L(C)$	0.712	0.279
6. $L(B)$ then $L'(C)$	0.710	0.300

为了解决这个问题，下节我们讨论如何将 virtual evidence 方法和 IPFP 方法结合，并将其应用于贝叶斯网。

4.2.4 BN-IPFP 算法

已知给定贝叶斯网 G 和一组 soft evidences 集合 $\mathbf{R} = (R(Y^1), L, R(Y^m))$ 。 G 定义在随机变量 $X = (X_1, X_2, \dots, X_n)$ 上，具有联合分布 $P(X) = \prod_{X_i \in X} P(X_i | \pi_i)$ ，其中 $P(X_i | \pi_i)$ 为 X_i 的条件概率表； $Y^1, Y^2, \dots, Y^m \subseteq X$ 。假设 \mathbf{R} 是一致的，且满足对任意 $R(Y^j) \in \mathbf{R}$ 有 $P(Y^j) \ll R(Y^j)$ ，即 $\{y^j : P(y^j) = 0\} \subseteq \{y^j : R(y^j) = 0\}$ 。那么使用 \mathbf{R} 对 G 信任更新的问题可以描述为：寻找一个联合分布 $Q^*(X)$ ，使其满

足 \mathbf{R} 中的所有证据，同时最小化与 $P(X)$ 的 K-L 距离。

为此，我们将 IPFP 的思想引入到 virtual evidence 方法中：记初始分布为 $Q_0(X) = P(X)$ ，每一步迭代使用 \mathbf{R} 中的一个 soft evidence。如果当前分布 $Q_{k-1}(Y^j)$ 满足 $R(Y^j)$ ，则直接进入下一步迭代；否则使用 $R(Y^j)$ 计算似然率，并在 G 中添加 virtual node，同时更新 $Q_{k-1}(Y^j)$ 。算法描述如下：

BN-IPFP-1 算法：

1. 初始值： $Q_0(X) = P(X), k = 1$;
2. 重复以下过程直至收敛：
 - 2.1 $j = 1 + (k-1) \bmod m; l = 1 + \lfloor (k-1)/m \rfloor$;
 - 2.2 使用下述似然率构造 virtual evidence:

$$L_{j,l}(Y^j) = \frac{R(y_{(1)}^j)}{Q_{k-1}(y_{(1)}^j)} \cdot \frac{R(y_{(2)}^j)}{Q_{k-1}(y_{(2)}^j)} \cdot \dots \cdot \frac{R(y_{(j)}^j)}{Q_{k-1}(y_{(j)}^j)}$$

其中 $y_{(1)}^j, y_{(2)}^j, \dots, y_{(j)}^j \in \mathbf{Y}^j$ 为 Y^j 的实例化;

- 2.3 根据 virtual evidence 方法，使用 $L_{j,l}(Y^j)$ 更新 $Q_{k-1}(X)$ 获得 $Q_k(X)$;

- 2.4 $k = k + 1$;

由上述算法描述可以看出，BN-IPFP-1 算法的主要思想为第 2.2 步：计算一个似然率为 $L_{j,l}(y^j)$ 的 virtual evidence，其中下标 l 表示每 m 次迭代为 $R(y^j)$ 增加的 virtual 节点个数。

定理 4.4 给定初始贝叶斯网，并记 $P(X) = \prod_{X_i \in X} P(X_i | \pi_i)$ ，若证据集 $\mathbf{R} = (R(Y^1), L, R(Y^m))$ 一致，且对任意 $R(Y^j) \in \mathbf{R}$ 有 $P(Y^j) \ll R(Y^j)$ ，即 $\{y^j : P(y^j) = 0\} \subseteq \{y^j : R(y^j) = 0\}$ ；则算法 BN-IPFP-1 收敛。

证明：

由 (4.2.8) 式，算法 2.3 步对 $Q_{k-1}(X)$ 的更新为：

$$Q_k(X | u_j) = c \cdot Q_{k-1}(X) \cdot L(Y^j)。$$

将算法第 2.2 步计算的似然率带入上式可得

$$Q_k(X | u_j) = Q_{k-1}(X) \cdot \frac{R(Y^j)}{Q_{k-1}(Y^j)} \quad (4.2.11)$$

比较可知 (4.2.11) 与 I-Projection 为同样的计算，根据 2.2 节关于 IPFP 的讨论，由 (Pan et al. 2006) 可知，当约束集 $\mathbf{R} = (R(Y^1), L, R(Y^m))$ 一致时，BN-IPFP-1 收敛。

例 4.3 根据例 4.2 已知条件，运行算法 BN-IPFP-1，在第四步迭代后算法收敛于图 4.3 所示结果。其中收敛后的似然率分别为 $L(B) = (1.0 : 0.354)$ 和 $L(C) =$

(0.578 : 1.0)。

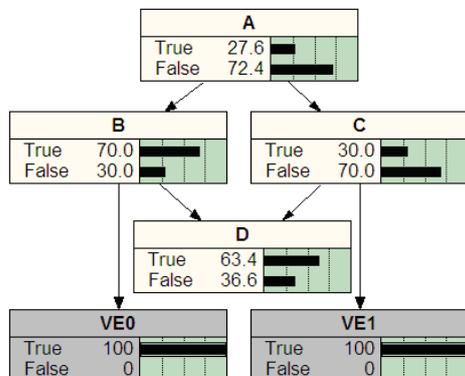


Fig. 4.3 Result BN of BN-IPFP-1 for example 4.3

注意到在 BN-IPFP-1 算法中，第 2.2 步中需要计算边缘概率 $Q_{k-1}(Y^j)$ ，在第 2.3 步需要在整个贝叶斯网中进行信任更新，所以当贝叶斯网规模增加时，算法将具有很高的复杂度。为此，我们提出了新的算法，其主要思想为：首先根据 \mathbf{R} 构造一个新的 soft evidence，记为 $R(Y)$ ，其中 $Y = \cup_j Y^j$ ；然后由式(4.2.9)将 $R(Y)$ 转化为 virtual evidence $L(Y)$ ，并根据 $L(Y)$ 对贝叶斯网进行信任更新。算法描述如下：

BN-IPFP-2 算法：

1. 由 $P(X)$ 生成 $P(Y)$ ；
2. 由 IPFP 算法，使用约束集 \mathbf{R} 更新 $P(Y)$ ，记收敛结果为 $Q^*(Y)$ ；
3. 根据式 (4.2.9) 将 $Q^*(Y)$ 转换为 virtual evidence，记其似然率为 $L(Y)$ ；
4. 使用 virtual evidence 方法，用 $L(Y)$ 更新 $P(X)$ 。

定理 4.5 给定初始贝叶斯网并记 $P(X) = \prod_{X_i \in X} P(X_i | \pi_i)$ ，若证据集 $\mathbf{R} = (R(Y^1), L, R(Y^m))$ 一致，且对任意 $R(Y^j) \in \mathbf{R}$ 有 $P(Y^j) \ll R(Y^j)$ ，即 $\{y^j : P(y^j) = 0\} \subseteq \{y^j : R(y^j) = 0\}$ ；则算法 BN-IPFP-2 收敛。

根据 IPFP 的收敛性证明，很容易得出定理 4.5。同时，由算法第 3 步的计算结合定理 4.1，使用 $L(Y)$ 更新 $P(X)$ 后获得的 $Q^*(X)$ 为 $P(X)$ 的 I-Projection，即 $Q^*(X)$ 与 $P(X)$ 具有最小的 K-L 距离。

例 4.4 根据例 4.2 已知条件，运行算法 BN-IPFP-2，在第四步迭代后算法收敛于图 4.4 所示结果。其中收敛后的似然率为 $L^*(B,C) = (0.578 : 1.0 : 0.205 : 0.354)$ 。

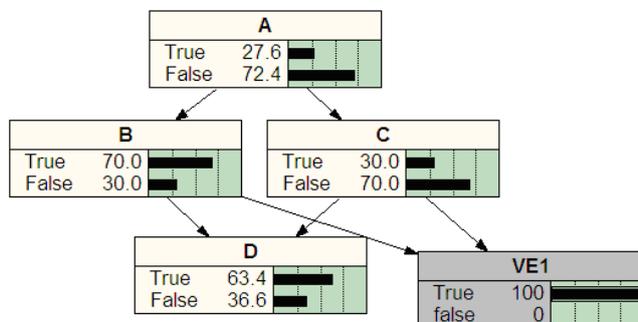


Fig. 4.4 Result BN of BN-IPFP-2 for example 4.4

4.2.5 证据不一致时的处理

对于同一事件的观察，由于观察者的不同，或者观察手段的不同，其观察结果也不尽相同；甚至同一观察者采用同样的观察手段，在不同时刻也会获得不同的观察结果，从而使得对于领域问题的观察证据之间存在相互矛盾或不一致的地方。本节结合第二章所讨论的约束集不一致时的知识合成算法，将其应用于 BN-IPFP，从而实现对证据不一致时贝叶斯网的信任更新。

首先回顾 2.3.3 节关于 SMOOTH 算法的性质：

- (1) SMOOTH 是对 IPFP 算法的扩展，它在约束集不一致时退化为标准 IPFP 算法；
- (2) SMOOTH 在约束集一致或不一致时均收敛；
- (3) 在约束集不一致时，SMOOTH 收敛结果最小化与初始条件的 I-Aggregate。

根据上述 SMOOTH 算法的性质，结合 4.2.3 节 BN-IPFP 算法，同样可以根据 SMOOTH 的算法思想对 BN-IPFP 算法进行改进，即根据贝叶斯网所决定的联合分布来修正证据，从而降低证据间的不一致度，并由修正后的证据进行信任更新。改进后的算法描述如下：

BN-SMOOTH-1 算法：

1. 初始值： $Q_0(X) = P(X)$, $k = 1$;
2. 重复以下过程直至收敛：
 - 2.1 $j = 1 + (k - 1) \bmod m$; $l = 1 + \lfloor (k - 1) / m \rfloor$;
 - 2.2 使用下述似然率构造 virtual evidence:

$$R(y^j) = (1 - \alpha)R(y^j) + \alpha Q_{k-1}(y^j)$$

$$L_{j,i}(Y^j) = \frac{R(y_{(1)}^j)}{Q_{k-1}(y_{(1)}^j)} \cdot \frac{R(y_{(2)}^j)}{Q_{k-1}(y_{(2)}^j)} \cdot \dots \cdot \frac{R(y_{(j)}^j)}{Q_{k-1}(y_{(j)}^j)}$$

