

修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院 情報理工学研究科 情報・ネットワーク工学専攻 博士前期課程		
氏 名	青木 健登	学籍番号	2231001
論 文 題 目	条件付き独立テスト二分木による ベイジアンネットワーク分類器の k-best モデル平均		
<p style="text-align: center;">要 旨</p> <p>Deep Learning や Random Forest などに代わる予測器としてベイジアンネットワーク分類器 (BNC) が注目されている。Sugahara ら (2021) は、少数データの場合でも分類精度が安定する BNC として、最大の周辺尤度をもつ Augmented Naive Bayes (ANB) を厳密学習する手法を提案した。しかし、彼らの手法は計算コストが高く数十変数の ANB 学習が限界である。一方、因果モデルの分野では、計算効率が高い構造学習法として、条件付き独立テスト (CI テスト) とエッジの方向付けにより構造を学習する制約ベースアプローチが知られている。近年、Sugahara ら (2022) は制約ベースアプローチを ANB の学習に拡張することで、1000 変数の大規模 ANB を学習できる手法を提案した。CI テストでは、2 ノード間が独立・従属モデルの周辺尤度の比による Bayes factor の対数 (対数 Bayes factor) の値が 0 以上か否かでエッジの有無を判定する。対数 Bayes factor の値が 0 から乖離するほどモデル選択の信頼性は向上するが、0 付近の場合は信頼性に乏しい。一方、先行研究では信頼性を考慮せず構造学習を行っており、改善の余地がある。そこで本論では、対数 Bayes factor の値が 0 近傍の閉区間に含まれる CI テストに対応するエッジの有無についてモデル平均を行う手法を提案する。全ての候補構造を用いたモデル平均は計算が困難であるため、周辺尤度上位 k 個の構造 (k-best) を探索してモデル平均を計算する。しかし、k-best の探索は全ての候補構造を比較する必要があり計算コストが高い。そこで本論では場合分け二分木の根から葉までのパスにより 1 つの有向グラフを表現し、周辺尤度上位 k 個のパス (ANB) を有効な枝刈りにより効率的に探索する手法を提案する。リポジトリ・データを用いた実験により、大規模変数データに対して提案手法が従来手法より有意に分類精度が高いことを示す。</p>			

令和五年度 修士論文

条件付き独立テスト二分木による
ベイジアンネットワーク分類器の k -best モデル平均

電気通信大学 大学院情報理工学研究科
情報・ネットワーク工学専攻 情報数理工学プログラム
学籍番号 2231001
青木 健登

主任指導教員 植野真臣 教授
指導教員 宇都雅輝 准教授

提出年月日 2024 年 1 月 29 日

目次

1	まえがき	3
2	ベイジアンネットワーク分類器	6
2.1	厳密解探索アプローチ	6
2.2	制約ベースアプローチ	8
3	信頼性が低い CI テストに対応するエッジの有無についてのモデル平均	10
3.1	CI テストの信頼性を考慮したエッジ表現	10
3.2	k -best の探索のための場合分け二分木	11
3.3	場合分け二分木のパスのスコア	12
3.4	k -best の探索のための場合分け二分木の枝刈り	14
3.5	k -best の探索のための場合分け二分木の深さ優先分枝限定法	17
4	評価実験	21
4.1	小規模変数データセットを用いた評価実験	22
4.2	大規模変数データセットを用いた評価実験	26
5	むすび	29

1 まえがき

分類器は Deep Learning アプローチと確率アプローチに大別される。前者は分類精度が高く、大規模変数にも適用できるという利点を持つが、数学的な性質が明らかにされておらず、モデルの解釈性も低いという問題がある。後者は真の分類確率に漸近収束する推定値を得ることができ、解釈性も高いという利点を持つが、分類精度が Deep Learning に劣っており、計算量も大きいという問題がある。確率アプローチでは、Naive Bayes[1] やロジスティック回帰が知られているが、説明変数間の関係を十分に表現できないためにあまり精度が高くない。Naive Bayes を一般化し、全ての変数間の条件付き独立性を表現するベイジアンネットワーク分類器 (Bayesian Network Classifier: BNC) [2] は確率アプローチの中でも解釈性が最も高いモデルの一つである。トップカンファレンス AAAI の最新研究 [3] では、BNC の精度向上が報告されており、Deep Learning や Random Forest などに代わる予測器として注目されている。

BNC は、離散確率変数をノードとし、ノード間の条件付き従属関係を非循環有向グラフ (Directed Acyclic Graph: DAG) で表した確率的グラフィカルモデルである。BNC の DAG 構造は一般にデータから推定する必要があり、この問題を構造学習と呼ぶ。構造学習では、最大の周辺尤度を持つ構造を探索する厳密解探索アプローチが従来から行われてきた [4, 5]。Sugahara ら [6, 7] は、データが少ない場合、このような BNC について目的変数の親変数が多くなってしまい、パラメータ学習に用いるデータがスパースになり、分類精度が著しく低下してしまうと指摘した。この問題を解決するために、彼らは目的変数が全ての説明変数の子にもつ Augmented Naive Bayes (ANB) [2] を厳密学習する方法を提案した。これにより、データが少ない場合でも分類精度を改善できることを示した。しかし、そもそも周辺尤度による厳密学習は漸近的に最適な構造に収束するが、データ数が少ない場合、推定される構造の事後確率が低くなり、分類精度が劣ってしまう。このように各構造の事後確率が低い場合は、モデル平均を行うことにより推論精度が向上できることが知られている [8]。しかし、モデル平均を BNC にそのまま適用すると、学習データから考えられる全ての候補構造について計算する必要があるため、現実的に計算が困難となる。この計算困難性を解決する先行研究として、学習スコアが最適な順に k 個の構造を選択し、得られた k 個の構造のモデル平均を行う k -best 法が知られている [9, 10]。さらに、Sugahara ら [11, 12] はアンサンブル法と k -best 法を組み合わせるモデル平均を行う BNC を提案し、有効性を報告している。しかし、厳密解探索アプローチは数十変数の ANB の学習が限界であるため大規模変数には適用できない。

一方、因果モデルの分野では、計算効率が高い構造学習法として制約ベースアプローチが知られている。制約ベースアプローチは、2ノード間の条件付き独立テスト (Conditional Independence Test: CI テスト) により得られる無向グラフに、オリエンテーションルール [13] による辺の方向付けを行うことで DAG を学習する。制約ベースアプローチでは、PC アルゴリズム [14], MMHC アルゴリズム [15], RAI アルゴリズム [16] などが提案されている。従来の制約ベースアプローチは漸近一致性をもたないことが問題であったが、Natori らは RAI アルゴリズムの CI テストに Bayes factor を用いることで漸近一致性を有する大規模ネットワーク構造学習を実現している [17, 18]。そして近年、Sugahara ら [19, 20] は、Natori らの手法を ANB 構造を仮定した BNC の学習に拡張し、1000 変数程度の大規模 ANB の厳密学習を実現している。

Sugahara ら [19, 20] の手法で用いる Bayes factor は、二つのモデルの周辺尤度の比を求めることで厳密なモデル選択が可能な手法である。Bayes factor を用いた CI テストでは、2ノード間が独立・従属モデルの周辺尤度の比による Bayes factor の対数 (対数 Bayes factor) が 0 以上か否かでエッジの有無を判定する。対数 Bayes factor の値が 0 から乖離するほどモデル選択の信頼性は向上する [21] が、0 付近の場合は信頼性は乏しい。一方、Sugahara らは CI テストの信頼性を考慮せずに構造学習を行っており、改善の余地がある。

以上の問題を考え、本論では対数 Bayes factor の値が予め設定した 0 近傍の閉区間 $[-\epsilon, \epsilon]$ に含まれる CI テストは信頼性が低いと判定し、その信頼性の低い CI テストに対応するエッジが存在する場合と存在しない場合の全てのパターンについてのモデル平均を行う手法を提案する。変数数が大きくなると対数 Bayes factor の値が閉区間 $[-\epsilon, \epsilon]$ に含まれるエッジの組合せ数も指数的に増加するため、全ての候補構造を用いたモデル平均は困難である。この問題を解決するため、提案手法では候補構造から周辺尤度上位 k 個の構造 (k -best) を選択してモデル平均を行う。しかし、 k -best の探索は全ての候補構造を比較する必要があるため計算コストが高い。そこで本論では計算コストを削減する枝刈りを導入するため、場合分け二分木を用いた k -best の探索を提案する。この場合分け二分木では、対数 Bayes factor の値が閉区間 $[-\epsilon, \epsilon]$ に含まれるエッジを節点とし、各節点ごとにそのエッジが存在しないことを表す 0-枝と存在することを表す 1-枝で二分する。本論では、この場合分け二分木の根から葉までのパスにより 1 つの有向グラフを表現し、周辺尤度上位 k 個のパス (ANB 構造) を有効な枝刈りにより効率的に探索する手法を提案する。

提案手法は計算コストを効率的に削減できるため、大規模 ANB についてモデル平均が計算できるようになり、分類精度を向上できる。リポジトリ・データを用いた実験により、

大規模変数データに対して提案手法が従来手法より有意に分類精度が高いことを示す.

2 ベイジアンネットワーク分類器

ベイジアンネットワークは、確率変数をノードとし、ノード間の条件付き従属関係を非循環有向グラフ (Directed Acyclic Graph: DAG) で表した確率的グラフィカルモデルである。ベイジアンネットワークにおける一つのノードを目的変数とし、その他のノードを説明変数とした分類器をベイジアンネットワーク分類器 (Bayesian Network Classifier: BNC) という [2]。今、 $n + 1$ 個の離散確率変数集合 $V = \{X_0, X_1, \dots, X_n\}$ において、各変数 X_i は r_i 個の状態集合 $\{1, \dots, r_i\}$ から一つの値をとるとし、各変数 X_i が値 k をとるとき、 $X_i = k$ と書く。また、BNC の構造を G とし、 G における変数 X_i の親変数集合を $Pa(X_i, G)$ とする。さらに、 θ_{ijk} を $Pa(X_i, G)$ が j 番目のパターンをとったとき ($Pa(X_i, G) = j$ と書く) に $X_i = k$ となる条件付き確率 $P(X_i = k \mid Pa(X_i, G) = j, G)$ を示すパラメータとし、 $\Theta = \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \{\theta_{ijk}\}$ とする。BNC では、次式のように同時確率分布 $P(X_0, X_1, \dots, X_n \mid G, \Theta)$ を各変数の条件付き確率の積に分解して表せる。

$$P(X_0, X_1, \dots, X_n \mid G, \Theta) = \prod_{i=0}^n P(X_i \mid Pa(X_i, G), G, \Theta).$$

パラメータ推定値として、期待事後確率推定値 (Expected a Posteriori: EAP) が最もよく用いられる。今、全変数に値が割り当てられたサンプルからなる学習データを D と表す。パラメータの事前分布にディリクレ分布を仮定すると、EAP は次式で与えられる。

$$\hat{\theta}_{ijk} = \frac{\alpha_{ijk} + N_{ijk}}{\alpha_{ij} + N_{ij}}.$$

ここで、 N_{ijk} は D において $X_i = k$ かつ $Pa(X_i, G) = j$ となる頻度を表し、 α_{ijk} はディリクレ事前分布のハイパーパラメータを表す。また、 $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$ 、 $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$ である。今、 X_0 を目的変数とし、 X_1, \dots, X_n を説明変数とする。BNC では説明変数のデータ $\mathbf{x} = \langle x_1, \dots, x_n \rangle$ が与えられたとき、目的変数の推定値 \hat{c} は以下のように得られる。

$$\hat{c} = \arg \max_{c \in \{1, \dots, r_0\}} P(c \mid \mathbf{x}, G, \Theta, D). \quad (1)$$

