令和元年度 情報数理工学プログラム卒業論文概要

平成 28 年度 入学	学籍番号 1610210
指導教員 植野 真臣	氏名 菊地 康介
題目 Tree width 制約下での tts 最小化コーダルグラフの探索	

概要

ベイジアンネットワークの厳密学習は NP 困難であり、60 変数程度が限界であった。しかし、近年、Natori ら (2017) は数千のベイジアンネットワーク学習を可能とするアルゴリズムを提案している。一方、推論はジョインツリーアルゴリズムの厳密計算量 tts を最小にする極小コーダル化を探索する厳密手法で使われる BT アルゴリズムは百変数程度が限界である。厳密手法では、ベイジアンネットワークのグラフ構造 G に存在する全ての PMC を列挙し、最小の tts を持つ極小コーダル化したグラフを動的計画法により、探索している。しかし、千変数を超える大規模ベイジアンネットワークのグラフ構造から PMC を全て列挙することは困難であるが、Li ら (2017) の研究によると、tts を最小にする極小コーダル化されたグラフの tree width は G の tree width と一致する場合が多いことが示されている。そこで、本研究では、G の tree width を k としたときに、k+1 以下のノード数を持つ PMC のみを列挙し、列挙した PMC を用いて構成される極小コーダル化の内、最小の tts を求める手法を提案する。シミュレーション実験により、17 種類のベイジアンネットワークにおいて、提案手法は従来の近似手法よりも小さな tts を求めることができ、厳密手法では求められなかった千変数を超えるベイジアンネットワークにおいて、小さい tts が求められることを示す。

電気通信大学情報理工学域
I 類 (情報系) 情報数理工学プログラム卒業論文

Tree width 制約下での tts 最小化コーダルグラフの探索

2020 年 2 月 17 日

情報数理工学プログラム

学籍番号 1610210

菊地康介

指導教員 植野 真臣

目次

1 はじめに 2

2 用語の定義 4

2.1 ベイジアンネットワークについて 4

2.2 無向グラフについて 4

3 ジョインツリーアルゴリズム 5

4 ハイブリッド法 9

5 BT アルゴリズムによる tts の計算 10

5.1 tts(total table size) 10

5.2 BT アルゴリズム 10

6 提案手法 12

7 実験 15

8 むすび 16

1　はじめに

ベイジアンネットワークは、離散確率変数をノードとし、ノード間の依存関係を非循環有向グラフ (Directed Acyclic Graph: DAG) で表現した確率的グラフィカルモデルの一つである。

各離散変数は、その親変数と CPT(Conditional Probabities Tables) と呼ばれる、条件付き確率表によって関係づけられる。このことにより、同時確率分布をコンパクトな形で表せることが、ベイジアンネットワークの特徴の一つである。

ベイジアンネットワークの厳密学習は NP 困難問題であり [18]、動的計画法 [19-23],A*探索 [24], 整数計画法 [25] などのアプローチにより厳密学習法が考案されているが、60 ノード程度のベイジアンネットワークの学習しかできなかった。しかし、漸近一致性を有する大規模ベイジアンネットワークの構造学習を行う方法が開発され [26-28]、1000 ノード程度のネットワークを学習できるようになった。

ベイジアンネットワーク上で行われる確率推論は重要な研究課題である。ベイジアンネットワークの厳密な周辺事後確率分布の計算は NP-hard であることが知られている [1]。また、近似的に推論することでさえも一般的には困難であるとされている [15]。このことが、ベイジアンネットワークの広範な応用への妨げとなっている。また、ベイジアンネットワークを MLF 式に変換し、MLF 式を表す ZDD を生成することにより、ベイジアンネットワークの厳密推論を行う方法が存在するが、ZDD を構築する時間が長いという問題点がある [16,17]。この問題に対して、確率推論の計算量を軽減するために、ジョインツリーアルゴリズムが考案されている [2-4]。ジャンクションツリーはモラル化されたベイジアンネットワークの木分解であり、ジャンクションツリー上で確率伝搬を行うことにより、厳密な周辺事後確率を効率的に計算できる。ジャンクションツリーアルゴリズムの厳密計算量は tts(total table size) で計算される。

最小の tts となる、木分解を探索することは、最小の tts を持つ極小コーダル化されたグラフの探索と同一である。既存研究 [10] では、BT アルゴリズムを用いて極小コーダル化の探索を行っている。

BT アルゴリズムとは、グラフに存在する PMC(potential maximal clique) をすべて列挙し、列挙した PMC から、tree width,minimum-fillin,generalized and fractional hypertreewidth,tts などの問題の各スコアを最適にするような、極小コーダル化されたグラフを動的計画法で探索するアルゴリズムである。[5]BT アルゴリズムは 2001 年ごろに考案されたアルゴリズムであるが、近年まで実際に実装されたことはなかった。しかし、2018 年頃に実際に実装され、実用上有用なアルゴリズムであることがわかった。[10] において、100 変数程度のベイ