其中 $y_{(1)}^j, y_{(2)}^j, \dots, y_{(j)}^j \in \mathbf{Y}^j$ 为 Y^j 的实例化;

2.3 根据 virtual evidence 方法, 使用 $L_{j,i}(Y^j)$ 更新 $Q_{k-1}(X)$ 获得 $Q_k(X)$;

2.4 $k = k + 1$;

定理 4.6 给定初始贝叶斯网及证据集证据 $\mathbf{R} = (R(Y^1), L, R(Y^m))$, 记 $P(X) = \prod_{X_i \in X} P(X_i | \pi_i)$, 若对任意 $R(Y^j) \in \mathbf{R}$ 有 $P(Y^j) \ll R(Y^j)$, 即 $\{y^j : P(y^j) = 0\} \subseteq \{y^j : R(y^j) = 0\}$; 则算法 BN-SMOOTH-1 收敛。

定理 4.6 的证明与 2.3.3 节 SMOOTH 算法证明类似, 这里不再赘述。

同样地, 将 SMOOTH 引入 BN-IPFP-2 算法, 可得下述算法:

BN-SMOOTH-2 算法:

1. 由 $P(X)$ 生成 $P(Y)$;
2. 由 SMOOTH 算法, 使用约束集 \mathbf{R} 更新 $P(Y)$, 记收敛结果为 $Q^*(Y)$;
3. 根据式 (4.2.9) 将 $Q^*(Y)$ 转换为 virtual evidence, 记其似然率为 $L(Y)$;
4. 使用 virtual evidence 方法, 用 $L(Y)$ 更新 $P(X)$ 。

定理 4.7 给定初始贝叶斯网及证据集证据 $\mathbf{R} = (R(Y^1), L, R(Y^m))$, 记 $P(X) = \prod_{X_i \in X} P(X_i | \pi_i)$, 若对任意 $R(Y^j) \in \mathbf{R}$ 有 $P(Y^j) \ll R(Y^j)$, 即 $\{y^j : P(y^j) = 0\} \subseteq \{y^j : R(y^j) = 0\}$; 则算法 BN-SMOOTH-2 收敛。

定理 4.7 的证明即为 SMOOTH 算法的收敛证明。

4.3 小结

本章讨论了在给定对事件的观察下, 如何进行贝叶斯网信任更新的问题, 即已知给定证据变量, 计算查询变量的后验概率。详细讨论了不同类型的证据, 及其之间的关系。重点描述了两类不确定性证据: soft evidence 和 virtual evidence。前者指不确定性的证据, 表现为证据变量处于不同状态的概率分布; 后者指关于证据的不确定性, 即关于观察结果本身是不确定性的。两种不确定性证据互相区别, 并且遵循不同的信任更新机制。

分析了几种不同的信任更新算法: Jeffrey's rule、virtual evidence method 和 IPFP 方法。分析表明, 在给定单一 soft/virtual evidence 时的信任更新, 等价于在单一约束条件下的 IPFP 方法。针对 virtual evidence 方法

在多重证据时进行贝叶斯网信任更新的局限性，将 IPFP 方法与 virtual evidence 方法进行结合，解决了多重证据时的信任更新。引入了知识合成算法 SMOOTH，并将其与 virtual evidence 结合，解决了证据间不一致时的更新问题。理论和实验结果验证了算法的正确性。与已有 soft evidence 更新算法（如 big clique 方法(Valtorta et al. 2002)等）相比，新的算法不要求特别的贝叶斯网推理机制，即新的算法可以方便地应用于现有的贝叶斯网推理工具。

第5章 贝叶斯网在语义网本体推理中的应用

本章回顾了基于贝叶斯网的本体不确定性推理研究概况，着重介绍并扩展了一种基于贝叶斯网的本体不确定性推理方法 (BayesOWL)。该方法通过将本体转化为贝叶斯网，同时将关于本体的不确定性知识通过知识集成的方法合成到转化后的贝叶斯网中，实现了本体推理到贝叶斯网推理的转化。我们通过采用本体推理工具，扩展了该方法对更一般本体的支持；分析了转化后贝叶斯网中的语义；通过引入改进后的贝叶斯网知识合成算法，解决了对不一致概率知识的合成问题；最后，实现了该方法的原型系统，此系统为从事该领域研究工作的人员提供了一个实用工具。

5.1 引言

近年来随着对语义网研究的不断深入，作为语义网核心的本体技术，被广泛应用于现实问题当中，而基于本体问题的研究也受到人们日益广泛的关注 (Osterwalder et al. 2002; Chabalier et al. 2007; Bottazzi et al. 2006; Batchelor et al. 2009; Schlicht et al. 2008)。本体是对问题领域的概念及概念之间关系的形式化表示，这种对知识的形式化描述，使语义网能够方便地定义网上信息的语义，从而使得机器能够根据这些明确定义的语义实现 Web 信息的自动处理、集成和重用。然而现实中人们所获得的信息和知识，往往含有大量的不准确、不完全、不一致的地方，即获得的信息和知识是不确定的，所以在解决实际问题时常常需要对这种不确定性进行处理（即如何表示不确定性并进行不确定性推理）。由于对本体的描述，即本体语言 (Mcguinness et al. 2004) (OWL, [Web Ontology Language](#))，是基于描述逻辑 (Baader et al. 2003) 的，而描述逻辑本身并未提供对不确定性的处理方法 (Cali et al. 2008; Koller et al. 1997; Ding et al. 2006)，这使得本体在处理不确定性时受到很大限制。

目前针对本体不确定性推理问题的研究主要为对描述逻辑的扩展。基于对描述逻辑扩展方法的研究主要包括两个方面：基于概率论的概率描述逻辑 (Lukasiewicz 2007; Cali et al. 2008; Udrea et al. 2005) 和基于模糊理论的模糊描述逻辑 (Stoilos et al. 2007; Bobillo et al. 2008)。他们通过对描述逻辑进行扩展并提供相应的推理机制，使扩展后的描述逻辑能够表示不确定性，并在此基础上进行不确定性推理。

基于贝叶斯理论的贝叶斯网络（又称信度网络或因果关系网络）于 1985 年由 Pearl 提出，它是一种概率图解模型。贝叶斯网通过一个有向无环图（DAG）来表示随机变量及其之间的条件独立关系。有向无环图蕴含了条件独立假设，即给定其父节点集，每一个变量独立于它的非子孙节点。节点之间的连接关系代表了贝叶斯网的条件独立语义。贝叶斯网中的每一个随机变量对应一个条件概率表（CPT），而所有的条件概率表通过链式规则确定了一个关于网络中所有随机变量的联合概率分布，即关于领域知识的概率知识库。随着对其研究的不断深入，贝叶斯网络已经成为不确定性知识表达和推理领域最有效的模型之一，并因而常被用来作为本体语言的底层逻辑来实现描述逻辑的概率扩展。将本体推理问题转化为贝叶斯网络推理问题的方法，既保留了本体语言在描述问题领域方面的优势，又利用了贝叶斯网络的推理能力。

Comment [Y12]: Shorten this paragraph. BN has been introduced early so it does not need to be repeated here.

5.2 研究现状

贝叶斯网是使用概率论进行本体不确定性问题研究中一种比较常用的方法。关于这种方法，目前的研究一般分为两种，一是扩展 OWL 本体语言，即使用贝叶斯网络来描述不确定性知识，并将其作为描述逻辑的底层逻辑；二是将本体推理问题转换为贝叶斯网推理问题。目前基于贝叶斯网的概率描述逻辑，或者通过改变现有的 OWL 本体语言使其能够表示不确定性，或者改变贝叶斯网以弥补其在知识表示方面的不足。

P-CLASSIC 于 1997 年由 Koller 等提出。它是对描述逻辑语言 CLASSIC 的一种概率扩展。P-CLASSIC 知识库中定义了一系列的 p-classes，每一个 p-class 为一个定义在基本属性上的贝叶斯网。通过定义这些 p-class 来表示本体中的不确定性。同时，P-CLASSIC 提供了一种有效的推理程序来计算本体中的概率包含关系。

PR-OWL (Costa et al. 2005) 是一种基于多实体贝叶斯网 (MEBN, Laskey 2007) 的描述逻辑概率扩展。MEBN 基于贝叶斯逻辑，它结合了一阶逻辑和贝叶斯网。PR-OWL 在贝叶斯网中用参数化的标志 (MFlags) 来表达知识，从而实现了将不确定性知识在一阶逻辑解释中的表示。

OntoBayes (Yang et al. 2007) 是另一种将 OWL 和贝叶斯网结合起来的方法。它通过增加对 OWL 本体语言类型的定义，来在 OWL 本体中描述概率和依赖关系，并将这种描述与 BN 网络结构相结合，以实现对本体的扩展。同时，OntoBayes 支持多值随机变量及定义在随机变量上的联合概率分布。

Ding 和 Peng 提出了一种理论框架，即 BayesOWL (Ding et al. 2004, 2006)。

BayesOWL 定义了一系列结构转换规则，根据本体定义的概念类及其之间的语义关系，将本体转化为贝叶斯网。BayesOWL 还提供了一种方法，将已知的不确定性知识合成到转换后的贝叶斯网中。作为结果的贝叶斯网构成了一个综合本体知识和概率知识的知识库。BayesOWL 通过定义转换规则，将本体推理问题直接转换为贝叶斯网推理问题，既未改变本体语言的底层逻辑又利用了贝叶斯网的推理能力，解决了分类学本体的不确定性问题。

5.3 BayesOWL 框架

本体是对共享概念化的正式的、清楚的规范，是关于领域知识的概念及概念间关系的规范描述。它提供了对领域建模所需的共享词典，即存在的对象（或概念）的类型，属性和其相互关系。而如何将本体中的概念类及其语义关系映射为贝叶斯网中对应元素，是将本体推理转化为贝叶斯网推理的首要问题。

5.3.1 结构转换规则

1 概念类的转换

根据 OWL 本体的相关定义，本体中的概念类为个体的集合，一个个体是否属于该类可用一个二元变量来表示。因此，在进行映射时，可以将本体中的概念类转化为贝叶斯网中的二元变量，即本体中定义的概念类 A ，对应于贝叶斯网中的一个二元变量节点 $A:(a, \bar{a})$ 。该节点有两个状态：是和非，分别表示一随机选取的个体属于和不属于该类。该类节点称为概念类节点。

