

## 6. 近似推論アルゴリズム

植野真臣  
電気通信大学  
大学院情報システム学研究所

## 1. ポリツリーアルゴリズム

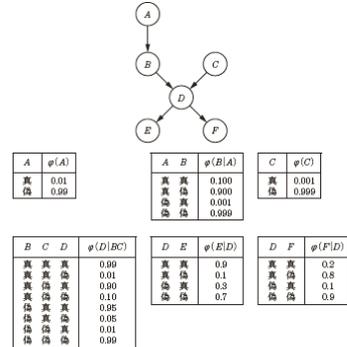


図 5.1 連結台木のベイジアンネットワーク<sup>1)</sup>

## 2. ポリツリーアルゴリズム

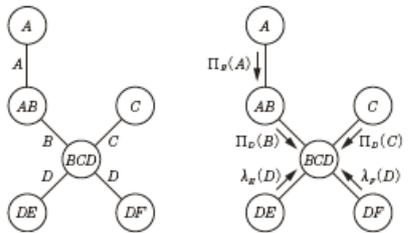


図 5.2 ポリツリーアルゴリズムによるメッセージパッシング

この条件下でのジョイントリーアルゴリズムの確率更新をポリツリーアルゴリズムと呼ぶ。さらに、ポリツリーアルゴリズムでのノード  $U$  から子ノード  $X$  へのメッセージを  $\pi_X(U)$  と書き、因果サポート (causal support) と呼ぶ。また、子ノード  $Y$  から親ノード  $X$  へのメッセージを  $\lambda_Y(X)$  と書き、 $Y$  から  $X$  への診断サポート (diagnostic support) と呼ぶ (図 5.2 右図)。

$U = \{U_1, \dots, U_i, \dots, U_{pa_X}\}$ ,  $Y = \{Y_1, \dots, Y_j, \dots, Y_{ch_X}\}$  とし,  $pa_X$  を  $X$  の親ノード数,  $ch_X$  を  $X$  の子ノード数とする。結果として、ノード  $X$  の親ノード  $U$  と子ノード  $Y_j$  の同時周辺分布は以下のように書ける。

$$p(XU, e) = \lambda_e(X) \phi(X | U) \prod_i \pi_X(U_i) \prod_j \lambda_{Y_j}(X). \quad (5.1)$$

ここで、 $\lambda_e(X)$  は、 $e$  がエビデンス  $e$  に一致していれば  $\lambda_e(X) = 1$ 、それ以外で 0 となるダミー変数である。

これより、以下が成り立つ。

$$\lambda_X(U_i) = \sum_{x \in \mathcal{X}(U_i)} \lambda_e(X) \phi(X | U) \prod_{k \neq i} \pi_X(U_k) \prod_j \lambda_{Y_j}(X), \quad (5.2)$$

$$\pi_{Y_j}(X) = \sum_U \lambda_e(X) \phi(X | U) \prod_i \pi_X(U_i) \prod_{k \neq j} \lambda_{Y_k}(X). \quad (5.3)$$

一般にノードはすべての隣接ノードからメッセージを受け取った後のみ、隣接ノードにメッセージを送る。一つしかメッセージがない場合、メッセージを受け取った直後にそのノードにメッセージを送る。

したがって、隣接ノードが単一の場合、そのノードはただちに隣接ノードにメッセージを送信する。すなわち、ノード  $X$  が単一親ノード  $U$  しかもたない葉ノードであれば、メッセージは以下のように送信される。

$$\lambda_X(U) = \sum_X \lambda_e(X) \phi(X | U).$$

また、単一子ノード  $Y$  しかもたない親ノード  $X$  のメッセージは以下のように送信される。

$$\pi_Y(X) = \lambda_e(X) \phi(X)$$

## 例

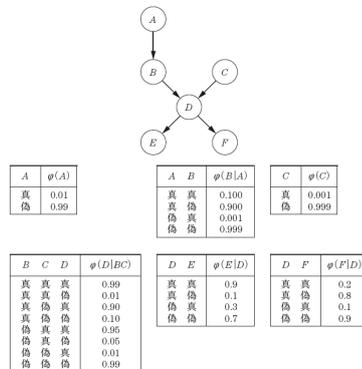


図 5.1 連結台木のベイジアンネットワーク<sup>1)</sup>

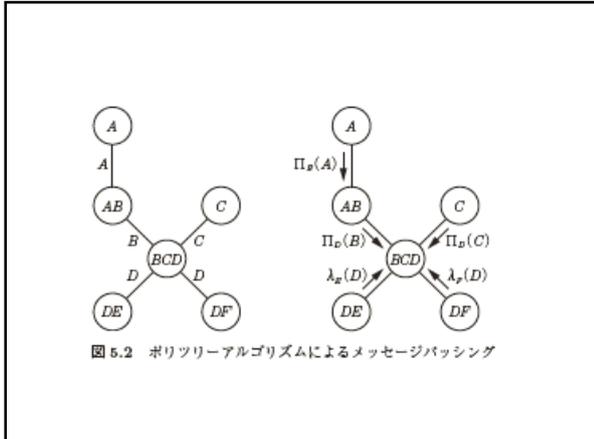


図 5.2 は、ノード E が真であるエビダンスを所与としたとき、ノード D への確率更新のプロセスを示している。このとき、つぎの  $\pi$  メッセージが得られる。