ジアンネットワークであれば、数秒で最小の tts を持つコーダル化されたグラフを発見できることが示されている。しかし、PMC の数は $O(1.7347^n)$[7] で増加することが知られており、これは、グラフに存在する maximal clique の数の $O(3^{\frac{n}{3}})$ より大きく、大規模なベイジアンネットワークでは、全ての PMC を列挙することは困難である。

一方、tts が最小となるベイジアンネットワークの tree width は、モラル化されたベイジアンネットワークの tree width と一致する場合が多いことが報告されている。[8] また、tree width の大きさを k とすると、k+1 以下の大きさの PMC は、グラフ全体の PMC の数よりも大幅に少ないことが示されている。[9] このことから、本研究では、PMC を、モラル化されたベイジアンネットワークの tree width の大きさを k としたとき、k+1 以下の大きさのもののみを列挙し、その下で最小の tts をもつ極小コーダル化されたグラフを動的計画法で求める手法を提案する。この手法により、与えられた無向グラフ G の最小の tts を近似的に求めることができる。列挙する PMC を制限することにより、以下の利点が得られる。

- 厳密手法では、時間、空間計算量が大きすぎるために計算できない、1000 変数を超えるベイジアンネットワークで、小さい tts を求めることができる
- 求められた tts が厳密手法と一致する場合が多い

また、近似的に小さい tts を求める方法として、ハイブリッド法が既存手法として存在している。実験を行い既存手法で求められる最小の tts よりも提案手法で求めた tts の方が小さいことを示した。

2　用語の定義

2.1　ベイジアンネットワークについて

n 個の変数集合 $\boldsymbol{x} = \{x_1, x_2, \cdots, x_n\}$ をもつベイジアンネットワークは、(G, Θ) で表現される。[12]

- G は $\boldsymbol{x}$ に対応するノード集合によって構成される非循環有効グラフ (directed acyclic graph,DAG) ネットワーク構造と呼ばれる。
- Θ は、G の各ノードに対応する条件付き確率パラメータ集合 $\{p(x_i|\Pi_i, G)\}, (i = 1, \cdots, n)$ である。ただし、Π_i は変数 x_i の親変数集合を示している。

2.2　無向グラフについて

本論文では、有限でありシンプルな連結無向グラフについて扱う。また、$G = (V, E)$ を連結無向グラフとし、G の頂点集合と、辺集合をそれぞれ $V(G)$、$E(G)$ と表記する。$G[S]$ を $S \subset V(G)$ かつ、$V(G[S]) = S$ かつ、$E(G[S]) = E(G) \cap \{\{u, v\}|u \in S, v \in S\}$ を満たす G の誘導部分グラフとする。G に存在するすべての二頂点が互いに辺で結ばれているとき、G を完全であるという。$G - S$ が元のグラフより連結成分の数が多いとき、S を separator という。また、S の頂点部分集合が、separator になっていないとき、S を minimal separator という。G に存在するすべての minimal separator の集合を $\delta(G)$ で表す。また、$G - S$ に存在する連結成分を component という。G に存在する閉路の内、隣合わない二頂点に辺が引かれているとき、その辺をコードという。G に存在するすべての長さ 4 以上の閉路がコードを持つとき G をコーダルグラフという。$V(H) = V(G)$ かつ $E(G) \subset E(H)$ を満たすコーダルグラフ H を G に辺を追加することによって作成することをコーダル化という。このうち $E(H)$ に含まれる辺を一つでも削除すると、コーダルグラフでならなくなるコーダル化を極小コーダル化という。部分集合 $\omega \subset V(G)$ について、$G[\omega]$ が完全であるとき、ω をクリークという。G 中に存在するほかのクリーク ω' が $\omega \subset \omega'$ を満たさないとき、ω を極大クリークという。G を極小コーダル化した時に、極大クリークになるものを potential maximal clique(PMC) という。

3　ジョインツリーアルゴリズム

本章では、ベイジアンネットワークの確率推論に用いられるジョインツリーアルゴリズムについて説明する。ジョインツリーアルゴリズムは大きく二つの工程に分かれている。一つ目の工程は与えられたベイジアンネットワークの DAG 構造からジョインツリーを構築する工程。二つ目は構築したジョインツリー上で確率伝搬を行う工程である。

初めに、ベイジアンネットワークの DAG 構造から、ジョインツリーへの変換方法について説明する。例として、図 1 に示した asia と呼ばれるベイジアンネットワークの変換を例として挙げる。