2.1 厳密解探索アプローチ

BNC の構造はデータから推定する必要があり、この問題を構造学習と呼ぶ。構造学習では、候補構造から最大の周辺尤度を持つ構造を探索する厳密解探索アプローチが従来か

ら行われてきた [4, 5]. パラメータの事前分布にディリクレ分布を仮定すると, 周辺尤度は次のように閉形式で表される.

$$P(D | G) = \prod_{i=0}^n \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}. \quad (2)$$

式 (2) の周辺尤度を Bayesian Dirichlet (BD) と呼ぶ. Buntine[22] はハイパーパラメータを $\alpha_{ijk} = \alpha/(r_i q_i)$ とした Bayesian Dirichlet equivalent uniform (BDeu) と呼ばれるスコアを提案しており, 最もよく用いられるスコアである. ここで, α は Equivalent Sample Size (ESS) と呼ばれる事前知識の重みを示す疑似サンプルである. BDeu の対数は次の性質を満たす.

$$\log BDeu(G) = \sum_{i=0}^n \text{Score}_i(Pa(X_i, G)). \quad (3)$$

ここで, $\text{Score}_i(Pa(X_i, G))$ は変数 X_i と $Pa(X_i, G)$ のみに依存する関数であり, ローカルスコアと呼ばれる. $\text{Score}_i(Pa(X_i, G))$ は次式で表せる.

$$\begin{aligned} & \text{Score}_i(Pa(X_i, G)) \\ &= \sum_{j=1}^{q_i} \left(\log \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} + \sum_{k=1}^{r_i} \log \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \right). \end{aligned} \quad (4)$$

一般的なベイジアンネットワークを用いた BNC を General Bayesian Network (GBN) と呼ぶ. Sugahara ら [6, 7] は, データが少ないとき生成モデルの GBN では目的変数の親変数が多くなってしまい, パラメータ学習に用いるデータがスパースになるため, 分類精度が著しく低下してしまうことを実験で示した. この問題を解決するため, 彼らは目的変数が全ての説明変数の親変数になる Augmented Naive Bayes (ANB) [2] を生成モデルとして厳密学習する手法を提案した. 彼らの手法はデータが少ない場合でも安定した分類精度を得ることができ, GBN や識別モデルの BNC より有意に分類精度が高いことを示した.

しかし, そもそも周辺尤度による厳密学習は漸近的に最適な構造に収束するが, データが少ない場合, 推定される構造の事後確率が低くなり, 分類精度が劣ってしまう. このように各構造の事後確率が低い場合は, モデル平均を行うことにより推論精度が向上できる [8]. BNC におけるモデル平均では, 考えられる全ての候補構造について平均して次式で \hat{c} を推定する.

$$\hat{c} = \arg \max_{c \in \{1, \dots, r_0\}} \sum_{G \in \mathcal{G}} p(D|G)p(c|\mathbf{x}, G, D).$$

ここで、 G は候補構造の集合である。しかし、 G の構造数は、学習データの変数数に対して超指数的に増加するため、計算時間が現実的ではない。この計算困難性を解決する先行研究として、学習スコアが最適な順に選択した k 個の構造についてのモデル平均を行う k -best 法が知られている [9, 10]。しかし、 k -best 法は類似の構造を選択する傾向にあり、推論精度の向上が限定的となってしまう。そこで Sugahara ら [11, 12] は、得られる構造が類似しないように、非復元抽出法により元のデータから生成した T 個の各データについて k -best 法で k 個の構造を選択し、得られた $T \times k$ 個の構造についてのモデル平均を行う BNC を提案し、有効性を報告している。しかし、厳密解探索アプローチは数十変数の構造学習が限界であるため、大規模変数には適用できない。

2.2 制約ベースアプローチ

因果モデルの分野では、計算コストを大幅に削減できる構造学習法として制約ベースアプローチが知られている。制約ベースアプローチ学習は、2 ノード間の条件付き独立テスト (Conditional Independence Test: CI テスト) により得られる無向グラフに、オリエンテーションルール [13] による辺の方向付けを行うことで DAG を学習する。制約ベースアプローチとして、PC アルゴリズム [14]、MMHC アルゴリズム [15]、RAI アルゴリズム [16] などが提案されてきた。従来の制約ベースアプローチは漸近一致性をもたないことが問題であったが、Natori ら [17, 18] は、RAI アルゴリズムに Bayes factor を用いた CI テスト [23] を組み込むことで、漸近一致性を有する大規模構造学習を実現した。

Sugahara ら [19, 20] は、Natori らの手法の初期構造や CI テストの実行範囲に制約を設けることで ANB を学習する手法 (以下、RAI-ANB と呼ぶ) を提案している。RAI-ANB の基本的な動作は以下のとおりである。今、グラフを $G = (V, E)$ と表し、 V 、 E はそれぞれ G に含まれる変数集合、エッジ集合を表す。ここで、 G は有向エッジと無向エッジを併せもつとする。また、 $G_{ex} = (V_{ex}, E_{ex})$ を RAI アルゴリズムによって分割された部分グラフとする。 X_0 を目的変数とする。

1. 説明変数集合からなる完全無向グラフとデータ D を入力する。
2. 各次数の CI テストにおいて、Bayes factor を用いた CI テスト [23] を行い、 X と Y が条件付き独立と判定されるとき、 XY 間のエッジを削除する。ANB では目的変数と説明変数は接続されているため、CI テストの条件部に目的変数 X_0 を必ず含める。
3. (2) で得られたグラフに対してオリエンテーションルール [13] を適用して方向付け



図 1: 独立モデルと従属モデルの例

を行う。

4. 方向付けの結果から部分グラフ G_{ex} に分割する。
5. 各部分グラフで再帰的に RAI-ANB を呼び出す。

この手続きにより，説明変数集合から構成される DAG が出力される．最後に，目的変数から全ての説明変数へエッジを接続することで，ANB 構造を得ることができる．RAI-ANB は厳密解探索アプローチで学習できない 1000 変数の大規模 ANB を学習でき，厳密学習した ANB に劣らない分類精度を達成している．

RAI-ANB で用いる Bayes factor は，二つのモデルの周辺尤度の比を求めることで厳密なモデル選択が可能な手法である．例として，ベイジアンネットワークの二変数 X, Y において X と Y 間について各変数の共通親変数集合を Z としたときの従属モデルを G_1 ，独立モデルを G_2 とし，それぞれ図 1 の (a)，(b) に示す．このときの Bayes factor を $BF(X, Y|Z)$ とすると，対数 Bayes factor は次のように表せる．

$$\log BF(X, Y|Z) = \log BDeu(G_1) - \log BDeu(G_2). \quad (5)$$

Bayes factor を用いた CI テスト [23] では，式 (5) の対数 Bayes factor が負のとき X と Y が Z を所与として条件付き独立であると判定する．対数 Bayes factor の値が 0 から乖離するほどモデル選択の信頼性は向上する [21] が，0 付近の場合は信頼性は乏しい．一方,RAI-ANB では CI テストの信頼性を考慮せずに構造学習を行っており，改善の余地がある．

3 信頼性が低い CI テストに対応するエッジの有無についてのモデル平均

本論では、対数 Bayes factor の値が 0 近傍で信頼性が低い CI テストに対応するエッジが存在する場合と存在しない場合の全てのパターンについてのモデル平均を行う手法を提案する。具体的には、まず RAI-ANB[19, 20] による構造学習を行う。このとき、対数 Bayes factor の値が 0 近傍の閉区間に含まれる CI テストに対応するエッジをメモリに記録する。次に、それらのエッジが RAI-ANB の学習結果の構造に存在するかしないかの全てのパターンから、周辺尤度上位 k 個の構造を選択してモデル平均を行う。

3.1 CI テストの信頼性を考慮したエッジ表現

Kass ら [21] は、対数 Bayes factor の値が 0 から乖離するほどモデル選択の信頼性が向上すると解釈している。本論では彼らの解釈を CI テストに適用し、対数 Bayes factor の値により CI テストの信頼性を評価する。閾値を $\varepsilon > 0$ とすると、式 (5) の対数 Bayes factor に対して次式が成り立つとき、変数集合 \mathbf{Z} を所与とした 2 変数 X, Y 間の CI テストの信頼性が低いと判定する。

$$\log BF(X, Y | \mathbf{Z}) \in [-\varepsilon, \varepsilon]. \quad (6)$$

提案手法では、まず RAI-ANB による構造学習を行い、次の条件のいずれかを満たすエッジをメモリに記録する。

- (a) RAI-ANB で削除されたエッジであり、条件付き独立と判定した CI テストで式 (6) が成立。
- (b) RAI-ANB で削除されなかったエッジであり、実施した少なくとも一つの CI テストで式 (6) が成立。
- (c) RAI-ANB で削除されなかったエッジであり、実施した全ての CI テストで式 (6) が不成立。

条件 (a) または (b) を満たすエッジを信頼性の低い CI テストに対応するエッジと判定し、その集合を $E_{\text{unreliable}}$ とおく。また、条件 (c) を満たすエッジを信頼性の高いエッジと判定し、その集合を E_0 と書き、DAG G_0 を $G_0 = (V, E_0)$ とおく。ここで、変数集合 $V = \{X_0, X_1, \dots, X_n\}$ とし、目的変数を X_0 とする。

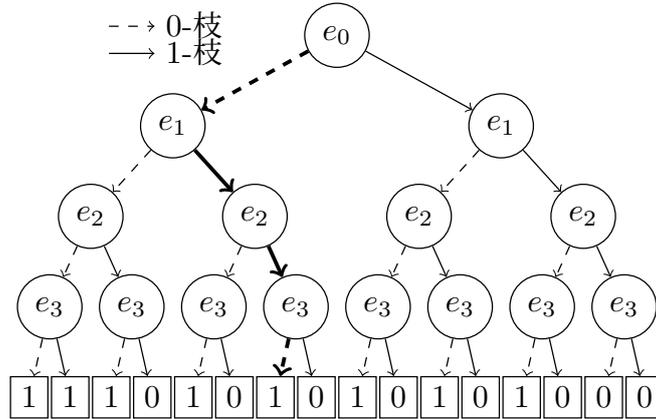


図 2: エッジの有無のパターンを表す場合分け二分木の例

3.2 k -best の探索のための場合分け二分木

提案手法では、信頼性の低いエッジの集合 $E_{\text{unreliable}}$ の各エッジが構造 G_0 に存在するかしないかの全てのパターンについてモデル平均を行い、分類精度を向上させる。変数数が大きくなると、このパターン数は指数的に増加するため、全ての候補構造を用いたモデル平均は困難である。そこで提案手法では k -best 法 [9, 10] のアイデアを適用し、候補構造から選んだ周辺尤度上位 k 個の構造（以下、 k -best）についてモデル平均を計算する。しかし、 k -best の探索には全ての候補構造を比較する必要があり計算コストが膨大になる。本論では、計算コストを削減する枝刈りを導入するために、 k -best の探索を場合分け二分木のスコア上位 k 個のパスの探索として定式化する。