表 5.1 変数 D のファミリーの同時周辺分布

B	C	D	$p(B, C, D, e)$
真	真	真	$1.7751 \times 10^{-10}$
真	真	偽	$5.9700 \times 10^{-9}$
真	偽	真	$1.6103 \times 10^{-9}$
真	偽	偽	$5.9640 \times 10^{-9}$
偽	真	真	$8.5530 \times 10^{-10}$
偽	真	偽	$1.4970 \times 10^{-9}$
偽	偽	真	$8.9721 \times 10^{-9}$
偽	偽	偽	$2.9611 \times 10^{-1}$

$\pi_D(A=1) = 0.01, \quad \pi_D(A=0) = 0.99,$   
 $\pi_D(B=1) = 0.00199, \quad \pi_D(B=0) = 0.99801,$   
 $\pi_D(C=1) = 0.001, \quad \pi_D(C=0) = 0.999.$

ただし、これまでと同様に変数の値は真のとき 1、偽のとき 0 と示す。また、以下の  $\lambda$  メッセージを得る。

$\lambda_E(D=1) = 0.9, \quad \lambda_E(D=0) = 0.3,$   
 $\lambda_F(D=1) = 1, \quad \lambda_F(D=0) = 1.$

変数 D のファミリーについての同時周辺分布は以下のように簡単に計算でき、

$p(B, C, D, e) = \phi(D|B, C)\pi_D(B)\pi_D(C)\lambda_E(D)\lambda_F(D)$

表 5.1 を得る。表を縦にすべて足し合わせることで、 $p(e) \approx 0.3076$  が得られる。

また、D から C へのメッセージを計算し、 $p(C, e) = \sum_{B, D} p(B, C, D, e)$  を用いれば、以下のように簡単に変数 C の周辺分布も得ることができる。

$\lambda_D(C=1) = 0.8700, \quad \lambda_D(C=0) = 0.3071,$   
 $p(C=1, e) = 0.0009, \quad p(C=0, e) = 0.3067$

結果として、条件付き周辺確率  $BEL(XU) = p(XU | e)$  を求めるためには以下のように正規化すればよい。

$$BEL(XU) = \eta \lambda_e(X) \phi(X | U) \prod_i \pi_X(U_i) \prod_j \lambda_{Y_j}(X).$$

ここで、 $\eta$  は  $BEL(XU) = p(XU | e)$  が確率分布となるようにするための (和が 1 となるようにするための) 正規化定数である。

## ルーピーベリフプロパゲーション

ポリツリーアルゴリズムは複雑結合木のための特殊なアルゴリズムであった。これを強引に、一般的なベイジアンネットワークの構造である複雑結合グラフに適用してみよう。いま、図 5.3 への適用を考えよう。まず、メッセージが単一ノードのみに伝播されるように、ノード E からノード C へメッセージ  $\lambda_E(C)$  を送ることから始めるとしよう。この場合、つぎにはどのメッセージも送られなくなってしまふ。なぜならば、隣接ノードからのすべてのメッセージが送られて来ないとメッセージを送信できないからである。

図 5.3 ルーピーベリフプロパゲーションの動作例

そこで、足りない情報を適当な値で補い、繰り返しメッセージパッシングを繰り返すことで、一般的な複雑結合グラフでの確率更新を実現するというアイデアが提案された。このアルゴリズムをルーピーベリフプロパゲーション (loopy belief propagation) と呼ぶ (Frey and MacKay 1997, Murphy, Weiss and Jordan 1999)。アルゴリズム 19 にルーピーベリフプロパゲーションの疑似コードを示している。まず、2. ですべてのメッセージを一様分布に初期化する。すなわち、最初は、エビダンスの隣接ノード以外の変数は、この初期値が各ノードに送信されたと仮定してメッセージを隣接ノードに送信する。t 回目の繰返しでは、t-1 回目に他の隣接ノードから受信したメッセージを用いて、すべての隣接ノードにメッセージを送信する。そして、メッセージが収束するまでこれを繰り返すというアルゴリズムである。

アルゴリズム 19 (ルーピーベリフプロパゲーション:  $LBP(G, \theta)$ )

- Input: ベイジアンネットワーク  $(G, \theta)$
- $e$ : エビダンス
- Output: approximate marginals  $BEL = p(XU | e)$

```

main
1.  $t \leftarrow 0$ 
2. すべてのメッセージを一様分布  $\pi_X^0, \lambda^0$  に初期化
3. while メッセージが収束していない do
4.    $t \leftarrow t + 1$ 
5.   for 観測ノード集合 U をもつすべての変数 X do
6.     for 各観測ノード変数  $U_i \in U$  do
7.        $\lambda_{X_i}^t(U_i) = \eta \sum_{X \setminus \{U_i\}} \lambda_e(X)$ 
8.      $\phi(X | U) \prod_{i \in U} \pi_X^{t-1}(U_i) \prod_j \lambda_{Y_j}^{t-1}(X)$ , ここで  $\eta$  は正規化係数
9.   end for
10.  for 各子ノード  $Y_j$  do
11.      $\pi_{Y_j}^t(X) = \eta \sum_{U \setminus \{Y_j\}} \lambda_e(X | U) \prod_i \pi_X^{t-1}(U_i)$ 
12.      $\times \prod_{k \neq j} \lambda_{Y_k}^{t-1}(X)$ 
13.   end for
14. return ファミリー XU について
     $BEL(XU) = \eta \lambda_e(X) \phi(X | U) \prod_i \pi_X^t(U_i) \prod_j \lambda_{Y_j}^t(X)$ 
    を出力
15. end procedure
    
```

