

递推信度传播算法 — 按良序的信度传播

陈峰¹ 刘红¹ 徐文立¹

摘要 针对循环信度传播算法在多环的贝叶斯网中迭代次数较多且不一定收敛的问题, 提出了递推信度传播算法. 它与循环信度传播及其推广算法的区别就在于按某一特定顺序(良序)进行信度传播. 该算法经过一轮信度传播便达到不动点, 显著降低了计算量. 按这种顺序传播信度等价于去掉网络中某些边而解除了网络中的环, 从而使信度不再出现环流. 此算法得到的不动点与循环信度传播算法在收敛时得到的不动点是一致的, 也就是网络的 Bethe 自由能的最小值点. 最后, 实验验证本文所提的算法在实际应用中能有效地降低推理的复杂度.

关键词 贝叶斯网络, 信度传播, 良序, Bethe 自由能

DOI 10.3724/SP.J.1004.2010.01091

Recursive Belief Propagation — Belief Propagation in a Well-order

CHEN Feng¹ LIU Hong¹ XU Wen-Li¹

Abstract Aimed at the loopy belief propagation's unknown convergence and too many iterations on Bayesian networks with cycles, an algorithm named recursive belief propagation (RBP) is proposed. Different from the classical belief propagation, a special order (well-order) is followed by RBP. Passing messages according to this order is equivalent to breaking the cycles sequentially, so the messages cannot circulate and the fixed point is arrived at in one trial. Furthermore, this fixed point is the same as the result of the loopy belief propagation or the minimal point of Bethe free energy. At last, we prove the good performance of the proposed algorithm by experiments.

Key words Bayesian network, belief propagation (BP), well-order, Bethe free energy

自从上世纪 80 年代 Pearl 提出贝叶斯网络及循环信度传播 (Belief propagation, BP) 算法^[1-2] 以来, 概率图模型推理就成为人工智能领域的一个研究热点. BP 算法在无环的图模型中可以得到精确的边缘概率或者后验概率, 但在有环的图模型中只能得到近似的结果, 甚至不收敛^[3-4]. 2000 年之后, 推理问题被解释为自由能最小化问题, BP 算法的不动点被证明是 Bethe 自由能的极小值点^[5], 这一思想的提出, 对研究 BP 算法的收敛性和设计更好的近似算法等问题都起到了重要作用. 近几年来, 学者们研究出了很多其他的 Bethe 自由能最小化的方法^[6-9], 不同的 Bethe 自由能最小化方法对应于不同的信息传播策略. 本文的核心思想也是设计一种收敛的且简单易行的信度传播策略.

在有环的图模型中使用 BP 算法, 信息将在环中循环传播, 信息很可能在重复的路径中传播, 一方面使得传播过程变得冗余, 另一方面, 也可能使得信

息来回振荡而不收敛. 目前解决此类问题的代表方法有基于树的再参数方法^[10]、基于树的序列再加权方法^[11] 等, 它们较经典的 BP 算法而言, 提高了收敛性, 但迭代次数仍然很高. 也有一些算法选择性地去掉一些边, 使得图模型变成一棵树^[12-13], 这类算法由于去边的依据不明确, 最后的结果不一定好. 与以上各种方法不同, 本文从最小化 Bethe 自由能的角度出发, 构造一种使得 Bethe 自由能取最小值的信度传播策略. 该策略选择一种良序进行信息传播, 正是因为这种特定的传播顺序, 使得此算法一次性到达不动点. 按此策略在一个环中传播信息就相当于将此环中的一条边去掉. 当模型中含有许多环时, 则按此策略依次递推地去掉这些环, 每次都使得环上的 Bethe 自由能取最小值, 从而使网络的 Bethe 自由能取得最小值. 该算法的信息传播过程不仅揭示了信度传播的本质, 也更清晰地揭示了误差形成的原因. 当去掉的边的依赖性不强时, 误差较小.

本文第 1 节介绍贝叶斯网及其目前流行的推理算法; 第 2 节提出用于计算边缘概率的递推信度传播算法并分析其复杂度; 第 3 节将递推信度传播算法推广到计算后验概率的情形; 第 4 节举例说明了递推信度传播算法的过程、不同传播顺序的迭代次数比较以及分析误差形成的原因; 第 5 节比较了递推信度传播算法与其他常用近似推理算法的性能; 第 6 节将本文算法应用于视频行为识别中, 体现了

收稿日期 2009-02-23 录用日期 2010-03-17
Manuscript received February 23, 2009; accepted March 17, 2010

国家自然科学基金 (60772050), 国家科技重大专项资助项目 (2009ZX02001) 资助

Supported by National Natural Science Foundation of China (60772050), Important National Science and Technology Specific Projects (2009ZX02001)

1. 清华信息科学与技术国家实验室 清华大学自动化系 北京 100084
1. Tsinghua National Laboratory for Information Science and Technology, Department of Automation, Tsinghua University, Beijing 100084

本算法的实用性.

1 贝叶斯网及其推理

贝叶斯网 (Bayesian network)^[14-15], 也称信度网 (Belief network), 是不确定性知识表达与推理的有力工具. 它采用有向图的形式表达多个变量的联合概率, 如图 1 所示. 图中的节点表示变量, 而边表示变量间的概率依赖关系.