場合分け二分木を用いるためには、木の節点となる変数の順序を決めなければならない。本論では $E_{\text{unreliable}}$ の各エッジを場合分け二分木の節点とするため、 $E_{\text{unreliable}}$ 上の順序 \prec が必要となる。ただし、ここでの議論では \prec は任意の順序でよい。順序 \prec を所与としたときの l 番目 ($l = 0, 1, \dots, L - 1$) のエッジを e_l とおく。ここで L は $E_{\text{unreliable}}$ のエッジ数である。本論では、エッジ e_l を場合分け二分木の深さ l の各節点とする。そして各節点ごとに、 e_l が存在しないことを表す 0-枝と存在することを表す 1-枝で二分する。このとき、場合分け二分木の根から葉までのパスは集合 E_{add} ($E_{\text{add}} \subseteq E_{\text{unreliable}}$) を表現する。 E_{add} を構造 $G_0 = (V, E_0)$ に追加するエッジの集合とみなすと、パスは有向グラフ $(V, E_0 \cup E_{\text{add}})$ に対応する。また、葉はその有向グラフが DAG であるとき 1、そうでないとき 0 とする。1 の値をもつ葉を葉 1 と呼ぶ。場合分け二分木の例を図 2 に示す。図 2 中に太線で示したパスは部分集合 $\{e_1, e_2\}$ を表現し、有向グラフ $(V, E_0 \cup \{e_1, e_2\})$ に対

応する。葉が 1 の値をもつためその有向グラフは DAG である。このように、候補構造の集合は場合分け二分木の根から葉 1 までのパスの集合として表現できる。

3.3 場合分け二分木のパスのスコア

提案手法では周辺尤度上位 k 個の構造を探索する。ここで、対数周辺尤度は式 (3) のように各変数 $X_i \in V$ のローカルスコアの和に分解できる。 X_i のローカルスコアは X_i と X_i の親変数集合のみに依存する。したがって、 X_i の親変数集合が全ての候補構造で等しい場合、 X_i のローカルスコアは全ての候補構造で等しくなる。よって、変数集合 V から、親変数集合が全ての候補構造で等しい変数を除いた集合を V' とおくと、 k -best を得るには V' の各変数のローカルスコアの総和 ($\log BDeu(G, V')$ とおく) が上位 k 個の構造 G を求めればよい。 V' の変数数を n' とおき、 $V' = \{X_1, X_2, \dots, X_{n'}\}$ とすると $\log BDeu(G, V')$ は次式で表せる。

$$\log BDeu(G, V') = \sum_{i=1}^{n'} \text{Score}_i(\text{Pa}(X_i, G)). \quad (7)$$

本論では、根から葉までのパスのスコアを、パスに対応する構造 G の $\log BDeu(G, V')$ とする。 $\log BDeu(G, V')$ は、式 (7) のように各 $X_i \in V'$ と X_i の親変数集合の組に依存する項に分解される。そのため、根から葉までのパスのスコアも、 X_i と X_i の親変数集合の組に対応するパスのスコアごとに分解できる。ここで、 X_i の親変数集合は X_i に接続する各エッジの有無に対応する。そのため、 X_i と X_i の親変数集合の組を、 X_i に接続する各エッジの有無を表すパスで表現したい。しかし、 $E_{\text{unreliable}}$ 上の順序 \prec が任意の順序である場合、このパスは存在する保証がない。

そこで本論では、以下で定義する辞書式順序を用いる。辞書式順序を定義するために、変数集合 V において $X_0 < X_1 < \dots < X_n$ と定義する。この $<$ は変数 X_i の添字の大小のみで決まる。この $<$ を用いて、任意の $X \leftarrow Y, Z \leftarrow W \in E_{\text{unreliable}}$ に対して

$$X \leftarrow Y \prec Z \leftarrow W \Leftrightarrow X < Z \vee (X = Z \wedge Y < W)$$

と定義する。 \prec を $E_{\text{unreliable}}$ 上の辞書式順序という。例えば $X_1 \leftarrow X_3 \prec X_1 \leftarrow X_4 \prec X_2 \leftarrow X_1$ である。

辞書式順序を用いる場合の、 X_i に接続する各エッジの有無を表すパスの例を示す。二変数 $X_i, X_j \in V$ に対して、エッジ $X_i \leftarrow X_j$ を $e_{i,j}$ と表すことにする。辞書式順序を用いると、エッジ $e_{1,2}, e_{1,3}, e_{2,3}, e_{2,4}$ を節点とする場合分け二分木は図 3 のようになる。図

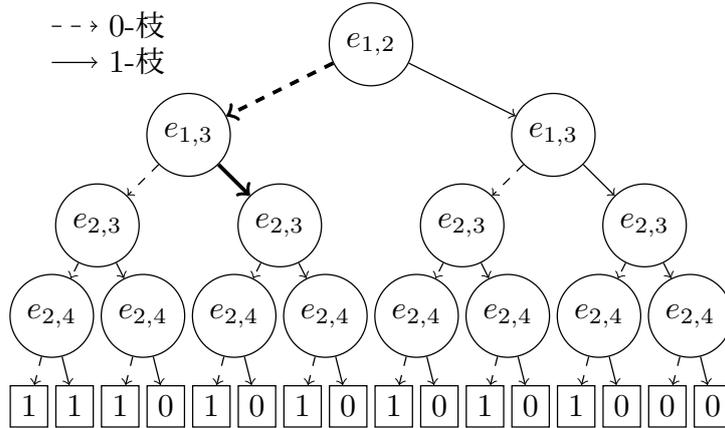


図 3: 辞書式順序を用いるときの場合分け二分木の例 ($e_{1,2} = X_1 \leftarrow X_2, e_{1,3} = X_1 \leftarrow X_3, e_{2,3} = X_2 \leftarrow X_3, e_{2,4} = X_2 \leftarrow X_4$)

3 より, X_1 に接続するエッジ $e_{1,2}, e_{1,3}$ の有無が, 節点 $e_{1,2}$ から節点 $e_{2,3}$ までのパスで表されることがわかる. また, X_2 に接続するエッジ $e_{2,3}, e_{2,4}$ の有無が, 節点 $e_{2,3}$ から葉までのパスで表されることがわかる.

このように, 辞書式順序を用いることで, 根から葉までのパスを X_i に接続する各エッジの有無を表すパスごとに分解できる. 本論では, これらのパスを用いて変数 X_i と X_i の親変数集合の組を表現する. 具体的には, エッジ $e_{i,j}$ が存在することを表すパスにより, 構造 G_0 にエッジ $e_{i,j}$ を追加した場合の X_i の親変数集合を表現する. 例を図 4 に示す. この図は, 図 3 における節点 $e_{1,2}$ から節点 $e_{2,3}$ までの各パスと親変数集合の対応を表している. 例えば, 図 3 に太線で示したパスは $e_{1,2}$ が存在せず $e_{1,3}$ が存在することを表すため, 構造 G_0 にエッジ $e_{1,3}$ を追加した場合の X_1 の親変数集合 $Pa(X_1, G_0) \cup \{X_3\}$ に対応する.

このように, 根から葉までのパスは, 変数 X_i と X_i の親変数集合の組に対応するパスごとに分解できる. 次に, この分解を用いてパスのスコアを計算する方法を説明する. はじめに, X_i に接続するエッジのうち, 場合分け二分木の節点として深さが最小であるものを e_{i,j_{\min}^i} , 深さが最大であるものを e_{i,j_{\max}^i} とおく. 例えば, 図 3 において $e_{1,j_{\min}^1} = e_{1,2}, e_{1,j_{\max}^1} = e_{1,3}, e_{2,j_{\min}^2} = e_{2,3}, e_{2,j_{\max}^2} = e_{2,4}$ である. このとき, 節点 e_{i,j_{\min}^i} から節点 e_{i,j_{\max}^i} の子節点までのパスは, 変数 X_i に接続する各エッジの有無を表すため, X_i と X_i の親変数集合の組に対応する. よって, 節点 e_{i,j_{\min}^i} から節点 e_{i,j_{\max}^i} の子節点までのパスを辿ることで, 式 (4) で表される X_i のローカルスコアを計算できる.

このことを利用して, 次のようにスコアを計算する. まず, 根 (節点 e_{1,j_{\min}^1}) から子節点を展開していき, e_{1,j_{\max}^1} の子節点を展開する時点で変数 X_1 のローカルスコアを計算す

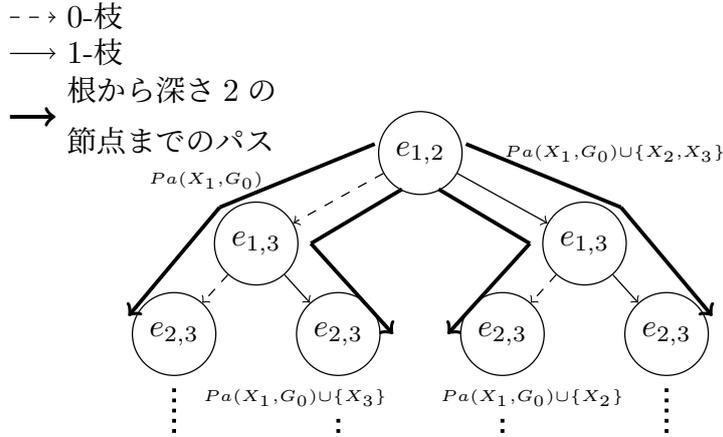


図 4: パスに対応する親変数集合の例 ($e_{1,2} = X_1 \leftarrow X_2, e_{1,3} = X_1 \leftarrow X_3$)

る．次に，同様に節点 e_{2,j_{\min}^2} から子節点を展開していき， e_{2,j_{\max}^2} の子節点を展開する時点で変数 X_2 のローカルスコアを計算する．以降，1 の値をもつ葉を展開するまで同様に探索とローカルスコアの計算を行う．葉を展開するとき，各変数 X_i のローカルスコアが得られているため，それらを足し合わせることでパスが表す構造 G の $\log BDeu(G, V')$ を得る．これをパスのスコアとする．

3.4 k -best の探索のための場合分け二分木の枝刈り

前節の定式化より，根から葉 1 までの $\log BDeu(G, V')$ が上位 k 個のパスにより k -best が得られる．パス探索には，深さ優先探索に枝刈りを組み込んだ深さ優先分枝限定法 (以下，DFBnB) を用いる．DFBnB は，制限時間の不足等により探索を打ち切る場合でも，その時点で得られた構造を用いてモデル平均を計算できる [3, 24]．

DFBnB では 2 つの枝刈りを行う．1 つ目は，非循環制約による枝刈りである．場合分け二分木の根から葉までのパスには，非循環制約を満たさないパスが存在する可能性がある．このパスを排除するために，パスの探索において 1-枝を辿る度に非循環制約を評価し，制約を満たさない場合は子節点を展開しない．

さらに DFBnB では，スコアを用いた枝刈りを行う．前節で述べたように，変数 X_i のローカルスコアは節点 e_{i,j_{\min}^i} から節点 e_{i,j_{\max}^i} の子節点までのパスを探索することで求まる．よって，節点 e_{i,j_{\min}^i} を展開する時点で，根から節点 e_{i,j_{\min}^i} までのパスに対応する変数 X_1, X_2, \dots, X_{i-1} のローカルスコアの総和 g が求まる．このとき，その節点 e_{i,j_{\min}^i} から 1 の値をもつ葉までのパスを探索する場合に得られる $X_i, X_{i+1}, \dots, X_{n'}$ のローカルスコアの総和の上限値 h_i が既知であれば，その節点 e_{i,j_{\min}^i} から 1 の値をもつ葉までのパ

スを辿ったときに得られる構造 G の $\log BDeu(G, V')$ の上限値を $g + h_i$ で推定できる。 $g + h_i$ が探索済みの構造のうち上位 k 個目の $\log BDeu(G, V')$ を下回る場合、その節点を通るパスはスコア上位 k 個のパスでないため、節点を枝刈りする。

