

# ベイジアンネットワーク学習アルゴリズム TPDA の高速化

## Fast Bayesian Network Learning Algorithm based on TPDA

森下 民平\*1\*2      植野 真臣\*2  
MORISHITA, Mimpei      UENO, Maomi

\*1株式会社シーエーシー  
CAC Corporation

\*2電気通信大学大学院 情報システム学研究所  
Graduate School of Information Systems, The University of Electro-Communications

We propose a constraint-based learning algorithm of Bayesian networks (BNs) for large data based on TPDA that uses monotone DAG faithful (MDF) assumption. By exploiting MDF assumption, we can reduce computational complexity of the algorithm. Our experimental results show the time effectiveness of the proposed method while keeping accuracy.

### 1. はじめに

本研究は、大規模データ (多変数で大量のデータ) 環境下でも実用的かつ安定した時間で動作するベイジアンネットワーク (BN) 学習アルゴリズムの開発を目的とする。本研究では、もっとも大規模データ処理に適した BN 学習アルゴリズム TPDA (Three Phase Dependency Analysis)[Cheng 02] を改良し計算量を減らすことにより、実行時間および実行時間の標準偏差を低く抑える手法を提案する。

BN の構造学習は、変数の数に対して NP 困難 [Chickering 96; Chickering 04] であり、大規模データ環境への BN 適用は困難な課題である。BN 学習アルゴリズムは、構造のベイズ予測分布スコアを最大化するサーチ&スコアベース手法と、確率変数間の条件付き独立テスト (CI テスト) により確率的依存関係を抽出することを目的とした制約ベースの 2 種類の手法に大別される。大規模データ処理では、より計算効率の高い制約ベース手法を用いるのが合理的である。TPDA は、大規模データ環境下でもっとも高速に動作する制約ベース学習アルゴリズムである。TPDA は、MDF (monotone DAG faithful) と呼ぶ条件を導入し、CI テストの回数を  $O(N^4)$  に抑えることを可能とした。(ここで  $N$  は変数の数)。

しかし TPDA の問題点は、MDF の仮定下における切断集合の探索方法が冗長で計算量が大きいことである。ここで任意の 2 つの確率変数の切断集合とは、当該 2 変数を互いに条件付き独立とする条件変数集合のことをいう。一般に、制約ベース学習における確率変数間の CI テストは、与えられた条件変数集合から、切断集合となる部分集合を探索することにより行う。TPDA は、与えられた条件変数集合の部分集合サイズの降順に切断集合を探索し、最大で  $N-2$  段階の探索を必要とする。

提案手法は、TPDA 同様 MDF を仮定し、条件変数集合の部分集合サイズの昇順に、最大でも 3 段階の探索で切断集合の探索と存在判定を行うというものである。これにより提案手法は、TPDA よりも少ない計算量で BN を学習できる。

TPDA と提案手法の比較評価実験により、第一に、提案手法は構造推定精度を同等に保ったまま実行時間の大幅な短縮と安定化が可能であることを示す。第二に、提案手法により、

連絡先: 森下 民平, (株) シーエーシー, 東京都中央区日本橋箱崎町 24-1, Phone 03-6667-8070, Fax 03-5641-3213, mimpei@cac.co.jp

1,000 個の確率変数を持つ BN をデスクトップ PC 上で充分実用的な時間で学習できることを示す。

### 2. 関連研究

Chickering ら [Chickering 06] は、TPDA が用いる MDF の仮定は、制約ベースの学習手法で通常用いられる faithful の仮定とは両立しないことを証明した。しかし TPDA は、学習精度と速度の両面において未だに最先端の学習アルゴリズムと同等の成績を示し、特に大量データ環境では他よりも高速に動作することが実験的に示されており [Brown 05; Tsamardinos 06] 実用的な価値は高い。

MMHC [Tsamardinos 06] を改良した PMMS [Brown 05] は TPDA と同じ CI テスト回数  $O(N^4)$  ながら MDF よりも広く用いられる faithful の仮定で動作するよう設計されている。しかしデータ数が大きくなると TPDA より学習時間がかかることが実験的に確かめられており、大規模データ処理には適さない。