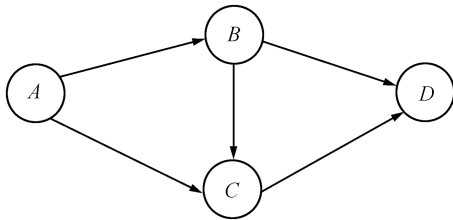


图 1 一个典型的贝叶斯网
Fig. 1 A typical Bayesian network

贝叶斯网的结构体现了变量间的条件独立性, 即一个节点在其父节点条件下与其他的祖先节点独立. N 个节点的贝叶斯网表达了对节点联合概率的如下分解:

$$P(X_{1:N}) = \prod_{i=1}^N P(X_i | \pi(X_i)) \quad (1)$$

其中, $\pi(X_i)$ 表示节点 X_i 的所有父节点. 例如, 图 1 的贝叶斯网表达的联合概率分解是 $P(ABCD) = P(A)P(B|A)P(C|AB)P(D|BC)$.

由于贝叶斯网能如此紧凑地表达联合概率, 基于它可以有效地进行概率推理, 包括计算边缘概率和后验概率. 对于这个问题, 目前为止最主流的算法是信度传播算法. 该算法的主要思想是每个节点利用邻居节点传来的信度和自己的条件概率更新自身的信度, 并将结果再传递给邻居节点. 该算法成立的一个重要假设就是在某个节点条件下, 其父节点和子节点是条件独立的, 这意味着该算法只有在树状结构的图中才能得到准确的结果. 对于有环的贝叶斯网, 目前尚未有既准确又高效的推理算法. 信度传播算法虽然可以作为一种近似的推理算法, 然而, 无论是信度传播算法本身还是它的推广与改进, 均存在迭代次数较多甚至不收敛等问题.

2 递推信度传播算法

相对于循环信度传播算法, 本文提出了递推信度传播算法. 其根本思想是选择一种特定的节点顺序 (良序) 进行信度传播, 一次性到达信息传播的不动点, 这样就可以减少计算量.

在提出递推信度传播之前, 首先介绍良序的概

念. 由图 1 所示的贝叶斯网络可以看到, 这个网络虽然简单, 但是包含两个环. 图中所有节点的一个排列中, 如果父节点总是在孩子节点之前, 则称这个排列为此贝叶斯网络的一个良序. 例如, 图 1 所示的贝叶斯网的一个良序是 $ABCD$.

在网络的所有这些节点中, 根节点的边缘概率是已知的, 根节点正好在良序的起点上. 很显然, 在良序中根节点之后的第一个节点的边缘概率很容易求得. 如图 1 中已知 $P(A)$, 所以下一步可求 $P(B)$, 即 $P(B) = \sum_A P(A)P(B|A) = \sum_A \lambda_{A \rightarrow B} P(B|A)$, 这里 $\lambda_{A \rightarrow B} = P(A)$ 称为由 A 传到 B 的信息.

经过这一步之后, 节点 B 的边缘概率就是已知的了, 下一步就可以求节点 C 的边缘概率, 即 $P(C) \approx \sum_B P(B) \sum_A P(A)P(C|AB) = \sum_B \lambda_{B \rightarrow C} \sum_A \lambda_{A \rightarrow C} P(C|AB)$, 这里得到的 $P(C)$ 是真实的节点 C 的边缘概率的近似, 这种近似等价于去掉了从 A 到 B 的边. 得到节点 C 的边缘概率之后, 就可以求取节点 D 的边缘概率, 即 $P(D) \approx \sum_C P(C) \sum_B P(B)P(D|BC) = \sum_C \lambda_{C \rightarrow D} \sum_B \lambda_{B \rightarrow D} P(D|BC)$, 这一步近似等价于去掉了从 B 到 C 的边. 此时, 网络中所有节点的边缘概率都已经求出. 在节点 C 和 D 的边缘概率计算中都引入了近似处理, 而这种近似处理就是动态地去掉一些边, 如 AB 和 BC , 实际上也就是依次解除了环 ABC 和 BCD . 这样做可以保证信息传播不会出现环流.

对于一般的网络, 可以根据上述计算边缘概率的过程归纳为如下的递推信度传播算法:

步骤 1. 信息由根节点向其中一个只以根节点作为父节点的孩子节点传播, 得到这个孩子节点的边缘概率, 然后去掉从根节点到这个孩子节点的边, 得到一个新的贝叶斯网;

步骤 2. 在新的贝叶斯网中重复步骤 1;

步骤 3. 当网络中的边全部去掉时, 算法终止, 此时得到网络中所有变量的边缘概率.

这个算法中的计算边缘概率的顺序是贝叶斯网的一个良序. 信度传播顺序可由图 2 示意, 并且该顺序可以在进行信度传播之前确定.

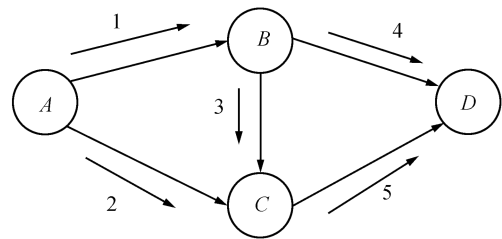


图 2 求边缘概率的信度传播顺序
Fig. 2 The order for marginal probabilities

该算法总是先求得父节点的边缘概率, 再递推出孩子节点的边缘概率, 其递推式可表达为

$$b(X_i) = \sum_{\pi(X_i)} P(X_i|\pi(X_i)) \prod_{j \in \pi(X_i)} b(X_j) \quad (2)$$

其中, $b(X_j)$ 表示节点 X_j 的边缘概率的一个近似, 也是由节点 X_j 向节点 X_i 传播的信息. 该递推式等价于去掉了 X_j 的父节点之间的边.