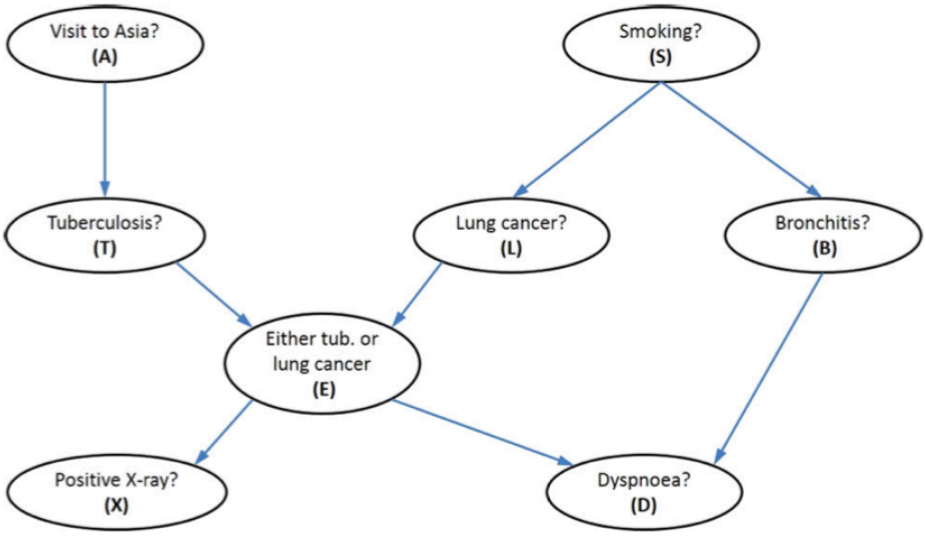

図 1: アジア

変換は次の 4step で行われる。

1. ベイジアンネットワークをモラル化する。ここで、モラル化とは、有効グラフにおいて、ある二頂点が共通の頂点を指す場合に、その二頂点間に無向辺を引き、(図 2 の左側) そのあとに、有効辺を無向辺に変換する操作を指す。(図 2 の右側)

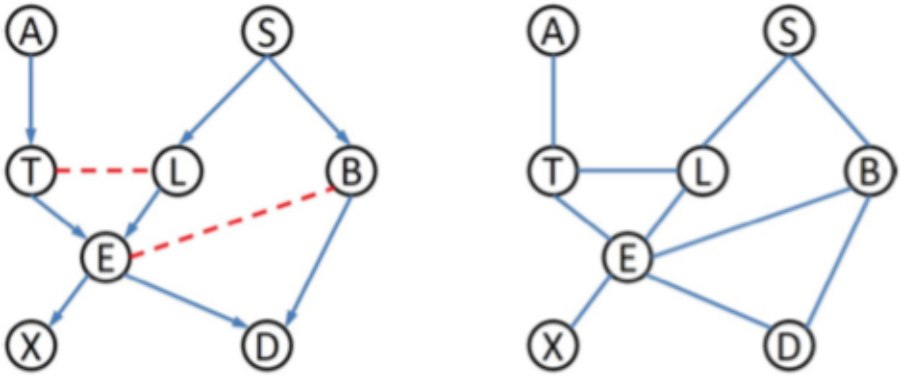

図 2: ベイジアンネットワークのモラル化

2. モラル化されたグラフをコーダル化する。

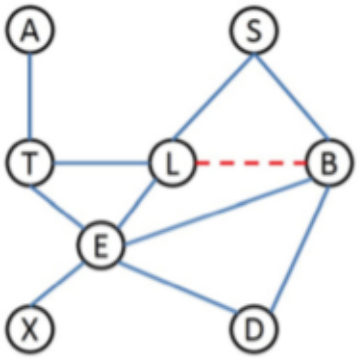

図 3: グラフのコーダル化

3. コーダル化されたグラフに存在するすべての maximal clique を列挙する。
4. 列挙した maximal clique を頂点とし、頂点間に以下の性質を満たすように辺を引く。
   - ジョインツリーに存在する二つの maximal clique,C_i, C_j について、$C_i \cap C_j$ が、C_i と C_j の path の間にのみ存在する。

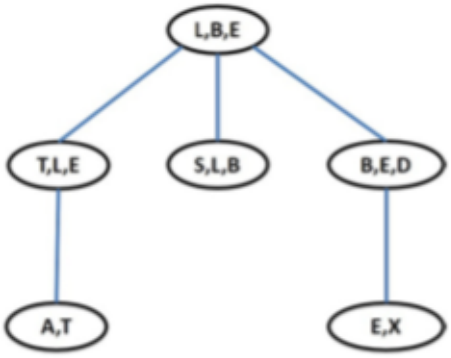

図 4: グラフのジョインツリー化

このとき、step1 と step3 は一意に決定できるが、step2 と step4 は一意に決定はできない。特に step2 は、コーダル化問題と呼ばれ、最適な極小コーダル化を選択することがベイジアンネットワークの推論の速さに大きくかかわる。step4 において、Jensen[14] らは最適なジャンクションツリーの構築を提案している。

次にジョインツリー上での確率伝搬について説明する。ジョインツリー上での確率伝搬の工程は以下の 3step で行われる。

1. ジョインツリー上の辺すべてに対して、辺の両端にある頂点に対応する maximal clique の共通頂点のラベルを図 5 のように紐づける。このラベルは separator となっている。
2. ジョインツリー上の maximal clique C_i に対して、ポテンシャル ϕ_i を作る。また、ベイジアンネットワークの CPT を用いて、ポテンシャル ϕ_{ij} を全てのセパレーターに紐づける。(すべての値を一つにする)
3. ジョインツリー上の頂点間で separator を通して、確率伝搬を行う。例えば、図 4 の場合、maximal clique C_i から C_j にメッセージ送るとする。このとき、図 6 のように separatorS_{ij} を用いて、ポテンシャル ϕ_i を除して、ϕ'_{ij} を得る。メッセージ ϕ'_{ij}/ϕ_{ij} は ϕ_j に受け取られる。また、separatorϕ_{ij} は ϕ'_{ij} に取り換えられる。