2 逻辑关系的转换

本体中对于概念类的定义，一般有两种方法。一种是直接指定类名（或引用）和限制列表；另一种定义方法是使用类表达式。OWL 支持基本的集合操作，即合集，交集和补集，分别表示为 “owl:unionOf”，“owl:intersectionOf” 和 “owl:complementOf”。其他的类构造子，如 “rdfs:subClassOf”，“owl:equivalentClass” 和 “owl:disjointWith” 称为类公理。在 BayesOWL 中，我们仅考虑以上六种类构造子（表 5.1）。

Table 5.1 Concept constructor supported in BayesOWL

Constructor	DL Syntax	Class Axiom	Logical Operator
rdfs:subClassOf	$C_1 \sqsubseteq C_2$	✓	
owl:equivalentClass	$C_1 \equiv C_2$	✓	
owl:disjointWith	$C_1 \sqcap \neg C_2$	✓	
owl:unionOf	$C_1 \sqcup \dots \sqcup C_n$		✓

owl:intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	√
owl:complementOf	$\neg C$	√

如果类 C 为 $C_i (i = 1, \dots, n)$ 的子类, 那么在转换后的贝叶斯网中, 该子类关系由图 5.1 给出的子网表示。该子网由节点 C 和 C_i 及从 C_i 到 C 的有向箭头组成。其中的 C 的条件概率表设置为: 当其父节点至少有一个处于“否”的状态时, C 取值为否; 否则 C 取真值的概率值默认为 0.5。表 5.2 给出了当 $n = 2$ 时, C 的条件概率表。

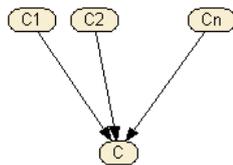


Fig. 5.1 Subnet of Subclass relation

Table 5.2 Conditional probability table of node C in figure 5.1

C_1	C_2	True	False
True	True	0.5	0.5
True	False	0.0	1.0
False	True	0.0	1.0
False	False	0.0	1.0

根据表 5.1, 除子类关系外, 其他五类构造子各定义了一种逻辑关系。为在转换后的 BN 中保持这种逻辑关系, 在贝叶斯网中增加新的一类节点, 即逻辑节点 (Logic Node, 简称为 L-Node)。该逻辑节点的父节点为所有与该逻辑关系相关的概念类节点。因为逻辑节点表达了概念类间的逻辑关系, 所以它的条件概率表可完全由其父节点状态的逻辑运算结果确定。下面我们介绍如何生成逻辑节点及设置其条件概率表。

如果类 $C_i (i = 1, \dots, n)$ 为等价类, 那么转换后的贝叶斯子网为图 5.2 所示 ($n=2$ 时)。该逻辑节点取值为是, 当且仅当所有父节点取相同值 ($c_1 c_2 \dots c_n \vee \bar{c}_1 \bar{c}_2 \dots \bar{c}_n$), 否则取值为否。表 5.3 给出了当 $n = 2$ 时, 逻辑节点的条件概率表。

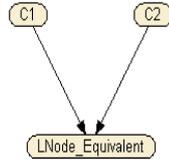


Fig. 5.2 Subnet of Equivalent relation

Table 5.3 Conditional probability table of L-Node in figure 5.2

C ₁	C ₂	True	False
True	True	1.0	0.0
True	False	0.0	1.0
False	True	0.0	1.0
False	False	1.0	0.0

其他逻辑节点的条件概率表可以类似地由其布尔表达式生成 (Ding 2006), 即:

- 若 C_1 和 C_2 不相交, 则对应逻辑节点取真值条件为 $\bar{c}_1c_2 \vee c_1\bar{c}_2 \vee \bar{c}_1\bar{c}_2$;
- 若 C 为 C_1 和 C_2 的并集, 则对应逻辑节点取真值条件为 $cc_1c_2 \vee c\bar{c}_1c_2 \vee cc_1\bar{c}_2 \vee \bar{c}c_1\bar{c}_2$;
- 若 C 为 C_1 和 C_2 的交集, 则对应逻辑节点取真值条件为 $cc_1c_2 \vee \bar{c}c_1c_2 \vee \bar{c}c_1\bar{c}_2 \vee \bar{c}\bar{c}_1\bar{c}_2$;
- 若 C_1 和 C_2 互补, 则对应逻辑节点取真值条件为 $\bar{c}_1c_2 \vee c_1\bar{c}_2$ 。

由上述讨论可见, 转换后的贝叶斯网包含了两类节点: 一类由本体定义的概念类转换而来, 称为概念类节点; 另一类由概念类间的逻辑关系转换而来, 称为逻辑节点。根据结构转换规则, 概念类节点的层次关系反映了本体中概念类的包容关系。根据转换规则, 所有逻辑节点均为叶节点, 即增加逻辑节点并不会在转换后的网络中产生环。环的情况只可能出现在两个概念类之间是互为子(父)类关系时, 此时互为子(父)类的两个类之间的逻辑关系可以用等价类替代, 从而避免在贝叶斯网中产生环。

3 隐含逻辑关系的处理

本节讨论如何通过扩展原始的 BayesOWL 框架, 来实现对更为一般的本体(如 OWL DL)的支持。我们知道在一般的本体中存在匿名概念类, 其定义是通过属性约束来实现的, 这种概念类没有类名, 相应地在结构转换时也没有与之对应的概念类节点。然而, 由于匿名类的存在, 本体中其他概念类之间可能存在通过匿名类构成的隐含逻辑关系。并且, 由于属性之间的关系, 本体中的概

念类也会因为具有不同的属性而存在隐含逻辑关系。例如，在 W3C 组织定义的本体“Wine”¹中，类“RedBurgundy”和“WhiteWine”之间并无显式定义的逻辑关系，但是由于两者分别是匿名类“hasColor=#Red”和“hasColor=#White”的子类，而这两个匿名类是不相交的，因此类“RedBurgundy”和类“WhiteWine”也是不相交的。

借助 Semantic Web 现有的本体推理工具（例如 Jena, FaCT, Pellet 等），可以寻找这种因匿名类和类属性而产生的逻辑关系，这样就可以在 BayesOWL 中实现对属性的处理，即通过寻找本体中因属性而产生的概念类之间的逻辑关系，并通过设置逻辑节点在生成的贝叶斯网中保证这种逻辑关系的正确性，来实现对本体中属性的处理。然而，由此获得的逻辑关系集合中存在大量冗余，即集合中某些逻辑关系蕴涵于其他逻辑关系中。如果简单地将这些冗余逻辑关系全部在贝叶斯网中用逻辑节点表示，作为结果的贝叶斯网络规模则会大大增加。为处理这种情况，需要对逻辑关系集合进行冗余移除操作。

用 S_{ONT} 表示本体中所有显式和隐式定义的关系集合，定义 $S_{DAG} \subseteq S_{ONT}$ 为满足以下条件的关系集合：1) 所有 S_{ONT} 中的关系蕴涵于 S_{DAG} ；2) S_{DAG} 中的任何关系 L 一定不蕴涵于集合 $S_{ONT} \setminus \{L\}$ 。下列冗余移除程序从 S_{ONT} 中移除冗余逻辑关系并获得 S_{DAG} ：

冗余移除程序

第一步：处理等价关系。首先将所有类按等价关系分成各个等价组，并在每个等价组中选出一个类作为代表；在 S_{ONT} 中用等价组代替所有的等价关系，对 S_{ONT} 中的其它关系，用选出的代表类替换其所代表的其他等价类；移除 S_{ONT} 中相同的关系。

第二步：处理不相交关系。在 S_{ONT} 中移除如下关系：若 $C \cap D = \emptyset \in S_{ONT}$, $A \cap B = \emptyset \in S_{ONT}$, 且 $A \subseteq C$ and $B \subseteq D$, 则移除 $A \cap B = \emptyset$ 。

第三步：处理子类关系。移除子类关系中非直接父子类关系。

例 5.1 在一个本体中定义了如下逻辑关系：

$$\{A \equiv B, B \equiv C, C \sqsubseteq E, D \sqsubseteq F, G \sqsubseteq B, E \sqcap F = \perp\},$$

那么根据本体推理工具可以得到：

$$S_{ONT} = \{A \equiv B, B \equiv C, C \equiv A, C \sqsubseteq E, B \sqsubseteq E, A \sqsubseteq E, G \sqsubseteq A, G \sqsubseteq B, G \sqsubseteq C, G \sqsubseteq E, D \sqsubseteq F, E \sqcap F = \perp, A \sqcap D = \perp, B \sqcap D = \perp, G \sqcap D = \perp, G \sqcap F = \perp, C \sqcap D = \perp, A \sqcap F = \perp, B \sqcap F = \perp, C \sqcap F = \perp, D \sqcap E = \perp\}.$$

根据冗余移除程序第一步， A, B, C 形成等价组，用该等价组替代 S_{ONT} 中的等价关系；若选取 A 作为该等价组的代表，用 A 替换 S_{ONT} 所含其它关系中的

¹ <http://www.w3.org/TR/owl-guide/wine.rdf>

B, C , 移除相同关系后得到 $\{A \equiv B \equiv C, A \sqsubseteq E, G \sqsubseteq A, G \sqsubseteq E, E \sqcap F = \perp, A \sqcap D = \perp, G \sqcap D = \perp, G \sqcap F = \perp, A \sqcap F = \perp, D \sqcap E = \perp\}$; 根据第二步得到 $\{A \equiv B \equiv C, G \sqsubseteq A, A \sqsubseteq E, G \sqsubseteq E, D \sqsubseteq F, E \sqcap F = \perp\}$, 根据第三步则最终得到 $S_{DAG} = \{A \equiv B \equiv C, A \sqsubseteq E, G \sqsubseteq A, D \sqsubseteq F, E \sqcap F = \perp\}$, 这些逻辑关系可以通过三个子类关系和两个逻辑节点来表示。

