

否定枝を含む shared BDD 上で動作する EM アルゴリズム

The EM algorithm that works on shared BDDs with negative edges

石島正和^{1*} 亀谷由隆¹ 佐藤泰介¹ 湊真一²
Ishihata Masakazu¹ Yoshitaka Kameya¹ Taisuke Sato¹ Shin-ichi Minato²

¹ 東京工業大学大学院情報理工学研究科

¹ Graduate School of Information Science and Engineering, Tokyo Institute of Technology

² 北海道大学大学院情報科学研究科

² Graduate School of Information Science and Technology, Hokkaido University

Abstract: We extend the BDD-EM algorithm, which is an expectation-maximization (EM) algorithm working on binary decision diagrams (BDDs), for shared BDDs (SBDDs) with negative edges. BDDs are used as a compact expression of Boolean formulas and moreover the use of SBDDs and negative edges is expected to reduce the time and space in the case where we have some similar partial structures. We show that the proposed algorithm which utilizes SBDD with negative edges for bipartite noisy-OR network reduces time and space in executing the EM algorithm.

1 はじめに

Binary decision diagram (BDD) [1, 2] は論理関数をコンパクトに表現する有向非循環グラフである。BDD は多方面で使われているが、BDD の新たな応用として、BDD によって表現された確率モデル上で動作する expectation-maximization (EM) アルゴリズム [4] である BDD-EM アルゴリズムが提案されている [12]。EM アルゴリズムは不完全データに対して最尤推定を行うアルゴリズムであり、BDD-EM アルゴリズムは観測命題 F が確率的命題変数 X_1, \dots, X_k の BDD で表現されるとき、 F の観測値から隠れ変数である X_1, \dots, X_k が真偽をとる確率を BDD のサイズに比例する計算量で学習する。BDD-EM アルゴリズムが扱う BDD は ROBDD [1, 2] に限られおり、複数の ROBDD を 1 つの ROBDD に統合して表現する Shared BDD (SBDD) [8] や互いに否定関係にある部分グラフを共有する否定枝 [1] を含む BDD は扱えない。SBDD や否定枝は複数の BDD を同時に構築する際の計算量とそれを表現するメモリを削減できることから、これらの利用により BDD-EM アルゴリズムの計算量と使用するメモリを削減する効果が期待できる。そこで本論文は BDD-EM アルゴリズムを否定枝を含む SBDD に対しても実行できるように拡張する。

本論文では、2 章で我々が扱う問題とその問題に対する EM アルゴリズムの定式化を行い、3 章で BDD,

SBDD, 否定枝について説明した後、4 章で BDD-EM アルゴリズムを紹介する。そして 5 章で BDD-EM アルゴリズムを否定枝を含む SBDD に拡張する提案アルゴリズムを示し、6 章で二部 noisy-OR ネットワークにおいて提案手法を用いることで確かに BDD-EM アルゴリズムの計算量が削減できることを確認する。最後に関連研究を 7 章、まとめを 8 章で述べる。

2 準備

2.1 問題設定

ここでは本論文が扱う問題を定義する。我々は N 個の観測可能な事象とそれらの背後にある k 個の観測不可能な事象を定義し、各時刻 t において観測可能な N 個の事象のうちいずれかについてそれが起こったかどうかを観測する¹。 N 個の観測可能な事象は N 個の確率的命題変数 F_1, \dots, F_N により表現され、 $F_i=1$ ($F_i=0$) は F_i が表す事象が起こったこと (起こらなかったこと) を表す。集合 $\{F_1, \dots, F_N\}$ を F により表し、これらを観測命題と呼ぶ。 $F^{(t)} (\in F)$ を時刻 t に観測した事象を表す観測命題、その観測値を $f^{(t)} (\in \{0, 1\})$ とすれば、時刻 1 から T までの観測列 \mathcal{O} は $\{(F^{(1)}, f^{(1)}), \dots, (F^{(T)}, f^{(T)})\}$ と表現される。これに対して k 個の観測不可能な事象を k 個の確率的命題変数 X_1, \dots, X_k により表し、これらを基本命題と呼ぶ。基本命題は互いに独立に確率的

*連絡先：東京工業大学大学院情報理工学研究科計算工学専攻
〒152-8552 東京都目黒区大岡山 2-12-1
E-mail: ishihata@mi.cs.titech.ac.jp

¹同時に 2 つ以上の事象を観測できる場合、それらをまとめて 1 つの事象とすることでこの条件に当てはめることが可能である。

に0,1の値をとり、基本命題の集合 V_F に対する値の割り当て $X_1 = x_1, \dots, X_k = x_k$ ($x_1, \dots, x_k \in \{0,1\}$) を $\phi = \{(X_1, x_1), \dots, (X_k, x_k)\}$ と表現する. 本論文ではすべての観測命題 $F_i \in F$ は基本命題 V_F の論理関数により表現できるとする. 従って, V_F に対する値の割り当て ϕ が決まれば, すべての $F_i \in F$ の値も一意に決まる. これより, $X_i \in V_F$ が真偽をとる確率 $P(X_i = x)$ ($x \in \{0,1\}$) が分かれば, F_i が真偽をとる確率も計算できる. 本論文では, 時刻1から T までの観測列 \mathcal{O} を得たとき, 観測列 \mathcal{O} を得る確率を最大化するように各基本命題 $X_i \in V_F$ が真偽をとる確率を学習することを目的とする.

例として次のような遅刻に関する問題を考える. 大学生 A, B はときどき学校に遅刻し, A, B の共通の友人 C は彼らが遅刻するかどうかを毎朝観測する. ただし, 彼らは毎朝同じ授業を取っているとは限らないので, C は A, B 両方の遅刻の有無を観測できる日もあれば, どちらか一方しか観測できない日もある. 例えば C の五日間の観測が「 A は遅刻した」「 B は遅刻した」「 A は遅刻しなかった」「 A, B ともに遅刻した」「 B は遅刻しなかった」であったとする. すると観測命題 L_A, L_B, L_{AB} をそれぞれ「 A は遅刻した」「 B は遅刻した」「 A, B ともに遅刻した」と定義すれば, 観測列 \mathcal{O} は $\{(L_A, 1), (L_B, 1), (L_A, 0), (L_{AB}, 1), (L_B, 0)\}$ と表現できる. これに対して, A, B の遅刻の原因となる観測できない事象がいくつか存在する. A は毎朝バスで通学し, B は晴れの日には自転車, 雨の日にはバスで通学するでしょう. 道路が渋滞するとバスが遅れ, バスに乗っている者は遅刻する. また, A, B はともに自分の時計が壊れていると遅刻する. 基本命題 H, R, C_A, C_B をそれぞれ「道路が渋滞する」「雨が降る」「 A の時計が壊れている」「 B の時計が壊れている」と定義すれば, L_A, L_B, L_{AB} は以下のように基本命題の論理関数で表現される.

$$\begin{aligned} L_A &\Leftrightarrow C_A \vee H \\ L_B &\Leftrightarrow C_B \vee (R \wedge H) \\ L_{AB} &\Leftrightarrow (C_A \vee H) \wedge (C_B \vee (R \wedge H)) \end{aligned}$$

この例における問題とは, 5日間の観測列 \mathcal{O} と観測命題と基本命題の関係を表す上記の論理関数から基本命題 H, R, C_A, C_B が真偽を取る確率を学習することである.