ここで問題となるのはメッセージがいつ収束するかであるが、残念ながら、ルービープリアリーフプロバゲーションでは収束の保証性はない。さらに、収束率は、メッセージスケジュール (message schedule) と呼ばれるメッセージが更新される順序に依存する。アルゴリズム 19 は 1 回の繰返しですべてのメッセージの収束を待つので、並列スケジュール (parallel schedule) と呼ばれ、正確な更新順序があらかじめ既知の場合も反映されないアルゴリズムである。これに比べ、各メッセージごとに収束させる逐次スケジュール (sequential schedule) はより柔軟なアルゴリズムであるといえる。

図 5.3 で、例えば、以下の順序でメッセージが計算されるとしよう。

$$\pi_B(A), \pi_C(A), \pi_D(B), \pi_D(C), \pi_E(C), \lambda_D(B), \lambda_B(A), \lambda_E(C), \lambda_D(C), \lambda_C(A).$$

例えば、 $\pi_E(C)$  の計算を  $\pi_C(A)$  と  $\lambda_D(C)$  を用いて計算する場合、繰返し  $t$  の時点で  $\pi_C(A)$  は  $\pi_E(C)$  より先に計算されているのでそれを用いられるが、 $\lambda_D(C)$  は一つ前のセッション  $t-1$  の時点のメッセージを用いる。

実際には、最適なメッセージスケジュールはわからないので、逐次スケジュールでは、繰返し時点  $t$  ごとに順序をランダムに変化させるということが行われる。しかし、ルービープリアリーフプロバゲーションではどのように慎重に順序を選んだとしても収束の保証はないし、たとえ収束してもよりよい収束点が存在する可能性がある。

5.3 カルバック・ライブラーダイバージェンスによる評価

二つの確率分布の距離を計測するためにカルバック・ライブラー (KL) ダイバージェンス (Kullback-Libler divergence, KL divergence) が一般に用いられる。例えば、エビタンス  $\theta$  を所与としたときの事後分布  $p(X|\theta)$  とその近似  $p(X|\phi)$  の KL ダイバージェンスは

$$KL(p(X|\theta), p(X|\phi)) = \sum_x p(x|\theta) \log \frac{p(x|\theta)}{p(x|\phi)} \quad (5.4)$$

と定義できる。KLダイバージェンスは非負の量であり、二つの分布が一致するときのみ最小値 0 となる。ただし、厳密には KL ダイバージェンスは対象性が成り立たず、距離空間ではない。

この KL ダイバージェンスを用いると真の分布  $p(X|\theta)$  とその近似  $p(X|\phi)$  について以下のような定理が成り立つ (Yedida, Freeman and Weiss 2000)。

定理 31  $p(X)$  をファミリー  $XU$  をもつベイジアンネットワーク  $\{G, \Theta\}$  とする。真の分布  $p(X|\theta)$  とその近似  $p(X|\phi)$  の KL ダイバージェンスは以下のように書くことができる。

$$KL(p(X|\theta), p(X|\phi)) = -ENT^*(X|\theta) - \sum_{XU} E_{p_\theta}(\log \lambda_\theta(X)) \phi(X|U) + \log p(\theta).$$

ここで、 $ENT^*(X|\theta) = -\sum_x p(x|\theta) \log p(x|\theta)$  は  $p(X|\theta)$  のエントロピー、 $E_{p_\theta}(\log \lambda_\theta(X)) \phi(X|U) = \sum_{XU} p(xu|\theta) \log \lambda_\theta(x) \phi(x|u)$  は近似された条件付き確率で重みづけられた真の条件付き確率パラメータを示している。

KL ダイバージェンスを最小化する近似  $p(X|\phi)$  を求めるのが、KL ダイバージェンス最小化について以下の最適問題である。

系 1  $p(X|\theta)$  は以下を最大化するとき、 $KL(p(X|\theta), p(X|\phi))$  を最小化する。

$$ENT^*(X|\theta) + \sum_{XU} E_{p_\theta}(\log \lambda_\theta(X)) \phi(X|U). \quad (5.5)$$

これより、以下の性質が導かれる。

- $p(X|\theta)$  は期待値  $E_{p_\theta}(\log \lambda_\theta(X)) \phi(X|U)$  を大きくするほど近似がよい。
- エビタンスを得た後のネットワークの近似確率  $p(X|\phi)$  が、一般分布に近づくほど近似がよい。

アルゴリズム 19 では、以下のようなファミリーごとのファクターに因子分解されている。

$$p(X|\theta) = \prod_{XU} \frac{p(XU|\theta)}{\prod_{U \in \Theta} p(U|\theta)}. \quad (5.6)$$

ここで、 $XU$  はネットワーク中のファミリー集合、 $U$  は対応するファミリー集合に現れる各親ノードを範囲とする。すなわち、一つの項は一つのファミリーとそれに含まれる一つの親ごとを単位として割り振られている。

ベイジアンネットワークがポリアリーフ構造でない場合、 $p(XU|\theta)$  に対して、ポリアリーフ構造による近似  $p(XU|\phi)$  を当てはめることになる。