枝刈りには、各 $i \in \{1, 2, \dots, n'\}$ に対して、節点 e_{i,j_{\min}^i} から 1 の値をもつ葉までのパスを探索して得られる $X_i, X_{i+1}, \dots, X_{n'}$ のローカルスコアの総和の上限値が必要になる。ここで、 e_{i,j_{\min}^i} をラベルとする節点のうち根から 0-枝のみを辿って到達できる節点を $e_{i,j_{\min}^i}^*$ とおくと、次の定理が成り立つ。

定理 3.1 $i \in \{1, 2, \dots, n'\}$ とする。節点 e_{i,j_{\min}^i} から 1 の値をもつ葉までのパスを探索して得られる変数 $X_i, X_{i+1}, \dots, X_{n'}$ のローカルスコアの総和は、節点 $e_{i,j_{\min}^i}^*$ から 1 の値をもつ葉までのパスを探索して得られる変数 $X_i, X_{i+1}, \dots, X_{n'}$ のローカルスコアの総和の最大値 h_i^* 以下になる。

定理 3.1 の証明では、以下の定義を用いる。

定義 3.1 $i, j \in \{1, \dots, n\}$ とする。場合分け二分木において、 $e_{i,j}$ をラベルとする任意の節点から葉までの 2 つのパスについて、それらのパスが表すエッジの有無のパターンが等しいとき、その 2 つのパスは等価であるという。

例えば図 5 に太線で示した 4 つのパスは等価である。また、 $e_{i,j}$ ($i, j \in \{1, \dots, n\}$) をラベルとする節点のうち、根から 0-枝のみを辿って到達できる節点を $e_{i,j}^*$ とおく。定理 3.1 の証明のため、以下の 2 つの補題を証明する。

補題 3.1 $i, j \in \{1, \dots, n\}$ とする。 $e_{i,j}$ をラベルとする任意の節点から 1 の値をもつ葉までの各パスに対して、節点 $e_{i,j}^*$ から 1 の値をもつ葉までの等価なパスが存在する。

証明 $i, j \in \{1, \dots, n\}$ とする。場合分け二分木において、 $e_{i,j}$ をラベルとする節点を任意にとり、その節点から 1 の値をもつ葉までの任意のパスをパス 1 とおく。また、節点 $e_{i,j}^*$ から葉までのパスのうち、パス 1 と等価なパスをパス 2 とおく。次に、場合分け二分木の根からパス 1 の終点までのパスが表現する DAG を G_1 とおく。同様に、根からパス 2 の終点までのパスが表現する有向グラフを G_2 とおく。 G_2 が DAG であることを証明する。

場合分け二分木において、節点 $e_{i,j}$ の祖先の節点のラベルであるエッジを集めてできる集合を E_1 、節点 $e_{i,j}$ の子孫の節点のラベルであるエッジを集めてできる集合を E_2 とおく。 G_2 において E_1 の各エッジは存在せず、 G_2 における $E_2 \cup \{e_{i,j}\}$ の各エッジの有無は G_1 と等しいため、 G_2 は G_1 の部分グラフである。よって、 G_1 が DAG であるとき

G_2 も DAG である. \square

補題 3.2 $i \in \{1, 2, \dots, n'\}$ とする. e_{i, j_{\min}^i} をラベルとする任意の節点から 1 の値をもつ葉までの 2 つのパスについて, 2 つのパスが等価であるとき, 2 つのパスを探索して得られる $X_i, X_{i+1}, \dots, X_{n'}$ のローカルスコアの総和は等しい.

証明 $i \in \{1, 2, \dots, n'\}$ とする. e_{i, j_{\min}^i} をラベルとする任意の節点から 1 の値をもつ葉までのパスを 2 つとり, パス 1, 2 とおく. パス 1 が表現する変数 $X_i, X_{i+1}, \dots, X_{n'}$ の親変数集合をそれぞれ $Pa'_i, Pa'_{i+1}, \dots, Pa'_{n'}$ とおく. このとき, パス 1 を探索して得られる変数 $X_i, X_{i+1}, \dots, X_{n'}$ のローカルスコアの総和は $\sum_{s=i}^{n'} Score_s(Pa'_s)$ と表せる. パス 1, 2 は等価であるから, パス 2 が表現する変数 $X_i, X_{i+1}, \dots, X_{n'}$ の親変数集合はパス 1 のものと一致する. よって, パス 2 を探索して得られる変数 $X_i, X_{i+1}, \dots, X_{n'}$ のローカルスコアの総和は $\sum_{s=i}^{n'} Score_s(Pa'_s)$ と表せる. したがって, パス 1, 2 を探索して得られる変数 $X_i, X_{i+1}, \dots, X_{n'}$ のローカルスコアの総和は等しい. \square

以上の定義, 補題を用いて, 以下のように定理 3.1 を証明する.

証明 背理法を用いて証明する. $i \in \{1, 2, \dots, n'\}$ とする. 節点 e_{i, j_{\min}^i} から 1 の値をもつ葉までのあるパス (パス 1 とおく) を探索して得られる変数 $X_i, X_{i+1}, \dots, X_{n'}$ のローカルスコアの総和 (h とおく) が, 節点 e_{i, j_{\min}^i} から 1 の値をもつ葉までのパスを探索して得られる変数 $X_i, X_{i+1}, \dots, X_{n'}$ のローカルスコアの総和の最大値 (h_i^* とおく) より大きいと仮定する. このとき, 補題 3.1 より, 節点 e_{i, j_{\min}^i} から 1 の値をもつ葉までのパスのうち, パス 1 と等価なパス (パス 2 とおく) が存在する. 補題 3.2 より, パス 1 とパス 2 を探索して得られる $X_i, X_{i+1}, \dots, X_{n'}$ のローカルスコアの総和は等しいため, パス 2 を探索して得られる $X_i, X_{i+1}, \dots, X_{n'}$ のローカルスコアの総和は h である. ここで, 仮定より $h_i^* < h$ が成り立つ. これは, h_i^* が節点 e_{i, j_{\min}^i} から 1 の値をもつ葉までのパスを探索して得られる変数 $X_i, X_{i+1}, \dots, X_{n'}$ のローカルスコアの総和の最大値であることと矛盾する. \square

定理 3.1 より, 提案手法では節点 e_{i, j_{\min}^i} から 1 の値をもつ葉までの探索により得られる $X_i, X_{i+1}, \dots, X_{n'}$ のローカルスコアの総和の最大値 h_i^* を保存する. その保存した h_i^* の値を, 他の e_{i, j_{\min}^i} の節点を展開する際の枝刈りに用いる. h_i^* を求めるためには, 節点 e_{i, j_{\min}^i} を根とする部分木を探索する必要がある. そこで, 節点 e_{i, j_{\min}^i} を除く各節点 e_{i, j_{\min}^i} に枝刈りを適用する.

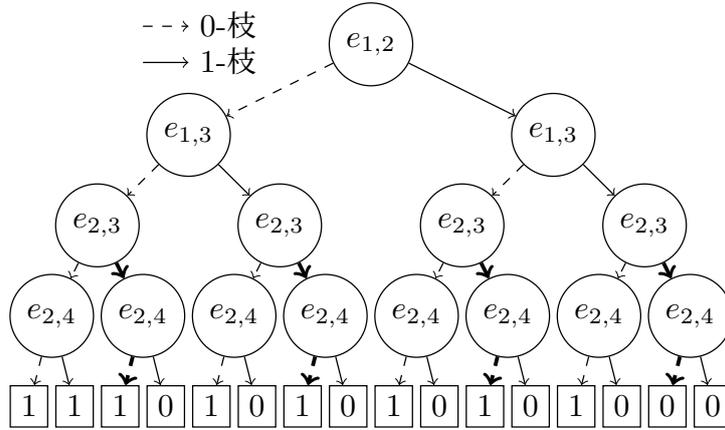


図 5: 等価なパスの例

3.5 k -best の探索のための場合分け二分木の深さ優先分枝限定法

提案アルゴリズムを説明する．まず， $\Pi_i = Pa(X_i, (V, E_{\text{unreliable}}))$ とおく．また， Π_i の各変数の添字を昇順に並べたときの m 番目の添字を第 m 成分にもつベクトルを π_i とし， π_i の第 m 成分を π_i^m で表すとす．また， Π_i の変数数を M_i とおく．例えば，図 3 の場合分け二分木では， $\pi_1 = (2, 3), \pi_2 = (3, 4), M_1 = 2, M_2 = 2$ となる．よって， $e_{1,2}, e_{1,3}, e_{2,3}, e_{2,4}$ はそれぞれ $e_{1,\pi_1^1}, e_{1,\pi_1^2}, e_{2,\pi_2^1}, e_{2,\pi_2^2}$ と表せる．このとき，エッジ $e_{i,\pi_i^1}, e_{i,\pi_i^2}, \dots, e_{i,\pi_i^{M_i}}$ の有無により候補構造における変数 X_i の親変数集合が定まる．また，前節の説明に用いた j_{\min}^i, j_{\max}^i はそれぞれ $\pi_i^1, \pi_i^{M_i}$ と一致する．

次の手順で探索する．構造 $G_0 = (V, E_0)$ における変数 X_i の親変数集合を Pa_i^0 とおく．はじめに，探索したうち $\log BDeu(G, V')$ 上位 k 個の構造 G を保存する変数 *structure*，それらの構造 G の $\log BDeu(G, V')$ を保存する変数 *score*，節点 $e_{i,j_{\min}^i}^*$ から 1 の値をもつ葉までのスコアの最大値を保存する変数 h_i^* を用意する．各節点にはその節点までに追加したエッジの集合を表す変数 E_{add} ，その節点までに追加したエッジにより変数 X_i の親になる変数の集合を表す Pa_i ，探索している構造のローカルスコアの総和を保存する変数 g ，節点のラベルであるエッジの終点を表す変数 i ，節点のラベルであるエッジの始点を表す変数 m を用意する．葉を展開した際に k -best の条件 (非循環制約や $\log BDeu(G, V')$ 上位 k 番目の制約) を満たすとき，*structure* と *score* を更新する．これにより， k -best を探索できる．

具体的には，次のように 5 つのステップで構成される．

1. 根を展開する (図 6 (a)). 根ではまだエッジが選択されていないため, 変数 E_{add} , Pa_i , $structure$, $score$ の値を空のリストとする. また, 変数 g の値を 0, h_i^* の値を $-\infty$, i の値を 1, m の値を 1 とする.
2. 次に, 根から 0-枝優先で深さ優先的に子節点を展開する. 0-枝を辿る場合, 変数 E_{add} , Pa_i の値は親節点の変数と同じ値とする (図 6 (b)). また, 以下の処理 1 を行う.
 [処理 1]
 ($m < M_i$ のとき) 子節点に変数 X_i に接続するエッジがあるため, m の値を一つ加算する (図 6 (b)).
 ($m = M_i$ のとき) $e_{i,\pi_i^1}, e_{i,\pi_i^2}, \dots, e_{i,\pi_i^{M_i}}$ の有無が定まっているため変数 X_i のローカルスコアが定まる (図 6 (c)). g に X_i のローカルスコア $Score_i(Pa_i^0 \cup Pa_i)$ を加算し, i の値を一つ加算し, m の値を 1 にする.
 以降, 0-枝優先で深さ優先的に子節点を展開する.
3. 葉を展開したとき ($i = n' + 1$ のとき), $structure$ に保存されている構造が k 個未満であるとき, 構造 $(V, E_0 \cup E_{add})$ を $structure$ に追加し, g を $score$ に追加する (図 6 (d)). $structure$ に保存されている構造が k 個であるとき, $score$ に保存されている値の最小値より g が大きい場合, その最小値を $score$ から削除し, その削除した値に対応する構造を $structure$ から削除する. そして, g を $score$ に追加し, 構造 $(V, E_0 \cup E_{add})$ を $structure$ に追加する. また, 各 $i \in \{2, 3, \dots, n'\}$ に対して, 節点 e_{i,j_{\min}^i} から 1 の値をもつ葉までのパスのスコアの最大値 h_i^* を更新する.
4. 1-枝を辿る場合, E_{add} に e_{i,π_i^m} を追加し, Pa_i に $X_{\pi_i^m}$ を追加し, グラフ $(V, E_0 \cup E_{add})$ が非循環制約を満たすか判定する. 制約を満たすとき (図 6 (e)), 0-枝を辿る場合と同様に, (2) で説明した [処理 1] を行う. 非循環制約を満たさないとき (図 6 (f)), 子節点を展開せず枝刈りする.
5. 根から少なくとも 1 つの 1-枝を辿って到達できる節点 e_{i,π_i^1} ($= e_{i,j_{\min}^i}$) を展開する場合, $structure$ の構造数が k であればスコアによる枝刈りを適用する (図 6 (g)). $g + h_i^*$ が暫定の k -best のスコアを下回る場合, この展開は行われず枝刈りされる.