2.2 EM アルゴリズム

2.1 節で定義した問題では, 各時刻のに観測した観測命題とその値の対の列 $\mathcal{O} = \{(F^{(1)}, f^{(1)}), \dots, (F^{(T)}, f^{(T)})\}$ が与えられる. しかし, そのときの基本命題 V_F に対する値の割り当て ϕ を知ることはできない. このような不完全な観測データから最尤法を用いて確率値を学習するアルゴリズムとして, EM アルゴリズム [4] が知ら

れている. EM アルゴリズムは観測列 \mathcal{O} に対する対数尤度を最大化するような各基本命題 $X_i \in V_F$ が真偽をとる確率 $P(X_i = x)$ ($x \in \{0,1\}$) を **Expectation step** (E-step) と **Maximization step** (M-step) を繰り返すことで推定する. ここでは 2.1 節で定義した問題に対して定式化された EM アルゴリズムを示す.

まず, EM アルゴリズムの最大化の対象である観測列 \mathcal{O} に対する対数尤度の計算式を導出する. 観測命題は基本命題の論理関数で表現されるため, 基本命題 $X_i \in V_F$ が真偽をとる確率 $P(X_i = x)$ ($x \in \{0,1\}$) から観測列 \mathcal{O} を得る確率を計算できる. $X_i \in V_F$ は互いに独立であるが, 中には同一の確率分布に従うものも存在する. $X_i, X_j \in V_F$ ($i \neq j$) が同一の確率分布に従うとき, X_i と X_j は i.i.d (独立同分布) であるという. V_F を互いに i.i.d. である基本命題の部分集合 π ($\subseteq V_F$) の集まりに分割したとき, 個々の部分集合 π を **i.i.d. 集合** と呼び, すべての i.i.d. 集合の族を Π_F と表現する. $X \in V_F$ が属する i.i.d. 集合を π_X で表すと, X の確率分布はパラメータ $\theta_{(\pi_X, x)}$ ($x \in \{0,1\}, 0 \leq \theta_{(\pi_X, x)} \leq 1$) を用いて以下で表現される.

$$P(X = x) = \theta_{(\pi_X, x)} \quad (\theta_{(\pi_X, 0)} + \theta_{(\pi_X, 1)} = 1)$$

すべてのパラメータの集合を $\theta = \{\theta_{(\pi, x)} \mid \pi \in \Pi_F, x \in \{0,1\}\}$ と書くと, θ が与えられたときに V_F が割り当て ϕ が指定した値をとる確率 $P_\theta(\phi)$ は以下のように計算できる.

$$P_\theta(\phi) = \prod_{(X, x) \in \phi} \theta_{(\pi_X, x)}$$

Φ を V_F に対するすべての値の割り当ての集合, $\Phi(F_i, f_i)$ ($F_i \in F, f_i \in \{0,1\}$) を $\phi \in \Phi$ のうち $F_i = f_i$ となる割り当ての集合とすれば, 観測列 \mathcal{O} を得る確率 $P_\theta(\mathcal{O})$ は以下のように計算される.

$$\begin{aligned} P_\theta(\mathcal{O}) &= \prod_{(F^{(t)}, f^{(t)}) \in \mathcal{O}} P_\theta(F^{(t)} = f^{(t)}) \\ &= \prod_{(F^{(t)}, f^{(t)}) \in \mathcal{O}} \sum_{\phi \in \Phi(F^{(t)}, f^{(t)})} P_\theta(\phi) \\ &= \prod_{(F^{(t)}, f^{(t)}) \in \mathcal{O}} \sum_{\phi \in \Phi(F^{(t)}, f^{(t)})} \prod_{(X, x) \in \phi} \theta_{(\pi_X, x)} \end{aligned}$$

観測対 (F_i, f_i) が観測列 \mathcal{O} に表れる回数を $\delta_{(F_i, f_i)}$ で表し, $\delta_{(F_i, f_i)} \geq 1$ である観測対の集合を \mathcal{O} と表現する. すると現在のパラメータ θ における観測列 \mathcal{O} に対する対数尤度 $\log L_{\mathcal{O}}(\theta)$ は次式で与えられる.

$$\begin{aligned} \log L_{\mathcal{O}}(\theta) &= \log P_\theta(\mathcal{O}) \\ &= \sum_{(F_i, f_i) \in \mathcal{O}} \delta_{(F_i, f_i)} \log \left[\sum_{\phi \in \Phi(F_i, f_i)} \prod_{(X, x) \in \phi} \theta_{(\pi_X, x)} \right] \quad (1) \end{aligned}$$

最後に観測列 \mathcal{O} に対する対数尤度 $\log L_{\mathcal{O}}(\theta)$ を極大化するパラメータ $\hat{\theta}$ は以下で定式化される E-step と M-step を交互に繰り返すことで推定される。

E-step: 条件付き期待値 $\eta_{\theta}^x[\pi]$ ($\pi \in \Pi_F$, $x \in \{0, 1\}$) を計算する。

$$\eta_{\theta}^x[\pi] = \sum_{(F_i, f_i) \in \mathcal{O}} \delta_{(F_i, f_i)} \sum_{\phi \in \Phi_F(F_i, f_i)} \sigma_{\pi, x}(\phi) P_{\theta}(\phi | F_i = f_i) \quad (2)$$

ここで $\sigma_{\pi, x}(\phi) = |\{X \in \pi | (X, x) \in \phi\}|$ である。

M-step: パラメータ $\theta_{(\pi, x)}$ ($\pi \in \Pi_F$, $x \in \{0, 1\}$) を

$$\hat{\theta}_{(\pi, x)} := \eta_{\hat{\theta}}^x[\pi] / (\eta_{\hat{\theta}}^1[\pi] + \eta_{\hat{\theta}}^0[\pi]) \quad (3)$$

に更新する。

式 (2) の条件付き期待値 $\eta_{\theta}^x[\pi]$ は「観測列 \mathcal{O} を得たとき、 $X \in \pi$ が値 x で表れる回数の期待値」を意味する。

2.1 節の遅刻の例題に対して EM アルゴリズムを実行しよう。まず基本命題 $V_F = \{H, R, C_A, C_B\}$ を i.i.d. 集合に分割する。 C_A と C_B は A, B の時計が壊れていることを表しているが、 A と B は同じ型の時計を使っていると仮定すれば、 C_A と C_B は i.i.d. であると考えることができる。すると V_F は $\pi_H = \{H\}$, $\pi_R = \{R\}$, $\pi_C = \{C_A, C_B\}$ の 3 つの i.i.d. 集合に分割でき、パラメータは $\theta_{(\pi_H, x)}$, $\theta_{(\pi_R, x)}$, $\theta_{(\pi_C, x)}$ ($x \in \{0, 1\}$) の 6 つとなる。これよりこの例題に対する E-step は 6 つの条件付き期待値 $\eta_{\hat{\theta}}^x[\pi_H]$, $\eta_{\hat{\theta}}^x[\pi_R]$, $\eta_{\hat{\theta}}^x[\pi_C]$ ($x \in \{0, 1\}$) を計算し、M-step は条件付き期待値を用いて $\theta_{(\pi_H, x)}$, $\theta_{(\pi_R, x)}$, $\theta_{(\pi_C, x)}$ を更新する。この 2 ステップをパラメータが十分に収束するまで交互に繰り返すことで、観測列に対する尤度を極大化するパラメータを得ることができる。

しかし、ここで定式化した EM アルゴリズムを実行するには困難がある。対数尤度 (式 (1)) や条件付き期待値 (式 (2)) の計算において、 $\Phi(F_i, f_i)$ ($F_i \in F$, $f_i \in \{0, 1\}$) は F_i の真理値表の $F_i = f_i$ となる行の集合に対応する。このため式 (1)(2) の計算は真理値表に沿って行われ、その計算量は基本命題の数 k に対して指数オーダーとなる。この困難を解決するため、論理関数を真理値表よりコンパクトに表現する手法として知られている BDD を用いて確率、条件付き期待値計算を行う BDD-EM アルゴリズムが提案されている [12]。3 章では BDD について説明し、4 章で BDD-EM アルゴリズムについて説明する。