このとき、式 (5.6) から近似確率  $p(X|\phi)$  のエントロピーは以下となる。

$$ENT^*(X|\theta) = -\sum_{XU} p(xu|\theta) \log \frac{p(xu|\theta)}{\prod_{U \in \Theta} p(u|\theta)}. \quad (5.7)$$

ここで、 $xu$  はそのファミリーの値を範囲とし、 $u$  はその値に対応した親ノードパターンの値  $u$  の中の一つの親ノードの値を範囲とする。

定理 32  $p(X)$  をファミリー集合  $XU$  をもつベイジアンネットワーク  $\{G, \Theta\}$  から発生されるネットワークの変数の同時確率分布とする。以下の式が最大値として収束したとき、 $\mu_u = BEL(u)$  および  $\mu_{xu} = BEL(xu)$  は収束している。

$$ENT^*(X|\theta) + \sum_{XU} E_{p_\theta}(\log \lambda_\theta(X)) \phi(X|U) = -\sum_{XU} \mu_{xu} \log \frac{\mu_{xu}}{\prod_{u \in \Theta} \mu_u} + \sum_{XU} \mu_{xu} \log \lambda_\theta(X) \phi(X|U). \quad (5.8)$$

ただし、

$$\sum_u \mu_u = \sum_{xu} \mu_{xu} = 1.$$

各ファミリーと各親ノードを固定して、あるファミリー  $Y$  の値  $y$  について

$$\sum_{xu=y} \mu_{xu} = \mu_y$$

が成り立つ。

すなわち、この定理では、周辺確率  $\mu_u = BEL(u)$  および  $\mu_{xu} = BEL(xu)$  の収束が、KL ダイバージェンス中の式 (5.8) の値も最大値に収束することを示している。

5.4 ジョイングラフ近似によるルービープリアリーフプロバゲーション

ルービープリアリーフプロバゲーションでは、一般的なネットワークの確率更新に、複結合木での確率更新を強引に繰り返すことにより、周辺事後確率を近似的に求めるという手法であったが、収束の保証はない。そこで、複結合木構造の近似をより緩和して一般化することにより、より収束確率を向上させることができると考えられる。厳密解であるジョイントリー構造からジョイングラフによる近似を考える (Yedida, Freeman and Weiss 2000, 2005)。

周辺事後分布  $p(X|\theta)$  は、ジョイントリーのクラスター  $C$  とセパレータ  $S$  によって以下のように書ける。

$$p(X|e) = \frac{\prod_C p(C|e)}{\prod_S p(S|e)} \tag{5.9}$$

このとき、以下の定理が成り立つ。

定理 33  $p(X|e)$  が以下の周辺事後分布 (5.9) をもつとき、そのエントロピーは

$$ENT(X|e) = \sum_C ENT(C|e) - \sum_S ENT(S|e)$$

となる。

すなわち、クラスターとセパレータの同時確率が利用可能であれば、KL ダイバージェンスが容易に計算できることである。結果として、KL ダイバージェンスを最大化することは、以下の式を最大化することである。

$$ENT(X|e) + \sum_{x \in \mathcal{X}} E_p(\log \lambda_x(X)p(x|U)) = - \sum_C \sum_{c \in \mathcal{C}} \mu_c \log \mu_c + \sum_S \sum_{s \in \mathcal{S}} \mu_s \log \mu_s + \sum_{x \in \mathcal{X}} \sum_{u \in \mathcal{U}} \mu_{xu} \log \lambda_x(X)p(x|u) \tag{5.10}$$

ただし、 $\mu_c, \mu_s, \mu_{xu}$  は、それぞれ、 $p(C|e), p(S|e), p(xu|e)$  を示している。クラスターとセパレータがジョイントリーのものもある場合、式 (5.9) は厳密に成り立つ。ここでは、ジョイントリーの一般形であるジョイングラフ (定義 57: [共通のノードをもつ二つのクリーク間すべてにエッジが引かれているクリークグラフ]) を導入することにより、よりよい近似としてのルーベリーブーフプロバゲーションを導く。

定義 57 より、ジョイントリーは「二つのクラスターに属するすべての共通ノードがそれらの間に存在する路のすべてのクラスターにも属している木」を示しているが、ジョイングラフは、二つのクラスターに属するすべての共通ノードがそれらの間に存在する路のすべてのクラスターにも属している必要はなく、いくつかのクラスターに属していればよい。結果として、ジョイングラフは、ジョイントリーの厳密な条件をより緩和したグラフであることがわかる。

結果として、ジョイントリーでの周辺事後分布 (5.9) は厳密であり、これをジョイングラフで近似した周辺事後分布は以下のように書ける。

$$p'(X|e) = \frac{\prod_C p(C|e)}{\prod_S p'(S_{ij}|e)} \tag{5.11}$$

当然、ジョイングラフがジョイントリー構造であれば式 (5.11) は厳密である。ジョイングラフの一種であるデュアルジョイングラフ (dual joingraph) を用いれば、分解 (5.11) は積結合木構造による近似 (5.7) を表現することもできる。

定義 82 以下のグラフ  $g$  をデュアルジョイングラフと定義する。

- $g$  はベイジアンネットワーク  $(G, \theta)$  を無向グラフ化したグラフである。
- ベイジアンネットワーク  $(G, \theta)$  の各ファミリー  $XU$  がデュアルジョイングラフのクリーク  $C_i$  の要素になる。
- セパレータ  $S_{ij} = U$ 、ただし、 $U \rightarrow X$ 。