### 3. ベイジアンネットワーク

特に断らない限り、本論では以下の表記を用いる。

確率分布  $P$  において、 $\mathbf{X}$  が  $\mathbf{Z}$  を所与として  $\mathbf{X}$  から条件付き独立であることを  $Ind_P(\mathbf{X}; \mathbf{Y}|\mathbf{Z})$  と書く。このとき  $\mathbf{Z}$  を条件付き独立関係の条件変数集合と呼ぶ。

$N$  個の離散確率変数  $X_1, \dots, X_N$  からなり、各変数  $X_i$  が状態  $k \in \{0, \dots, r_i - 1\}$  をとる定義域  $\mathbf{U}$  を考える。 $\mathbf{U}$  上の BN は、 $B = \langle \mathcal{G}, \Theta \rangle$  と定義される。ここで最初の要素  $\mathcal{G} = \langle \mathbf{V}, \mathbf{E} \rangle$  は、確率変数集合  $\mathbf{U} = \{X_1, \dots, X_N\}$  に対応する頂点集合  $\mathbf{V}$  とその変数間の依存関係を表現した有向辺集合  $\mathbf{E}$  からなる有向非循環グラフを表す。グラフ  $\mathcal{G}$  は、変数間の独立関係を表現している。次の成分  $\Theta$  は、パラメータ  $\theta_{ijk} = P(X_i = k | \Pi_i = j)$  の集合である。ここで  $\Pi_i$  は変数  $X_i$  のグラフ  $\mathcal{G}$  上の親変数集合を表し、 $\Pi_i = j$  は  $\Pi_i$  が  $j$  番目の状態値を取ることをいう。BN  $B$  は、変数集合  $\mathbf{U}$  上のユニークな同時確率分布を以下のように定める。

$$P(X_1, \dots, X_N | B) = \prod_{i=1}^N P(X_i | \Pi_i, \mathcal{G}) \quad (1)$$

グラフ  $\mathcal{G}$  上で、各辺  $X_{(i)} \rightarrow X_{(i+1)}$  または  $X_{(i)} \leftarrow X_{(i+1)}$  が存在し、各頂点が高々一度しか現れない頂点の順序

列  $\langle X_{(1)}, X_{(2)}, \dots, X_{(n)} \rangle$  を経路と呼び,  $\pi$  で表す.  $\mathcal{G}$  上の  $X$  と  $Y$  間のすべての経路に含まれる変数集合を  $X$  と  $Y$  の経路集合と呼ぶ. 経路  $\pi$  中の任意の  $\{X_{(i-1)}, X_{(i)}, X_{(i+1)}\}$  について, 辺  $X_{(i-1)} \rightarrow X_{(i)}$  と  $X_{(i)} \leftarrow X_{(i+1)}$  がともに存在するとき,  $X_{(i)}$  は経路  $\pi$  の位置  $i$  における合流点と呼ぶ. 非循環有向グラフ  $\mathcal{G}$  上の経路  $\pi$  は, 条件変数集合  $\mathbf{Z}$  を所与として経路上の各変数  $X_{(i)}$  が次のどちらかの性質を持つときにアクティブ経路と呼ぶ. すなわち, (1)  $X_{(i)}$  が位置  $i$  で合流点でなくかつ  $X_{(i)}$  が  $\mathbf{Z}$  に含まれないとき, または, (2)  $X_{(i)}$  が位置  $i$  で合流点でかつ  $X_{(i)}$  もしくは  $\mathcal{G}$  における  $X_{(i)}$  の子孫が  $\mathbf{Z}$  に含まれるときである.

変数集合  $\mathbf{X}$  に含まれる任意の変数と  $\mathbf{Y}$  に含まれる任意の変数とに条件変数集合  $\mathbf{Z}$  を所与としてグラフ  $\mathcal{G}$  におけるアクティブ経路が存在しないとき,  $\mathbf{X}$  と  $\mathbf{Y}$  は条件変数集合  $\mathbf{Z}$  を所与として有向分離されるという.