这里有一个问题: 在计算过程中采用了去边的策略, 引入了近似, 那么能否保证所得到的结果是我们需要的呢? 下面的定理说明以良序进行的传播结果刚好是系统的 Bethe 自由能极小值点. 需要补充说明的是, Bethe 自由能极小值点本身也是近似结果, 但等同于 BP 算法的结果. 这说明本文提出的方法虽然引入近似, 但在结果上等价于 BP 算法, 而计算量大为减少.

定理 1. 对任意的贝叶斯网, 按递推信度传播算法求得的各节点的边缘概率, 正好是以各节点的家族覆盖团为因子节点的因子图^[2] 所对应的变分自由能 (即 Bethe 自由能) 的极小值点.

证明. 见附录. □

如果父节点之间是完全连接的, 假设每个节点的取值为 K 种, 精确推理需 $|\pi(X_j)|K^{|\pi(X_j)|+1}$ 次乘法和 $K(K^{|\pi(X_j)|} - 1)$ 次加法, 而采用递推信度传播算法仅需 $|\pi(X_j)|K^2$ 次乘法和 $|\pi(X_j)|(K^2 - K)$ 次加法, 复杂度从指数函数降到线性函数. 对于图 1 而言, 假设每个节点的取值为 K 种, 用这个算法计算出所有边缘概率仅需 $2K^3 + K^2$ 次乘法和 $K(K - 1) + 2K(K^2 - 1)$ 次加法. 如果不按此顺序传播信息, 即使是在无环的图模型中, 一般而言, 要进行的循环传播次数也正比于网络的直径^[1]. 而本文所提算法只需一次信息传播, 所以计算量大幅度减少.

3 后验概率的求取

前一节介绍了边缘概率的计算, 而后验概率可以看成为计算好的边缘概率与反向传来的信息的合成. 一个节点的后验概率可由贝叶斯公式计算, 即

$$P(X_i|E = e) \propto P(X_i, E_i^+) P(E_i^-|X_i) \quad (3)$$

其中, E_i^+ 表示 X_i 之前的证据, E_i^- 表示 X_i 之后的证据, $E = e = \{E_i^+, E_i^-\}$. 这里的 $P(X_i, E_i^+)$ 可在求边缘概率的程序中遇到证据变量时不求全加和而直接代入证据的值计算得出, 而 $P(E_i^-|X_i)$ 可由反向信息传播得到. 这里为节省冗余计算, 反向传播的起点选择在所有证据均收到前向信息的时刻. 于是得到如下计算后验概率的递推信度传播算法:

步骤 1. 按照贝叶斯网的一种良序进行信息传

播, 直到信息传到了所有的证据变量, 然后, 信息分两个方向传播, 其中一个方向进入步骤 2, 另一个方向进入步骤 3;

步骤 2. 继续沿良序前向传播, 当信息传到最后一个节点时, 前向传播终止;

步骤 3. 从最后遇到的这个证据变量开始逆良序沿路返回, 进行后向信息传播, 当所有证据信息传到良序的第一个节点时, 后向传播终止;

步骤 4. 当前后向传播均终止时, 按贝叶斯公式返回所有节点的后验信度.

上述步骤 2 和步骤 3 可以并行进行. 例如图 1 所示的贝叶斯网, 设证据为 $C = c$, 则计算其他非证据节点后验概率的传播算法如下:

- 1) $\lambda_{A \rightarrow B} = P(A), b(B) = \sum_A \lambda_{A \rightarrow B} P(B|A), \lambda_{A \rightarrow C} = P(A), \lambda_{B \rightarrow C} = b(B);$
- 2) $\lambda_{B \rightarrow D} = b(B), b(D) = \sum_B \lambda_{B \rightarrow D} P(D|Bc);$
- 3) $\lambda_{C \rightarrow A} = \sum_B \lambda_{B \rightarrow C} P(c|AB), \lambda_{C \rightarrow B} = \sum_B \lambda_{A \rightarrow C} P(c|AB);$
- 4) return $b(A) \propto P(A)\lambda_{C \rightarrow A}, b(B) \propto b(B)\lambda_{C \rightarrow B}, b(D) = b(D).$

本算法的信息传播过程可用图 3 示意, 其中 1, 2, 3 对应于步骤 1; 4, 5 对应于步骤 2; 6, 7 对应于步骤 3. 由于节点 $C = c$ 是证据节点不要求后验概率, 所以图中 2 和 3 表示的信息传播过程是可以省略的.

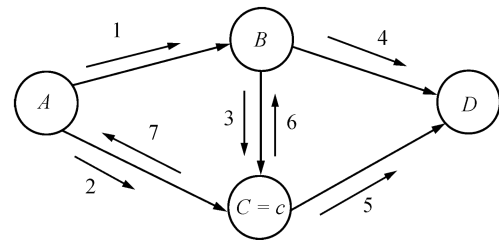


图 3 计算后验概率的顺序

Fig. 3 The order for posteriors

和 Pearl 的信度传播算法相比, 本算法选择了一种特定的顺序, 在证据变量之前的变量间进行了双向的信息传播, 而在证据变量之后的变量只进行了单向传播, 因此复杂度将降低. 特别地, 如果证据变量是根节点, 那么, 经过一个方向的信息传播就可以得到所有的后验概率.