例えば、図 5.4 (a) のベイジアンネットワークを考えよう。このベイジアンネットワークに対応するデュアルジョイングラフは、図 5.4 (b) になる。図のデュアルジョイングラフに似て、周辺事後分布の近似を計算すると、以下のようになる。

図 5.4 ベイジアンネットワークとデュアルジョイングラフ

$$p'(X|e) = \frac{p'(A|e)p'(B|e)p'(ABC|e)p'(ABD|e)p'(CDE|e)}{p'(A|e)p'(B|e)p'(C|e)p'(D|e)}$$

この近似は、ルーベリーブーフプロバゲーションで用いた近似、つまり積結合木による近似、と等価のものになり、width も 3 に抑えられている。一方、厳密であるジョイントリーによる周辺事後分布は以下のとおりである。

$$p(X|e) = \frac{p(ABC|e)p(ABD|e)p(ABCD|e)p(CDE|e)}{p(ABC|e)p(ABD|e)p(CD|e)}$$

厳密解では、width は 4 となり計算量は増加することがわかる。最後に、制約を緩和した近似であるジョイングラフ構造による周辺事後分布は以下のとおりである。

$$p'(X|e) = \frac{p'(ABC|e)p'(ABD|e)p'(ACD|e)p'(CDE|e)}{p'(B|e)p'(AC|e)p'(AD|e)p'(CD|e)}$$

積結合木によるポリツリー近似よりよい近似であるにもかかわらず、width は 3 に抑えられていることがわかるであろう。このように、ルーベリーブーフプロバゲーションにおけるポリツリー近似をジョイングラフに置き換えることにより、よりよい解が得られることが期待される。

結果として、ジョイングラフ近似を用いたルーベリーブーフプロバゲーションの確率更新式は以下で与えられる。

図 5.4 (a) ジョイントリー (b) ジョイングラフ

図 5.4 図 5.4 のネットワークに対応したジョイントリーとジョイングラフ

$$BEL(C_i) = \eta \phi_i \prod_k M_{ki} \tag{5.12}$$

$$M_{ij} = \eta \sum_{c \in \mathcal{C}_i} \phi_i \prod_{k \in \mathcal{C}_j} M_{ki} \tag{5.13}$$

ここで、 $BEL(C_i) = p(C_i|e)$ 、 $\phi_i$  はクラスター  $i$  中の対応するすべての CPT の値の積、 $M_{ij}$  はクラスター  $i$  からクラスター  $j$  に送信されるメッセージ、 $\eta$  は正規化定数を示している。また、セパレータの周辺事後分布の近似  $p'(S_{ij}|e)$  も以下のように計算できる。

$$BEL(S_{ij}) = \eta_j M_{ij} \tag{5.14}$$

結果として、図 5.4 に対応したジョイントリーとジョイングラフは図 5.5 のようになる。

アルゴリズム 20 (ジョイングラフルーベリーブーフプロバゲーション:  $JLBP(G, \Phi_i)$ )

- Input: ジョイングラフ  $G$ 、クラスター  $i$  中の対応するすべての CPT の値の積  $\phi_i$
- Output: approximate marginals  $BEL = p(C_i)$

main

- $t \leftarrow 0$
- すべてのメッセージ  $M_{ij}^t$  を一様分布に初期化
- while メッセージが収束していない do
- $t \leftarrow t + 1$
- for ジョイングラフの各アーク  $i \rightarrow j$  do
- $M_{ij}^t \leftarrow \eta \sum_{c \in \mathcal{C}_i} \phi_i \prod_{k \in \mathcal{C}_j} M_{ki}^{t-1}$
- $M_{ji}^t \leftarrow \eta \sum_{c \in \mathcal{C}_j} \phi_j \prod_{k \in \mathcal{C}_i} M_{ki}^{t-1}$
- end for
- end while
- return  $BEL(C_i) = \eta \phi_i \prod_k M_{ki}^t$  を出力
- end procedure

定理 34 ジョイングラフルーベリーブーフプロバゲーションの周辺事後分布の収束値  $\mu_{c_i} = BEL(c_i)$  と  $\mu_{s_{ij}} = BEL(s_{ij})$  のとき、以下のエントロピーを最大化する。

$$ENT'(X|e) + \sum_{c_i} E_{p'}(\log \phi_i) = - \sum_{c_i} \sum_{c_i} \mu_{c_i} \log \mu_{c_i} + \sum_{S_{ij}} \sum_{s_{ij}} \mu_{s_{ij}} \log \mu_{s_{ij}} + \sum_{c_i} \sum_{c_i} \mu_{c_i} \log \phi_i \tag{5.15}$$

ただし、

$$\sum_{c_i} \mu_{c_i} = \sum_{s_{ij}} \mu_{s_{ij}} = 1.$$

ただし、ジョイングラフルーベリーブーフプロバゲーションは、ポリツリー近似によるルーベリーブーフプロバゲーションよりも近似精度の改善はできるが、それでもまだ、大局的な一貫性は保証されないことに留意してほしい。