3 Binary decision diagram

Binary decision diagram (BDD) [1, 2] は論理関数 F をコンパクトに表現する有向非循環グラフである。

命題変数 X_1, \dots, X_k の論理関数 F を表現する BDD は X_1, \dots, X_k のいずれかでラベル付けされた変数節点と 1 又は 0 でラベル付けされた定数節点によって構成される。節点 n の変数ラベルを $Var(n)$ で表わす。すべての変数節点 n は 2 つの子節点 1-child と 0-child を持ち、それぞれを $Ch_1(n)$, $Ch_0(n)$ で表す。 n から $Ch_1(n)$ ($Ch_0(n)$) に至る辺のことを **1-枝 (0-枝)** と呼び $\langle n, 1 \rangle$ ($\langle n, 0 \rangle$) と書く。一方始点が存在しない辺を入力枝と呼び、入力枝が指す節点を根節点と呼ぶ。 F を表現する BDD の根節点から f 定数節点 ($f \in \{0, 1\}$) に至るすべてのパスの集合を \mathcal{P}_f^F で表し、パス $\rho \in \mathcal{P}_f^F$ は F が値 f を取る X_1, \dots, X_k に対する値の割り当てを表す。パス ρ は節点と枝が交互に表れる順序で表現され、 ρ が表現する値の割り当て ϕ_{ρ} は ρ に含まれる枝を用いて $\{(Var(n), x) | \langle n, x \rangle \in \rho\}$ と表現される。

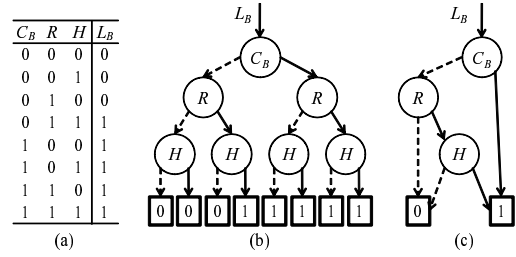


図 1: L_B を表す (a) 真理値表, (b)BDT, (c)ROBDD

図 1 の (a)(b) は遅刻の例の論理関数 $L_B \Leftrightarrow C_B \vee (R \wedge H)$ を表した真理値表と対応する BDD である。図 1(b) のように完全二分木である BDD のことを binary decision tree (BDT) と呼ぶ。BDD 中の実線は 1-枝を、破線は 0-枝を表す。図 1(b) 上のパス

$$\rho = \{C_B, \langle C_B, 0 \rangle, R, \langle R, 0 \rangle, B, \langle H, 0 \rangle, \boxed{0}\}$$

は値の割り当てが $\{(C_B, 0), (R, 0), (H, 0)\}$ であるとき $L_B = 0$ であることを表し、これは L_B の真理値表の 1 行目に対応する。BDT 上の根節点から定数節点に至るパスは真理値表の各行と 1 対 1 に対応する。

同じ論理関数を表現する BDD は複数種類存在する。例えば図 1(c) も図 1(b) と同様に L_B を表現する BDD である。図 1(c) の BDD は BDT と比較して節点数が少ないが、これは BDT 中のいくつかの節点は簡略化規則によって削除または共有化されるからである。節点 n の両枝が同じ節点を指しているとき n は削除され、節点 n, n' において $Var(n) = Var(n')$, $Ch_x(n) = Ch_x(n')$ ($x \in \{0, 1\}$) であるとき n と n' は共有化される。それ以上簡略化できない BDD のことを reduced BDD と呼び、更にパスに表れる変数の順序 (変数順序) が一意であるとき、これを **reduced orderd BDD (ROBDD)** と呼ぶ。ROBDD は論理関数と変数順序に対して一意に決まる。図 1(c) は変数順序を C_B, R, H としたとき

の L_B を表す ROBDD である。図 1(c) 上のパス $\rho = \{C_B, \langle C_B, 1 \rangle, \boxed{1}\}$ は、 $C_B = 1$ のとき $L_B = 1$ であることを表すが、これは L_B の真理値表の下 4 行に対応する。このように ROBDD は真理値表を圧縮する効果がある。ROBDD のサイズは変数順序に依存するが、最適な変数順序を見つけることは NP-hard であることが知られているが [5], よい変数順序を発見するヒューリスティックも多く提案されている [8]。本論文では特に断りのない限り ROBDD を扱うことにし、以後 ROBDD を BDD と呼ぶ。

遅刻の例の観測命題 $F = \{L_A, L_B, L_{AB}\}$ を BDD で表現すると図 2(a) のようになる。このように論理関数の集合は BDD の集合で表現される。しかし、この 3 つの BDD はすべて $Var(n) = H, Ch_1(n) = \boxed{1}, Ch_0(n) = \boxed{0}$ なる節点 n を持っている。複数の論理関数を複数の BDD で表現する場合、複数の BDD に同型な部分グラフが表れることがある。このような冗長性を解消するため、複数の BDD を効率的に扱う技術として shared BDD と否定枝技法が提案されている。以下ではこの 2 つの技術について説明する。

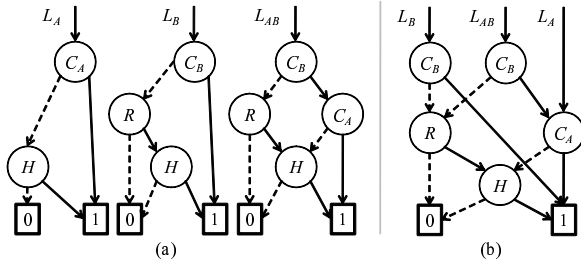


図 2: L_A, L_B, L_{AB} を表現する ROBDD(a) と SBDD(b)

3.1 Shared BDD

Shared BDD (SBDD)[8] は複数の BDD を統合して 1 つの BDD として表現する。例えば、図 2(a) の L_A, L_B, L_{AB} を表す 3 つの BDD は、図 2(b) のように 1 つ SBDD で表現できる。 N 個の論理関数 F_1, \dots, F_N を表現する SBDD は N 本の入力枝を持ち、 F_i に対応する入力枝が指す節点を「 F_i の根節点」と呼ぶ。 F_i の根節点から f 定数節点 ($f \in \{0, 1\}$) に至るすべてのパスの集合を $\mathcal{P}_f^{F_i}$ と表現し、パス $\rho \in \mathcal{P}_f^{F_i}$ は値の割り当てが ϕ_ρ であるとき $F_i = f$ であることを表現する。例えば図 2(b) の SBDD において、 L_A の根節点 C_A から 1 定数節点に至るパス $\rho = \{C_A, \langle C_A, 1 \rangle, \boxed{1}\}$ は $C_A = 1$ のとき $L_A = 1$ であることを表す。

複数の BDD を 1 つの SBDD で統合的に表現することは、部分グラフを共有化しメモリを削減するだけでなく、同じ BDD を再構築する時間を削減する効果も

ある。複数の論理関数を SBDD で表現するには、すべての論理関数で共通の変数順序を定義する必要がある。SBDD のサイズも BDD と同様、変数順序に依存するが SBDD に対して有効な変数順序を発見するヒューリスティックが提案されている [8]。複数の BDD を SBDD で表現することで、多くの場合は節点数を削減することが可能だが、変数順序や表現する論理関数によっては SBDD を利用することでかえって節点数が増加することもある。