在 Wine 本体上根据以上步骤进行实验, 结果如表 5.4 所示。

Table 5.4 Number of relations in Wine ontology

	显式定义	S_{ONT}	S_{DAG}
逻辑关系个数	99	6641	388

从表 5.4 可以看出, Wine 本体中显式定义的逻辑关系数为 99, 由 Pellet 生成的 S_{ONT} 中逻辑关系个数为 6641。而在移除冗余逻辑关系之后的 S_{DAG} 中, 其个数降为 388, 仅为 S_{ONT} 的 6%。这样在将 Wine 本体转换为贝叶斯网后, 其网络节点个数为 342。其中, 概念类节点个数为 126, 逻辑节点个数为 216。

4 结构转换规则

综上所述, 我们给出扩展后的 BayesOWL 结构转换规则:

- 1) 概念类。本体中定义的每一个概念类, 在贝叶斯网中对应一个二元概念类节点。
- 2) 概念类间关系。首先生成 S_{ONT} 集合, 利用冗余移除程序生成 S_{DAG} 。
- 3) 网络结构。根据 S_{DAG} 关系集合, 生成贝叶斯网络结构 (子类由箭矢表示, 其它关系由逻辑节点表示)。
- 4) 条件概率表。设置节点默认初始条件概率表。

根据上述结构转换规则, 可以将任意给定的本体转换为贝叶斯网, 网络结构表示了本体的概念类层次结构。其中逻辑节点及其条件概率表描述了概念类之间的逻辑关系。

例 5.1 根据文献 (Ding et al. 2004) 中的 “nature.owl” 例子, 图 5.3 给出了按照扩展后的结构转换规则生成的贝叶斯网。

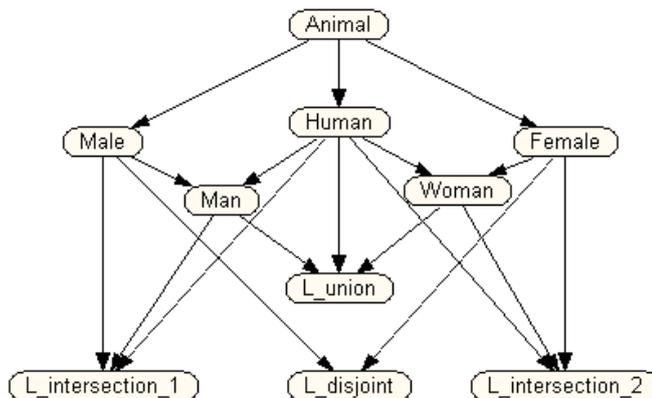


Fig. 5.3 Translated BN from 'Nature' ontology

同样地，我们按照结构转换规则，由“wine”本体生成了对应的由 342 个节点组成的贝叶斯网¹。

5.3.2 BayesOWL 中的语义关系

由 BayesOWL 获得的贝叶斯网的语义，建立在关于概念类节点的联合概率分布上，即对于所有概念类节点的每一个实例化，有一个概率值与其对应，并且该概率值满足： $Q(x_c) = P(x_c | LT)$ 。

其中， P 为转换后的贝叶斯网， Q 为关于所有概念类的联合概率分布， $x_c = \{x_1, \dots, x_n\}$ 表示所有类节点的一个实例化， LT 表示所有的逻辑节点取值为真。

描述逻辑的解释由一个对象的领域和一个解释函数组成。在 BayesOWL 语义中，对于本体的每一个解释，存在一个函数 P ，它将该解释映射为一个 0 到 1 之间的值。例如在描述逻辑中，假设有一个这样的解释 I “所有动物(A)中不是雌性(F)的集合”，那么解释 I 就可以表示为 “ $A \sqcap \neg F$ ”。如果用 M 表示这个集合，在 BayesOWL 语义中，将存在一个对应于解释 I 的概率分布 $P(M)$ ，它给出了任意一个对象属于或不属于这个集合的概率。

根据 W3C 的定义，OWL 中的每一个抽象句法对应于一个解释。表 5.5 列出了 BayesOWL 中使用到的三个非公理型抽象句法及其对应的解释。

Table 5.5 Interpretations for 3 embedded constructors used in BayesOWL

Abstract Syntax	Interpretation
unionOf(C_1, \dots, C_n)	$EC(C_1) \cup \dots \cup EC(C_n)$
intersectionOf(C_1, \dots, C_n)	$EC(C_1) \cap \dots \cap EC(C_n)$
complementOf(C)	$O - EC(C)$

其中 EC 表示 OWL 类和数据类型的统一资源标识 (URI)， O 表示 $EC(owl:Thing)$ ， $Thing$ 为所有 OWL 类的父类。

事实上，每个内含构造都可以转化为一个公理型的解释。例如，“complementOf”即对应于公理 “ComplementClass(C_1, C_2)” 的一个解释，且关于该解释的条件为 “ $EC(C_1) = O - EC(C_2)$ ”。根据表 5.1，BayesOWL 支持 OWL 本体语言中的 6 种构造子。表 5.6 给出了将这 6 种构造子转化为公理型解释的结果。

Table 5.6 Conditions on interpretations used in BayesOWL

Directive	Conditions on Interpretations
-----------	-------------------------------

¹ <http://mail.ustc.edu.cn/~syzhang/wine.dne>

SubClasses(C_1, C_2)	$EC(C_1) \subseteq EC(C_2)$
EquivalentClasses(C_1, \dots, C_n)	$EC(C_i) = EC(C_j)$, for $1 \leq i < j \leq n$
DisjointClasses(C_1, \dots, C_n)	$EC(C_i) \cap EC(C_j) = \{\}$, for $1 \leq i < j \leq n$
UnionClasses($C, (C_1, \dots, C_n)$)	$EC(C) = EC(C_1) \cup \dots \cup EC(C_n)$
IntersectionClasses($C, (C_1, \dots, C_n)$)	$EC(C) = EC(C_1) \cap \dots \cap EC(C_n)$
ComplementClasses(C_1, C_2)	$EC(C_1) = O - EC(C_2)$

对于每个公理型解释的条件, 在 BayesOWL 语义中存在一个概率条件与之对应。即在 BayesOWL 中, 每一个公理解释的条件被映射为一个概率条件。表 5.7 列出了这种映射关系。

Table 5.7 Conditions on interpretations used in BayesOWL

Conditions on Interpretations	Conditions on Probabilities
$EC(C_1) \subseteq EC(C_2)$	$P(c_2 c_1) = 1$
$EC(C_i) = EC(C_j)$, for $1 \leq i < j \leq n$	$P(c_i LT) = P(c_j LT)$, for $1 \leq i < j \leq n$
$EC(C_i) \cap EC(C_j) = \{\}$, for $1 \leq i < j \leq n$	$P(c_i, c_j LT) = 0$, for $1 \leq i < j \leq n$
$EC(C) = EC(C_1) \cup \dots \cup EC(C_n)$	$P(c LT) = P(c_1 \vee \dots \vee c_n LT)$
$EC(C) = EC(C_1) \cap \dots \cap EC(C_n)$	$P(c LT) = P(c_1, \dots, c_n LT)$
$EC(C_1) = O - EC(C_2)$	$P(c_1 LT) = 1 - P(c_2 LT)$

定理 5.1 由 BayesOWL 转换生成的贝叶斯网保证了表 5.7 中概率条件的合法性。

证明:

假设由 BayesOWL 结构转换生成的贝叶斯网包含 5.3.1 节描述的 6 种逻辑关系, 根据各逻辑关系生成 6 种对应的子网:

I. 对应“`rdfs:subClassOf`”的子网, 其中 C_1 为 C_2 的子集, 则:

$$P(c_2 | c_1) = \frac{P(c_1, c_2)}{P(c_1)} = \frac{P(c_1, c_2)}{P(c_1, c_2) + P(c_1, \bar{c}_2)}$$

$$= \frac{P(c_1, c_2)}{P(c_1, c_2) + P(c_2)P(c_1 | \bar{c}_2)}$$

考虑到若 C_1 父节点有一个为 False, 则 C_1 为 False, 即 $P(c_1 | \bar{c}_2) = 0$, 所以 $P(c_2 | c_1) = 1$ 。

II. 对应“`owl:equivalentClass`”的子网, 其中类 C_i ($i = 1, \dots, n$) 为等价类, 则对于任意 $1 \leq i < j \leq n$, 有

$$P(c_i | LT) = \frac{P(c_i, LT)}{P(LT)} = \frac{\sum_{c_j} P(c_i, C_j, LT)}{P(LT)} = \frac{P(c_i, c_j, LT) + P(c_i, \bar{c}_j, LT)}{P(LT)}, \text{ 且}$$

$$= \frac{P(c_i, c_j, LT) + P(c_i)P(\bar{c}_j)P(LT | c_i, \bar{c}_j)}{P(LT)}$$

$$\begin{aligned}
 P(c_j | LT) &= \frac{P(c_j, LT)}{P(LT)} = \frac{\sum_{C_i} P(c_j, C_i, LT)}{P(LT)} = \frac{P(c_j, c_i, LT) + P(c_j, \bar{c}_i, LT)}{P(LT)} \\
 &= \frac{P(c_j, c_i, LT) + P(c_j)P(\bar{c}_i)P(LT | c_j, \bar{c}_i)}{P(LT)}.
 \end{aligned}$$

考虑到若 C_i 和 C_j 取值不同，则逻辑节点概率值为 0，即 $P(LT | c_i, \bar{c}_j) = P(LT | c_j, \bar{c}_i) = 0$ 。因此有 $P(c_i | LT) = P(c_j | LT)$ 。

III. 对应“owl:disjointWith”的子网，其中 C_i, C_j 为不相交关系，则对任意 $1 \leq i < j \leq n$ ，有

$$P(c_i, c_j | LT) = \frac{P(c_i, c_j, LT)}{P(LT)} = \frac{P(c_i)P(c_j)P(LT | c_i, c_j)}{P(LT)}.$$