### 5.5 エッジ削除アルゴリズム

ベイジアンネットワーク研究のトップカンファレンスである Conference on Uncertainty in Artificial Intelligence (UAI) の 2010 年の開催で、与えられたデータを用いて隠れたデータを推論するという課題で、推論アルゴリズムのコンテストが行われたが、ここで見事に優勝したのは Choi, A. and Darwiche, A. である。そのときの彼らのコメントでは、「いろいろな経験から大規模なネットワークでの推論では、ルーピービリーフプロパゲーションを繰り返し、収束しない箇所を削除してまた繰り返すのが最も効率が良い」と口頭でコメントしている。これを精緻に実行するアルゴリズムが、エッジ削除アルゴリズム (edge deletion algorithm) である (Choi and Darwiche 2010)。

性がある。そこで、図 (a) のように与えられたネットワーク (ループ構造をもつネットワーク) に等価な構造を導出するために補助変数  $U$  と  $S$  を追加し、図 (b) のようにエッジを削除した構造 (緩和したネットワーク) を考える。このとき、 $U$  は  $U$  と同じ周辺分布を示すパラメータ  $\phi(U)$  をもつノード、 $S$  は  $X$  が  $U$  への隠れ変数 (  $(X|U)$  ) に対応するパラメータ  $\phi(S|U)$  をもち、ソフトエビデンス (確率値をエビデンスとして扱う) としても解釈できる。エッジ削除アルゴリズムでは、エッジを削除して補助の子ノード  $S$  を追加し、元の構造による各ノードの周辺分布を近似しようというアイデアである。

$$\phi(U) = p(U|e^i - S), \quad (5.16)$$

$$\phi(S|U) = \eta^i \phi^i(S|U). \quad (5.17)$$

ここで、式 (5.17) の  $\eta^i$  は正規化係数である。また、式 (5.16) の  $S$  は削除した各エッジのエビデンスの情報を含み、 $\phi(U)$  を求めるために削除したエッジによりエビデンスの情報が変わることを意味する。

式 (5.16) と式 (5.17) を満たすパラメータを計算するために以下の手続きを必要とする。まず、パラメータ  $\phi(U)$  および  $\phi(S|U)$  の初期値を一様分布の緩和したネットワーク  $G_0^i$  であるとする。  $\eta^i$  をこのネットワークの確率分布とする。各反復  $i > 0$  で、ネットワーク  $G_0^i$  のエッジのパラメータは、緩和したネットワーク  $G_{i-1}^i$  以下の式で厳密推論を行うことにより決定していく。

$$\phi(U) = p_{i-1}(U|e^i - S), \quad (5.18)$$

$$\phi(S|U) = \eta_{i-1}^i \phi^i(S|U). \quad (5.19)$$

ネットワーク  $G_i$  のパラメータがネットワーク  $G_{i-1}$  のパラメータと一致した場合にエッジのパラメータは収束したとみなし、これを緩和したネットワークのパラメータと見做す。収束するまで、これらの計算を繰り返し行う。

アルゴリズム 21 (削除エッジのパラメータ推定アルゴリズム: ED-BP( $G, \alpha, \Sigma$ ))

- Input: ベイジアンネットワーク  $\{G, \Theta\}$
- $e$ : エビデンス
- $\Sigma$ : 削除エッジ集合
- Output: approximate marginals  $BEL_t(XU) = p_t^i(X|U|e^i)$

main

1.  $t \leftarrow 0$
2.  $G_0^i \leftarrow G$  から  $\Sigma$  の要素を削除したグラフ
3.  $e^i \leftarrow e + G^i$  で追加した各ノード  $S$  の証拠  $\pm$
4. すべての補助変数パラメータを一様分布  $\phi(U)^0, \phi(S|U)^0$  に初期化
5. while 補助変数パラメータが収束していない do
6.      $t \leftarrow t + 1$
7.     for 各削除エッジ  $U \rightarrow X$  do
8.          $\phi(U)^t \leftarrow p_{t-1}^i(U|e^i - S)$
9.          $\phi(S|U)^t \leftarrow \eta_{t-1}^i \phi^i(S|U)$ , ここで  $\eta$  は正規化係数
10.         end for
11.     end while
12.     return  $BEL_t(XU) = p_t^i(X|U|e^i)$  を出力
13.     end procedure

#### 5.5.1 緩和

いま、ベイジアンネットワークを正確な推論に従うようにするため、エッジの削除によるネットワークの緩和 (relax) を考える。図 5.6 (a) は、変数  $U$  と  $X$  が等値  $U = X$  である制約をもつネットワークである。この等価制約を緩和すると、図 (b) のようにエッジを削除することができる。

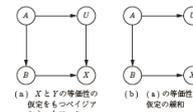


図 5.6 2 変数の等価制約をもつベイジアンネットワークの緩和

より一般的な状況 (図 5.7 (a)) では、このような等価制約はないので、エッジを削除することにより子ノード  $X$  は親ノード  $U$  からの直接の依存を失い、親ノード  $U$  は、子ノード  $X$  によって受け取られたエビデンスの情報を失う可能性がある。

