

## GPU によるベイジアンネットワーク学習の高速化

森 隆史 <sup>1,a)</sup> 吉廣 卓哉 <sup>1,b)</sup>

ベイジアンネットワークは、事象間の因果関係を条件付確率に基づいてモデル化した確率モデルであり、様々な分野で因果分析や意思決定に利用されている。優れたベイジアンネットワークを効率的に学習するアルゴリズムとして、順序制約に基づいたものや遺伝的アルゴリズムに基づいたもの等、複数のアプローチから多数のアルゴリズムが提案されている。しかし、ベイジアンネットワーク学習問題は NP 困難であり、事象数が多い大規模なベイジアンネットワークを学習するためには膨大な時間がかかる問題がある。本研究では、GPU を用いて学習アルゴリズムを並列化することで、従来よりも高速にベイジアンネットワークを学習する方法について検討する。

## 1. はじめに

ベイジアンネットワークモデルとは、事象を「ノード」その因果関係を「リンク」で表した確率モデルであり、予測や原因推定に用いられる。ベイジアンネットワークで膨大なデータの解析を行うことで、医学、意思決定支援など様々な分野での応用が期待されている。

ベイジアンネットワークモデルは事象の観測データを元にモデルを学習する。モデルの候補を複数生成し、それぞれの候補に対し評価を行う。そして、評価の最も良かったモデルが、事象の観測データを良く表すベイジアンネットワークモデルとなる。つまり、実際の観測データを最も良く説明するモデルであり、未来の事象を予測するための確率モデルとして活用できる。

しかし、ベイジアンネットワークの事象の数が増加するにしたがって、探索すべきモデルの候補も増加する。結果、膨大なモデルの候補から最適なモデルを求めることになる。膨大なモデルの候補から最適なモデルを求める問題は NP 困難な問題であることがわかっている<sup>1)</sup>。このため、現実的な実行時間でそれなりに優れたモデルを求めるアルゴリズムが多数提案されているが、大規模なベイジアンネットワークの計算は未だに膨大な時間がかかる。

本研究では、GPU を用いて並列処理を行うことで負荷分散させ、大規模なベイジアンネットワークを高速に学習することが目的である。具体的には、観測データから候補となるモデルを複数生成し、各モデル候補をそれぞれ評価する。その各モデルの評価を GPU を用いて並列処理を行うことで高速に最適なモデルを求める。

## 2. ベイジアンネットワーク学習手法

最適なベイジアンネットワークを学習することは NP 困難な問題である。よって近似解を高速に求めるために、全順序関係を制約とした K2<sup>2)</sup>や、遺伝的アルゴリズムである PBIL-RS<sup>3)</sup>、CGA、UMDA などを用いた様々な学習手法が提案されている。K2 アルゴリズムは、あらかじめ事象間の全順序関係を知っておかなければならない。これを制約

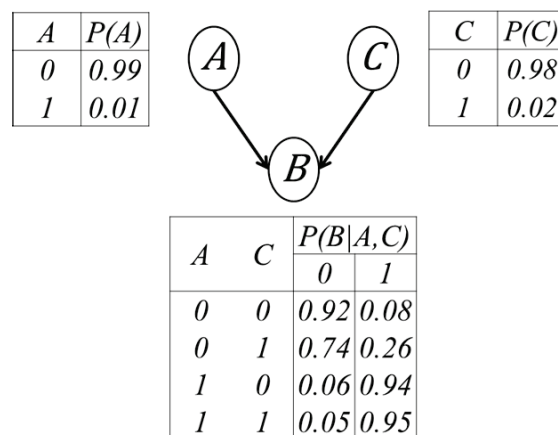


図 1 ベイジアンネットワークモデルの例

としてモデルの候補を生成することで、モデルの探索空間を小さくすることができる。また PBIL-RS は良いモデルの候補の特性を学習させ、再度モデルの候補を生成する。これを繰り返すことにより最適なモデルを求める。

一般にモデルの学習手順は、モデル候補の複数生成する。生成したモデルに対してそれぞれ情報量基準を計算し、それを「スコア」とする。そしてスコアの良いモデルを残す。これらの操作を繰り返すことで最適に近いモデルを求める。スコア算出について図 1 を用いて説明する。ノード A, B, C はそれぞれ事象を表しており、事象が生起するとき 1 をとり、しなければ 0 をとる。リンク A→B は因果関係を表し A を親ノード、B を子ノードと呼ぶ。A と C の生起確率は、それぞれ P(A) と P(C) でこれにより A と C 値が決まる。ノード B は A と C を親ノードに持つので、条件付確率 P(B|A,C) に従って B の値が決まる。図 1 のようなモデルの場合、A と C の親ノードは無く、B の親ノードは A と C であるので、モデルの親子関係（親がない場合も 1 つとする）は全部で 3 つとなる。この 3 つの親子関係について、観測データと各ノードの親子関係を元に、それぞれスコア算出し、足し合わせたものが図 1 のモデルのスコアとなる。よってノード数が増加するに従い、ノードの親子関係の数も増加するので、1 モデルのスコア算出に多くの時間がかかる。

1 和歌山大学  
a) s171053@center.wakayama-u.ac.jp  
b) tac@sys.wakayama-u.ac.jp

遺伝的アルゴリズム PBIL-RS に基づいたベイジアンネットワーク学習手法は各ノード間にリンクを張る確率を示したベクトル  $P$  を用意する。確率ベクトル  $P$  に基づいて確率的にモデルを複数生成する。そして、それぞれのモデルのスコアを算出し、ある閾値以上のスコアの良いモデルを選出する。その選出した複数ある良いモデルの各ノード間のパターンを確率ベクトル  $P$  に反映させ更新する。このようにモデルの生成、スコアの算出、確率ベクトルの更新という操作を 1 世代として、何世代も繰り返すことにより確率ベクトル  $P$  を収束させて最適なモデルを求める。