考虑到若 C_i, C_j 均为真，则逻辑节点概率取值为 0，所以有 $P(c_i, c_j | LT) = 0$ 。

IV. 对应于“owl:unionOf”的子网，其中 C 为 $C_i (i = 1, \dots, n)$ 的并集。不失一般性，假设 $i = 2$ 。考虑到 C_i 为 C 的子集，则若 C_i 取值为 True 且 C 取值为 False，则对应的逻辑节点概率为 0。则

$$\begin{aligned}
 P(c | LT) &= \frac{P(c, LT)}{P(LT)} = \frac{\sum_{c_1, c_2} P(c, C_1, C_2, LT)}{P(LT)}, \\
 &= \frac{P(c, c_1, c_2, LT) + P(c, c_1, \bar{c}_2, LT) + P(c, \bar{c}_1, c_2, LT) + P(c, \bar{c}_1, \bar{c}_2, LT)}{P(LT)},
 \end{aligned}$$

$$P(c_1, c_2 | LT) = \frac{P(c_1, c_2, LT)}{P(LT)} = \frac{P(c_1, c_2, c, LT)}{P(LT)},$$

$$P(c_1 | LT) = \frac{P(c_1, LT)}{P(LT)} = \frac{P(c_1, c_2, c, LT) + P(c_1, \bar{c}_2, c, LT)}{P(LT)}, \text{ 且}$$

$$P(c_2 | LT) = \frac{P(c_2, LT)}{P(LT)} = \frac{P(c_2, c_1, c, LT) + P(c_2, \bar{c}_1, c, LT)}{P(LT)}.$$

考虑到若所有 C_i 取值为 False，则对应逻辑节点取值为 False，即 $P(c, \bar{c}_1, \bar{c}_2, LT) = 0$ ，则有 $P(c | LT) = P(c_1 \vee c_2 | LT)$ 。

V. 对应于“owl:intersectionOf”的子网，其中 C 为 $C_i (i = 1, \dots, n)$ 的交集。不失一般性，假设 $i = 2$ 。有

$$P(c|LT) = \frac{P(c,LT)}{P(LT)} = \frac{\sum_{c_1, c_2} P(c, C_1, C_2, LT)}{P(LT)}, \text{ 且}$$

$$= \frac{P(c, c_1, c_2, LT) + P(c, c_1, \bar{c}_2, LT) + P(c, \bar{c}_1, c_2, LT) + P(c, \bar{c}_1, \bar{c}_2, LT)}{P(LT)}$$

$$P(c_1, c_2 | LT) = \frac{P(c_1, c_2, LT)}{P(LT)} = \frac{P(c_1, c_2, c, LT) + P(c_1, c_2, \bar{c}, LT)}{P(LT)}。$$

考虑到 C 可以视为 C_i 的子集, 即若 C_i 取值 False 且 C 取值 True, 则对应逻辑节点概率为 0, 所以有 $P(c|LT) = P(c_1, c_2 | LT)$ 。

VI. 对应“owl:complementOf”的子网, 其中 C_1, C_2 为互补关系, 有

$$P(c_1 | LT) = \frac{P(c_1, LT)}{P(LT)} = \frac{P(c_1, c_2, LT) + P(c_1, \bar{c}_2, LT)}{P(LT)}, \text{ 和}$$

$$P(c_2 | LT) = \frac{P(c_2, LT)}{P(LT)} = \frac{P(c_2, c_1, LT) + P(c_2, \bar{c}_1, LT)}{P(LT)}。$$

考虑到 $P(c_1 | LT) + P(c_2 | LT) = \frac{\sum_{c_1, c_2} P(C_1, C_2, LT)}{P(LT)} = 1$, 因此有 $P(c_1 | LT) = 1 - P(c_2 | LT)$ 。

综上, 定理 5.1 成立。

至此我们讨论了 BayesOWL 的结构转换规则及转换后生成的贝叶斯网的语义。需要指出的是, 根据贝叶斯网的 d-分离性, BayesOWL 转换结果的贝叶斯网可能隐含了条件独立关系。例如, 在 OWL 本体中, 拥有同一父类的子类之间一般是不相交的, 而在转换后的贝叶斯网中, 它们之间是条件独立的。如果没有矛盾存在, 那么可以将条件独立关系视为子类间的默认关系, 否则可以在贝叶斯网中加入不相交逻辑节点来表示它们间的关系。

5.3.3 BayesOWL 中概率知识的表示和合成

在 BayesOWL 中, 本体概念类之间的逻辑关系由 BN 网络结构表示, 这些关系间的不确定性由条件概率表来描述。而生成网络结构时设置的逻辑节点的默认条件概率表往往与实际不符。更准确的不确定性知识可由用户或领域专家给出或通过其它资源如 Web 获得。由专家提供的知识通常为概念类之间的联合概率或父子类之间的条件概率等, 而不是条件概率表。同样, 由于 Web 资源来源的多样性, 从网络资源 (例如万维网、Web 应用和 Web 服务等) 提取而来的概率知识往往是不完整、不精确的。为了构造条件概率表, 需要统一这些概率知

识的表示形式，并提出相应的机制将其合成到贝叶斯网的 CPT 中。

本节中我们首先讨论如何在 BayesOWL 的框架下，将概率知识以概率分布来表示。接着利用第三章讨论的概率知识合成算法，将其应用于 BayesOWL 中的网络节点条件概率表的构造。

1 概率知识的表示

不确定性存在于实际应用中的各个方面，然而在语义网的研究中，迄今为止仍未有一个普遍接受的方法来描述不确定性。对于一个本体，不确定性知识可以视为对概念类和类间关系的概率约束，并通过概率分布来表示。由于本体是通过本体语言来描述（例如 OWL），自然地在表示概率知识时，可以通过 OWL 本体语言的格式来描述。例如，对于概率 $P(c) = 0.5$ ，它包含了 3 个元素：变量 C 、表示变量特定状态的命题（ c ，某一随机提取的个体属于概念类 C ）和概率值（0.5）。那么在表示概率知识时，可以通过定义三个 OWL 类标签来表示概率知识。

类 “*Variable*”。表示随机变量；“*Variable*” 具有属性 “*hasClass*”，该属性指向这个变量对应的本体中的概念类；

类 “*Proposition*”。描述 “*Variable*” 的特定状态的命题；“*Proposition*” 具有两个属性 “*hasVariable*” 和 “*hasState*”，分别指向对应变量和命题对应变量的状态；

类 “*Probability*”。描述概率值；“*Probability*” 具有 3 个属性：“*hasProposition*”、“*hasCondition*” 和 “*hasValue*”。三个属性集的势分别为 “ ≥ 1 ”、“ ≥ 0 ” 和 “ $=1$ ”。

利用上述定义的三个类型，能够完整描述以概率分布表示的任何概率知识。例如在 “*Nature*” 本体中，对概率知识 $P(\text{Male}=\text{True}, \text{Human}=\text{True} \mid \text{Animal}=\text{True}) = 0.026$ ，可以描述为如下形式：

```
<owl:Variable rdf:ID="male">
  <hasClass>Male</hasClass>
</owl:Variable>
<owl:Variable rdf:ID="human">
  <hasClass>Human</hasClass>
</owl:Variable>
<owl:Variable rdf:ID="animal">
  <hasClass>Animal</hasClass>
</owl:Variable>
<owl:Proposition rdf:ID="m1">
```

```

    <hasVariable>male</hasVariable>
    <hasState>True</hasState>
</owl:Proposition>
<owl:Proposition rdf:ID="h1">
    <hasVariable>human </hasVariable>
    <hasState>True</hasState>
</owl:Proposition>
<owl:Proposition rdf:ID="a1">
    <hasVariable>animal</hasVariable>
    <hasState>True</hasState>
</owl:Proposition>
<owl:Probability rdf:ID="P(m1,h1|a1)">
    <hasPposition>m1</hasPposition>
    <hasPposition>h1</hasPposition>
    <hasCondition>a1</hasCondition>
    <hasValue>0.026</hasValue>
</owl:Probability>

```

利用这种表示方法的另一个好处是，由于这种表示方法遵循本体语言的格式，所以可以利用处理本体文件的方法来生成或解析概率文件。另外，因为概率文件独立于本体文件，所以概率知识可以发布到 Web 上，从而实现共享与重用。

2 BayesOWL 中条件概率表的构造

根据 5.3.1 节的介绍，由 OWL 本体转化的贝叶斯网包括两种节点：概念类节点和逻辑节点。其中，逻辑节点的条件概率表完全由其所表示的逻辑运算结果确定。对于概念类节点的条件概率表，可以根据已知概率知识和 E-IPFP 算法来构造。在构造概念类变量的条件概率表时，注意到转换后的贝叶斯网只有在所有逻辑节点取真值时，OWL 本体语义才得到满足，所以在实现 E-IPFP 算法时，对于算法中 2.2 步，应该在逻辑节点全部设置为“True”的子空间（记为 LT ）来进行，即 2.2 步的计算公式应为：

$$Q_k(x, LT) = Q_{k-1}(x, LT) \cdot \frac{R(y^j)}{Q_{k-1}(y^j | LT)} \quad (5.3.1)$$

可以看出，利用上式求得的联合概率分布在所有逻辑节点取真值时，其边缘分布满足给定约束条件。对于 D-IPFP 算法，可以做同样的修正。显然地，根据定理 3.1，当约束条件相互一致且与贝叶斯网络结构一致时，使用式 (5.3.1)

Formatted: Highlight

Formatted: Highlight

修改后的 E-IPFP 算法收敛。

例 5.2 根据图 5.3 给出的本体例子，给出以下一致的约束集：

- $P(\text{Animal}) = 0.560$
- $P(\text{Male, Human} | \text{Animal}) = 0.511$
- $P(\text{Female, Human} | \text{Animal}) = 0.258$
- $P(\text{Man} | \text{Animal, Human}) = 0.664$
- $P(\text{Woman} | \text{Animal, Human}) = 0.336$