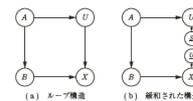


図 5.7 一般的なループ構造の緩和

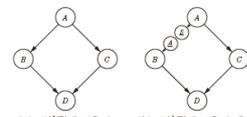
さらに、ネットワーク  $G$  および各反復  $i$  で各変数の同時周辺分布は以下のようになる。

$$BEL_t(X) = p_t^i(X|e^i), \quad (5.20)$$

$$BEL_t(XU) = p_t^i(X|U|e^i). \quad (5.21)$$

また、ループ構造で最大のボツリ構造になるように任意のノード間のエッジを削除した場合、緩和された構造での事後確率は元の構造でのルーピービリーフプロパゲーションの収束値に等しいことが証明されている (Choi and Darwiche 2006)。以上のパラメータ推定アルゴリズムをアルゴリズム 21 に示した。

例 47 図 5.8 は、エッジ  $A \rightarrow B$  を削除し、アルゴリズム 21 によってエッジのパラメータを推定した結果 (表 5.2) のネットワークを示している。ここで、



(a) エビデンス  $e: D=1$  (b) エビデンス  $e: D=1, S=1$

A	$\phi(A)$	A B	$\phi(B A) = \phi^i(B A)$	A C	$\phi(C A)$
真	0.8	真	0.8	真	0.5
偽	0.2	真	0.2	真	0.5
		偽	0.4	偽	1.0
		偽	0.6	偽	0.0

B	C	D	$\phi(D BC)$	A S	$\phi(S A)$
真	真	真	0.1	真	0.333
真	真	偽	0.0	真	0.250
真	偽	真	0.3	真	0.667
真	偽	偽	0.7	偽	0.333
偽	真	真	0.0	真	0.0
偽	真	偽	0.1	真	0.0
偽	偽	真	0.8	偽	0.0
偽	偽	偽	0.2	偽	0.0

変数  $D$  が真のとき、 $D=1$ 、変数  $D$  が偽のとき、 $D=0$  と書く。他の変数も同様とする。今回の場合は、4 回の繰返し後に収束している。

表 5.2 図 5.8 (b) で推定されたエッジパラメータ

		t=0	t=1	t=2	t=3	t=4
$\varphi(\underline{g}=1 A)$	A=1	0.5000	0.3496	0.3440	0.3438	0.3438
	A=0	0.5000	0.6504	0.6560	0.6562	0.6562
$\varphi(\underline{A})$	A=1	0.5000	0.8142	0.8257	0.8262	0.8262
	A=0	0.5000	0.1858	0.1743	0.1738	0.1738

一方、図 5.9 1 番左のベイジアンネットワークを考えよう。すべてのエッジについて緩和した構造が左から 2 番目の構造であり、いくつかのエッジを再生させた構造が 3 番目、4 番目の構造である。左から 4 番目の構造は、ループを緩和した最大のボリフリー構造であり、最もよい近似になると考えられる。

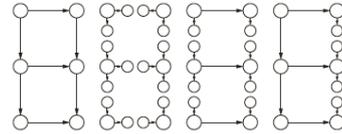


図 5.9 ベイジアンネットワークの緩和

エッジ削除アルゴリズムでは、ルービ-ビリーフプロパゲーションによってうまく表現できないノード間のエッジを回復させ、エッジ削除してもうまく収束するノード間のみの構造を意図的に探索するという手法であり、現在、最もよいといわれる手法である。

5.5.2 再生

エッジ削除アルゴリズムでは、どのエッジを削除するのが最も重要である。各エッジ間の条件付きパラメータを計算し、その結果から削除してもよいエッジを探せばよい。削除した後の確率パラメータの相互情報量を計算し、その値が高い順にエッジの重要度を定めることができる。相互情報量は、2変数が独立であれば最小値 0 となり、従属性が高ければ高いほど大きな値をとるので、削除されたエッジの 2 変数間の相互情報量が高い場合、それらは独立ではないのでエッジが必要であると考えられ、エッジを再生 (recover) しなければならない。

例えば、図 5.7 の例では、 $U$  と  $V$  の相互情報量  $MI(U, V)$  を計算すればよい。

$$MI(U, V) = \sum_{\underline{u}} \sum_{\underline{v}} p(\underline{U}, \underline{V} | e) \log \frac{p(\underline{U}, \underline{V} | e)}{p(\underline{U} | e)p(\underline{V} | e)} \quad (5.22)$$

結果として、相互情報量の高いエッジは削除せずに、再生させて構造を作成し、その構造上で補助変数のパラメータを計算し、事後確率を求めればよい。

エッジ削除アルゴリズムをアルゴリズム 22 に示す。ただし、このアルゴリズムには多数のバージョンが存在している。アルゴリズム 22 のループができた場合にジョイントリーアルゴリズムを適用するアルゴリズムが最もよい近似を得ることが知られている (Choi and Darwiche 2010)。この意味でも、大規模なジョイントリーの構成アルゴリズムはより重要である。

アルゴリズム 22 (エッジ削除アルゴリズム:  $ED(G, e)$ )

- Input: ベイジアンネットワーク  $\{G, \Theta\}$
- e: エビデンス
- Output: approximate marginals  $BEL(XU) = p(X|U|e)$