4 算例分析

本节先举例说明递推信度传播的过程, 然后说明按其他的顺序传播需要的迭代次数较多, 最后分析了误差形成的原因.

4.1 含 3 个节点的图模型

一个贝叶斯网总可以合并为如下的 3 个节点的图模型 (或者是图中再减少一些边), 其中 A 是网络中的根节点, B 是只以 A 为父节点的节点, C 为网络中其他的节点合成的大节点, 因此以图 4 为例分析信度传播过程具有代表性意义. 该贝叶斯网对应的联合概率分解为 $P(ABC) = P(A)P(B|A)P(C|AB)$.

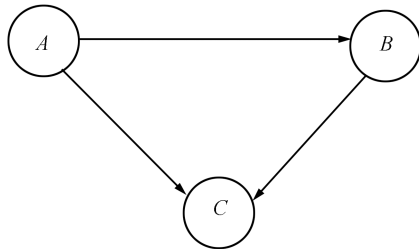


图 4 含有 3 个节点的有环贝叶斯网

Fig. 4 A cyclic Bayesian network with 3 nodes

下面计算使得这个网络的 Bethe 自由能最小化的边缘概率. 可将此联合概率分解为两个势函数之积, 即 $P(ABC) = \Psi(AB)\Psi(ABC)$, 其中 $\Psi(AB) = P(A)P(B|A)$, $\Psi(ABC) = P(C|AB)$. 于是, 该贝叶斯网可用图 5 所示的因子图^[2]表示, 其中圆圈表示变量节点, 方框表示因子节点.

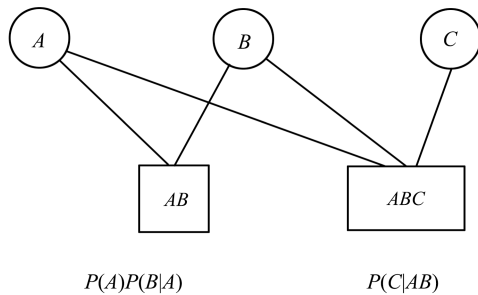


图 5 因子图

Fig. 5 Factor graphs

根据该因子图, 构造 Bethe 自由能, 求出最小值对应的不动点方程, 然后由不动点方程, 按照良序设计出如下的递推信度传播算法.

步骤 1. 初始化. 令 $n(\cdot) = m(\cdot) = 1$, 初始值选任意正数均可, 为简单起见取为 1.

步骤 2. 第一次迭代. 迭代的公式与经典的信度传播算法完全一致, 但按良序进行信度传播.

$$m_{AB \rightarrow A} = \sum_B P(A) P(B|A) n_{B \rightarrow AB} = P(A)$$

$$m_{AB \rightarrow B} = \sum_A P(A) P(B|A) n_{A \rightarrow AB} = P(B)$$

$$n_{A \rightarrow ABC} = m_{AB \rightarrow A} = P(A)$$

$$n_{B \rightarrow ABC} = m_{AB \rightarrow B} = P(B)$$

$$n_{C \rightarrow ABC} = 1$$

$$m_{ABC \rightarrow C} \propto \sum_{AB} P(C|AB) P(A) P(B)$$

$$m_{ABC \rightarrow B} \propto \sum_{AC} P(C|AB) P(A) = 1$$

$$m_{ABC \rightarrow A} \propto \sum_{BC} P(C|AB) P(B) = 1$$

$$n_{B \rightarrow AB} = m_{ABC \rightarrow B} \propto 1$$

$$n_{A \rightarrow AB} = m_{ABC \rightarrow A} \propto 1$$

(4)

从上式可知, 迭代之后得到的 $n_{B \rightarrow AB}$ 和 $n_{A \rightarrow AB}$ 与迭代之前的一样, 如果再进行第二次迭代, 把第一次的结果无论代入哪个递推方程, 结果都将不再改变, 因此按上式表示的顺序迭代, 一次就可以收敛. 这个顺序可用图 6 表示, 其中先按 1, 2, 3, 4, 5 的顺序传播, 再按它的逆向 6, 7, 8, 9, 10 反向传播信息.

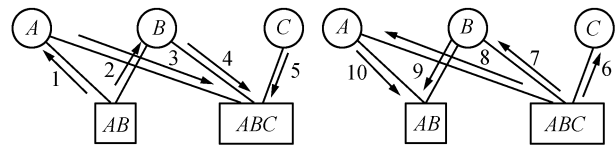


图 6 信息传播顺序

Fig. 6 The order of message passing

根据迭代的信息不动点, 得到如下的边缘分布:

$$b(A) \propto P(A)$$

$$b(B) \propto P(B)$$

$$b(C) \propto \sum_B P(B) \sum_A P(C|AB) P(A)$$

(5)

事实上, 当 $b(A) = P(A)$ 已知时, 上述 10 步中只需第 2 步和第 6 步就可以了. 这里的第 2 个迭代式正好对应递推信度传播算法中的步骤 1, 第 6 个迭代式也是在去掉边之后再重复步骤 1. 也就是说, 这个迭代过程得到的结果, 等价于先后在如图 7 所示的两个无环的贝叶斯网上进行信度传播.