3.2 否定枝

論理関数 F の 1 定数節点と 0 定数節点を入れ替えると、 $\neg F$ を表現する BDD となる。つまり、 F と $\neg F$ の BDD は定数節点を除いて同型である。互いに否定同士の関係にある BDD を共有化する手法として否定枝技法[1]が提案されている。否定を表す記号 \circ が付加された枝である否定枝はそれ以下の部分グラフにおいて論理を反転させる効果を持つ。例えばパス ρ が 1 定数節点に至るとき、 ρ が否定枝を奇数本含む場合、 ρ においてその 1 定数節点は論理的に 0 定数節点を意味する。ROBDD の一意性が失われないよう、否定枝の使用は 1-枝にのみ制限されており、1-枝が指す 1 定数節点はすべて否定枝で 0 定数節点に変換する。

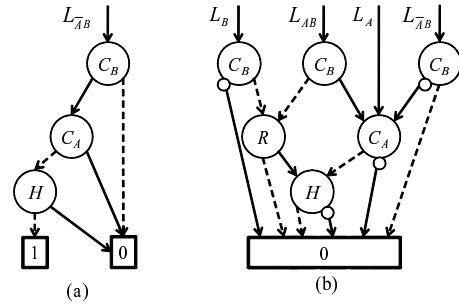


図 3: (a) $L_{\bar{A}B}$ の BDD, (b) $L_A, L_B, L_{AB}, L_{\bar{A}B}$ の SBDD

例えば遅刻の例題において先の 5 日間の観測に加えて「 A は遅刻したが B は遅刻しなかった」を観測したとする。この観測事象を表現する観測命題 $L_{\bar{A}B}$ は基本命題を用いて $L_{\bar{A}B} \Leftrightarrow \neg(C_A \vee H) \wedge (C_B \vee (R \wedge H))$ と表現できる。図 3(a) は $L_{\bar{A}B}$ を表現する BDD であるが、これと先の観測命題を表現した SBDD (図 2(b)) を比較すると共有可能な変数節点は存在しない。しかし、図 3(a) の枝 $\langle C_B, 1 \rangle$ 以下の部分グラフは $\neg L_A$ を表しており、これは否定枝を用いて図 2(b) の L_A を表す部分グラフと共有化できる。図 3(b) は観測命題 $F = \{L_A, L_B, L_{AB}, L_{\bar{A}B}\}$ を否定枝を含む SBDD で表現したものである。 F を 4 つの BDD で表現すると変数節点の総数が 12 個となるのに対して、否定枝を含む SBDD で表現すると変数節点は 6 つに削減される。

4 BDD-EM アルゴリズム

4.1 BDD-EM アルゴリズムの修正

2.2 節で定式化した EM アルゴリズムを実行するには観測列 \mathcal{O} に対する対数尤度 (式 (1)) 及び条件付き期待値 (式 (2)) を計算する必要がある。しかし、式 (1)(2) は観測命題 $F_i \in F$ を表現する真理値表に沿って計算を行うため、その計算量は基本命題の数 k に対して指数オーダーとなる。そこで真理値表を圧縮する効果のある BDD を用いてこの計算時間を短縮する **BDD-EM** アルゴリズムが提案されている [12]。BDD-EM アルゴリズムは各観測命題 $F_i \in F$ を BDD Δ_{F_i} で表現し、その上で動的計画法を用いて確率計算を行うことで式 (1)(2) で定義される対数尤度、条件付き期待値を高速に計算する。BDD-EM アルゴリズムの E-step の計算量は観測命題を表現する BDD の総節点数に比例する。

簡略化規則により BDD 中の節点を削除することは真理値表を圧縮する効果がある一方、条件付き期待値計算を困難にする。文献 [12] で記述される BDD-EM アルゴリズムは削除された節点を数値的に復元することで式 (2) の条件付き期待値を計算する。これに対して、BDD Δ_{F_i} ($F_i \in F$) の集合 Δ_F を用いて EM アルゴリズムを定式化しなおす。これは文献 [7] での確率変数の変換を参考にしており、この変換により条件付き期待値 $\eta_{\theta}^x[\pi]$ ($\pi \in \Pi_F$, $x \in \{0, 1\}$) の定義を式 (4) に書き換えられ、削除された節点を考慮することなく計算する。

$$\eta_{\theta}^x[\pi] = \sum_{(F_i, f_i) \in \mathcal{O}} \delta_{(F_i, f_i)} \sum_{\rho \in \mathcal{P}_{f_i}^{F_i}(\pi, x)} \sigma_{\pi, x}(\phi_{\rho}) P_{\theta}(\phi_{\rho} | F_i = f_i) \quad (4)$$

式 (4) の $\eta_{\theta}^x[\pi]$ は「観測列 \mathcal{O} を得たとき、 $Var(n) \in \pi$ なる節点 n の x -枝 $\langle n, x \rangle$ が BDD 上のパス中に表れる回数の期待値」を意味し、これを用いた EM アルゴリズムは対数尤度の単調増加が保証される (証明略)。

Δ_F を用いて定式化し直した BDD-EM アルゴリズムの E-step と M-step は Proc. 1 と Proc. 2 によって実行される。Proc. 1 中の BACKWARD(Δ_{F_i}) は Δ_{F_i} のすべての節点 n に対して後ろ向き確率 $\mathcal{B}_{\theta}^1[n]$, $\mathcal{B}_{\theta}^0[n]$ を、FORWARD(Δ_{F_i}) は前向き確率 $\mathcal{F}_{\theta}[n]$ を計算し、Proc. 2 はそれらの確率を用いて式 (4) で定義される条件付き期待値を計算する。そして MSTEP(Π_F) は計算した条件付き期待値を用いてすべてのパラメータ $\theta_{(\pi, x)}$ を式 (3) に従って更新する。後ろ向き・前向き確率と条件付き期待値の計算法については 4.2, 4.3 節で述べる。

4.2 後ろ向き確率、前向き確率の計算

節点 n の後ろ向き確率 $\mathcal{B}_{\theta}^x[n]$ ($x \in \{0, 1\}$) とは n から x 定数節点に至る確率である。定義より明らかに定

Procedure 1 ESTEP(Δ_F)

```

1: for all  $\Delta_{F_i} \in \Delta_F$ 
2:   BACKWARD( $\Delta_{F_i}$ );
3:   FORWARD( $\Delta_{F_i}$ );
4:   EXPECTATION( $\Delta_{F_i}$ );
5: end for

```

Procedure 2 MSTEP(Π_F)

```

1: for all  $\pi \in \Pi_F$ ,  $x \in \{0, 1\}$ 
2:    $\theta_{(\pi, x)} := \eta_{\theta}^x[\pi] / (\eta_{\theta}^1[\pi] + \eta_{\theta}^0[\pi])$ ;
3: end for

```

数節点 $\boxed{1}$, $\boxed{0}$ において以下が成り立つ。

$$\mathcal{B}_{\theta}^1[\boxed{1}] = 1, \quad \mathcal{B}_{\theta}^1[\boxed{0}] = 0, \quad \mathcal{B}_{\theta}^0[\boxed{1}] = 0, \quad \mathcal{B}_{\theta}^0[\boxed{0}] = 1$$

これに対して Δ_{F_i} のすべての変数節点の集合を N_{F_i} で表すと、 $n \in N_{F_i}$ の後ろ向き確率はその子節点の後ろ向き確率を用いて以下により再帰的に計算できる ($x \in \{0, 1\}$)。

$$\mathcal{B}_{\theta}^x[n] = \theta_{(\pi_n, 0)} \mathcal{B}_{\theta}^x[Ch_0(n)] + \theta_{(\pi_n, 1)} \mathcal{B}_{\theta}^x[Ch_1(n)]$$

ここで π_n は n にラベル付けられている変数 $Var(n)$ が属する i.i.d. 集合 $\pi_{Var(n)}$ の省略形である。 N_{F_i} を Δ_{F_i} の構造に従ってトポロジカル順序を与えれば、後ろ向き確率は動的計画法に基づいて順序の遅い節点から計算することにより、 $|N_{F_i}|$ に比例する時間で計算される。