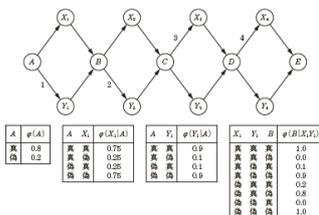
main

1.  $\Sigma \leftarrow$  最大ボリフリー構造にするために削除したエッジの集合
2.  $G' \leftarrow G$  から  $\Sigma$  の要素を削除したグラフ
3. アルゴリズム 21 (ED-BP) を用いて各変数  $X$  の事後確率を計算
4. while  $G'$  で正確な推論が可能である限りエッジを再生 do
5. 各削除エッジ  $U \rightarrow X$  で相互情報量  $MI$  を計算
6.  $\Sigma \leftarrow$  相互情報量トップ  $K$  個を除くエッジ集合
7.  $G' \leftarrow G'$  から  $\Sigma$  の要素を削除したグラフ
8. end while
9. return  $BEL(XU) = p(X|U|e)$  を出力
10. end procedure

いま、図 5.10 のネットワークから  $A \rightarrow Y_1(1)$ ,  $B \rightarrow Y_2(2)$ ,  $C \rightarrow X_3(3)$ ,  $D \rightarrow X_4(4)$  を削除し、図 5.11 の緩和したネットワークの再生を考える。

アルゴリズム 22 によって図 5.11 のネットワークでエッジを一つずつ再生した場合の事後確率を表 5.3 に示す。表 5.4 に示すように、エッジの再生は最も大きい相互情報量をもつエッジの順に行う。

このとき、 $B \rightarrow Y_2(2)$  のエッジを再生しても各変数の事後確率が改善されない。



A	$p(A)$	A, X <sub>1</sub>	$p(X_1 A)$	A, Y <sub>1</sub>	$p(Y_1 A)$	X <sub>1</sub> , Y <sub>1</sub> , B	$p(B X_1, Y_1)$
真	0.8	真	0.75	真	0.9	真	1.0
偽	0.2	真	0.25	偽	0.1	真	0.0
		偽	0.25	真	0.1	偽	0.1
		偽	0.75	偽	0.9	偽	0.0

図 5.10 変数  $X_1, \dots, X_4$ , 変数  $Y_1, \dots, Y_4$  および変数  $B, C, D$  が 2 階的 CPD をもつネットワーク (エビデンスは  $X_3=0, B=1$  とする)

ここに注意してほしい。実際、このエッジを最初に再生した場合に事後確率が改善されないことを表 5.5 に示す。表 5.5 は、最も小さい相互情報量をもつエッジの順に再生した場合の事後確率である。表 5.6 は再生過程での相互情報量を示した。

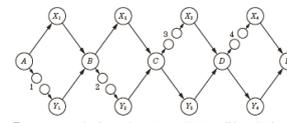


図 5.11 図 5.10 のネットワーク内の四つのエッジを削除し、緩和したネットワーク

表 5.3 エッジを再生することによる各ノードの事後確率 (右端の列は正しい事後確率)

事後確率	{1, 2, 3, 4}	{2, 3, 4}	{2, 3}	{2}	{}
$p(A=1 e)$	0.7114	0.7290	0.7300	0.7331	0.7331
$p(B=1 e)$	0.3565	0.4236	0.4258	0.4429	0.4429
$p(C=1 e)$	0.2405	0.2733	0.2775	0.2910	0.2910
$p(D=1 e)$	0.5824	0.6008	0.6100	0.5917	0.5917

表 5.4 削除されたエッジの相互情報量 (相互情報量の大きい順にエッジを再生)

相互情報量	{1, 2, 3, 4}	{2, 3, 4}	{2, 3}	{2}	{}
$A \rightarrow Y_1(1)$	$1.13 \times 10^{-3}$				
$B \rightarrow Y_2(2)$	0	0	0	0	0
$C \rightarrow X_3(3)$	$6.12 \times 10^{-4}$	$7.36 \times 10^{-4}$	$6.98 \times 10^{-4}$		
$D \rightarrow X_4(4)$	$1.12 \times 10^{-3}$	$1.08 \times 10^{-3}$			

表 5.5 エッジを再生することによる各ノードの近似事後確率 (右端の列は正しい事後確率)

事後確率	削除エッジ				
	{1,2,3,4}	{1,3,4}	{1,4}	{4}	{}
$p^i(A=1 e^i)$	0.7114	0.7114	0.7160	0.7324	0.7331
$p^i(B=1 e^i)$	0.3585	0.3585	0.3681	0.4412	0.4429
$p^i(C=1 e^i)$	0.2405	0.2405	0.2564	0.2879	0.2910
$p^i(D=1 e^i)$	0.5824	0.5824	0.5689	0.5848	0.5917

表 5.6 削除されたエッジの相互情報量 (相互情報量の小さい順にエッジを再生)

相互情報量	削除エッジ				
	{1,2,3,4}	{1,3,4}	{1,4}	{4}	{}
$A \rightarrow Y_1(1)$	$1.13 \times 10^{-3}$	$1.13 \times 10^{-3}$	$9.89 \times 10^{-4}$		
$B \rightarrow Y_2(2)$	0				
$C \rightarrow X_3(3)$	$6.12 \times 10^{-4}$	$6.12 \times 10^{-4}$			
$D \rightarrow X_4(4)$	$1.12 \times 10^{-3}$	$1.12 \times 10^{-3}$	$1.14 \times 10^{-3}$	$1.11 \times 10^{-3}$	

一般に、ジョイントツリーアルゴリズムを用いて、削除できるエッジは削除して  $width$  の小さいクリークを作成していくことが重要である。しかし、最大クリークに対応するループのエッジが削除できるかどうかはわからないので、やはり、4章で述べたジョイントツリーアルゴリズムの計算量削減も重要課題である。