提案の DFBnB により k -best を探索するアルゴリズムを Algorithm 1 に示す.

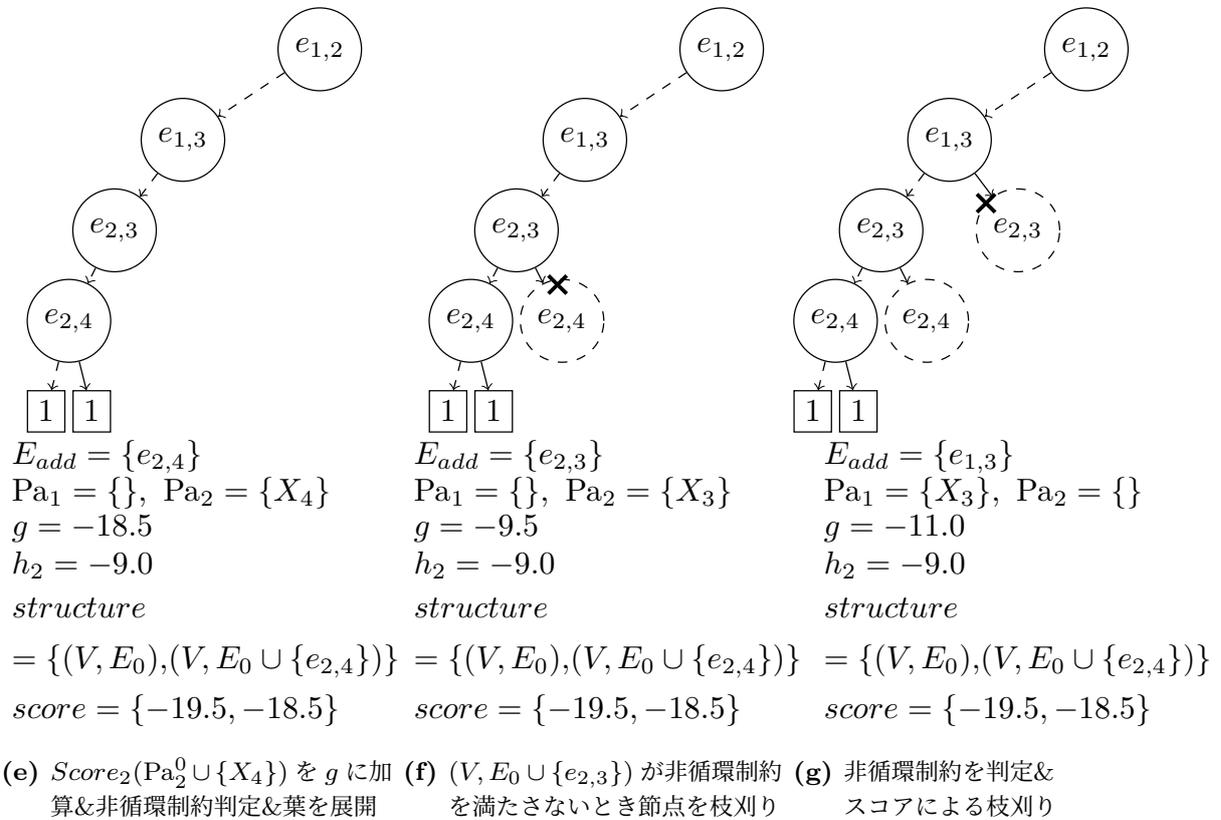
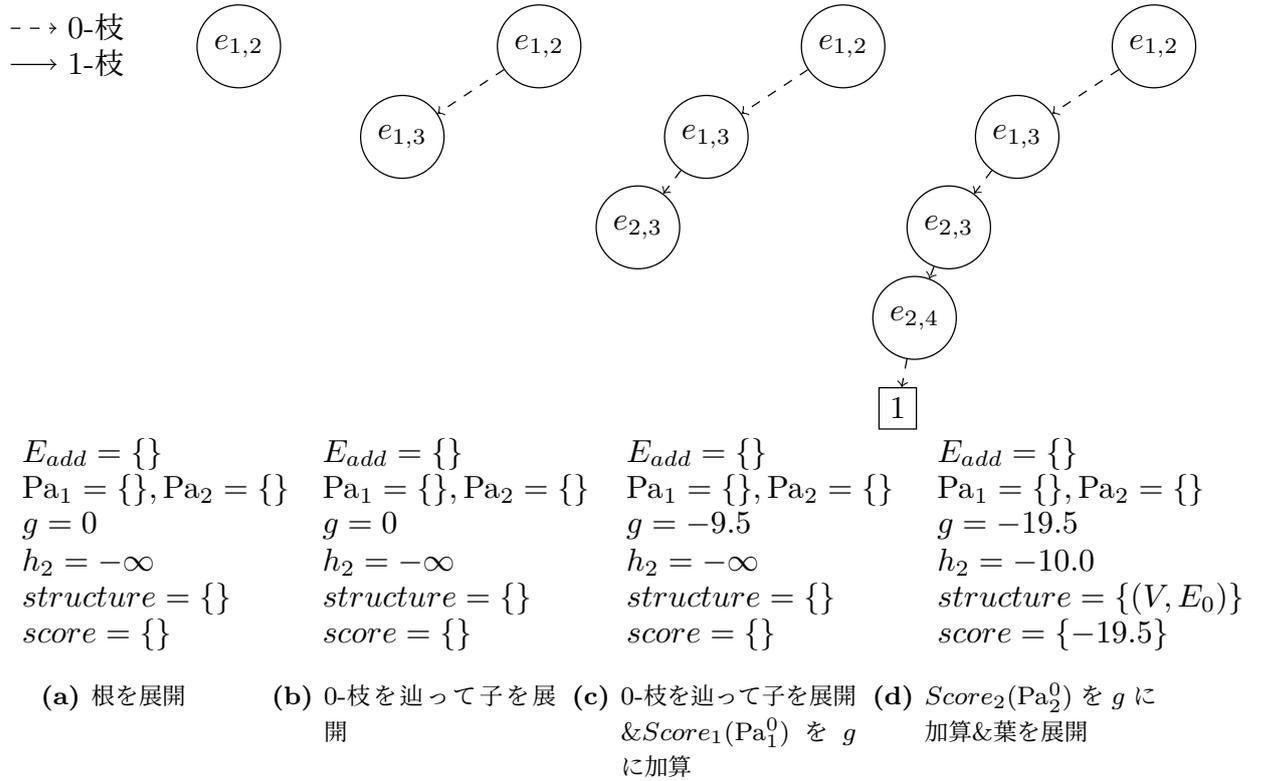


図 6: 深さ優先分枝限定法 (DFBnB) の動作例 ($k = 2$)

Algorithm 1 proposed depth-first branch and bound (DFBnB) for finding the k -best structures

```

1: function DFBnB( $m, E_{\text{unreliable}}, D, \text{structure}, \text{score}, G_0, n', i, h, LS, Pa, E_{\text{add}}, M, \pi$ )
2:   if  $i == n' + 1$  then ▷ 葉を展開した場合
3:      $\text{score}_G \leftarrow \sum_{s=1}^{n'} LS[s]$ 
4:     if  $\text{structure}$  のサイズが  $k$  未満 then
5:        $\text{structure}$  に  $(V, E_0 \cup E_{\text{add}})$  を追加
6:        $\text{score}$  に  $\text{score}_G$  を追加
7:     else if  $\text{score}_G > \text{score}$  の最小値 then
8:        $\text{score}, \text{structure}$  を更新
9:     end if
10:    for  $i' = 2, \dots, n' - 1$  do ▷  $h_i^*$  の更新
11:       $\text{tmp} \leftarrow \sum_{s=i}^{n'} LS[s]$ 
12:      if  $h[i'] < \text{tmp}$  then
13:         $h[i'] \leftarrow \text{tmp}$ 
14:      end if
15:    end for
16:  else ▷ 葉以外の節点を展開した場合
17:    for  $b = 0, 1$  do ▷ 0-枝, 1-枝
18:      if  $b == 1$  then
19:         $E_{\text{add}} \leftarrow E_{\text{add}} \cup \{e_{i, \pi_i^m}\}$  ▷ エッジ  $e_{i, \pi_i^m}$  を追加
20:         $\text{Pa}[i] \leftarrow \text{Pa}[i] \cup \{X_{\pi_i^m}\}$ 
21:        if  $(V, E_0 \cup E_{\text{tmp}})$  が有向閉路をもつ then ▷ 枝刈り
22:          Continue
23:        end if
24:      end if
25:      if  $m == M_i$  then ▷ スコアによる枝刈りを適用する深さ
26:         $LS[i] \leftarrow \text{Score}_i(\text{Pa}(X_i, G_0) \cup \text{Pa}[i])$ 
27:        if  $\text{structure}$  のサイズが  $k$  then
28:           $g \leftarrow \sum_{s=1}^i LS[s]$ 
29:          if  $g + h[i + 1] < \text{score}$  の最小値 then ▷ 枝刈り
30:            Continue
31:          end if
32:        end if
33:        DFBnB( $1, E_{\text{unreliable}}, D, \text{structure}, \text{score}, G_0, n', i + 1, h, LS, \text{Pa}, E_{\text{add}}, M, \pi$ )
34:      else ▷ スコアによる枝刈りを適用しない深さ
35:        DFBnB( $m + 1, E_{\text{unreliable}}, D, \text{structure}, \text{score}, G_0, n', i, h, LS, \text{Pa}, E_{\text{add}}, M, \pi$ )
36:      end if
37:    end for
38:     $E_{\text{add}} \leftarrow E_{\text{add}} \setminus \{e_{i, \pi_i^m}\}$ 
39:     $\text{Pa}[i] \leftarrow \text{Pa}[i] \setminus \{X_{\pi_i^m}\}$ 
40:  end if
41: end function
42: function MAIN( $E_{\text{unreliable}}, D, G_0, n', M, \pi$ ) ▷  $G_0 = (V, E_0), M = \{M_1, \dots, M_{n'}\},$ 
43:    $\pi = \{\pi_i, \dots, \pi_{n'}\}$ 
44:    $\text{score} \leftarrow \{\}$ 
45:    $\text{structure} \leftarrow \{\}$ 
46:    $h \leftarrow \{-\infty, \dots, -\infty\}$ 
47:    $LS \leftarrow \{0, \dots, 0\},$ 
48:    $\text{Pa} \leftarrow \{\emptyset, \dots, \emptyset\}$ 
49:    $E_{\text{add}} \leftarrow \emptyset$ 
50:   DFBnB( $1, E_{\text{unreliable}}, D, \text{structure}, \text{score}, G_0, n', 1, h, LS, \text{Pa}, E_{\text{add}}, M, \pi$ )
51: return  $\text{structure}, \text{score}$ 