则运行 E-IPFP 算法后，收敛于图 5.4 所示贝叶斯网。

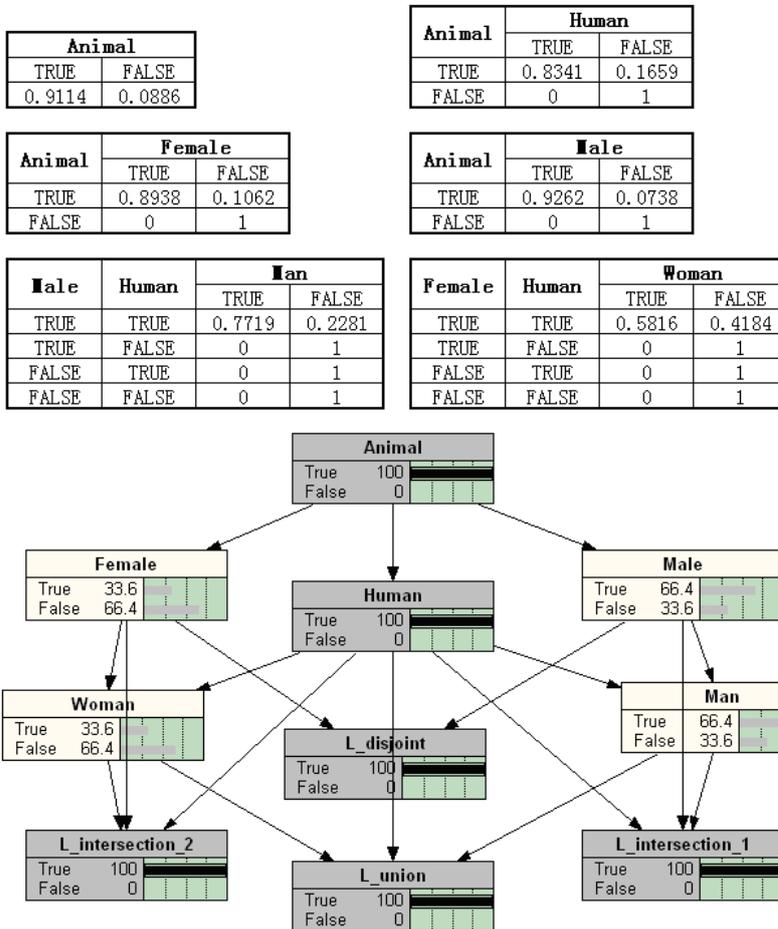


Fig. 5.4 Result BN for 'Nature' ontology example (consistent)

若约束集不一致，可使用 3.2.3 节给出的 E-IPFP-SMOOTH 算法进行知识合成。

例 5.3 根据图 5.3 给出的本体例子，给出以下不一致的约束集：

- $P(\text{Animal}) = 0.560$
- $P(\text{Male, Human} | \text{Animal}) = 0.511$
- $P(\text{Female, Human} | \text{Animal}) = 0.258$
- $P(\text{Man} | \text{Animal, Human}) = 0.664$
- $P(\text{Woman} | \text{Animal, Human}) = 0.56$

由于“Man”和“Woman”为互补关系，而本例中的最后两个条件并不满足这一关系，即约束集是不一致的。在运行 E-IPFP-SMOOTH 算法后，收敛于图 5.4 所示贝叶斯网。

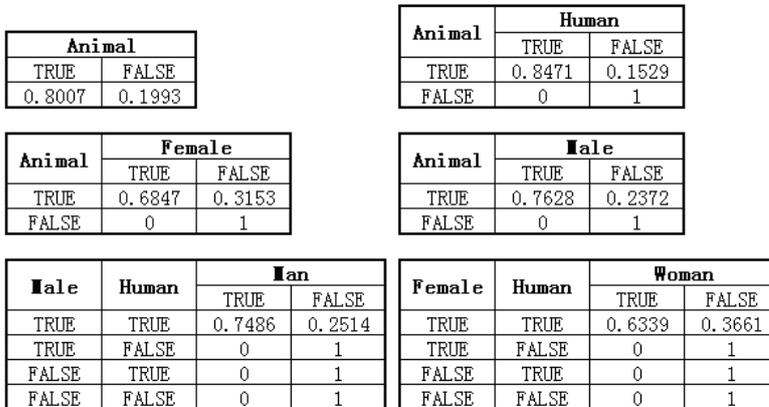


Fig. 5.5 Result BN for 'Nature' ontology example (inconsistent)

由图 5.5 可见，收敛后的贝叶斯网中，约束条件分别修改为 $P(\text{Man} | \text{Animal, Human}) = 0.636$ 和 $P(\text{Woman} | \text{Animal, Human}) = 0.364$ ，满足互补关系。

5.4 BayesOWL 理论框架的实现

根据 5.3 节的介绍，我们将 BayesOWL 的实现分为三部分：解析器(Parser)、结构构造器和 CPT 构造器。其中解析器由 *T-Parser* 和 *P-Parser* 组成，分别用来解析 OWL 本体文件和概率知识文件；结构构造器是 BayesOWL 结构转换规则的实现，它根据 *T-Parser* 的解析结果按照结构转换规则生成贝叶斯网；CPT 构造器用来设置贝叶斯网络节点的条件概率表，它首先根据 5.3.1 节中讨论，为节点条件概率表设置默认值，如果已知概率知识，则将 *P-Parser* 解析结果作为概率约束，通过 5.3.2 节所述的知识合成算法来设置条件概率表。BayesOWL 流程图如图 5.6 所示。

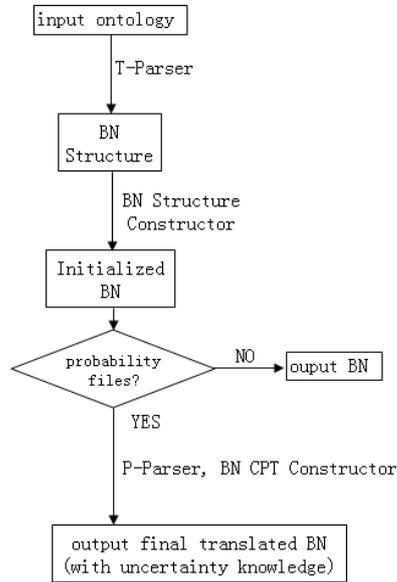


Fig. 5.6 BayesOWL framework

BayesOWL 的输入包括两部分，一为定义本体的 OWL 本体文件，一为概率知识文件。其输出为结果贝叶斯网。即若无概率知识文件，则将设置了默认条件概率表的贝叶斯网作为结果输出；若已知概率知识，则 CPT 构造器通过调用 P-Parser 的解析结果，根据 E-IPFP/D-IPFP 算法，将概率知识合成到贝叶斯网中并将合成后的贝叶斯网作为结果输出。系统体系结构如图 5.7 所示。

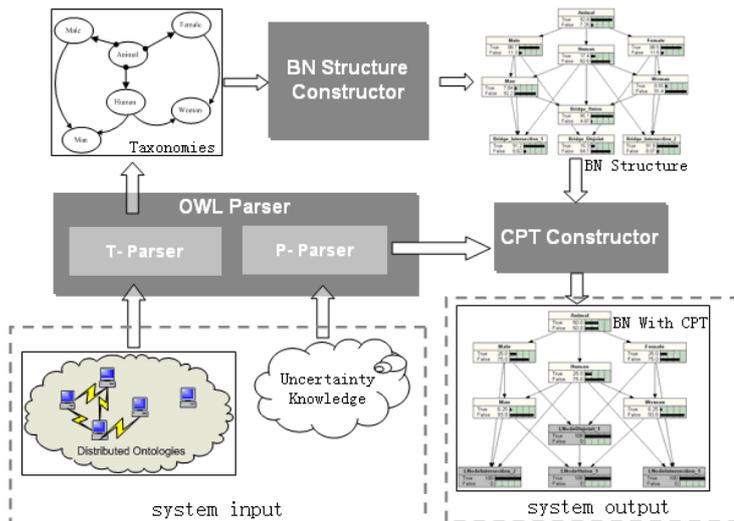


Fig. 5.7 BayesOWL system architecture

系统实现的过程中，T-Parser 在生成逻辑关系集合 S_{ONT} 时，我们采用了 Pellet¹工具来寻找本体概念类之间的相互逻辑关系。在实现 E-IPFP/D-IPFP 算法时，涉及到贝叶斯网的部分我们采用了 Norsys 公司开发的 Netica²工具。图 5.8 给出了 BayesOWL 原型系统的应用程序接口 (API)³。

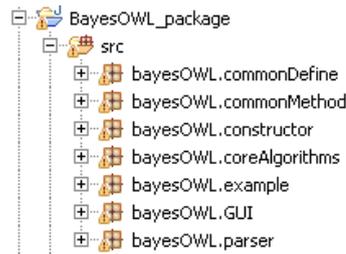


Fig. 5.8 BayesOWL application programming interface

5.5 小结

本章介绍并扩展了一种处理语义网本体不确定性的理论框架，BayesOWL。该理论框架通过一系列结构转换规则，将给定的 OWL 本体转换为贝叶斯网，并通过设置逻辑节点保留了本体中概念类之间的语义关系。通过对结构转换规则的扩展，使得 BayesOWL 能够支持更为一般的本体。同时在本体描述语言的基础上，提出了一种表示不确定性知识的方式。并通过引入贝叶斯网知识合成算法，实现了将关于本体的不确定性知识合成到转化后的贝叶斯网。最后通过对原型系统的实现，从而实现了利用贝叶斯网对本体推理问题的求解，为从事该领域的研究人员提供了一个有用的工具。

与其他本体不确定性推理方法相比，BayesOWL 的方法并不需要改变 OWL 本体语言及其底层逻辑，同时又保留了基于描述逻辑的本体描述语言在问题描述方面的能力和贝叶斯网强大的推理能力。

然而，限于贝叶斯网在表达方面的局限性，如何完全地将本体映射为贝叶斯网，即如何在贝叶斯网中表达 OWL 本体的其他部分（如概念类的属性、个体等）依然是一个待解决的问题。另外，现有的关于本体的不确定知识一般由领域专家给出，考虑到现有网络资源的开放性，如何利用文本分析、信息检索等方法自动提取关于本体的不确定性知识，是另外一个研究方向。

¹ <http://clarkparsia.com/pellet>

² <http://www.norsys.com/>

³ <http://www.csee.umbc.edu/~ypeng/BayesOWL/index.html>

第6章 论文总结

虽然关于贝叶斯网不确定性推理的理论和应用研究已经很多，但是如前文所述，目前这些研究依然存在有待进一步解决的问题。本文主要针对贝叶斯网在不确定性信息下的知识合成与信任更新进行研究，提出了一些算法。所做主要工作和成果可总结如下：