于是, 图 4 所示的有环贝叶斯网可以分解为如图 7 所示的两个无环的贝叶斯网. 在分解之后的两个贝叶斯网上先后执行循环信度算法与在原网络上进行循环信度传播算法的结果完全一样, 得到的边缘概率是原网络的 Bethe 自由能的最小值点.

至此, 我们说明了对于一个三节点的环, 递推信

度传播算法求出的边缘概率是网络的 Bethe 自由能在约束条件下的最小值点.

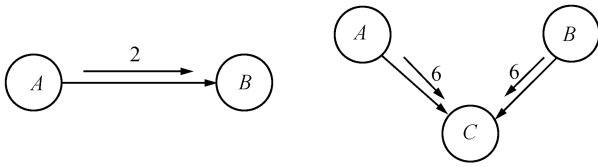


图 7 两个无环图

Fig. 7 Two acyclic Bayesian networks

4.2 不同传播顺序的迭代次数比较

以图 1 所示的贝叶斯网为例, 节点的顺序共有 $P_4^4 = 24$ 种, 如果按这些顺序正向传播或反向传播各算一次迭代, 那么, 为了求得不动点对应的边缘概率需要的迭代次数见表 1.

表 1 不同传播顺序的迭代次数

Table 1 Iteration times for different orders

顺序	次数	顺序	次数	顺序	次数
ABCD	1	BACD	3	CABD	3
ABDC	2	BADC	4	CADB	3
ACBD	3	BCAD	3	CBAD	3
ACDB	3	BCDA	3	CBDA	3
ADBC	2	BDAC	4	CDAB	3
ADCB	2	BDCA	4	CDBA	3
DABC	2	DBAC	4	DCAB	2
DACB	2	DBCA	4	DCBA	2

从表 1 中可以看出, 为求图 1 中的边缘概率, 用 Pearl 传播算法收敛到不动点需要的传播次数最多是 4, 即网络的直径 (网络最长路径的长度), 最少是按良序传播, 仅需 1 次. 对于某个顺序, 需要传播的次数的规律是把该顺序按“正向-反向-正向-反向...”的顺序排列, 如果排到第 n 次就出现良序, 则需要 n 次传播. 例如 ACBD, 排列 ACBD-DBCA-ACBD 已出现良序 ABCD 了, 所以需要 3 次传播. 如果采用并行传播算法, 需要并行传播的次数也正比于网络的直径. Pearl 信度传播需要的迭代次数即使在收敛的情况下也正比于网络的直径, 而本算法总是只需要传播一次信息. 即使使用基于树的再参数化方法或者基于树的再加权算法, 对于该问题也至少选择两棵生成树, 迭代的次数也至少是递推信度传播算法的两倍以上.

总之, 采用良序传播所需迭代次数是最少的. 对于有些贝叶斯网络, 良序不一定是唯一的, 这样采取任何一种良序传播所需的传播次数都是一次.

4.3 信度传播的误差分析

有了递推信度传播算法, 就可以分析 Bethe 自由能最小化算法产生误差的原因了. 对图 1 所示的贝叶斯网使用递推信度传播算法, 节点 A 和节点 B 的边缘概率的计算结果是精确的, 节点 C 的边缘概率由 $P(C) = \sum_{AB} P(C|AB)P(A)P(B|A)$, 近似为 $\hat{b}(C) = \sum_{AB} \hat{P}(C|AB)P(A)P(B)$. 令 $M = \sup_{A,B} \frac{P(B|A)}{P(B)} = \sup_B \frac{1}{P(B)} \sup_A P(B|A)$, 则真实分布与近似分布的 KL 距离为:

$$KL(P(C) \parallel \hat{b}(C)) = \sum_C P(C) \log \frac{P(C)}{\hat{b}(C)} \leq \sum_C P(C) \log M = \log M$$

显然, 当 A 和 B 独立时, 误差为 0 (当 A 和 B 相关性不强时, 则选择此法, 否则将 A 和 B 合为 1 个节点考虑).

图 1 中各节点的条件概率表如图 8 所示.

A	0	1	A \ B	0	1
0	0.4	0.6	0	0.6	0.4
1			1	0.1	0.9

A \ B	0	1	B \ C	0	1
00	0.8	0.2	00	0.2	0.8
01	0.1	0.9	01	0.3	0.7
10	0.7	0.3	10	0.6	0.4
11	0.2	0.8	11	0.1	0.9

图 8 图 1 的条件概率表

Fig. 8 Conditional probability tables of Fig. 1

根据递推信度传播算法, 可求得各节点的边缘概率. 当 $P(B=0|A=1)$ 的值 (对应于图 8 右上角表中 $A=1$ 的那一行) 从 0.1 变化到 0.9 时, 得到 $P(C)$ 的真实值与近似值之差如图 9 所示. 从图 9 可以看出, 当 $P(B=0|A=1) = 0.6$ 时, 真实值与近似值之差为 0, 这是因为此时 A 和 B 独立, 算得的节点 C 的边缘概率为精确的. 当 $P(B=0|A=1)$ 取值越远离时, 误差将越大.