```

4 評価実験

本章では、提案手法の有効性を示すために評価実験を行う。具体的には、リポジトリデータを用いて提案手法と他の BNC 学習手法の分類精度を比較する。

本章では以下の 8 つの手法を比較する。

- Naive Bayes
- TAN: 対数尤度を最適化する TAN を学習 [2]
- exact-ANB: BDeu スコアで ANB を厳密学習
- kbest-ANB: kbest 法 [10] により, BDeu スコアの上位 k 個の ANB を選択してモデル平均
- Ens-kbest-GBN: 非復元抽出法により元のデータから生成した T 個の各データについて k-best 法で k 個の GBN を選択し, 得られた $T \times k$ 個の GBN を用いてモデル平均 [12]
- RAI-GBN: Bayes factor を用いた RAI で GBN を学習 [17][18]
- RAI-ANB: Bayes factor を用いた RAI で ANB を学習 [19]
- no-pruning: 提案手法から枝刈りを除いた手法により k 個の ANB を選択してモデル平均
- proposal: 提案手法により k 個の ANB を選択してモデル平均

exact-ANB は BDeu スコアを最大化する ANB を動的計画法 [5] を用いて厳密に学習した。BDeu スコアと Bayes factor の ESS の値は, Ueno[25] の提案に従い事後分散を最大にする 1.0 とした。いずれの手法においても, 構造学習後の BNC のパラメータは全て EAP で推定した。各手法の実行環境は, Ubuntu 20.04 を OS とする計算機 (Intel(R) Core(TM) i9-10900K 3.70GHz, RAM 64GB) である。また, Sugahara ら [19, 20] と同様に実行時間の上限を 6 時間と定め, それを超えた場合に計算を打ち切った。ただし, 提案手法および no-pruning が制限時間に到達した場合は, その時点で得られたスコア上位 k 個の構造をモデル平均に用いた。先行研究 [10][12] を参考に $k = 10$, $T = 10$ とした。

また, 本実験では解析の妥当性を示すために各 BNC の構造学習と分類に対して 10 分割交差検証を行った。10 分割交差検証では, データを 10 分割した後, 1 個をテストデータ, 他の 9 個を学習データとして学習と分類を計 10 回繰り返す。テストデータの各説明変数に対し, 推定された目的変数の値が正しく分類されている割合を算出し, 10 分割交差検証におけるその平均を分類精度とした。また, 提案手法の有効性を示すために, 分類精

表 1: 小規模変数のデータにおける各手法の分類精度 (太字は最大の分類精度)

No.	Dataset	Variables	Sample-size	Naive-Bayes	TAN	exact-ANB	kbest-ANB	Ens-kbest-GBN	RAI-ANB	no-pruning			
										$\epsilon = \log 3$	$\epsilon = \log 20$	proposal $\epsilon = \log 3$	proposal $\epsilon = \log 20$
1	Balance	5	625	0.7184	0.8496	0.9168	0.9168	0.9120	0.9168	0.9168	0.9168	0.9168	0.9168
2	banknote	5	1372	0.8433	0.8819	0.8812	0.8812	0.8812	0.8812	0.8812	0.8812	0.8812	0.8812
3	Hayes-Roth	5	132	0.8258	0.6742	0.8106	0.8106	0.7879	0.8106	0.8106	0.8106	0.8106	0.8106
4	iris	5	150	0.7133	0.8267	0.8133	0.8267	0.8200	0.8067	0.8067	0.8133	0.8067	0.8133
5	lenses	5	24	0.7083	0.6667	0.6250	0.6250	0.8333	0.6250	0.6250	0.6250	0.6250	0.6250
6	Car	7	1728	0.8079	0.9462	0.9398	0.9363	0.9236	0.9398	0.9398	0.9392	0.9398	0.9392
7	liver	7	345	0.6319	0.6551	0.6580	0.6725	0.6203	0.6638	0.6580	0.6580	0.6580	0.6580
8	monk1	7	432	0.7500	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
9	mux6	7	64	0.5469	0.6094	0.5469	0.5625	0.5313	0.5469	0.5469	0.5625	0.5469	0.5625
10	led7	8	3200	0.7278	0.7344	0.7288	0.7288	0.7325	0.7288	0.7288	0.7288	0.7288	0.7288
11	HTRU2	9	17898	0.8513	0.9141	0.9141	0.9141	0.9141	0.9084	0.9084	0.9084	0.9084	0.9084
12	Nursery	9	12960	0.8807	0.9317	0.9184	0.9183	0.9285	0.9367	0.9363	0.9330	0.9363	0.9330
13	pima	9	768	0.6797	0.7214	0.7188	0.7188	0.7201	0.7188	0.7188	0.7096	0.7188	0.7096
14	Breast Cancer	10	277	0.7004	0.6968	0.7256	0.7184	0.7148	0.7184	0.7112	0.7112	0.7112	0.7112
15	Breast Cancer Wisconsin	10	683	0.9780	0.9517	0.9766	0.9766	0.9766	0.9766	0.9766	0.9766	0.9766	0.9766
16	cmc	10	1473	0.4603	0.4800	0.4657	0.4650	0.4494	0.4773	0.4820	0.4854	0.4820	0.4854
17	glass	10	214	0.4860	0.6168	0.5888	0.5888	0.5794	0.5981	0.5935	0.6028	0.5935	0.6028
18	shuttle-small	10	5800	0.7900	0.9707	0.9714	0.9716	0.9693	0.9702	0.9705	0.9705	0.9705	0.9705
19	threeOF9	10	512	0.7969	0.8926	0.8574	0.8867	0.8691	0.8477	0.8535	0.8516	0.8535	0.8516
20	Tic-Tac-Toe	10	958	0.6681	0.7557	0.8528	0.8559	0.8382	0.7547	0.7526	0.7547	0.7526	0.7547
21	MAGIC	11	19020	0.7130	0.7768	0.7867	0.7872	0.7862	0.7583	0.7802	0.7802	0.7802	0.7802
22	Solar flare	11	1389	0.5918	0.8193	0.8251	0.8251	0.8431	0.8143	0.8121	0.8121	0.8121	0.8121
23	heart	14	270	0.8259	0.8148	0.8296	0.8296	0.8074	0.8222	0.8185	0.8185	0.8185	0.8185
24	wine	14	178	0.9213	0.9213	0.9326	0.9270	0.9438	0.9270	0.9326	0.9326	0.9326	0.9326
25	cleve	14	296	0.8311	0.7973	0.8311	0.8209	0.8209	0.8277	0.8311	0.8311	0.8311	0.8311
26	australian	15	690	0.8246	0.8377	0.8391	0.8362	0.8464	0.8449	0.8435	0.8391	0.8435	0.8391
27	crx	15	653	0.8315	0.8254	0.8407	0.8407	0.8484	0.8407	0.8484	0.8423	0.8484	0.8423
28	EEG	15	14980	0.5722	0.6298	0.6903	0.6895	0.6806	0.6696	0.6675	0.6682	0.6675	0.6682
29	Congressional	17	232	0.9095	0.9353	0.9483	0.9526	0.9655	0.9353	0.9397	0.9397*	0.9397	0.9526
30	zoo	17	101	0.9901	0.9307	0.9703	0.9703	0.9703	0.9703	0.9604	0.9604*	0.9604	0.9604
31	pendigits	17	10992	0.7999	0.8936	0.9290	0.9294	0.9269	0.9091	0.9124	0.9144	0.9124	0.9144
32	letter	17	20000	0.4460	0.5209	0.5943	0.5942	0.5727	0.5604	0.5610	0.5634	0.5610	0.5634
33	ClimateModel	19	540	0.8611	0.9185	0.8574	0.8907	0.8722	0.9222	0.9222	0.9222	0.9222	0.9222
34	Image Segmentation	19	2310	0.7338	0.8043	0.8251	0.8260	0.8199	0.8056	0.8039	0.8056	0.8039	0.8056
35	lymphography	19	148	0.8514	0.7838	0.7905	0.7973	0.7230	0.8649	0.8649	0.8514	0.8649	0.8514
36	vehicle	19	846	0.4232	0.5816	0.6170	0.6111	0.5674	0.5898	0.5957	0.6064	0.5957	0.6064
37	german	21	1000	0.7170	0.7300	0.7390	0.7420	OT	0.7420	0.7410	0.7430	0.7410	0.7430
38	bank	21	30488	0.7458	0.8778	0.8954	OT	OT	0.8901	0.8904	0.8912*	0.8904	0.8904
39	waveform-21	22	5000	0.7886	0.7932	0.7952	OT	OM	0.7832	0.7854	0.7882	0.7854	0.7882
40	Mushroom	22	5644	0.9963	0.9998	1.0000	OT	OM	0.9998	1.0000	1.0000	1.0000	1.0000
41	spect	23	263	0.7266	0.8052	0.7978	OM	OM	0.8052	0.8090	0.8277*	0.8090	0.8277
	average			0.7479	0.7993	0.8108	N/A	N/A	0.8075	0.8082	0.8092	0.8082	0.8095
	p-value			0.0001	0.0510	0.1685	N/A	N/A	0.0890	-	-	-	-

度の多重検定手法として標準的に用いられる Hommel 法による多重検定 [26] を行い p 値を求め、各手法、各データセットに対して、構造学習にかかる平均計算時間を求めている。

4.1 小規模変数データセットを用いた評価実験

本節では、厳密解探索アプローチを適用可能な小規模変数のデータセットを用いて各手法で学習した BNC の分類精度と各手法の計算時間の比較を行う。提案手法で用いる閾値

表 2: 小規模変数のデータにおける RAI-ANB において条件 a, b を満たすエッジ数と提案の枝刈り回数

No.	Dataset	条件 a, b を 満たすエッジ数		枝刈り回数	
		$\varepsilon = \log 3$	$\varepsilon = \log 20$	$\varepsilon = \log 3$	$\varepsilon = \log 20$
1	Balance	0.0	0.0	0.0	0.0
2	banknote	0.0	0.6	0.0	0.0
3	Hayes-Roth	0.0	0.0	0.0	0.0
4	iris	3.2	4.0	0.4	0.5
5	lenses	0.1	1.5	0.0	0.0
6	Car	0.3	0.5	0.0	0.0
7	liver	1.3	2.8	0.1	2.4
8	monk1	0.0	0.0	0.0	0.0
9	mux6	0.0	2.3	0.0	0.0
10	led7	0.0	0.0	0.0	0.0
11	HTRU2	1.6	3.8	0.0	3.5
12	Nursery	0.2	0.4	0.0	0.0
13	pima	1.3	4.7	0.1	12.0
14	Breast Cancer	0.5	1.1	0.0	0.0
15	Breast Cancer Wisconsin	0.0	0.0	0.0	0.0
16	cmc	1.6	3.4	0.1	0.9
17	glass	2.0	4.8	1.9	25.5
18	shuttle-small	1.2	5.5	0.0	0.2
19	threeOf9	1.7	6.1	0.0	49.8
20	Tic-Tac-Toe	0.3	1.5	0.0	0.0
21	MAGIC	0.4	1.2	0.0	0.0
22	Solar Flare	1.3	5.3	0.1	14.9
23	heart	2.3	6.6	0.7	25.0
24	wine	2.9	13.1	2.3	114.4
25	cleve	1.8	4.6	0.2	12.9
26	australian	2.6	15.0	1.7	207.2
27	crx	1.7	18.5	1.3	759.6
28	EEG	2.7	8.6	0.0	2.9
29	Congressional	16.5	56.7	960.6	1763084.3
30	zoo	28.0	67.9	10191.1	599414.6
31	pendigits	3.9	9.6	0.0	6.0
32	letter	0.9	2.6	0.0	0.0
33	ClimateModel	0.4	2.2	0.0	0.6
34	Image Segmentation	5.2	16.7	15.0	490.3
35	lymphography	8.1	29.4	21.6	4551.6
36	vehicle	9.2	23.6	24.7	1806.7
37	german	2.8	7.0	1.3	24.1
38	bank	26.9	42.4	106937.5	574266.1
39	waveform-21	12.4	28.5	194.4	18744.2
40	Mushroom	10.1	35.5	61.1	58414.9
41	spect	22.1	69.8	1478.3	461350.8
	average	4.3	12.4	2924.3	84960.6

ε の値は, Kass ら [21] を参考に $\log 3, \log 20$ とした. 実験には, 変数数 5 から 23 をもつ 41 個のデータセットを用いた.