## 4. TPDA

TPDA (Three Phase Dependency Analysis) [Cheng 02] は, 効率的に動作することを目的として開発された制約ベース学習アルゴリズムである. 他の多くの制約ベース手法と同様に, TPDA は確率変数間の CI テストに条件付き相互情報量を用いる.  $X$  と  $Y$  の間の  $\mathbf{Z}$  を所与とした確率分布  $P$  における条件付き相互情報量は, 以下のように定義される.

$$\text{Inf}_P(X; Y | \mathbf{Z}) = \sum_{x, y, \mathbf{z}} P(x, y, \mathbf{z}) \log \frac{P(x, y | \mathbf{z})}{P(x | \mathbf{z})P(y | \mathbf{z})} \quad (2)$$

ここで条件変数集合が  $\emptyset$  の場合を相互情報量  $\text{Inf}_P(X; Y)$  と表す.

制約ベースアルゴリズムの実行時間のほとんどは確率変数間の CI テストに費やされる. そのため TPDA は, 極力 CI テストの回数を減らすように設計されている. 具体的には, MDF 条件の導入と 3 フェーズに分かれた処理とを行う.

$\text{Active}_{\mathcal{G}}(X; Y | \mathbf{Z})$  を, グラフ  $\mathcal{G}$  において条件変数集合  $\mathbf{Z}$  を所与とした  $X, Y$  間のアクティブな経路集合を表すものとする. このとき確率分布  $P$  は  $\mathcal{G}$  について, 以下の式が成立するときまたそのときに限り  $\text{MDF}$  (*monotone DAG faithful*) であると言う.

$$\begin{aligned} \text{Active}_{\mathcal{G}}(X; Y | \mathbf{Z}) \subseteq \text{Active}_{\mathcal{G}}(X; Y | \mathbf{Z}') \\ \implies \text{Inf}_P(X; Y | \mathbf{Z}) \leq \text{Inf}_P(X; Y | \mathbf{Z}') \end{aligned} \quad (3)$$

TPDA は,  $X, Y$  間の辺の要否をテストする際, 所与とする  $\mathbf{Z}$  の中から  $\text{Inf}(X; Y | \mathbf{C}) < \varepsilon$  となる部分集合  $\mathbf{C}$  を探し, 見つければ辺  $\{X, Y\}$  は存在しないと判断する. この  $X$  と  $Y$  を有向分離する集合  $\mathbf{C}$  を  $X$  と  $Y$  の切断集合と呼ぶ. 与えられた変数集合  $\mathbf{Z}$  に切断集合が含まれるか否かをテストすることを切断集合存在テストと呼ぶ. 切断集合存在テストにおいて, 各  $\mathbf{Z}' \subseteq \mathbf{Z}$  を所与としたとき,  $\text{Inf}(X; Y | \mathbf{Z}')$  の成立を判定すること, すなわち TPDA においては  $\text{Inf}(X; Y | \mathbf{Z}') < \varepsilon$  の成立を判定することを CI テストと呼ぶ.

MDF を仮定しない場合, 切断集合存在テストには最大で  $2^{|\mathbf{Z}|}$  回  $\text{Inf}(X; Y | \cdot)$  を計算する必要がある. そのため,  $\mathbf{Z}$  のサイズが大きくなるにつれ, 計算は急速に困難になる. TPDA は MDF を仮定しこの問題に対処する. すなわち,  $\mathbf{Z}$  に含まれる変数が MDF の仮定により切断集合に含まれないと判断したならば, 当該変数をその探索からは以後外すことにより部分集合の組み合わせ爆発を回避する.

TPDA は, 切断集合存在テストの際, 条件変数集合の部分集合を部分集合サイズの降順に列挙し, 最大  $N - 2$  段階の探索を行う.

TPDA は, 以下の 3 フェーズにわかれた処理を行い, 最後に記録した切断集合を用いオリエンテーションルールに基づいて辺の方向を決める.

(1) Drafting: Chow & Liu アルゴリズム [Chow 68] を用いてベースとなる木構造を構築する.