每解除一个环时, 都存在上述的近似. 所以在含有很多环时, 在良序后面的节点的边缘概率将受其父节点的边缘概率的影响, 这种累计的误差有时会越来越大, 也有时是前后互相抵消的. 但总的来说, 只有当进行了近似的那些边所确定的依赖性本身就不强时, 算法才得到较好的结果. 如图 10 所示, 当 $P(B=0|A=1)$ 的值从 0.1 变化到 0.9 时, 得到

$P(D)$ 的真实值与近似值之差并不是像图 9 一样的趋势. 这就说明 $P(D)$ 的误差与 A 和 B 的独立性的关系并不是很明确, 有可能 A 和 B 的依赖性越强, $P(D)$ 的误差反而越小.

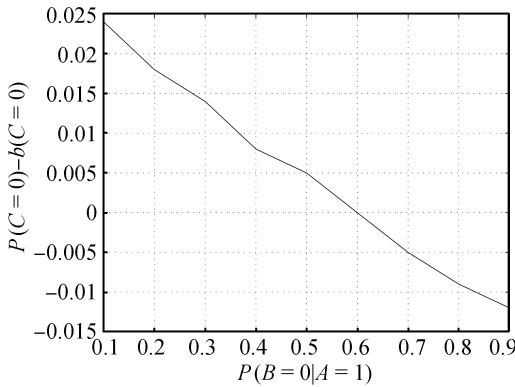


图 9 $P(C)$ 的误差
Fig.9 Errors of $P(C)$

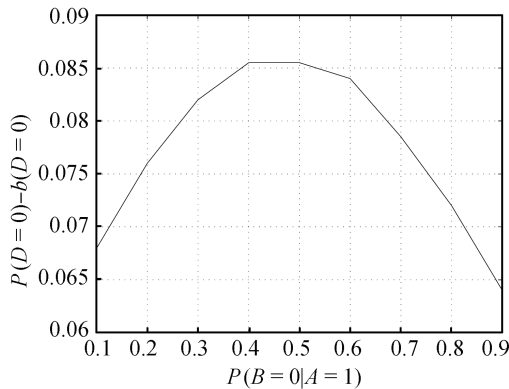


图 10 $P(D)$ 的误差
Fig.10 Errors of $P(D)$

5 性能比较

本节将本文所提的推理算法与几种常见近似推理算法的性能进行了比较.

近似推理算法可分为确定性近似和不确定性近似两类. 确定性近似推理算法的主要代表是变分推理算法, 而不确定性的代表是似然加权与吉布斯采样等随机抽样算法. 总体来讲, 变分推理的时间复杂度较低, 但对于环较多或者结构特别复杂的网络误差较大, 而采样算法适应性较好, 但收敛速度较慢.

本文所提算法属于确定性推理算法. 图 11 为一个常见的有环动态贝叶斯网的三个节点的后验概率的绝对误差曲线. “RBP” 表示递推信度传播算法的误差 (也就是循环信度传播算法的误差, 因为它们的不动点是一样的), “Gibbs100” 表示吉布斯采样 100 个样本时的误差, “Lw100” 表示似然加权采样 100 个样本时的误差. 从图 11 中可以看出, 在这样

的样本数目选择下, 递推信度传播算法的误差并不比采样算法的误差大. 从执行时间上看, “RBP” 用了 27 ms, 而 “Gibbs100” 用了 40 ms, “Lw100” 用了 110 ms. 这说明达到同样的计算精度时, 采样算法所需的时间一般比递推信度传播的时间多. 而且, 当采样样本数更多时, 后两种算法需要的时间成倍增加. 确定性近似推理算法由于其复杂度较低而成为近年来概率推理的研究热点.

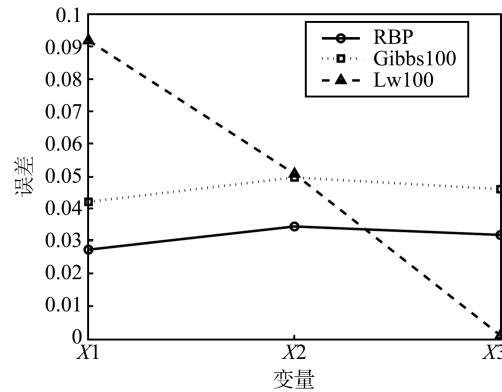


图 11 三种算法的误差
Fig.11 Errors of three algorithms

6 算法应用

现实问题中常常遇到结构非常复杂的贝叶斯网络, 对推理复杂度的要求很高. 例如, 在基于视频的行为识别采用的贝叶斯网络复杂度很高. 这种很高的复杂度常常影响它的实时应用. 图 12 中的动态贝叶斯网是本课题组用来进行行为识别的一个例子^[16].