表 1 に, 各データセットに対する各手法の分類精度を示している. 表中の"OT"は計算時間が上限に到達し, 計算が打ち切られたことを表す. 表中の"OM"はメモリ上限に到

表 3: データセット No.1~36 に対する各手法の分類精度 (太字は最大の分類精度)

	Naive-Bayes	TAN	exact-ANB	kbest-ANB	Ens-kbest-GBN	RAI-ANB	no-pruning			
							proposal		proposal	
							$\varepsilon = \log 3$	$\varepsilon = \log 20$	$\varepsilon = \log 3$	$\varepsilon = \log 20$
average	0.7414	0.7935	0.8060	0.8084	0.8055	0.8025	0.8031	0.8035*	0.8031	0.8039
p-value	0.0001	0.1796	0.3371	0.2863	0.4840	0.4295	-	-	-	-

達し、計算が打ち切られたことを表す。表中の"*"は制限時間に到達した時点で得られた k -best でモデル平均を行い、分類した結果を示す。表 1 の下部に示されている "average" は全データセットに対する各手法の平均分類精度であり、"p 値"は多重検定により得られた p 値を示している。各データセットに対し、各手法において最も高い分類精度を太字で示している。また表 2 に、RAI-ANB で実施した CI テストにおいて条件 a, b を満たすエッジの数と提案手法における枝刈りの回数を示す。

表 1 より、提案手法の平均分類精度は Naive Bayes, TAN, RAI-ANB より高いことがわかる。特に Naive Bayes に対しては有意水準 5% のもとで有意に分類精度が高かった。提案手法が Naive-Bayes, TAN より高精度であるのは、Naive Bayes, TAN は説明変数間の構造に制約を設けるのに対し、提案手法が学習する ANB は制約を設けないため、表現力が高いことが要因である。また提案手法が RAI-ANB より高精度である理由は、対数 Bayes factor の値が設定した 0 近傍の閉区間に含まれるエッジが存在する場合と存在しない場合についてモデル平均することで、RAI-ANB で実施した CI テストの誤りによる分類精度低下を抑制できたためである。さらに提案手法の分類精度は、閾値 $\varepsilon = \log 3$ よりも $\varepsilon = \log 20$ の方が高かった。これは表 2 に示すように、閾値 ε を大きくすると対数 Bayes factor の値が閉区間 $[-\varepsilon, \varepsilon]$ に含まれる CI テストに対応するエッジ数が増加し、CI テスト結果が誤っているエッジの検出漏れが少なくなるためである。

また、 k best-ANB と Ens- k best-GBN はデータセット 37~41 番で計算時間やメモリ使用量の上限を超えたことによる計算打ち切りが発生したため、平均分類精度は "N/A" となっている。これらの手法は計算コストが高いうえ、計算打ち切りが発生した場合に構造が 1 つも得られないため、変数数が大きいデータセットでは分類ができない。一方、提案手法は全てのデータセットで分類ができた。

そこで提案手法と k best-ANB, Ens- k best-GBN の分類精度を比較するため、これらの手法で学習できたデータセット 1~36 番に対する各手法の分類精度を表 3 に示す。表 3 より、 k best-ANB と Ens- k best-GBN の分類精度は提案手法より高かった。したがって、変数数が 20 未満の場合は提案手法よりもこれらのモデル平均手法の方が有効である。

表 4: 小規模変数のデータにおける各手法の計算時間 [s]

No.	Dataset	Variables	Sample-size	Naive-Bayes	TAN	exact-ANB	kbest-ANB	Ensemble-kbest-GBN	RAI-ANB	proposal			
										no-pruning		proposal	
										$\epsilon = \log 3$	$\epsilon = \log 20$	$\epsilon = \log 3$	$\epsilon = \log 20$
1	Balance	5	625	0.00	0.01	0.06	0.02	0.86	0.01	0.02	0.02	0.01	0.01
2	banknote	5	1372	0.00	0.01	0.06	0.01	0.84	0.02	0.04	0.04	0.02	0.02
3	Hayes-Roth	5	132	0.00	0.00	0.05	0.01	0.69	0.01	0.01	0.01	0.01	0.01
4	iris	5	150	0.00	0.00	0.05	0.01	0.66	0.01	0.03	0.02	0.01	0.01
5	lenses	5	24	0.00	0.00	0.04	0.01	0.50	0.00	0.01	0.01	0.00	0.00
6	Car	7	1728	0.00	0.01	0.10	0.04	1.67	0.01	0.10	0.03	0.07	0.01
7	liver	7	345	0.00	0.01	0.07	0.02	1.05	0.01	0.03	0.03	0.01	0.01
8	monk1	7	432	0.00	0.01	0.08	0.02	1.18	0.01	0.01	0.01	0.01	0.01
9	mux6	7	64	0.00	0.00	0.07	0.02	0.91	0.01	0.01	0.01	0.01	0.01
10	led7	8	3200	0.00	0.02	0.13	0.05	2.29	0.01	0.11	0.03	0.07	0.02
11	HTRU2	9	17898	0.00	0.04	0.24	0.12	4.28	0.11	0.32	0.25	0.19	0.14
12	Nursery	9	12960	0.00	0.09	0.45	0.26	7.05	0.04	0.10	0.10	0.05	0.05
13	pima	9	768	0.00	0.01	0.12	0.05	2.16	0.02	0.04	0.03	0.02	0.02
14	Breast Cancer	10	277	0.00	0.01	0.21	0.10	4.02	0.02	0.10	0.03	0.07	0.02
15	Breast Cancer Wisconsin	10	683	0.00	0.02	0.34	0.18	6.19	0.02	0.04	0.04	0.02	0.02
16	cmc	10	1473	0.00	0.01	0.32	0.13	5.55	0.02	0.04	0.04	0.02	0.02
17	glass	10	214	0.00	0.01	0.16	0.07	3.01	0.02	0.03	0.03	0.02	0.02
18	shuttle-small	10	5800	0.00	0.03	0.22	0.11	4.30	0.12	0.25	0.25	0.13	0.13
19	threeOf9	10	512	0.00	0.01	0.17	0.08	3.16	0.01	0.03	0.03	0.01	0.01
20	Tic-Tac-Toe	10	958	0.00	0.01	0.29	0.16	5.14	0.01	0.03	0.03	0.02	0.02
21	MAGIC	11	19020	0.00	0.05	0.46	0.29	8.54	0.33	0.76	0.70	0.42	0.37
22	Solar Flare	11	1389	0.00	0.02	0.34	0.21	6.73	0.05	0.10	0.10	0.05	0.05
23	heart	14	270	0.00	0.01	1.26	1.24	29.97	0.02	0.11	0.05	0.08	0.02
24	wine	14	178	0.00	0.01	0.90	0.94	25.15	0.02	0.05	0.08	0.02	0.02
25	cleve	14	296	0.00	0.01	1.46	1.35	57.01	0.02	0.04	0.04	0.02	0.02
26	australian	15	690	0.00	0.01	6.24	6.36	234.55	0.03	0.13	0.15	0.10	0.03
27	crx	15	653	0.00	0.01	13.63	14.61	512.91	0.03	0.14	6.03	0.09	0.04
28	EEG	15	14980	0.00	0.08	7.78	7.86	238.49	2.62	5.28	5.28	2.65	2.66
29	Congressional	17	232	0.00	0.01	6.72	9.39	304.35	0.05	0.32	21600.00*	0.11	44.35
30	zoo	17	101	0.00	0.01	4.66	7.04	207.92	0.05	2764.29	21600.00*	0.06	1.24
31	pendigits	17	10992	0.00	0.27	53.64	55.15	1614.72	4.53	9.09	9.09	4.56	4.56
32	letter	17	20000	0.00	1.07	291.69	288.93	6213.53	3.49	7.06	7.05	3.56	3.55
33	ClimateModel	19	540	0.00	0.01	120.79	133.10	3997.71	0.03	0.14	0.07	0.09	0.03
34	Image Segmentation	19	2310	0.00	0.05	117.88	129.25	3668.37	0.22	0.44	1.20	0.23	0.23
35	lymphography	19	148	0.00	0.01	51.27	65.37	2392.43	0.05	0.11	2101.42	0.05	0.06
36	vehicle	19	846	0.00	0.02	82.92	96.15	3381.64	0.15	0.32	38.73	0.16	0.17
37	german	21	1000	0.00	0.02	1539.57	1579.03	OT	0.06	0.19	0.12	0.12	0.06
38	bank	21	30488	0.00	0.39	8793.37	OT	OT	0.89	299.32	21600.00*	26.31	26.43
39	waveform-21	22	5000	0.00	0.08	6435.73	OT	OM	0.64	1.37	742.78	0.72	0.74
40	Mushroom	22	5644	0.00	0.13	2682.19	OT	OM	6.85	13.72	1009.97	6.87	10.83
41	spect	23	263	0.00	0.01	1265.07	OM	OM	0.07	288.64	21600.00*	0.14	2.17
average				0.00	0.06	523.92	N/A	N/A	0.50	79.80	2149.63	0.65	1.89
standard deviation				0.00	0.18	2022.87	N/A	N/A	1.38	434.12	6467.97	3.97	7.89

次に、各手法の計算時間を比較する。表 4 に各手法の構造学習の計算時間を示す。表 4 において"average"は全データセットに対する各手法の構造学習にかかる平均計算時間であり、"standard deviation"は各手法の計算時間の標準偏差を示している。Naive Bayes は構造学習の必要がないため、計算時間が 0.00 秒になっている。表 4 より、構造学習を要する手法の平均計算時間は TAN, RAI-ANB, 提案手法 ($\epsilon = \log 3$, $\epsilon = \log 20$) の順で

表 5: 大規模変数のデータにおける各手法の分類精度 (太字は最大の分類精度)