一方節点 n の前向き確率 $\mathcal{F}_{\theta}[n]$ とは Δ_{F_i} の根節点から n に至る確率である。 N_{F_i} にトポロジカル順序を与えたとき、順序が i である節点を n_i で表せば、 n_1 は根節点に一致する。定義より明らかに $\mathcal{F}_{\theta}[n_1] = 1$ が成り立つ。これより変数節点 n の前向き確率は、その親節点集合 $Pa_x(n) = \{n' \in N_{F_i} \mid Ch_x(n') = n\}$ ($x \in \{0, 1\}$) を用いて以下により再帰的に計算される。

$$\mathcal{F}_{\theta}[n] = \sum_{n' \in Pa_0(n)} \theta_{(\pi_{n'}, 0)} \mathcal{F}_{\theta}[n'] + \sum_{n' \in Pa_1(n)} \theta_{(\pi_{n'}, 1)} \mathcal{F}_{\theta}[n']$$

$n' \in Pa_x(n)$ は n より早いトポロジカル順序を持つことから、前向き確率は順序の早い節点から順に計算することで $|N_{F_i}|$ に比例する計算量で求められる。

4.3 条件付き期待値の計算

式 (4) で定義した条件付き期待値 $\eta_{\theta}^x[\pi]$ ($\pi \in \Pi_F$, $x \in \{0, 1\}$) は後ろ向き確率と前向き確率を用いて計算される。 $\mathcal{P}_f^{F_i}(n, x)$ ($F_i \in F$, $f \in \{0, 1\}$, $n \in N_{F_i}$) を Δ_{F_i} の根節点から f 定数節点に至るパスのうち $\langle n, x \rangle$ を含むパスの集合とすれば、式 (4) は以下のように書き換えられる。

$$\eta_{\theta}^x[\pi] = \sum_{(F_i, f_i) \in \mathcal{O}} \delta_{(F_i, f_i)} \sum_{n \in N_{F_i}(\pi)} \sum_{\rho \in \mathcal{P}_{f_i}^{F_i}(n, x)} P_{\theta}(\phi_{\rho} | F_i = f_i)$$

ここで $N_{F_i}(\pi)$ は π に属する基本命題でラベル付けられた節点の集合である。 $\eta_{\theta}^x[\pi]$ を計算するため、以下で定義される $p_n[x|F_i=f]$ を導入する。

$$p_n[x|F_i=f] \equiv \mathcal{F}_{\theta}[n] \theta_{(\pi_n, x)} \mathcal{B}_{\theta}^f[Ch_x(n)] / \mathcal{B}_{\theta}^f[n_1]$$

後ろ向き、前向き確率の定義より、 $\mathcal{B}_{\theta}^f[n_1]$ は F_i の根節点から f 定数節点に至る確率 $P_{\theta}(F_i=f)$ を意味し、 $\mathcal{F}_{\theta}[n] \theta_{(\pi_n, x)} \mathcal{B}_{\theta}^f[Ch_x(n)]$ はパス $\rho \in \mathcal{P}_f^{F_i}$ が $\langle n, x \rangle$ を含む確率を意味する。 $p_n[x|F_i=f]$ は「 $F_i=f$ を得たとき、枝 $\langle n, x \rangle$ が Δ_{F_i} 上のパスに含まれる確率」を意味し、

$$p_n[x|F_i=f] = \sum_{\rho \in \mathcal{P}_f^{F_i}(n, x)} P_{\theta}(\phi_{\rho}|F_i=f)$$

と書ける。これより $\eta_{\theta}^x[\pi]$ は $p_n[x|F_i=f]$ を用いて以下のように書き換えることができる。

$$\eta_{\theta}^x[\pi] = \sum_{F_i \in F} \sum_{n \in N_{F_i}(\pi)} \sum_{f \in \{0,1\}} \delta_{(F_i, f)} p_n[x|F_i=f] \quad (5)$$

式 (5) で定義される条件付き期待値は、Proc. 3 で定義される EXPECTATION(Δ_{F_i}) をすべての $\Delta_{F_i} \in \Delta_F$ に対して実行することで計算できる。 Δ_{F_i} に対する EXPECTATION(Δ_{F_i}) の計算量は $|N_{F_i}|$ に比例するので ESTEP(Δ_F) 全体の計算量は全 BDD のサイズの総和 $\sum_{F_i \in F} |N_{F_i}|$ に比例する。

Procedure 3 EXPECTATION(Δ_{F_i})

- 1: for $n \in N_{F_i}, x \in \{0,1\}$ do
 - 2: for all $f \in \{0,1\}$
 - 3: $p_n[x|F_i=f] = \mathcal{F}_{\theta}[n] \mathcal{B}_{\theta}^f[Ch_x(n)] \theta_{(\pi_n, x)} / \mathcal{B}_{\theta}^f[n_1]$;
 - 4: end for
 - 5: $\eta_{\theta}^x[n] += \delta_{(F_i, 0)} p_n[x|F_i=0] + \delta_{(F_i, 1)} p_n[x|F_i=1]$;
 - 6: end for
-

5 提案アルゴリズム

4章で説明した BDD-EM アルゴリズムは観測命題 F_1, \dots, F_N を N 個の BDD で表現し、それらの上で確率計算を行うことで BDD の総節点数に比例する計算量で E-step を実行した。その一方複数の BDD を効率的に表現する手法として SBDD や否定枝技法が提案されているので本論文では BDD-EM アルゴリズムを否定枝を含む SBDD 上で実行できるよう拡張する。

式 (4) の条件付き期待値 $\eta_{\theta}^x[\pi]$ ($\pi \in \Pi_F, x \in \{0,1\}$) は、観測命題 $F = \{F_1, \dots, F_N\}$ を表現した N 個の BDD を用いて EM アルゴリズムを定式化することにより得られたが、 F を表現する否定枝を含む SBDD Δ_F を用いて定式化したときも同じ式が得られる (証明略)。観測命題 $F = \{F_1, \dots, F_N\}$ を表現した SBDD Δ_F のすべての節点の集合を N_F で表すと、 $n \in N_F$ は複数の観測命

題によって共有されることがある。そこで SBDD 上で動作する BDD-EM アルゴリズムは確率 $p_n[x|F_i=f]$ に変わって、以下で定義される期待値 $E_{\theta}[\langle n, x \rangle]$ ($n \in N_F, x \in \{0,1\}$) を計算する。

$$E_{\theta}[\langle n, x \rangle] = \sum_{(F_i, f_i) \in O} \sum_{\rho \in \mathcal{P}_{f_i}^{F_i}(n, x)} \delta_{(F_i, f_i)} P_{\theta}(\phi_{\rho}|F_i=f_i)$$

$E_{\theta}[\langle n, x \rangle]$ は「観測列 O を得たとき、枝 $\langle n, x \rangle$ が Δ_F 上のパスに含まれる回数の期待値」を意味し、これを用いて式 (4) の $\eta_{\theta}^x[\pi]$ は以下のように書き換えられる。

$$\eta_{\theta}^x[\pi] = \sum_{n \in N_F(\pi)} E_{\theta}[\langle n, x \rangle] \quad (6)$$

$E_{\theta}[\langle n, x \rangle]$ は SBDD の節点 $n \in N_F$ に対して、否定枝を考慮した後ろ向き確率 $\mathcal{B}_{\theta}^1[n], \mathcal{B}_{\theta}^0[n]$ と、前向き期待値 $\mathcal{F}_{\theta}^1[n], \mathcal{F}_{\theta}^0[n]$ を用いて計算できる。これより SBDD に対する BDD-EM アルゴリズムの E-step は Proc. 4 のように、後ろ向き計算と前向き期待値を計算する BACKWARD(Δ_F) と FORWARD(Δ_F)、そして条件付き期待値を計算する EXPECTATION(Δ_F) を用いて実行できる。