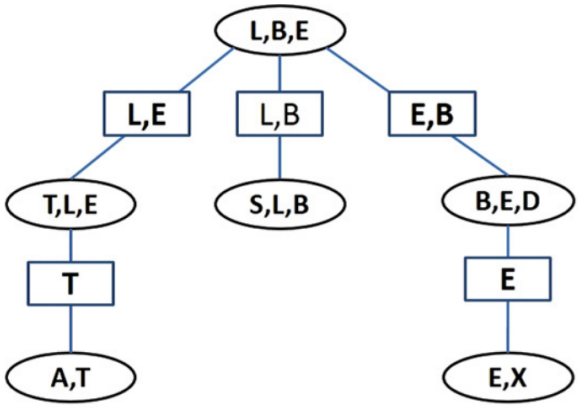

図 5: separator の追加

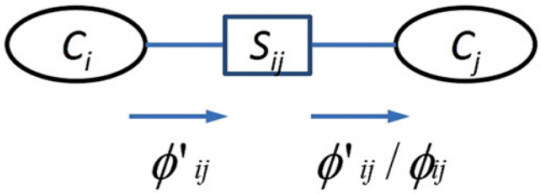

図 6: メッセージパッシング

確率伝搬の方法について説明する。最初にジョインツリー上の maximal clique を一つ選び、根とし、その maximal clique から確率伝搬をスタートする。メッセージパッシングは二つのフェーズに分かれる。一つ目の段階は集積フェーズと呼ばれるフェーズである。集積フェーズでは maximal clique が隣接する maximal clique の内、根に向かう maximal clique 以外の maximal clique からメッセージを受け取ると、根に向かう maximal clique へとメッセージを送る。これを根となる maximal clique がその隣接する maimal clique 全てからメッセージを受け取るまで行う。二つ目の段階は分配フェーズと呼ばれるフェーズである。このフェーズでは集積フェーズで根にメッセージを集めたあとに、根から、隣接する maximal clique にメッセージを伝搬する。この伝搬を葉となる maximal clique 全てに伝搬するまで行う。

上記に説明した通りにジョインツリー上で確率伝搬を行うことにより、効率的に確率推論を行えることが知られている。また、確率推論を行う際の厳密計算量は tts といい、tts を最小化することが確率推論を高速に行うことにつながる。

4　ハイブリッド法

連結無向グラフ G を極小コーダル化するための方法として、G のある頂点 v を一つ選び、その頂点に隣接している頂点間に辺がない場合に、辺を追加し、v を消去する。この操作を変数消去という。次に完全化した頂点を取り除いた後のグラフでも同様に頂点を一つ選び変数消去する。この操作を繰り返し、G から頂点がなくなるまで続ける。この操作で、加えた辺すべてをもとの G に加えると G が極小コーダル化されることが知られている。よって、変数を消去する順番を求めれば、G を極小コーダル化したグラフがわかる。この事実から、小さい tts を求めるために以下の二つのヒューリスティックな変数消去アルゴリズムが考案されている。Algorithm1 は min-degree heuristic と呼ばれ、tree width が 2 以下であれ

Algorithm 1 min-degree heuristic

1: $Order \leftarrow ()$
2: **for** i=1 to n **do**
3: 　$Order \leftarrow G$ の中で最小の次数を持つ頂点 v
4: 　v に隣接する頂点間に辺が引かれていない場合辺を引く
5: 　$G \leftarrow G \setminus v$
6: **end for**
7: **return** Order

ば、最小の tts を持つ極小コーダル化したグラフを求めることができることが知られている。Algorithm2 は min-fill heuristic と呼ばれ、Algorithm2 で、最小の辺の数を持つ頂点が複数

Algorithm 2 min-fill heuristic

1: $Order \leftarrow ()$
2: **for** i=1 to n **do**
3: 　$Order \leftarrow G$ の中で隣接する頂点間に辺が引かれていない場合辺を引いた場合に最小の辺の数を持つ頂点 v
4: 　v に隣接する頂点間に辺が引かれていない場合辺を引く
5: 　$G \leftarrow G \setminus v$
6: **end for**
7: **return** Order

ある場合に最小の次数を持つ頂点でタイブレークする方法が最も有効であると報告されてお

り [30]、ハイブリッド法と呼ばれる。

5　BT アルゴリズムによる tts の計算

5.1　tts(total table size)

上記で示したジョインツリー上の確率伝搬を行う際の厳密計算量を表す tts(total table size) について定義する。最初に ts(table size) について定義する。

定義 1.

クリーク C の ts(table size) とは、

$$ts(C) = \prod_{V_i \in C} |r_i| \tag{1}$$

で定義され、$|r_i|$ はノード V_i に対応する変数の状態数を示す。

次に tts(total table size) について定義する。

定義 2.