(2) Thickening: 足りない可能性のある辺をすべて付け加える. すなわち, 木構造のなかで辺がない変数ペアについて, 当該ペアの経路上の隣接変数を所与とし切断集合存在テストを行い, 存在しなければ辺を追加する.

(3) Thinning: 余分な辺を削除する. 各辺  $\{X, Y\}$  について,  $\{X, Y\}$  以外の経路が存在するなら, 当該経路上の  $X$  と  $Y$  の隣接変数と条件によってはさらにその隣接変数とを所与とし切断集合存在テストを行い, 辺  $\{X, Y\}$  の要否を決める.

### 4.1 TPDA の計算量

TPDA は, 確率変数の数を  $N$  とすると, CI テストの回数を最大で  $O(N^4)$  に抑えられる [Cheng 02]. しかし CI テストの回数は, 制約ベースアルゴリズムの計算量の上限ではない. 一般に, 制約ベースアルゴリズムの最大計算量は CI テストのパターン数で決まり, 変数の数  $N$  に対して指数オーダーである. ここでパターンとは, その CI テストにおける対象の 2 変数ならびに条件変数すべてが取りうる状態の組み合わせのことである.

まず, TPDA の最大計算量について考える.  $\mathbf{Z}$  を所与とした  $X$  と  $Y$  の切断集合存在テストを行うこと, すなわち  $\mathbf{Z}' \subseteq \mathbf{Z}$  を列挙しそれぞれについて CI テストを行うことを考える. 変数が取りうる状態の最大値を  $r$  とすると, 条件変数集合サイズ  $|\mathbf{Z}'|$  は最大で  $N - 2$ , CI テスト  $\text{Inf}(X; Y | \mathbf{Z}')$  のパターン数は  $r^{|\mathbf{Z}'|}$  である. TPDA では, 条件変数集合のサイズは第 1 段階目の  $|\mathbf{Z}'| = N - 2$  から 1 ずつ減少し, 最後の  $N - 2$  段階目では条件変数集合サイズは  $|\mathbf{Z}'| = 1$  となる. したがって TPDA の CI テストのパターン数は, 切断集合存在テストの各段階  $S = 1, 2, \dots, N - 2$  において  $r^N, r^{N-1}, \dots, r^3$  となる. よって TPDA の最大計算量は, 各段階の CI テスト数にパターン数を乗じ和を取った値, すなわち  $O(r^N + \sum_{i=2}^{N-2} ir^{i+1})$  である.

次に, TPDA の切断集合存在テストごとの最小計算量を考える. TPDA は, 変数  $X$  と  $Y$  の切断集合を条件変数集合  $\mathbf{Z}$  から探索する際, 条件変数集合の部分集合  $\mathbf{Z}' \subseteq \mathbf{Z}$  を部分集合サイズの降順に列挙し, 着目する部分集合が切断集合であるか否かを CI テストにより調べる. CI テストの計算量は, 所与とする変数集合サイズの指数オーダーで, 変数の取る状態数の最大値を  $r$ , 所与とする変数集合サイズ  $|\mathbf{Z}'|$  を最大の場合の  $N - 2$  とすると  $O(r^{|\mathbf{Z}'|})$  である. したがって, 条件変数集合の部分集合サイズの降順に切断集合を探索する TPDA では, 切断集合存在テストごとの最小計算量は  $O(r^{N-2})$  である.

## 5. 提案アルゴリズム

TPDA の問題点は, MDF の仮定下における切断集合存在テストの方法が冗長で計算量が大いことにある. 具体的には, 条件変数集合の部分集合サイズの降順に最大で  $N - 2$  段階の CI テストが必要で, 各段階の CI テストにおける条件変数集合サイズも,  $N - 2, \dots, 2, 1$  のすべてをテストする必要がある.