- (1) 较为系统地探讨了概率推理中的知识合成方法，揭示了各算法的关系，分析了他们的性质。
- (2) 提出了一种简单的基于 IPFP 概率知识合成算法 SMOOTH。该算法在标准 IPFP 算法的基础上，引入平滑因子，通过双向修改概率约束条件和目标联合概率分布，实现了算法的快速收敛。SMOOTH 算法在约束集一致时退化为标准 IPFP 算法，在约束集不一致时，收敛结果与 GEMA 算法相同。数值模拟结果表明，与已有算法（CC-IPFP, GEMA 等）相比，SMOOTH 收敛性能稳定，且计算复杂度低，同时可以调整平滑因子，实现算法的加速收敛。
- (3) 研究了贝叶斯网知识合成问题。在 Peng 和 Ding 的工作基础上，详细给出了 E-IPFP 算法的收敛性证明。同时，通过引入 SMOOTH 算法，实现了对 E-IPFP 算法的改进，使其能够处理不一致的情况。实验结果表明，改进后的算法不仅能够处理概率约束集不一致，同样能够在一定条件下处理约束条件与贝叶斯网络结构不一致。
- (4) 系统介绍了不确定性证据下贝叶斯网信任更新问题。探讨了已有的更新算法，揭示了各算法之间的关系。在 Peng 和 Pan 的工作基础上，研究了将 Pearl 的 virtual evidence 方法与 IPFP 方法相结合的信任更新算法。讨论了算法的收敛性能。对算法进行了扩展，使其能够处理多重证据，及证据相互不一致的情况。
- (5) 探讨了贝叶斯网在语义网本体不确定性推理中的应用。在 Ding 和 Peng 的工作基础上对 BayesOWL 理论框架进行了扩展与完善。扩展后的 BayesOWL 能够处理更为一般的本体（如 OWL DL），并能够处理更为普遍的概率知识。实现了 BayesOWL 的原型系统，为从事相关研究的人员提供了一个实用工具。

以上这些研究工作旨在推动贝叶斯网理论模型及其推理机制在现实中的应用。然而由于个人水平有限，在我们的研究工作中依然存在瑕疵和有待进一步完善的地方，如：

- (1) 不确定性知识的来源。在本文的研究中，我们假设已经获得关于问题领域的不确定性知识。然而现实中，海量的知识以各种各样的方式存在于

互联网，如何通过信息检索、文本分类等相关技术自动提取感兴趣的不确定性知识，将是一个非常有意义的研究方向。

- (2) 贝叶斯网知识合成的研究中，我们维持了网络结构的不变，即假设网络结构能够很好的描述问题领域的模型。然而，当不确定性知识与网络结构不一致，且来自于非常可靠的数据源时，则有可能是网络结构并不能反映问题模型。如何结合贝叶斯网结构学习的相关算法，[通过修改网络结构](#)来实现不确定性知识的合成，同样是一个很有意义的研究方向。
- (3) 进行贝叶斯网信任更新时，若证据为混合的，即既有 hard evidence，又有 soft evidence 和 virtual evidence 时，如何进行信任更新，依然有待进一步地研究。

历时两个多月，终于完成了这篇毕业论文的写作。在这两个多月的工作中，阅读了大量与本文研究相关的论文和技术报告，并完成了这篇论文的文字写作、绘图和排版工作。几年的科研生活，让我经历了一个从入门到深入，从简单到复杂的过程，锻炼了分析问题与解决问题的能力，是我人生道路上最为印象深刻的一段历程。

参考文献

- Baader F, Calvanese D, et al. 2003. *The Description Logic Handbook: Theory, Implementation, Applications*. Cambridge University Press, Cambridge, UK.
- Batchelor C, Bittner T, et al. 2009. The RNA Ontology (RNAO): An ontology for integrating RNA sequence and structure data. *First International Conference on Biomedical Ontology*.
- Bloemke M, 1998. *Agent Encapsulated Bayesian Networks*, PhD thesis, Department of Computer Science, University of South Carolina.
- Bobillo F, Straccia U, 2008. fuzzyDL: An expressive fuzzy description logic reasoner, *International Conference on Fuzzy Systems (FUZZ-08)*.
- Bock HH, 1989. A Conditional Iterative Proportional Fitting (CIPF) Algorithm with Applications in the Statistical Analysis of Discrete Spatial Data, *Bull. ISI, Contributed papers of 47th Session in Paris*, vol. 1, pp. 141-142.
- Bottazzi E, Ferrario R, 2006. Preliminaries to a dolce ontology of organizations, *International Journal of Business Process Integration and Management*.
- Brachman R, Borgida A, et al. 1991. Living with CLASSIC: When and How to Use a KL-ONE-Like Language, In John Sowa, editor, *Principles of Semantic Networks*.
- Bwon DT, 1959. A Note on Approximation to Discrete Probability Distributions, *Information and Control*, vol. 2, pp. 386-392.
- Cali A, Lukasiewicz T, 2008. An approach to probabilistic data integration for the Semantic Web, in *Proceedings of ISWC-URSW Conference*.
- Chabalier J, Mosser J, 2007. Integrating biological pathways in disease ontologies, in *Proceedings of the 12th World Congress on Health (Medical) Informatics – Building Sustainable Health Systems*.
- Chan H and Darwiche A, 2003. Revisiting the Problem of Belief Revision with Uncertain Evidence, in *Proceedings of 18th International Joint Conference on Artificial Intelligence*.
- Combettes PL, 1993. The Foundations of Set Theoretic Estimation, *Proceedings of IEEE*, vol. 81, no. 2.
- Costa PCG, Laskey KB, and Laskey KJ, 2005. PR-OWL: A Bayesian ontology language for the semantic web, in *Proceedings of Workshop on Uncertainty Reasoning for the Semantic Web (URSW) at the 4th International Semantic Web Conference (ISWC)*, Galway, Ireland.
- Costa PCG, 2005. *Bayesian Semantics for the Semantic Web*, PhD thesis, Dept. of System Engineering and Operations Research, George Mason University, Fairfax, VA.

- Cramer E, 2000. Probability Measures with Given Marginals and Conditionals: I-projections and Conditional Iterative Proportional Fitting, *Statistics and Decisions*, vol. 18, pp. 311-329.
- Csiszár I, 1975. I-divergence Geometry of Probability Distributions and Minimization Problems, *The Annals of Probability*, 3(1): 146-158.
- Csiszár I, 1967. Information-type measures of difference of probability distributions and indirect observation, *Studia Sci. Math. Hungar.*, vol 2, pp. 229-318.
- Deming WE and Stephan FF, 1940. On a least square adjustment of a sampled frequency table when the expected marginal total are known, *Ann. Math. Statist.* 11, 427-444.
- Ding Z, Peng Y and Pan R, 2006. BayesOWL: Uncertainty Modeling in Semantic Web Ontologies, in *Soft Computing in Ontologies and Semantic Web*, Springer-Verlag.
- Ding Z, Peng Y and Pan R, 2004. A Bayesian Approach to Uncertainty Modeling in OWL Ontology, in *Proceedings of the International Conference on Advances in Intelligent Systems – Theory and Applications*, Nov. 15-18, Luxembourg.
- Ding Z, Peng Y, 2004. A Probabilistic Extension to Ontology Language OWL, *Proc. of the 37th Hawaii International Conference on System Sciences*, Big Island, HI.
- Ding Z, 2006. BayesOWL: A Probabilistic Framework for Semantic Web, PhD thesis, Dept. of Computer Science and Electrical Engineering, University of Maryland Baltimore County.
- Doan A, Madhavan J et al. 2003. Learning to Match Ontologies on the Semantic Web, *VLDB Journal*, 303-319.
- Fienberg SE, 1970. An Iterative Procedure for Estimation in Contingency Tables, *Ann. Math. Statist.*, vol. 41, no. 3, pp. 907-917.
- Fukushige Y, 2004. Representing Probabilistic Knowledge in the Semantic Web, Position paper for the W3C Workshop on Semantic Web for Life Sciences, Cambridge, MA, USA.
- Haberman S, 1974. *The Analysis of Frequency Data*, Chicago: University of Chicago Press.
- Iusem AN, De Pierro AR, 1991. On the convergence of Han's method for convex programming with quadratic objective, *Mathematical Programming*, 52: 265-284.
- Jeffrey R, 1983. *The Logic of Decisions*, 2nd Edition, Chicago: University of Chicago Press.
- Jensen F, 1994. *An Introduction to Bayesian Networks*, Aalborg University.
- Jirousek R, Preucil S, 1995. On the Effective Implementation of the Iterative Proportional Fitting Procedure, *Computational Statistics & Data Analysis*, vol. 19, 177-189.
- Jirousek R, Vomlel J, 1995. Inconsistent Knowledge Integration in a Probabilistic Model, *Proceedings of workshop "Mathematical Models for handling partial knowledge in A.I."*, New York, Plenum Publ. Corp.
- Kim Y, Valtorta M, 2004. A Prototypical System for Soft Evidential Update, *Applied Intelligence*,

- 21, 1 (July-August 2004), 81-97.
- Koller D, Levy A and Pfeffer A, 1997. P-CLASSIC: A Tractable Probabilistic Description Logic, In Proceedings of AAAI, pp 390-397.
- Kruihof R, 1937. Telefoonverkeersrekening, De Ingenieur 52, E15-E25.
- Kullback S and Leibler RA, 1951. On Information and Sufficiency. Annals of Mathematical Statistics, vol. 22, pp. 79-86.
- Kyburg HE, 1987. Bayesian and non-Bayesian evidential updating. Artificial Intelligence, 31:271-293.
- Laskey KB, 2007. MEBN: A Language for First-Order Bayesian Knowledge Bases, Artificial Intelligence, 172(2-3).
- Laskey KJ and Laskey KB, 2008. Uncertainty Reasoning for the World Wide Web: Report on the URW3-XG Incubator Group, Fourth International Workshop on Uncertainty Reasoning for the Semantic Web.
- Lukasiewicz T, 2009. Uncertainty Reasoning for the Semantic Web, Web Reasoning and Rule Systems: Third International Conference on Scalable Uncertainty Management.
- Lukasiewicz T, Straccia U, 2008. Tightly coupled fuzzy description logic programs under the answer set semantics for the Semantic Web, Int. J. Semantic Web Inf. Syst. 4(3), 68-89.
- Lukasiewicz T, 2007. Probabilistic description logic programs. International Journal Approximate Reasoning 45(2), 288-307.
- McGuinness DL, Harmelen FV, 2004. OWL web ontology language overview, W3C recommendation.
- Meditskos G, Bassiliades N, 2008. Rule-based OWL Ontology Reasoning Using Dynamic ABOX Entailments, in Proceeding of the 2008 conference on ECAI.
- Mitra P, Noy NF and Jaiswal AR, 2004. OMEN: A Probabilistic Ontology Mapping Tool, In Workshop on Meaning Coordination and Negotiation at the Third International Conference on the Semantic Web (ISWC-2004). Hiroshima, Japan.
- Oh S, Marks II RJ, 1993. Alternating projection onto fuzzy convex sets, in Proceedings of IEEE Conference on Fuzzy Systems, V. 81, pp. 148-155.
- Osterwalder A, Pigneur Y, 2002. An e-Business Model Ontology for Modeling e-Business, 15th Bled Electronic Commerce Conference.
- Pan J, 2007. A flexible ontology reasoning architecture for the semantic web, IEEE Transactions on Knowledge and Data Engineering.
- Pan R, Ding Z, Yu Y and Peng Y, 2005. A Bayesian Network Approach to Ontology Mapping, in Proceedings of the Fourth International Semantic Web Conference (ISWC2005), Galway,