連結無向グラフ G の tts(total table size) とは、グラフ中の全クリークについての ts の合計として

$$tts(G) = \sum_{C \in G} ts(C) \tag{2}$$

で定義される。

上記の章で定義した tts はモラル化されたベイジアンネットワークの極小コーダル化の方法によって大きく異なる。そのため、最小または、小さい tts を持つ極小コーダル化されたグラフを探すことが、ベイジアンネットワークの確率推論の計算の高速化につながる。

5.2　BT アルゴリズム

BT アルゴリズムとは、[5] で提案されたアルゴリズムである。コーダルグラフの minimal searator が、互いに互いを分離しないという性質と、minimal separator が maximal clique を分離しないという性質を用いて動的計画法を行い、高速に極小コーダル化グラフのなかから、目的に合ったものを求めるアルゴリズムである。具体的には、BT アルゴリズムは二つの工程に分かれている。初めに、与えられた連結無向グラフ G の PMC をすべて列挙する。次に、列挙した PMC を用いて、条件に最的な極小コーダル化したグラフを構築する。

BT アルゴリズムの疑似コードを記した。Algorithm3 の 4 から 11 行目で、(Ω, S, C) の三つ組を与えられた連結無向グラフ G に存在するすべての PMC について作成する。ここで、

Algorithm 3 BT algorithm

1: $BTALGORITHM(G, cliqueCost, mergeCost)$
2: $\Pi \leftarrow EnumeratePMCs(G)$
3: $T \leftarrow \{\}$
4: **for each** $\Omega \in \Pi$ **do**
5: **for each** $D \in C(G \setminus \Omega)$ **do**
6: $S \leftarrow N(D)$
7: $C \leftarrow The component of G \setminus S such that \omega \subset S \cup C$
8: $T \leftarrow T \cup \{\Omega, S, C\}$
9: **end for**
10: $T \leftarrow T \cup \{(\Omega, \emptyset, V(G))\}$
11: **end for**
12: sort T in increasing order of$|S \cup C|$
13: $dp[(S, C)] \leftarrow \infty forall(S, C)$
14: **for each** $(\Omega, S, C) \in T$ **do**
15: $cost \leftarrow cliqueCost(\Omega, S)$
16: **for each** $C' \in C(G[C \setminus \Omega])$ **do**
17: $S' \leftarrow N(C')$
18: $cost \leftarrow mergeCost(cost, dp[(S', C')])$
19: **end for**
20: **if** $cost < dp[S, C]$ **then**
21: $dp[(S, C)] \leftarrow cost$
22: $optChoice[(S, C)] \leftarrow \Omega$
23: **end if**
24: **end for**
25: **return** $dp[(\emptyset, V(G))]$

Ω は PMC、S は minimal-separator、$G \setminus S$ に存在する component であり、$S \subset \Omega \subset S \cup C$ を満たす。ここで、10 行目で S をϕ として、G 全体を一つの block としてみた、ダミーノードが挿入されている。このダミーノードは、S が ϕ ではない三つ組を全て計算した後に、最適な block の組み合わせを探索するために必要になる。
この三つ組を 12 行目で block の大きさ、$|S \cup C|$ の大きさで昇順にソートする。13 行目で、各 block に対して変数を一つ用意し、初期値として∞を代入する。

Algorithm 4 Reconstructin the optimal triangulation

1: $RECONSTRUCT(G, optChoice)$
2: $H \leftarrow G$
3: $Q \leftarrow \{(\{\}, V(G))\}$
4: **while** $Qnotempth$ **do**
5: $(S, C) \leftarrow pop(Q)$
6: $\Omega \leftarrow optChoice[(S, C)]$
7: $E(H) \leftarrow E(H) \cup \Omega^2$
8: **for each** $C' \in C(G[C \setminus \Omega])$ **do**
9: $S' \leftarrow N(C')$
10: $Q \leftarrow Q \cup \{(S', C')\}$
11: **end for**
12: **end while**
13: **return** H

14 行目から 24 行目でソートした三つ組に対して動的計画法を行う。15 行目にある cliqueCost 関数は PMC に対して与えられた問題に対するスコアが計算される。例えば、tts 最小化問題の場合 ts が計算される。18 行目にある mergeCost 関数では、三つ組に対して、三つ組に含まれる block、$S \cup C$ に Ω が含まれているときに最適な block、$S \cup C$ の tts が計算される。

Algorithm4 では、Algorithm3 において、optChoice に加えられた PMC を用いて、実際に極小コーダル化されたグラフを構成している。

6 提案手法

BT アルゴリズムを用いて、連結無向グラフ G の最小の tts を求める場合、G を tts が最小になるように極小コーダル化したグラフ H に含まれる PMC の条件は現在までには研究されていない。よって、最小の tts をもつ極小コーダル化されたグラフを構築するには、G に存在する PMC 全てを列挙する必要がある。しかし、ベイジアンネットワークが大規模になるにつれて、PMC の数は爆発的に増えるため、時間、空間計算量が膨大となり、最小の tts を計算することができない。