一方提案手法は、切断集合の探索を、条件変数の部分集合サイズの昇順に最大で3段階、対応する条件変数集合サイズも  $|Z'| = 1, 2, N - 2$  の3種類しかテストする必要がない。

以下、提案手法のアルゴリズムを示す。提案手法のアルゴリズムの骨格は、TPDAと同じである。異なるのは、切断集合存在テストにおける切断集合探索処理である。提案手法の切断集合探索処理を図1に示す。

```
// The function returns the cut-set of X and Y in Z.
Function CutSetSearch (variable X, variable Y, variable set
Z): variable set

1:  $Z^c := Z$ 
2: for all  $Z_i \in Z^c$  do
3:    $s_i := \text{Inf}(X; Y | \{Z_i\})$ 
4:   if  $s_i < \varepsilon$  then
5:     return  $\{Z_i\}$ 
6: for all  $\{Z_i, Z_j\}$  s.t.  $Z_i, Z_j \in Z^c, i \neq j$  do
7:    $s_{ij} := \text{Inf}(X; Y | \{Z_i, Z_j\})$ 
8:   if  $s_{ij} < \varepsilon$  then
9:     return  $\{Z_i, Z_j\}$ 
10:  if  $s_{ij} > s_i$  then
11:     $Z^c := Z^c \setminus \{Z_j\}$ 
12:  if  $s_{ij} > s_j$  then
13:     $Z^c := Z^c \setminus \{Z_i\}$ 
14:  $s := \text{Inf}(X; Y | Z^c)$ 
15: return  $\begin{cases} Z^c & \text{if } s < \varepsilon, \\ \emptyset & \text{otherwise.} \end{cases}$ 
```

図1: CutSetSearch: 提案手法の切断集合探索処理

CutSetSearch 関数は、確率変数  $X, Y$ 、および条件変数集合  $Z$  を受け取り、 $X$  と  $Y$  の切断集合が  $Z$  に含まれればそれを返す関数である。まず1行目の  $Z^c$  は切断集合の候補集合であり、初期状態は  $Z$  のコピーである。2から5行目の第1段階では、条件変数の部分集合サイズ  $|Z'| = 1$  の場合の CI テストを行う。6から13行目までは第2段階、すなわち部分集合サイズを2として CI テストを行う。このとき MDF の仮定を用い、 $\text{Inf}(X; Y | Z_i) < \text{Inf}(X; Y | \{Z_i, Z_j\})$  なら  $Z_j$  を切断集合候補集合  $Z^c$  から削除する。第3段階の14から15行目では、残った  $Z^c$  をすべて所与として CI テストを行う。

### 5.1 提案手法の計算量

この節では、提案手法の計算量を示す。

まず提案手法の最大計算量を検討する。切断集合存在テストの最初の段階では、条件変数集合サイズは  $|Z'| = 1$  のため、CI テストすなわち  $\text{Inf}(X; Y | Z')$  の真偽判定で扱う変数の数は  $X$  と  $Y$  を含め3である。したがって、最初の段階における CI テストのパターン数は、 $r$  を変数の状態数の最大値とすると  $r^3$  である。同様に、次の段階ではパターン数は  $r^4$  となる。最後の3段階目では、最大の場合、CI テストで変数すべてを扱うため、パターン数は  $r^N$  である。各段階の CI テストの回数とパターン数を乗じ和を取ると、最大計算量は  $O(r^N + N^2 r^4)$  である。

提案手法の最大計算量  $O(r^N + N^2 r^4)$  と、TPDA の最大計算量  $O(r^N + \sum_{i=2}^{N-2} i r^{i+1})$  とを比較すると、その差分は両手法に共通な  $r^N$  を除いた部分、すなわち TPDA では  $O(\sum_{i=2}^{N-2} i r^{i+1})$ 、提案手法では  $O(N^2 r^4)$  である。TPDA では、計算量の差分  $O(r^{N-1})$  は  $N$  について指数オーダーなのに対し、提案手法では  $N, r$  いずれについても多項式オーダーである。

次に切断集合存在テストごとの最小計算量は、提案手法では  $O(Nr^3)$  である。なぜなら提案手法では1段階目までで切

断集合を発見すれば、 $N - 2$  回の CI テストで CI テストごとに  $r^3$  のパターンを計算すれば良いからである。

以上より、提案手法は TPDA より最大計算量および最小計算量ともに少ないことがわかる。

## 6. 実験結果

TPDA および提案アルゴリズムを Java を用いて実装し、実行時間と正解となるネットワーク構造との一致度合いを計測した実験結果を示す。

実験に用いた環境は、Intel Core 2 プロセッサ 2.67GHz の 4GB RAM 上で動作する Windows Vista Business SP2 で、Java 仮想マシンには最大 512MB のメモリを割り当てた。