Procedure 4 ESTEP(Δ_F)

- 1: BACKWARD(Δ_F);
 - 2: FORWARD(Δ_F);
 - 3: EXPECTATION(Δ_F);
-

否定枝を含む SBDD の節点 $n \in N_F$ の後ろ向き確率は BDD のそれと異なる。否定枝はそれ以下の部分グラフで論理を反転させるため、1定数節点に至るパス ρ に否定枝が奇数本含まれるとき、その1定数節点は論理的に0定数節点を意味する。節点 $n \in N_F$ に対する後ろ向き確率 $\mathcal{B}_{\theta}^x[n]$ ($x \in \{0,1\}$) は「節点 n から論理的に x 定数節点に至る確率」と定義し、以下で計算する。

$$\mathcal{B}_{\theta}^x[n] = \theta_{(\pi_n, 0)} \mathcal{B}_{\theta}^x[Ch_0(n)] + \begin{cases} \theta_{(\pi_n, 1)} \mathcal{B}_{\theta}^x[Ch_1(n)] & \langle n, 1 \rangle \text{ が通常枝} \\ \theta_{(\pi_n, 1)} \mathcal{B}_{\theta}^{1-x}[Ch_1(n)] & \langle n, 1 \rangle \text{ が否定枝} \end{cases}$$

後ろ向き確率は BACKWARD(Δ_F) (Proc. 5) によって $|N_F|$ に比例する時間で計算される。

SBDD 上で動作する BDD-EM アルゴリズムは条件付き期待値 $\eta_{\theta}^x[\pi]$ ($\pi \in \Pi_F, x \in \{0,1\}$) を枝 $\langle n, x \rangle$ ($n \in N_F$) の期待値 $E_{\theta}[\langle n, x \rangle]$ を用いて計算する。 $E_{\theta}[\langle n, x \rangle]$ を計算するために前向き期待値 $\mathcal{F}_{\theta}^1[n], \mathcal{F}_{\theta}^0[n]$ を定義する。 $\mathcal{F}_{\theta}^1[n]$ ($\mathcal{F}_{\theta}^0[n]$) は「観測列 O を得たとき、 $n \in N_F$ が Δ_F 上のパスに含まれる回数の期待値」のうち、節点 n から論理的に1定数節点 (0定数節点) に至るパスについてのみ考えた値である。 $\mathcal{P}_n^{F_i}(e)$ ($\mathcal{P}_n^{F_i}(o)$) を $\rho \in \mathcal{P}_n^{F_i}$ のうち否定枝を偶数 (奇数) 本含むパスの集合とすれば、

Procedure 5 BACKWARD(Δ_F)

```
1: // initialize
2:  $\mathcal{B}_\theta^1[\mathbb{1}] = 1$ ;  $\mathcal{B}_\theta^1[\mathbb{0}] = 0$ ;  $\mathcal{B}_\theta^0[\mathbb{1}] = 0$ ;  $\mathcal{B}_\theta^0[\mathbb{0}] = 1$ ;
3: // backward prob.
4: for  $i = |N_F|$  to 1 do
5:    $\pi = \pi^{var(n_i)}$ ;
6:   if  $\langle n_i, 1 \rangle$  is not a negative edge then
7:      $\mathcal{B}_\theta^1[n_i] = \theta_{(\pi,0)} \mathcal{B}_\theta^1[Ch_0(n_i)] + \theta_{(\pi,1)} \mathcal{B}_\theta^1[Ch_1(n_i)]$ ;
8:      $\mathcal{B}_\theta^0[n_i] = \theta_{(\pi,0)} \mathcal{B}_\theta^0[Ch_0(n_i)] + \theta_{(\pi,1)} \mathcal{B}_\theta^0[Ch_1(n_i)]$ ;
9:   else if  $\langle n_i, 1 \rangle$  is a negative edge then
10:     $\mathcal{B}_\theta^1[n_i] = \theta_{(\pi,0)} \mathcal{B}_\theta^1[Ch_0(n_i)] + \theta_{(\pi,1)} \mathcal{B}_\theta^0[Ch_1(n_i)]$ ;
11:     $\mathcal{B}_\theta^0[n_i] = \theta_{(\pi,0)} \mathcal{B}_\theta^0[Ch_0(n_i)] + \theta_{(\pi,1)} \mathcal{B}_\theta^1[Ch_1(n_i)]$ ;
12:   end if
13: end for
```

$\mathcal{F}_\theta^1[n]$, $\mathcal{F}_\theta^0[n]$ は以下の式によって計算される.

$$\begin{aligned} \mathcal{F}_\theta^1[n] &= \sum_{F_i \in F} \left(\sum_{\rho \in \mathcal{P}_n^{F_i}(e)} \delta_{(F_i,1)} P_\theta(\phi_\rho | F_i = 1) \right. \\ &\quad \left. + \sum_{\rho \in \mathcal{P}_n^{F_i}(o)} \delta_{(F_i,0)} P_\theta(\phi_\rho | F_i = 0) \right) \\ \mathcal{F}_\theta^0[n] &= \sum_{F_i \in F} \left(\sum_{\rho \in \mathcal{P}_n^{F_i}(e)} \delta_{(F_i,0)} P_\theta(\phi_\rho | F_i = 0) \right. \\ &\quad \left. + \sum_{\rho \in \mathcal{P}_n^{F_i}(o)} \delta_{(F_i,1)} P_\theta(\phi_\rho | F_i = 1) \right) \end{aligned}$$

前向き期待値は FORWARD(Δ_F) (Proc. 6) によって計算され, その計算量は SBDD のサイズ $|N_F|$ に比例する.

Procedure 6 FORWARD(Δ_F)

```
1: // initialize
2: for all  $n \in N_F$ 
3:    $\mathcal{F}_\theta^1[n] = 0$ ;  $\mathcal{F}_\theta^0[n] = 0$ ;
4: end for
5: for all  $F_i \in F$ 
6:    $n = \text{Root}(F_i)$ ; // The root node of  $F_i$ 
7:   if  $n$  is not referenced by a negative edge then
8:      $\mathcal{F}_\theta^x[n] += \delta_{(F_i,x)} / \mathcal{B}_\theta^x[n]$  for  $x \in \{0,1\}$ ;
9:   else if  $n$  is referenced by a negative edge then
10:     $\mathcal{F}_\theta^{1-x}[n] += \delta_{(F_i,x)} / \mathcal{B}_\theta^x[n]$  for  $x \in \{0,1\}$ ;
11:   end if
12: end for
13: // forward prob.
14: for  $i = 1$  to  $|N_F|$  do
15:   if  $\langle n_i, 1 \rangle$  is not a negative edge then
16:      $\mathcal{F}_\theta^x[Ch_1(n_i)] += \theta_{(\pi_{n_i},1)} \mathcal{F}_\theta^x[n_i]$  for  $x \in \{0,1\}$ ;
17:   else if  $\langle n_i, 1 \rangle$  is a negative edge then
18:      $\mathcal{F}_\theta^{1-x}[Ch_1(n_i)] += \theta_{(\pi_{n_i},1)} \mathcal{F}_\theta^x[n_i]$  for  $x \in \{0,1\}$ ;
19:   end if
20:    $\mathcal{F}_\theta^x[Ch_0(n_i)] += \theta_{(\pi_{n_i},0)} \mathcal{F}_\theta^x[n_i]$  for  $x \in \{0,1\}$ ;
21: end for
```