また、PMC を列挙するために、[5],[7] の方法では minimal-separator をすべて列挙する必要があるが、minimal-separator は $O(1.6181^n)$[29] で増えることが知られている。そのため、

minimal-separator 自体をすべて列挙できず、PMC を列挙すらできない場合がある。

一方、G の tts を最小にする極小コーダル化は G の tree width と一致する場合が多いことが知られている [8]。また、G の tree width の大きさを k としたとき、k+1 以下の PMC の数は、G に存在する PMC の数よりも大幅に少ない。以上のことから、本研究では、k+1 以下の PMC を全て、列挙し、その中で最小の tts を求める手法を提案する。k+1 以下の PMC をすべて列挙することにより、k+1 以下の大きさの PMC で構築される極小コーダル化したグラフを BT アルゴリズムですべて調べ上げ、その中から、最小の tts をもつ極小コーダル化したグラフを構築することができる。これを treewidth 制約下での tts 最小化と今後呼称する。

本論文では、[10] の論文に基づき、treewidth の大きさを k としたときに、k+1 以下の大きさの PMC を列挙する。具体的には、m を 1 から、順々に大きくしていき、大きさ m の PMC を SAT を用いて列挙していく。大きさ m の PMC を列挙するたびに、1 から、m までの大きさの全ての PMC を用いて、G を極小コーダル化できるかを、上述した BT アルゴリズムで確かめる。具体的には G の minimal-fillin が構築できないとき、上述した BT アルゴリズムは∞を返すので、∞を返さないときに極小コーダル化したグラフを構築で判定する。本研究では、変形した BT アルゴリズムを内で、treewidth を計算する cliqueCost と mergeCost に tts の計算を加えることにより、tree width 制約下での tts 最小化を実現する。提案アルゴリズムを Algorithm3 に示した。Algorithm1 の cost を計算する複数の変数 dp を tw と tts にそれぞれわけ、それぞれを dpTw,dpTts とする。それぞれで、動的計画法を行う。提案アルゴリズムで求めたいのは、tts を最小化する極小コーダル化したグラフである。よって、optChoice には、tts を最小にする PMC のみを加えればよい。より、13 行目から 15 行目で、dpTw の値を更新し、極小コーダル化できるかを調べる。16 行目から 19 行目で、dpTts の値を更新し、tts が最小となる PMC を判別する。

また、前処理として、[13] より、グラフを clique separator で分割することにより、探索空間を削減する。

Algorithm 5 PROPOSED algorithm

1: $PROPOSEDALGORITHM(G, cliqueCost, mergeCost)$
2: insert lines 2-12 in Algorithm1
3: $dpTw[(S,C)] \leftarrow \infty forall(S,C)$
4: $dpTts[(S,C)] \leftarrow \infty forall(S,C)$
5: **for each** $(\Omega, S, C) \in T$ **do**
6: $costTw \leftarrow cliqueCostTw(\Omega, S)$
7: $costTts \leftarrow cliqueCostTts(\Omega, S)$
8: **for each** $C' \in C(G[C \setminus \Omega])$ **do**
9: $S' \leftarrow N(C')$
10: $costTw \leftarrow mergeCostTw(costTw, dpTw[(S', C')])$
11: $costTts \leftarrow mergeCostTts(costTts, dpTts[(S', C')])$
12: **end for**
13: **if** $costTw < dpTw[S,C]$ **then**
14: $dpTw[(S,C)] \leftarrow costTw$
15: **end if**
16: **if** $costTts < dpTts[S,C]$ **then**
17: $dpTts[(S,C)] \leftarrow costTts$
18: $optChoice[(S,C)] \leftarrow \Omega$
19: **end if**
20: **end for**
21: **return** $dpTts[(\emptyset, V(G))], dpTw[(\emptyset, V(G))]$

7 実験

提案手法の効果を確認するため、以下の実験を行った。制限時間 3 時間の元で、既存手法の厳密な tts 最小化の方法とハイブリッド法と提案の tree width 制約下での tts 最小化手法の実行時間を 17 種類のベイジアンネットワークで比較した。ベイジアンネットワークレポジトリは http://www.bnlearn.com/bnrepository/にあるものを持ちいた。実験は一つのベイジアンネットワークにつき、10 回ずつ行いその平均をとった。また、探索する PMC の減少率を確認するため、既存手法と提案手法の PMC の数を記載し、tts の大きさを比べた。実験の結果を表 1 にまとめた。表 1 の各列について説明する。ベイジアンネットワークの列は、使用したベイジアンネットワークの名前を表している。V,E はそれぞれ、モラル化されたベイジアンネットワークの頂点数と辺の数を表している。tts の行では、厳密手法と提案手法を 10 回ずつ行いその平均を、ハイブリッド法は 1000 回行い、その最小の tts と平均の tts をだした。時間 [s] では、それぞれ、厳密手法と提案手法を 10 回ずつ行いその平均を、ハイブリッド法は 1000 回行い、その平均時間を出した。PMC の数では、厳密手法と提案手法で列挙する PMC の数を出した。