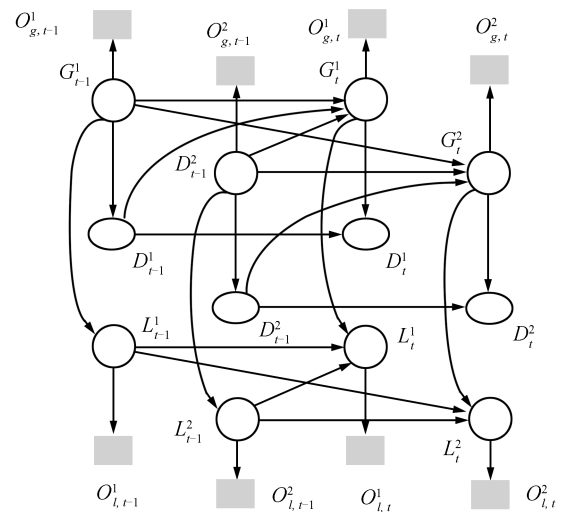


图 12 两人交互行为识别的 CHDS-DBN 模型
Fig.12 The CHDS-DBN model for recognizing interactive motion of two people

图 12 中, G 、 L 和 D 分别称为整体行为状态、局部行为状态以及驻留时间状态; G^c 、 L^c 和 D^c 分别表示第 c 个“单人集合”中的对应状态; G_t^c 、 L_t^c 、 D_t^c 分别表示时刻 t 第 c 个“单人集合”中的对应状态. 用本文所提算法算得的 4 种行为状态的后验概率如图 13 和图 14 所示.

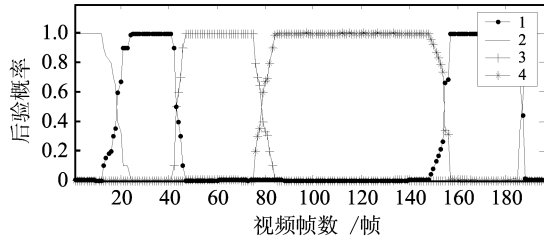


图 13 整体行为状态 G^1 的后验概率

Fig. 13 Posteriors for global motion state G^1

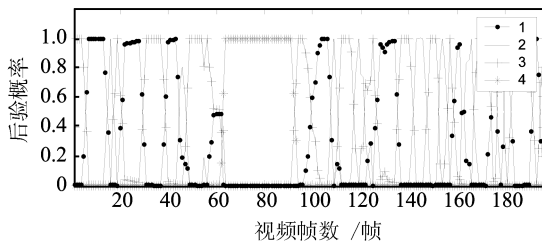


图 14 局部行为状态 L^1 的后验概率

Fig. 14 Posteriors for local motion state L^1

如果采用以往的信度传播算法, 模型的时间片有多少个就要进行多少轮的信息迭代, 而采用按良序的递推信度传播, 只需要从前往后传递先验信息, 再从后往前传递证据信息就可以算得后验概率. 也就是说, 具有 K 帧的视频, 只需要一轮信息传播, 而以往的信息传播算法需要的迭代次数正比于 K . 采用本文所提算法, 降低了信息迭代的次数, 有利于该算法在实时监控等领域中的应用.

7 结论

本文所提算法按某种顺序沿贝叶斯网的有向边正反向各传递一次信息, 就能得到所需的边缘概率和后验概率. 这种顺序正好是贝叶斯网的一种良序. 在一般情况下本文算法比 Pearl 信度传播算法迭代的次数要少得多, 而得到的结果和 Pearl 信度传播算法在收敛时是一致的, 并在一定的条件下, 误差不会很大. 在具体应用中, 本算法更具有实用性.

致谢

感谢美国联合技术公司的资助.

附录 定理 1 的证明

证明. 在一般情况下, 一个贝叶斯网表示的联合概率分解为 $P(X) = \prod_{i=1}^n P(X_i | \pi(X_i))$. 令 $C_i = \{X_i, \pi(X_i)\}$ 为节点 X_i 的家族覆盖团. 可选 C_i 作为因子节点构造相应的因子图. 于是相应的 Bethe 自由能的不动点方程^[3] 为:

$$\begin{aligned} n_{i \rightarrow C_i} &= \prod_{\{j | i \in C_j, j \neq i\}} m_{C_j \rightarrow i} \\ n_{i \rightarrow C_j} &= m_{C_i \rightarrow i} \prod_{\{k | i \in C_k, k \neq i, j\}} m_{C_k \rightarrow i} \\ m_{C_i \rightarrow i} &= \sum_{C_i \setminus i} P(X_i | \pi(X_i)) \prod_{\{j | j \in C_i, j \neq i\}} n_{j \rightarrow C_i} \\ m_{C_j \rightarrow i} &= \sum_{C_j \setminus i} P(X_j | \pi(X_j)) n_{j \rightarrow C_j} \prod_{\{k | k \in C_j, k \neq i, j\}} n_{k \rightarrow C_j} \\ &\forall j \in \{j | i \in C_j, j \neq i\} \end{aligned} \quad (A1)$$

其中, $n_{i \rightarrow C_j}$ 表示从变量节点 X_i 传到因子节点 C_j 的信息, $m_{C_j \rightarrow i}$ 表示由因子节点 C_j 传到变量节点 X_i 的信息. 在信息传播过程中, $n(\cdot)$ 和 $m(\cdot)$ 是不断更新的, 当它们不再变化时, 信息传播就收敛了, 此时称为信度传播已收敛到不动点. 这些不动点就应该满足式 (A1), 由满足式 (A1) 的信息 $n(\cdot)$ 和 $m(\cdot)$ 得到的边缘概率就是 Bethe 自由能的极小值点^[3, 5].