$\mathcal{B}_\theta^1[n]$ と $\mathcal{F}_\theta^1[n]$ の定義より, $\mathcal{F}_\theta^1[n] \theta_{(n,0)} \mathcal{B}_\theta^1[Ch_0(n)]$ は「観測列 \mathcal{O} を得たとき, 枝 $\langle n, 0 \rangle$ が Δ_F 上のパスに

含まれる期待値」のうち, $Ch_0(n)$ から 1 定数節点に至るパスについてのみ考えた値である. これより枝 $\langle n, 0 \rangle$ ($n \in N_F$) の期待値 $E_\theta[\langle n, 0 \rangle]$ は以下の式で計算できる.

$$E_\theta[\langle n, 0 \rangle] = \sum_{f \in \{0,1\}} \mathcal{F}_\theta^f[n] \theta_{(\pi_n,0)} \mathcal{B}_\theta^f[Ch_0(n)] \quad (7)$$

枝 $\langle n, 1 \rangle$ が通常枝であれば式 (7) と同様に $E_\theta[\langle n, 1 \rangle]$ を計算できる. しかし $\langle n, 1 \rangle$ が否定枝であるとき, それ以下で論理が入れ代わることを考慮して $\mathcal{B}_\theta^1[Ch_1(n)]$ と $\mathcal{B}_\theta^0[Ch_1(n)]$ を入れ替える必要がある. これより $E_\theta[\langle n, 1 \rangle]$ は以下の式で計算できる.

$$E_\theta[\langle n, 1 \rangle] = \begin{cases} \sum_{f \in \{0,1\}} \mathcal{F}_\theta^f[n] \theta_{(\pi_n,1)} \mathcal{B}_\theta^f[Ch_1(n)] & \langle n, 1 \rangle \text{ が通常枝} \\ \sum_{f \in \{0,1\}} \mathcal{F}_\theta^f[n] \theta_{(\pi_n,1)} \mathcal{B}_\theta^{1-f}[Ch_1(n)] & \langle n, 1 \rangle \text{ が否定枝} \end{cases} \quad (8)$$

式 (7)(8) より $\eta_\theta^x[\pi]$ は Proc. 7 によって N_F のサイズに比例する計算量で求められる. 後ろ向き確率, 前向き期待値の計算量も $O(|N_F|)$ であることから, 否定枝を含む SBDD に対する BDD-EM アルゴリズムの E-step の計算量は SBDD のサイズに比例する.

Procedure 7 EXPECTATION(Δ_F)

```
1: for  $i = 1$  to  $|N_F|$  do
2:   if  $\langle n_i, 1 \rangle$  is not a negative edge then
3:      $\eta_\theta^1[n_i] += \mathcal{F}_\theta^1[n_i] \mathcal{B}_\theta^1[Ch_1(n_i)] \theta_{(\pi_{n_i},1)}$ ;
4:      $+\mathcal{F}_\theta^0[n_i] \mathcal{B}_\theta^0[Ch_1(n_i)] \theta_{(\pi_{n_i},1)}$ ;
5:   else if  $\langle n_i, 1 \rangle$  is a negative edge then
6:      $\eta_\theta^1[n_i] += \mathcal{F}_\theta^1[n_i] \mathcal{B}_\theta^0[Ch_1(n_i)] \theta_{(\pi_{n_i},1)}$ ;
7:      $+\mathcal{F}_\theta^0[n_i] \mathcal{B}_\theta^1[Ch_1(n_i)] \theta_{(\pi_{n_i},1)}$ ;
8:   end if
9:    $\eta_\theta^0[n_i] += \mathcal{F}_\theta^1[n_i] \mathcal{B}_\theta^1[Ch_1(n_i)] \theta_{(\pi_{n_i},0)}$ ;
10:   $+\mathcal{F}_\theta^0[n_i] \mathcal{B}_\theta^0[Ch_1(n_i)] \theta_{(\pi_{n_i},0)}$ ;
11: end for
```

6 実験

複数の BDD 上で動作する BDD-EM アルゴリズムの計算量は全 BDD のサイズの総和 $\sum_{F_i \in F} |N_{F_i}|$ に比例し, 提案手法である否定枝を含む SBDD 上で動作する BDD-EM アルゴリズムの計算量は SBDD のサイズ $|N_{F_i}|$ に比例する. つまり, 観測命題を表現する複数の BDD を 1 つの SBDD に統合することで節点が削減できれば, その上で動作する EM アルゴリズムの計算量も削減される. 本論文では診断などでよく用いられている 2 部 noisy-OR ネットワークで表現される問題に対して, SBDD を用いることで BDD の節点数を削減できることを確認する.

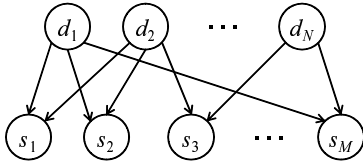


図 4: 2部 noisy-OR ネットワークの例

6.1 2部 noisy-OR ネットワーク

2部 noisy-OR ネットワーク (bipartite noisy-OR network) は複数の原因と複数の結果の関係を表現するベイジアンネットワークである. 図 4 は N 個の原因 $D = \{d_1, \dots, d_N\}$ と M 個の症状 (結果) $S = \{s_1, \dots, s_M\}$ の関係を表現する 2部 noisy-OR ネットワークである. 原因 $d \in D$ はそれぞれ独立に $0, 1$ の値を確率的に取り, 症状 $s \in S$ は N 個の原因のうちいくつかの原因に依存して確率的に $0, 1$ の値を取る. $d=1$ は原因 d が起きたことを表し, 同様に $s=1$ は症状 s が現れたことを意味する. 2部 noisy-OR ネットワークでは, 症状 $s \in S$ とそれを引き起こす原因 $Pa(s) \subseteq D$ の関係は **noisy-OR** により以下の論理関数により表現される.

$$s \Leftrightarrow \bigvee_{d \in Pa(s)} (d \wedge \neg I(d, s)) \quad (9)$$

ここで $I(d, s)$ ($s \in S, d \in Pa(s)$) は病気 d が症状 s を引き起こすことを妨げる事象を表現した妨害変数である. つまり s と $Pa(s)$ が noisy-OR 関係であるとき, s が真になるのは 1 つ以上の $d \in Pa(s)$ が真であり, その内のいずれかの妨害変数 $I(d, s)$ が偽であるときに限られる. 患者 k に対して各症状 $s \in S$ が表れているかを診断し, 表れている症状の集合を S_p^k , 表れていない症状の集合を S_n^k とすると, 患者 k の診断結果 R^k は式 (9) を用いて以下のように表現される.

$$\begin{aligned} R^k &\Leftrightarrow \left(\bigwedge_{s \in S_p^k} s \right) \wedge \left(\bigwedge_{s \in S_n^k} \neg s \right) \\ &\Leftrightarrow \left(\bigwedge_{s \in S_p^k} \bigvee_{d \in Pa(s)} (d \wedge \neg I(d, s)) \right) \\ &\quad \wedge \left(\bigwedge_{s \in S_n^k} \neg \left(\bigvee_{d \in Pa(s)} (d \wedge \neg I(d, s)) \right) \right) \quad (10) \end{aligned}$$

観測命題を $F = \{R^1, \dots, R^K\}$, 基本命題を $V_F = D \cup \{I(d, s) \mid s \in S, d \in Pa(s)\}$ とすれば, 観測命題 $R^k \in F$ は式 (10) のように基本命題の論理関数として書ける. K 人の患者に対する診断結果は $\mathcal{O} = \{(R^1, 1), \dots, (R^K, 1)\}$ と表現され, BDD-EM アルゴリズムを用いれば, \mathcal{O} を得る確率を最大化する $P(d=1)$ ($d \in D$) を学習できる.