表 1: 実験結果

				tts				時間 [s]			PMC の数	
ベイジアンネットワーク	$\|V\|$	$\|E\|$	tw	厳密手法	ハイブリッド法平均	ハイブリッド法最小	提案手法	厳密手法	ハイブリッド法	提案手法	厳密手法	提案手法
alarm	37	65	4	996	1072	1038	**996**	**0**	0.003	**0**	43	**43**
barley	48	126	7	17140796	2434804	23644556	**17140796**	0.774	**0.009**	1.416	10250	**1337**
child	20	30	3	642	704	**642**	**642**	**0**	**0**	**0**	642	**24**
hailfinder	56	99	4	9406	9958	9706	**9406**	0.03	**0.008**	0.054	1340	**200**
insurance	27	70	8	23880	48669	46872	**23880**	**0**	0.003	0.038	362	**75**
mildew	35	80	4	3400464	11318387	4261796	**3400464**	0.042	**0.003**	0.056	1695	**190**
pathfinder	109	208	6	182641	**182641**	**182641**	**182641**	**0**	0.006	0.15	126	**105**
water	32	123	9	3028305	3657180	3657180	**3028305**	0.055	**0.006**	0.15	1242	**139**
win95pts	76	225	8	2684	**2684**	**2684**	**2684**	0.085	**0.027**	0.225	1278	**197**
andes	223	626	?	?	389810	389752	?	>7200	**0.253**	>7200	?	?
diabetes	414	819	4	?	13738043	12377488	**9825960**	>7200	**0.491**	721.951	?	**3077**
pigs	442	806	13	?	709655	709344	?	>7200	**0.612**	>7200	?	?
link	715	1738	13	?	69562569	37594506	?	>7200	**2.669**	>7200	?	?
munin1	190	366	10	?	376853018	183627540	**65497886**	>7200	**0.118**	5483.57	?	**115320**
munin2	1004	1662	7	7	52336655	4900627	**2049942**	>7200	**2.77**	>7200	?	**622904**
munin3	1044	1745	7	?	3776234	3553553	**3078488**	>7200	**3.336**	6413.43	?	**49827**
munin4	1042	1843	8	?	27922023	27595098	**8859834**	>7200	**3.646**	>7200	?	**200558**

近似手法について、頂点数 100 以下のベイジアンネットワークでは、近似手法の方が、厳密手法よりも実行時間が増加していることが分かる。これは、大きさ k の PMC を列挙する際に PMC の条件を SAT に変換するため、そのオーバーヘッドが大きいためだと考えられる。また、tts は厳密かいが計算できるベイジアンネットワークでは、厳密手法と提案手法が一致していることが分かる。一方頂点数が 150 を超えるベイジアンネットワークでは、提

案した近似手法のみが制限時間 2 時間の下で、実行できた。また、厳密手法の場合、制限時間なしで実行したところ、メモリが足らずに計算が終わらなかった。そのため、今回の手法は、空間計算量の面で厳密手法より、優位な点があると言える。このことは、提案手法と既存手法の PMC の数を比べることにより、最大で child のベイジアンネットワークにおいて、96% の PMC の削減ができていることから分かる。また、ハイブリッド法で出せる tts と比べると提案手法はすべてのベイジアンネットワークで小さい tts を求められていることが分かる。

8 むすび

今回の提案では、近似手法であったが、実験結果より、treewidth 制約下での tts が最小の tts に一致する場合が非常に多いことが分かった。そのため、今回の手法で計算した tts を最小の tts であると、保証するアルゴリズムを考えたい。保証できないことが分かった場合は、treewidth 制約下で出した tts よりも小さい tts をもつ極小コーダル化したグラフに含まれる、treewidth の大きさを k とした時、k+2 以上の頂点を持つ PMC の条件を考えることによって、すべての PMC を列挙する必要がなくなると考えている。一番簡単な方法としては、treewidth 制約下で出した tts よりも大きい ts を持つ PMC は列挙する必要はないことが分かる。また、今回の実験では、treewidth を k としたとき、k+1 以下の PMC を SAT による方法で列挙したが、他の方法で列挙する場合も考えている。具体的には、連結無向グラフの treewidth の大きさが k かどうかを調べる判定アルゴリズムは、k+1 以下の PMC を列挙する時間よりもはるかに小さいことが分かっているので、グラフの treewidth を決定してから、k+1 以下のもののみを列挙するように、[11] のアルゴリズムを改良する方法を考えている。

謝辞

本論文の作成に当たり、熱心に指導してくださった、植野真臣教授、岡本吉央教授、川野秀一准教授に感謝いたします。

参考文献

[1] G.F. Cooper, The computational complexity of probabilistic inference using Bayesian belief networks, Artif. Intell. 42 (2–3) (March 1990) 393–405

[2] S.L. Lauritzen, D.J. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems, J. R. Stat. Soc., Ser. B 50 (2) (1988) 157–224

[3] F. Jensen, S. Lauritzen, K. Olesen, Bayesian updating in causal probabilistic networks by local computations, CSQ, Comput. Stat. Q. 4 (1990) 269–282

[4] P.P. Shenoy, G. Shafer, Axioms for probability and belief-function propagation, in: Uncertainty in Artificial Intelligence, Elsevier, Amsterdam, 1990, pp.169-198

[5] Vincent Bouchitté and Ioan Todinca.2001.Treewidth and minimum fillin:Grouping the minimal separators. SIAM J.Comput.31,1(2001),212–232.