以下证明由递推信度传播算法求得的信息也满足式 (A1). 按递推信度传播算法, 可求得

$$\begin{aligned} n_{i \rightarrow C_i} &= 1 \\ m_{C_j \rightarrow i} &= 1, \quad \forall j \in \{j | i \in C_j, j \neq i\} \\ n_{i \rightarrow C_j} &= b(X_i), \quad \forall j \in \{j | i \in C_j, j \neq i\} \\ m_{C_i \rightarrow i} &= \sum_{C_i \setminus i} P(X_i | \pi(X_i)) \prod_{\{j | j \in C_i, j \neq i\}} b(X_j) \end{aligned} \quad (A2)$$

其中, $b(X_i) \propto m_{C_i \rightarrow i} \prod_{\{j | i \in C_j, j \neq i\}} m_{C_j \rightarrow i}$. 式 (A2) 中第 1 式成立的原因是势函数本身就存放在因子节点上, 变量节点传给自己的家族因子的信息为 1; 第 2 式是由变量的家族因子传给它的父节点的信息, 是从后往前传的信息, 在求边缘概率时, 该信息为 1; 第 3 式为变量节点传给孩子节点的家族因子的信息, 应为自己的边缘概率; 第 4 式也就是正文中的式 (2), 是递推的关键式. 考虑到 $b(X_i) \propto m_{C_i \rightarrow i} \prod_{\{j | i \in C_j, j \neq i\}} m_{C_j \rightarrow i}$, 这是由信息计算边缘概率的公式, 然后将式 (A2) 代入式 (A1), 可知式 (A1) 是成立的. \square

References

- 1 Pearl J. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 1986, **29**(3): 241–288
- 2 Kschischang F, Frey B, Loeliger H. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 2001, **47**(2): 498–519
- 3 Heskes T. On the uniqueness of loopy belief propagation fixed points. *Neural Computation*, 2004, **16**(11): 2379–2413
- 4 Ihler A, Fisher J, Willsky A. Loopy belief propagation: convergence and effects of message errors. *The Journal of Machine Learning Research*, 2005, **6**(5): 905–936

- 5 Yedidia J S, Freeman W T, Weiss Y. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 2005, **51**(7): 2282–2312
- 6 Nishiyama Y, Watanabe S. Generalization of concave and convex decomposition in Kikuchi free energy. In: Proceedings of the 18th International Conference on Artificial Neural Networks. Prague, Czech Republic: Springer, 2008. 51–60
- 7 Goldberger J, Kfir H. Serial schedules for belief-propagation: analysis of convergence time. *IEEE Transactions on Information Theory*, 2008, **54**(3): 1316–1318
- 8 Hazan T, Shashua A. Convergent message-passing algorithms for inference over general graphs with convex free energies. In: Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence. Helsinki, Finland: AUAI Press, 2008. 264–273
- 9 Meltzer T, Globerson A, Weiss Y. Convergent message passing algorithms — a unifying view. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence. Montreal, Canada: AUAI Press, 2009. 393–401
- 10 Wainwright M, Jordan M. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 2008, **1**(1-2): 1–305
- 11 Kolmogorov V. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006, **28**(10): 1568–1583
- 12 Choi A, Darwiche A. A variational approach for approximating Bayesian networks by edge deletion. In: Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence. Cambridge, USA: AUAI Press, 2006. 80–89
- 13 Minka T, Yuan Q. Tree-structured approximations by expectation propagation. In: Proceedings of the Conference on Neural Information Processing Systems. Massachusetts, USA: The MIT Press, 2003. 1–8
- 14 Zhang Lian-Wen, Guo Hai-Peng. *Introduction of Bayesian Network*. Beijing: Science Press, 2005. 31–65 (张连文, 郭海鹏. 贝叶斯网引论. 北京: 科学出版社, 2005. 31–65)
- 15 He Ying-Jie, Liu Wei-Yi. Learning Bayesian network by first learning Markov network. *Journal of Computer Research and Development*, 2002, **39**(1): 87–99 (何盈捷, 刘惟一. 由 Markov 网到 Bayesian 网. 计算机研究与发展, 2002, **39**(1): 87–99)
- 16 Du You-Tian. Study on Human Activity Recognition Based on Probabilistic Network [Ph. D. dissertation], Tsinghua University, China, 2008 (杜友田. 基于概率网络的人行为识别研究 [博士学位论文], 清华大学, 中国, 2008)



陈 峰 清华大学自动化系副教授. 主要研究方向为概率图模型和视频图像分析. 本文通信作者.

E-mail: chenfeng@mail.tsinghua.edu.cn
(**CHEN Feng** Associate professor in the Department of Automation, Tsinghua University. His research interest covers visual surveillance, and image and video processing. Corresponding author of this paper.)



刘 红 清华大学自动化系硕士研究生. 2007 年获得江苏大学电气工程及其自动化专业学士学位. 主要研究方向为概率网络与模式识别.

E-mail: liuhong07@mails.tsinghua.edu.cn

(**LIU Hong** Master student in the Department of Automation, Tsinghua University. He received his bachelor degree from Jiangsu University in 2007. His research interest covers probabilistic graph model and pattern recognition.)



徐文立 清华大学自动化系教授. 主要研究方向为计算机视觉, 图像视频处理, 机器人和自动控制.

E-mail: xuwl@mail.tsinghua.edu.cn
(**XU Wen-Li** Professor in the Department of Automation, Tsinghua University. His research interest covers computer vision, image and video processing, robotics, and automatic control.)