本実験では、ベイジアンネットワーク構造学習のベンチマークとして頻りに用いられる表1に示す alarm, insurance, water, win95pts の4つのネットワークを学習した。ここで in/out は各頂点の最大の入次数と出次数、state は各変数の状態数の最小値と最大値を表す。

表1: 実験に用いたネットワーク構造

	V	E	in/out	state
alarm	37	46	4/5	2-4
insurance	27	52	3/7	2-5
water	32	66	5/3	3-4
win95pts	76	112	7/10	2-2

TPDA の条件付き相互情報量の閾値  $\varepsilon$  は、[Cheng 02] で推奨される 0.01 をそのまま使用した。データ数は、1,000 件から 50,000 件までのサンプルをそれぞれ 10 セットずつ用いた。各ネットワークについて実行時間の平均および標準偏差を計測した結果を図2に示す。

すべてのネットワークにおいて、提案手法は実行時間平均を改善しており、特に比較的 TPDA で実行時間がかかっている alarm と insurance では実行時間を大幅に改善している。また提案手法の実行時間の標準偏差は、すべての場合において極めて小さく、データによらず安定した時間で学習できることを示している。

提案手法の辺の誤検出数は、TPDA と比較して平均で alarm(+1.41), insurance(+0.17), water(+0.12), win95pts(+0.05) であり、各ネットワークにおける標準偏差 (1.32 から 3.95) からすると誤差の範囲と考えられる。提案手法と TPDA の学習精度がネットワークによりわずかに異なるのは、MDF の仮定が成立しない場合があるためと考えられる。

次に、変数の増加に伴う実行時間の変化の実験結果を示す。最大入出次数を 4-5、変数の状態数を 2 から 4 に固定し、変数の数を 100 から 1,000 まで増加させたネットワークを合成した。各ネットワークにつき、10,000 件のサンプルを 10 セットずつ生成し、実行時間を計測した。図3にその計測結果を示す。

TPDA は変数数が 400 個および 600 個以上のとき、いずれのサンプルセットも 3 時間以上学習が終了しないため実行を打ち切っている。一方提案手法では、1,000 個の変数を持つネットワークでも平均 26 分以下、標準偏差も 24 秒程度で学習が可能であることがわかる。