[6] Tuukka Korhonen, Jeremias Berg, and Matti Jarvisalo. Solving graph problems via potential maximal cliques: An experimental evaluation of the Bouchitte-Todinca al-gorithm. ACM J. Exp. Alg.,24(1:9), 2019.

[7] Fedor V. Fomin, Ioan Todinca, and Yngve Villanger. 2015. Large Induced Subgraphs via Triangulations and CMSO. SIAM J. Comput. 44, 1 (2015), 54–87. https://doi.org/10.1137/140964801

[8] Chao Li and Maomi Ueno. 2017. An extended depth-first search algorithm for optimal triangulation of Bayesian networks. International Journal of Approximate Reasoning 80(2017),294–312.

[9] Hisao Tamaki. 2017. Positive-instance driven dynamic programming for treewidth. In Proc. ESA (Leibniz International Proceedings in Informatics (LIPIcs)), Vol. 87. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 68:1–68:13.

[10] T. Korhonen, J. Berg, and M. J ¨ arvisalo. Enumerating potential maximal cliques via SAT and ASP. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, pages 1116–1122. International Joint Conferences on Artificial Intelligence Organization, 2019.

[11] Vincent Bouchitté and Ioan Todinca. 2002. Listing all potential maximal cliques of a graph. Theoretical Computer Science 276,1(2002),17–32.

[12] 植野真臣 (2013) ベイジアンネットワーク　コロナ社

[13] RobertE.Tarjan.1985. Decompositionbycliqueseparators. Discrete Mathematics 55,2(1985),221–232.

[14] F.V. Jensen, F. Jensen, Optimal junction trees, in: UAI, 1994, pp. 360–366.

[15] D. Roth, On the hardness of approximate reasoning, Artif. Intell. 82 (1996) 273–302, http://dx.doi.org/10.1016/0004-3702(94)00092-1.

[16] Shan Gao and Masakazu Ishihata and Shin-ichi Minato. 2018. Separate Compilation of Bayesian Networks for Efficient Exact Inference. 人工知能学会論文誌, vol33,no.6,pp.A-I35_1-15.

[17] S. Minato, K. Satoh, and T. Sato, “Compiling bayesian networks by symbolic probability calculation based on zero-suppressed BDDs,” Proc. 20th International Joint Conference of Artificial Intelligence (IJCAI-2007), pp.2550–2555, 2007.

[18] NP-Complete,” in Learning from Data: Artificial Intelligence and Statistics, vol.V, pp.121–130, Springer, 1996.

[19] R.G. Cowell, “Efficient maximum likelihood pedigree reconstruction,” Theoretical Population Biology, vol.76, no.4, pp.285–291, Dec. 2009.

[20] M. Koivisto and K. Sood, “Exact Bayesian structure discovery in Bayesian networks,” J. Machine Learning Research, vol.5, pp.549–573, Dec. 2004

[21] A. Singh and A. Moore, “Finding optimal Bayesian networks by dynamic programming,” Technical report, Carnegie Mellon University, pp.1–16, June 2005

[22] T. Silander and P. Myllymaki, “A simple approach for finding the globally optimal Bayesian network structure,” Proc. Uncertainty in Artificial Intelligence (UAI), pp.445–452, 2006.

[23] B. Malone, C. Yuan, and E.A. Hansen, “Memoryefficient dynamic programming for learning optimal Bayesian networks,” Proc. 25th AAAI Conference, pp.1057–1062, 2011.

[24] C. Yuan, B. Malone, and W. Xiaojian, “Learning optimal Bayesian networks using A* search,” Int. Joint Conference on Artificial Intelligence (IJCAI), pp.2186–2191, 2011.

[25] J. Cussens, “Bayesian network learning with cutting planes,” Proc. Uncertainty in Artificial Intelligence (UAI), pp.153–160, 2011.

[26] K. Natori, M. Uto, Y. Nishiyama, S. Kawano, and M. Ueno, “Constraintbased learning Bayesian networks using Bayes factor,” Advanced Methodologies for Bayesian Networks (AMBN) 2015, vol.LNAI 9505, pp.15–31, Springer, 2015.

[27] K. Natori, M. Uto, and M. Ueno, “Consistent learning Bayesian networks with thousands of variables,” Advanced Methodologies for Bayesian Networks (Proc. Machine Learning Research), vol.73, pp.57–68, 2017.

[28] 名取和樹，宇都雅輝，植野真臣，“Bayes factor を用いた RAI アルゴリズムによる大規模ベイジアンネットワーク学習，” 信学論（D），vol.J101-D, no.5, pp.754–768, May 2018.

[29] FedorV.FominandYngveVillanger.2008. Treewidthcomputationandextremalcombinatorics.InProc.ICALP (Lecture Notes in Computer Science),Vol.5125.Springer,210–221

[30] A. Darwiche, Modeling and Reasoning with Bayesian Networks, Cambridge University Press, 2009.