No.	Dataset	Variables	Sample size	Naive-Bayes	TAN	RAI-GBN	RAI-ANB	proposal
1	ionosphere	35	351	0.7493	0.9259	0.8803	0.9088	0.9288
2	kr-vs-kp	37	3196	0.8780	0.9255	0.9406	0.9524	0.9521*
3	Connect-4	43	67557	0.5417	0.7643	0.7337	0.7938	0.7938
4	Flowmeters D	44	180	0.8500	0.8500	0.6778	0.8500	0.8556*
5	movement libras	91	360	0.5028	0.5528	0.2278	0.5333	0.5583
6	dota2	117	102944	0.5960	0.5798	0.5442	0.5957	0.5960*
7	Musk1	167	478	0.6576	0.7563	0.6744	0.7983	0.8256*
8	Musk2	167	6598	0.7376	0.8412	0.8821	0.9615	0.9609*
9	Epileptic Seizure	179	11500	0.2346	0.1887	0.1187	0.3808	0.4090
10	mfeat-fac	219	2000	0.3520	0.4545	0.2630	0.4650	0.4700*
11	semeion	257	1600	0.8556	0.8688	0.4557	0.8776	0.8782
12	madelon	501	2000	0.5910	0.5380	0.6215	0.5905	0.6000
	average			0.6289	0.7019	0.5908	0.7257	0.7357
	p-value			0.0034	0.0028	0.0037	0.0064	-

短いことがわかる。no-pruning は閾値 ε の増加に伴い計算時間が大幅に増加し、 $\varepsilon = 20$ の場合は exact-ANB より平均計算時間が長くなってしまふ。これは条件 a, b を満たすエッジ数の増加に伴い、提案の場合分け二分木のサイズが指数的に増加し、計算コストが膨大になるためである。閾値 $\varepsilon = \log 20$ の場合の提案手法と no-pruning を比較すると、提案手法の計算時間は no-pruning に比べて非常に短い。これは枝刈りにより、場合分け二分木の探索空間を制限できたことが要因である。特に、データセット 29, 30, 38, 41 番では、no-pruning は制限時間 6 時間以内に学習できなかったが、提案手法では 45 秒以内に学習できた。これは表 2 に示すように、これらのデータセットでは提案手法で実施される枝刈りの回数が多く、探索空間を大幅に制限できたためである。以上より、提案の枝刈りにより場合分け二分木を用いた k -best 列挙の計算コストを削減できることが示された。

4.2 大規模変数データセットを用いた評価実験

本節では、厳密解探索アプローチで扱えない大規模変数のデータセットを用いた評価実験を行う。変数数 35~501 である 12 個のデータセットを用いて、従来手法で学習し

表 6: 大規模変数のデータにおける RAI-ANB において条件 a,b を満たすエッジ数

No.	Dataset	条件 a, b を 満たすエッジ数
1	ionosphere	16.5
2	kr-vs-kp	125.3
3	Connect-4	15.4
4	Flowmeters D	107.9
5	movement libras	127.4
6	dota2	280.0
7	Musk1	1406.5
8	Musk2	1190.8
9	Epileptic Seizure	102.1
10	mfeat-fac	502.0
11	semeion	323.8
12	madelon	50.5
	average	354.0

た BNC と提案手法で学習した BNC の分類精度を比較する。提案手法で用いる閾値 ε の値は, Kass ら [21] を参考に $\log 3$ とした。適用できないため, 厳密解探索アプローチは比較対象から除外した。表中の提案手法の列の "*" は制限時間に到達した段階で得られた k -best でモデル平均を行い, 分類した結果を示す。

表 5 より, 提案手法の分類精度が最も高いことがわかる。また, 提案手法は Naive Bayes, TAN, RAI-GBN, RAI-ANB に対して有意水準 5% のもとで有意に分類精度が高かった。したがって, 提案手法は 35~500 変数程度の大規模変数のデータに対して有効であることが示された。提案手法の分類精度が RAI-ANB より高い理由は, 信頼性の低い CI テストに対応するエッジの有無のパターンについてモデル平均を行うことで, RAI-ANB で実施した CI テストの誤りによる分類精度低下を抑制できたためである。また, 提案手法が学習する ANB は説明変数間の構造に制限を設けないが, Naive Bayes や TAN は構造に制限を設けるため, これらの BNC は ANB に比べて表現力が低下してしまい, 分類精度が低下する。

次に, RAI-ANB において条件 a, b を満たすエッジの数を表 6 に示す。表 2, 6 より, 大規模変数のデータでは小規模変数のデータに比べて条件 a, b を満たすエッジ数が多い。

これは大規模変数の場合はパラメータ数が増大し、データがスパースになるため、小規模変数の場合に比べて CI テストの信頼性が低くなる傾向があるためである [20, 27]. したがって、小規模変数のデータよりも大規模変数のデータの方が提案手法が有効であると考えられる.

以上より次の結果を得た.

- 実データを用いた評価実験において、提案手法は最大 500 変数の ANB のモデル平均を計算できた. 大規模変数のデータの場合、Naive Bayes, TAN, RAI-GBN, RAI-ANB より有意に分類精度が高かった.
- 提案の枝刈りにより、枝刈りを用いない場合に比べて、場合分け二分木を用いた k -best 探索の計算コストを削減できた.

5 むすび

本論文では、RAI-ANB が CI テストの信頼性を考慮せずに学習する問題を改善するため、信頼性が低い CI テストに対応するエッジが存在するかしないかの全てのパターンから周辺尤度上位 k 個の構造 (k -best) を選択し、モデル平均する手法を提案した。具体的には、(1) 対数 Bayes factor の値が予め設定した 0 近傍の閉区間に含まれる CI テストは信頼性が低いと判定し、(2) その信頼性の低い CI テストに対応するエッジが存在する場合と存在しない場合の全てのパターンを表す場合分け二分木を提案し、(3) この二分木から周辺尤度上位 k 個のパス (ANB 構造) を有効な枝刈りにより効率的に探索する手法を提案した。実データを用いた評価実験の結果、提案の枝刈りを行うことで場合分け二分木を用いた k -best 探索の計算コストを削減できた。それにより、提案手法は 500 変数の大規模 ANB のモデル平均を計算でき、大規模変数のデータにおいて RAI-ANB よりも分類精度が有意に高いことを示した。

今後の課題として、提案手法の計算コスト削減が挙げられる。場合分け二分木の圧縮表現として Zero-suppressed Binary Decision Diagrams (ZDD)[28] が知られている。場合分け二分木より節点数が少ない ZDD を探索グラフに用いることで、計算コストを削減できると考えられる。

謝辞

本論文を作成するにあたり，指導教員の植野真臣教授から，丁寧かつ熱心なご指導を賜りました．ここに感謝の意を表します．また，日頃から親身になって研究を支えていただいた宇都雅輝准教授に深謝いたします．そして，ゼミや日常の議論を通じて多くの示唆や知識を頂いた菅原聖太助教，研究室の先輩・同期・後輩に感謝いたします．

参考文献

- [1] Marvin Minsky. Steps toward Artificial Intelligence. Proceedings of the IRE, 49(1):8–30, January 1961.
- [2] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian Network Classifiers. Machine Learning, 29(2):131–163, November 1997.
- [3] Shouta Sugahara, Koya Kato, and Maomi Ueno. Learning bayesian network classifiers to minimize the class variable parameters. In Proceedings of the AAAI conference on artificial intelligence, 2024.
- [4] David Heckerman, Dan Geiger, and David M. Chickering. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. Machine Learning, 20(3):197–243, September 1995.
- [5] Tomi Silander and Petri Myllymäki. A simple approach for finding the globally optimal Bayesian network structure. In Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence, UAI’06, pages 445–452, Arlington, Virginia, USA, July 2006. AUAI Press.
- [6] Shouta Sugahara, Masaki Uto, and Maomi Ueno. Exact learning augmented naive bayes classifier. In Václav Kratochvíl and Milan Studený, editors, Proceedings of the Ninth International Conference on Probabilistic Graphical Models, volume 72 of Proceedings of Machine Learning Research, pages 439–450. PMLR, 11–14 Sep 2018.
- [7] 菅原 聖太 and 植野 真臣. Augmented Naive Bayes 制約を持つベイジアンネットワーク分類器の厳密学習. 電子情報通信学会論文誌 D, J103-D(4):301–313, 2020.
- [8] David Maxwell Chickering and David Heckerman. A comparison of scientific and engineering criteria for Bayesian model selection. Statistics and Computing, 10(1):55–62, January 2000.
- [9] Jin Tian, Ru He, and Lavanya Ram. Bayesian Model Averaging Using the k-best Bayesian Network Structures. In Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, pages 589–597, 2010.
- [10] Yetian Chen and Jin Tian. Finding the k-best equivalence classes of bayesian network structures for model averaging. In Proceedings of the AAAI conference on artificial intelligence, pages 2431–2438, 2014.

- [11] Shouta Sugahara, Itsuki Aomi, and Maomi Ueno. Bayesian Network Model Averaging Classifiers by Subbagging. In Proceedings of the 10th International Conference on Probabilistic Graphical Models, pages 461–472. PMLR, February 2020. ISSN: 2640-3498.
- [12] Shouta Sugahara, Itsuki Aomi, and Maomi Ueno. Bayesian network model averaging classifiers by subbagging. Entropy, 24(5), 2022.
- [13] Judea Pearl. Causality: Models, Reasoning and Inference. Cambridge University Press, 2000.
- [14] Peter Spirtes, Clark Glymour, and Richard Scheines. Causation, Prediction, and Search. MIT Press, 2000.
- [15] Ioannis Tsamardinos. The max-min hill-climbing bayesian network structure learning algorithm. Machine Learning, 65(1):31–78, 2006.
- [16] Raanan Yehezkel and Boaz Lerner. Bayesian network structure learning by recursive autonomy identification. Journal of Machine Learning Research, 10:1527–1570, 2009.
- [17] Kazuki Natori, Masaki Uto, and Maomi Ueno. Consistent learning bayesian networks with thousands of variables. In Advanced Methodologies for Bayesian Networks (Proc. Machine Learning Research), volume 73, pages 57–68, 2017.
- [18] 名取和樹, 宇都雅輝, and 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. 電子情報通信学会論文誌 D, J101-D(5):754–768, 2018.
- [19] Shouta Sugahara, Wakaba Kishida, Koya Kato, and Maomi Ueno. Recursive autonomy identification-based learning of augmented naive Bayes classifiers. In Proceedings of The 11th International Conference on Probabilistic Graphical Models, pages 265–276. PMLR, September 2022.
- [20] 菊谷成慎, 菅原聖太, 名取和樹, and 植野真臣. Augmented naive bayes による大規模ベイジアンネットワーク分類器学習. 電子情報通信学会論文誌 D, J104-D(1):65–81, 2021.
- [21] Robert E. Kass and Adrian E. Raftery. Bayes factors. Journal of the American Statistical Association, 90(430):773–795, 1995.
- [22] Wray Buntine. Theory refinement on bayesian networks. In Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence, UAI’91, page 52–60, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

- [23] H. Steck and T.S. Jaakkola. On the dirichlet prior and bayesian regularization. Neural Information Processing Systems (NIPS 2002), pages 697–704, 2002.
- [24] 加藤弘也, 菅原聖太, and 植野真臣. 深さ優先分枝限定法による目的変数パラメータ数を最小化するベイジアンネットワーク分類器学習. 電子情報通信学会論文誌 D, 2024.
- [25] Maomi Ueno. Robust learning bayesian networks for prior belief. In UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, pages 698–707. AUAI Press, 2011.
- [26] G. Hommel. A stagewise rejective multiple test procedure based on a modified bonferroni test. Biometrika, 75(2):383–386, 1988.
- [27] 本田 和雅, 名取 和樹, 菅原 聖太, 磯崎 隆司, and 植野 真臣. 推移性を利用した大規模ベイジアンネットワーク構造学習. 電子情報通信学会論文誌 D, J102-D(12):796–811, 2019.
- [28] Shinichi Minato. Zero-suppressed BDDs for set manipulation in combinatorial problems. In Proceedings of the 30th international Design Automation Conference, pages 272–277, 1993.