## 7. おわりに

本研究の結論は、以下の通りである。

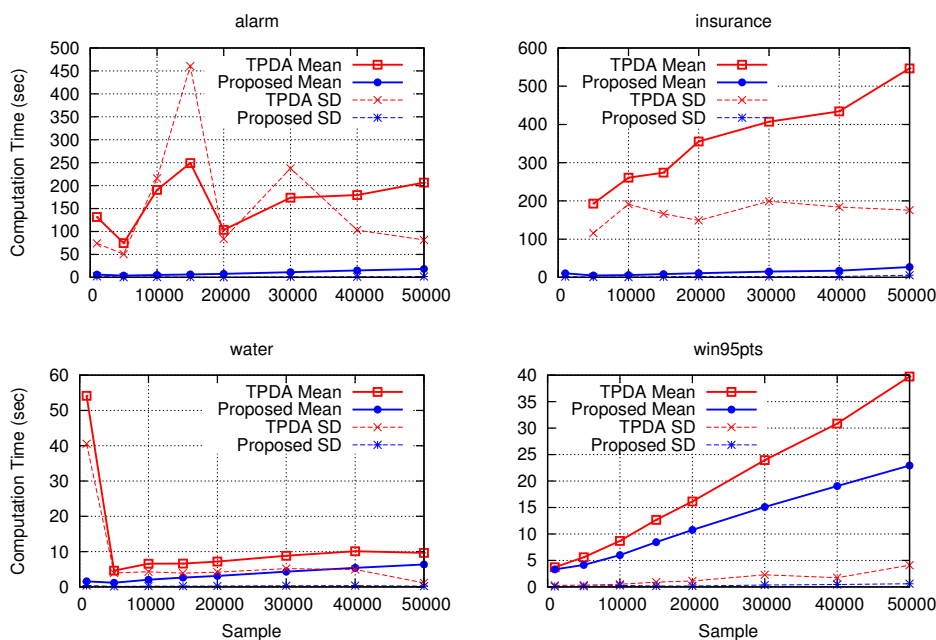


図 2: 実行時間 (秒) の平均と標準偏差

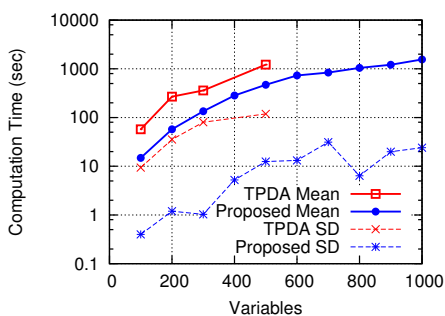


図 3: 変数の数と実行時間

(1) TPDA には, MDF を仮定した切断集合存在テストに冗長さがあることを指摘し, これを改善するアルゴリズムを提案した. (2) TPDA の最大計算量が  $O(r^N + \sum_{i=2}^{N-2} ir^{i+1})$  であるのに対し, 提案手法の最大計算量は  $O(r^N + N^2r^4)$  に抑えられることを示した. また切断集合存在テストごとの最小計算量は, TPDA は  $O(r^N)$  の指数オーダーであるのに対し, 提案手法では  $O(Nr^3)$  の多項式オーダーとなることを示した. (3) TPDA と提案手法の実装と比較評価実験を行い, その結果, 提案手法は学習精度を保ったまま実行時間を大幅に短縮できること, また実行時間の標準偏差を非常に小さく抑えられることを示した. さらに提案手法は, 1,000 個の変数からなるネットワークを標準的なデスクトップ PC 上でも充分実用的な時間で学習できることを示した.

今後の課題として, [Chickering 06] により faithful の仮定とは両立しないことが証明されている MDF の仮定を用いるのではなく, より信頼性の高い仮定は何かを突き止め, それを用いた学習アルゴリズムを開発したい.

## 参考文献

- [Brown 05] Brown, L. E., Tsamardinos, I., and Aliferis, C. F.: A Comparison of Novel State-of-the-Art Polynomial Bayesian Network Learning Algorithms, in *Proceedings of AAAI*, pp. 739–745 (2005)
- [Cheng 02] Cheng, J., Greiner, R., Kelly, J., Bell, D., and Liu, W.: Learning Bayesian networks from data: an information-theory based approach, *Artificial Intelligence*, Vol. 137, No. 1-2, pp. 43–90 (2002)
- [Chickering 96] Chickering, D. M.: Learning Bayesian networks is NP-complete, in *Learning from Data: Artificial Intelligence and Statistics V*, pp. 121–130, Springer-Verlag (1996)
- [Chickering 04] Chickering, D. M., Heckerman, D., and Meek, C.: Large-Sample Learning of Bayesian Networks is NP-Hard, *J. Mach. Learn. Res.*, Vol. 5, pp. 1287–1330 (2004)
- [Chickering 06] Chickering, D. M. and Meek, C.: On the incompatibility of faithfulness and monotone DAG faithfulness, *Artificial Intelligence*, Vol. 170, No. 8, pp. 653–666 (2006)
- [Chow 68] Chow, C. K. and Liu, C. N.: Approximating discrete probability distributions with dependence trees, *IEEE Trans. on Information Theory*, Vol. 14, No. 3, pp. 462–467 (1968)
- [Pearl 88] Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann (1988)
- [Spirtes 93] Spirtes, P., Glymour, C., and Scheines, R.: *Causation, Prediction, and Search*, Springer-Verlag (1993)
- [Tsamardinos 06] Tsamardinos, I., Brown, L. E., and Aliferis, C. F.: The max-min hill-climbing Bayesian network structure learning algorithm, *Machine Learning*, Vol. 65, pp. 31–78 (2006)