- Ireland, November 6-10.
- Pan R, Peng Y, and Ding Z, 2006. Belief Update in Bayesian Networks Using Uncertain Evidence, 18th IEEE International Conference on Tools with Artificial Intelligence, 441-444.
- Pan R, 2006. Semantically Linked Bayesian Networks: A Framework for Probabilistic Inference over Multiply Bayesian Networks, PhD thesis, Dept. of Computer Science and Electrical Engineering, University of Maryland Baltimore County.
- Pearl J, 1986. Fusion, Propagation, and Structuring in Belief Networks, In *Artificial Intelligence*, 29: 241-248.
- Pearl J, 1987. Evidential Reasoning Using Stochastic Simulation of Causal Models, In *Artificial Intelligence*, 32: 245-257.
- Pearl J, 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman, CA.
- Pearl J, 1990. Jeffery's Rule, Passage of Experience, and Neo-Bayesianism, in *Knowledge Representation and Defeasible Reasoning*, Kyburg HE, Loui RP, and Carlson GN, Eds. Boston: Kluwer Academic Publishers, pp. 245-265.
- Peng Y and Ding Z, 2005. Modifying Bayesian Networks by Probability Constraints, in *Proceedings of 21st Conference on Uncertainty in Artificial Intelligence (UAI-2005)*, Edinburgh, Scotland, July 26-29.
- Peng Y, Ding Z, 2005. Modifying Bayesian Networks by Probability Constraints, in *Proceedings of 21st Conference on Uncertainty in Artificial Intelligence*, Edinburgh, Scotland, July 26-29.
- Prasad S, Peng Y, and Finin T, 2002. A Tool for Mapping Between Two Ontologies Using Explicit Information, *Ontologies in Agent Systems (OAS) workshop*, International Conference on Autonomous Agents and Multi-Agent Systems, Bonogna, Italy.
- Rüschendorf L, 1995. Convergence of the Iterative Proportional Fitting Procedure, *Annual Statistics*, vol. 23, no. 4, pp. 1160-1174.
- Russell, Stuart J, Norvig, Peter, 2003. *Artificial Intelligence: A Modern Approach* (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall.
- Schlicht A, Stuckenschmidt H, 2008. Towards Distributed Ontology Reasoning for the Web, in *Proceedings of the 2008 IEEE/WIC/ACM International on Web Intelligence and Intelligent Agent Technology*.
- Sirin E, Parsia B, Cuenca Grau B, et al. 2007. Pellet: A practical OWL-DL reasoner, *Journal of Web Semantics*, 51-53.
- Stoilos G, Stamou G, et al. 2007. Reasoning with very expressive fuzzy description logics, *Journal of Artificial Intelligence Research*.

- Sure Y et al. 2005. The SWRC Ontology - Semantic Web for Research Communities, In Proceedings of the 12th Portuguese Conference on Artificial Intelligence (EPIA 2005), Springer, Covilha, Portugal.
- Udrea O, Yu D, et al. 2005. Probabilistic ontologies and relational databases, Lecture notes in computer science.
- Valtorta M, Kim Y, and Vomlel J, 2002. Soft Evidential Update for Probabilistic Multiagent Systems, International Journal of Approximate Reasoning, vol. 29, no. 1, pp. 71-106.
- Vomlel J, 1999. Methods of Probabilistic Knowledge Integration, PhD thesis, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University.
- Vomlel J, 2003. Integrating Inconsistent Data in a Probabilistic Model, Journal of Applied NonClassical Logics, pp. 1-20.
- Wagner C, 2002. Probability kinematics and commutativity, Philosophy of Science, vol. 69, pp. 266 – 278.
- Yang Y and Calmet J, 2005. OntoBayes: An Ontology-driven Uncertainty Model, in Proceedings of IAWTIC, pp 457-463.
- Yelland PM, 1999. Market Analysis Using a Combination of Bayesian Networks and Description Logics, Sun Microsystems Technical Report, TR-99-78.
- Zhang S, Peng Y and Wang X, 2008. An Efficient Method for Probabilistic Knowledge Integration, in Proceedings of The 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI-2008), Dayton, Ohio, USA.
- Zhang S, Sun Y and Peng Y, 2009. BayesOWL: A Prototype System for Uncertainty in Semantic Web, Proc. of the 2009 International Conference on Artificial Intelligence, Las Vegas, USA.
- 杨喜权, 曹雪亚等, 2008, 基于贝叶斯网络的本体不确定性推理, 计算机应用, Vol. 28.
- 张连文, 郭海鹏, 2006, 贝叶斯网引论, 科学出版社。
- 马少平, 朱小燕, 2004, 人工智能, 清华大学出版社。
- <http://www.w3.org/2001/sw/> Semantic Web Homepage.
- <http://www.w3.org/RDF/> W3C RDF Homepage.
- <http://www.w3.org/TR/rdf-syntax-grammar/> RDF/XML Syntax Specification.
- <http://www.w3.org/TR/rdf-mt/> RDF Semantics.
- <http://www.w3.org/TR/rdf-schema/> W3C RDF Schema Specification.
- <http://www.w3.org/TR/owl-ref/> OWL Web Ontology Language Reference.

.....

致 谢

在研究学习期间，我有幸得到了三位老师的教导，他们是：王晓蒲教授，彭云教授，霍剑青教授。三位老师深厚的学术功底，严谨的工作态度和敏锐的科学洞察力使我受益良多。在我完成论文的工作期间，老师们不仅认真仔细地指导我的论文，而且对我严格要求，鼓励我多思考，并给我提供了良好的科研条件。衷心感谢他们多年来给予我的悉心教导和热情帮助。

感谢实验室的杨旭、袁泉、郭玉刚等诸位师兄姐妹们，多年来他们一直给我提供了不少帮助，并在精神上给予了我很多的支持。

感谢 Baltimore 的朋友们，他们对我的学习、生活给予了很大的帮助。

.....

最后，谨以此文献给我的父亲和母亲。

2010 年 5 月

宋体 12 磅 (英文用 Times New Roman 体 12 磅), 行距 20 磅, 段前段后 0 磅, 学术论文书写格式同参考文献, “已发表论文”、“待发表论文”等字加粗

期间发表的学术论文与取得的研究成果

黑体 16 磅加粗居中, 单倍行距, 段前 24 磅, 段后 18 磅

在读期间发表的学术论文与取得的研究成果

已发表论文:

- [1] Zhang, S., Peng, Y. and Wang, X.: “An Efficient Method for Probabilistic Knowledge Integration”, in *Proceedings of The 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI-2008)*, Dayton, Ohio, USA, Nov. 3-5, 2008
- [2] Zhang, S., Sun, Y., Peng, Y. and Wang, X.: “BayesOWL: A Prototype System for Uncertainty in Semantic Web”, in *Proceedings of 2009 International Conference on Artificial Intelligence (ICAI-2009)*, Las Vegas, Nevada, USA, Jul. 13-16, 2009
- [3] Zhang, S., Sun, Y., Peng, Y. and Wang, X.: “A Practical Tool for Uncertainty in OWL Ontologies”, in *Proceedings of The Tenth IASTED International Conference on Artificial Intelligence and Applications (AIA2010)*, Innsbruck, Austria, Feb. 15-17, 2010

已接受论文:

- [1] Peng, Y., Zhang, S. and Pan, R.: “Bayesian Network Reasoning with Uncertainty Reasoning”, (Accepted by *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, IJUFKS*)

Font: 12 pt

Page VII: [7] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page VII: [7] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page VII: [7] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page VII: [7] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page VII: [7] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page VII: [7] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page VII: [8] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page VII: [8] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page VII: [8] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page VII: [8] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page VII: [8] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page VII: [8] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page VII: [9] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page VII: [9] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page VII: [10] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: 12 pt

Page VII: [10] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: 12 pt

Page VII: [11] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: 12 pt

Page VII: [11] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: 12 pt

Page VII: [12] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: 12 pt

Page VII: [12] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: 12 pt

Page VII: [13] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: 12 pt

Page VII: [13] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: 12 pt

Page VII: [14] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: 12 pt

Page VII: [14] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: 12 pt

Page VII: [15] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: Times New Roman, 12 pt

Page VII: [15] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: Times New Roman, 12 pt

Page VII: [15] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: Times New Roman, 12 pt

Page VII: [16] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: Times New Roman, 12 pt

Page VII: [16] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: Times New Roman, 12 pt

Page VII: [17] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: Times New Roman, 12 pt

Page VII: [17] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: Times New Roman, 12 pt

Page VII: [18] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: Times New Roman, 12 pt

Page VII: [18] Formatted	ypeng	8/2/2011 4:32:00 PM
--------------------------	-------	---------------------

Font: Times New Roman, 12 pt

Page IX: [19] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [19] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [20] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [20] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [21] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [21] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [22] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [22] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [23] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [23] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [23] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [23] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [23] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [23] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [24] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [24] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [25] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [25] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [26] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: Bold, French (France)

Page IX: [26] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: Bold, French (France)

Page IX: [26] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: Bold, French (France)

Page IX: [26] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: Bold, French (France)

Page IX: [26] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: Bold, French (France)

Page IX: [26] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: Bold, French (France)

Page IX: [27] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: Bold

Page IX: [27] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: Bold

Page IX: [27] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: Bold

Page IX: [27] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: Bold

Page IX: [27] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: Bold

Page IX: [28] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [28] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [29] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [29] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [30] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [30] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [31] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [31] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [32] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [32] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [33] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt

Page IX: [33] Formatted	ypeng	8/2/2011 4:32:00 PM
-------------------------	-------	---------------------

Font: 12 pt