6.2 実験と考察

原因 $D = \{d_1, \dots, d_N\}$ と症状 $S = \{s_1, \dots, s_M\}$ の関係を表す人工的な 2部 noisy-OR ネットワークを生成し,

そこからサンプリングされた 100 個の観測 R^1, \dots, R^{100} を複数の BDD と SBDD で表現し, それらの総節点数を比較する (実装の都合上, 否定枝は BDD と SBDD の双方で利用した). 人工的なネットワークにおいて, $Pa(s)$ ($s \in S$) は D からランダムに C 個以下選び, $P(d_i = 1) = 1/N$ ($d \in D$), $P(I(d, s) = 1) \in [0, 0.1]$ ($s \in S, d \in Pa(s)$) とした. この C を最大原因数と呼び, N, M, C を変化させることで BDD と SBDD のサイズがどのように変化するかを観測した.

図 5 は $N=10, C=5$ とし, M を $10, 20, \dots, 100$ と変化させたときの BDD と SBDD のサイズを表したグラフであり, 図 6 は $C=10$ としたときのグラフである. 本実験のように 1 つのネットワークから複数の観測を得る場合, それらを表現する BDD は共通の部分グラフを多く含むことが予想される. 図 5, 6 を見ると, 常に SBDD の方がコンパクトになっており, 確かに共通の部分グラフを多く含んでいたことが分かる. また, 図 5 と図 6 を比べると, 後者の方が SBDD を用いることで BDD をより圧縮できることがわかる. これは各症状の原因が増えたことで, 共通の原因を持つ症状が増え, より共有化できる部分グラフが増えたからと考えられる. 実験より, 2部 noisy-OR ネットワークで表される問題において, SBDD を用いた EM アルゴリズムを用いると確かに計算量が削減できることを確認した.

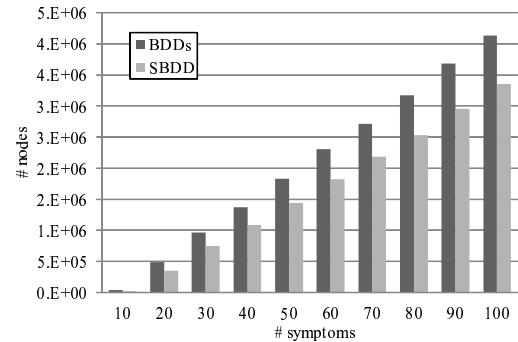


図 5: 実験結果 1 ($\max_{s \in S} |Pa(s)| = 5$)

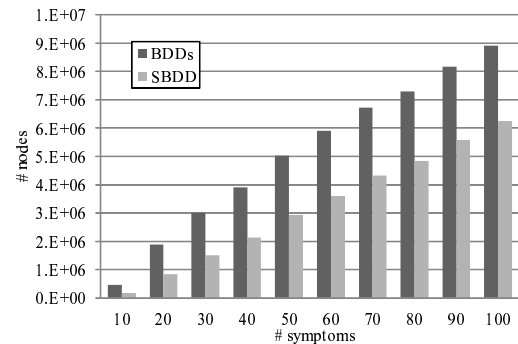


図 6: 実験結果 2 ($\max_{s \in S} |Pa(s)| = 10$)

7 関連研究

BDD を用いて確率計算を行う手法はいくつか提案されており, その一つとして BDD の一種である Zero-suppressed BDD を用いてベイジアンネットワークの確率計算を行う手法 [9] が挙げられるが, これを用いたパラメータ学習アルゴリズムはまだ提案されていない. ProbLog[3] は Prolog プログラムの各述語に確率を導入した論理形式であり, ProbLog プログラムは観測式を説明する基底アトムから成る BDD を用いて確率計算を行う. ProbLog による確率学習アルゴリズムが提案されているが [6, 11], それらは BDD-EM アルゴリズムのように任意の論理関数で表現された観測に対して EM アルゴリズムを実行することはできない. PRISM[10] は Prolog を基にしたモデル記述言語であり, BDD 似た説明グラフというデータ構造を用いることでベイジアンネットワークの確率計算や学習を行う. PRISM では扱う論理式の各選言子が互いに排反であるという仮定をおいているが, 文献 [12] あるいは本論文で提案した BDD-EM アルゴリズムを用いることでこの仮定を取り除くことが可能である.

8 まとめ

BDD-EM アルゴリズムは観測事象がいくつかの確率的命題の論理関数で表現されるとき, 観測事象を表現した BDD 上で EM アルゴリズムを実行することで各確率的命題が真偽をとる確率を学習する. 本論文では BDD-EM アルゴリズムを BDD をよりコンパクトに表現する手法として広く利用されてる Shared BDD と否定枝を扱えるよう拡張し, 提案アルゴリズムの計算量が SBDD のサイズに比例することを示した. また, 従来の BDD-EM アルゴリズムは条件付き期待値を計算する際に BDD の簡略化規則によって削除された節点に関する期待値を数値的に復元していたが, これが不要ないことについても述べた. 更に 2 部 noisy-OR ネットワークにおいて複数の観測事象を複数の BDD で表現する代わりに 1 つの SBDD で表現することで, 節点数を削減できることを確認し, 結果として BDD-EM アルゴリズムの計算量も削減できることを確かめた.

参考文献

- [1] S. B. Akers. Binary decision diagrams. *IEEE Trans. on Computers*, Vol. 27, No. 6, pp. 509–516, 1978.
- [2] R.E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Trans. on Computers*, Vol. 35, No. 8, pp. 677–691, 1986.
- [3] L. De Raedt, K. Angelika, and H. Toivonen. ProbLog: A probabilistic Prolog and its application in link discovery. In *Proc. of IJCAI'07*, pp. 2468–2473, 2007.
- [4] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Statistical Society B*, Vol. 39, pp. 1–38, 1977.
- [5] R. Drechsler and D. Sieling. Binary decision diagrams in theory and practice. *Int'l J. on Software Tools for Technology Transfer*, Vol. 3, pp. 112–136, 2001.
- [6] Bernd. Gutmann, Angelika. Kimmig, Kristian. Kersting, and Luc Raedt. Parameter learning in probabilistic databases: A least squares approach. In *Proc. of the 2008 European Conf. on Machine Learning and Knowledge Discovery in Databases (ECML08)*, pp. 473–488, 2008.
- [7] Y. Kameya and T. Sato. Efficient EM learning with tabulation for parameterized logic programs. In *Proc. of the 1st Int'l Conf. on Computational Logic (CL2000)*, 2000.
- [8] S. Minato, N. Ishiura, and S. Yajima. Shared binary decision diagram with attributed edges. *Proc. of ACM/IEEE Design Automation Conf.*, pp. 52–57, 1990.
- [9] S. Minato, K. Satoh, and T. Sato. Compiling Bayesian networks by symbolic probability calculation based on Zero-suppressed BDDs. In *Proc. of IJCAI'07*, pp. 2550–2555, 2007.
- [10] T. Sato and Y. Kameya. Parameter learning of logic programs for symbolic-statistical modeling. *J. of Artificial Intelligence Research*, Vol. 15, pp. 391–454, 2001.
- [11] Ingo Thon, Niels Landwehr, and Luc Raedt. A simple model for sequences of relational state descriptions. In *Proc. of the 2008 European Conf. on Machine Learning and Knowledge Discovery in Databases (ECML08)*, pp. 506–521, 2008.
- [12] 石畠正和, 亀谷由隆, 佐藤泰介, 湊真一. BDD 上の命題化確率計算に基づく EM アルゴリズム. 人工知能学会第 70 回基本問題研究会 (SIG-FPAI) 予稿集, SIG-FPAI-A801, pp. 15–22, 2008.