### 3. GPU による負荷分散手法

#### 3.1 GPU のアーキテクチャ

本研究では GPU を用いるため、想定する GPU のアーキテクチャを図 2 に示し、これを用いて説明する。GPU 上にはストリーミングマルチプロセッサ (SM) という演算部が複数ある。その中に最小単位の演算ユニットであるストリーミングプロセッサ (SP)、レジスタ、同 SM 内の SP が共有できるシェアードメモリが搭載されている。また GPU 上の各 SM が共有できるグローバルメモリがある。プログラムにおいて、最小の処理単位を「スレッド」といい、スレッドをまとめたものを「ブロック」という。スレッドは SP、ブロックは SM に対応する。GPU は膨大な処理を複数のスレッドに割当て並列に実行することで高速計算を実現している。またプログラム実行の流れは以下になる。

- (1) CPU (ホスト) 側で GPU (デバイス) 側のメモリの宣言確保を行う
- (2) デバイス側のグローバルメモリに並列処理したいデータを転送する
- (3) ホスト側で並列処理を行うプログラム (カーネル関数) 起動させ、デバイス側は結果データをグローバルメモリに保持する
- (4) デバイス側は結果データをホスト側に転送し、メモリ解放を行う

#### 3.2 提案アルゴリズム

提案手法では、ベイジアンネットワーク学習手順における複数のモデル候補のスコア算出に着目する。生成した各モデルを 1 ブロック、そのモデルの各ノードを 1 スレッドとして、並列処理することで高速に複数モデルのスコア算出を行う。

一般的なベイジアンネットワーク学習アルゴリズムは、次の手順に当てはめることができる。

- (a) 複数のモデル候補を生成する
- (b) 生成した複数モデルのスコア算出を CPU で逐次実行する
- (c) スコアの良いモデルを記憶し、(a)に戻る

従来手法では生成した複数モデルに対して 1 モデルずつス

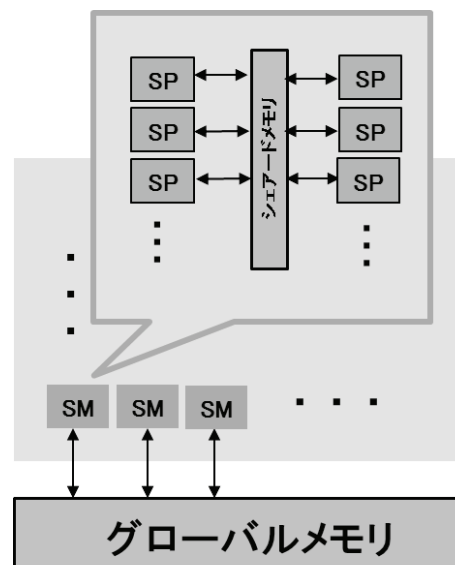


図 2 GPU のアーキテクチャ

```

1: global int model[x][m][m];
2: global int data[s][m];
3: global double score[x];
4:
5: shared double tmp_score;
6: i = blockIdx; j = threadIdx;
7: S = CreateParentSet(i,j)
8: For each assignment patterns T of S {
9:     CountingData(T)
10:    tmp_score += AIC(T)
11: }
12: score[i] += tmp_score;

```

図 3 提案する並列処理アルゴリズム

コアを算出するため、モデルの生成数の増加やノードが増加すると計算に多大な時間がかかる。

そこで提案手法では複数モデルのスコア算出である手順 (b) を並列処理することでより高速にベイジアンネットワークの学習を行う。提案手法では、手順 (b) を以下に示すように手順 (b1) ~ (b3) に分けることによって、並列処理を実現する。

- (a) 複数のモデル候補を生成する
- (b1) 生成した複数モデルのデータ、観測データを GPU (デバイス) 側のグローバルメモリに転送する
- (b2) 1 つのモデルを 1 ブロック、そのモデルの 1 つのノードを 1 スレッドに割当て、複数モデルのスコア算出を並列実行する
- (b3) 算出した複数モデルのスコアを CPU (ホスト) 側に転送する
- (c) スコアの良いモデルを記憶し、(a)に戻る

提案手法の手順(b2)について図3を用いて説明する。x はモデルの生成数、m はモデルのノード数、s は観測データのサンプル数を表す。model[x][m][m]は、手順(a)で生成した x 個のモデルの  $m \times m$  の隣接行列、data[s][m] はサンプル回数 s、事象 m 個の観測データ、score[x]は x 個の各モデルのスコアを示す。手順(b1)でそれぞれ model, data, score をグローバルメモリに確保しておく。5 行目の tmp\_score は、モデルの各ノードのスコアを一時保存する変数で、書き込み回数が多くなるためグローバルメモリより高速なシェアードメモリに領域を確保する。6 行目の i はブロック番号、j はスレッド番号を示し、ブロック i のスレッド j にモデル i のノード j の処理を割り当てる。7 行目の  $S = \text{CreateParentSet}(i, j)$  は、モデル i のノード j に対する親ノードの集合 S を作成する。8 行目の T は、S の各ノードが取り得る値の全組み合わせの集合を示す。図2のようなモデルで、ノード B のスコアを算出する場合、 $S = \{A, C\}$ 、 $T = \{00, 01, 10, 11\}$  となる。9 行目の CountingData(T)は、観測データ data 中に、集合 T のある 1 つの要素と合致するものを数える。そして、10 行目の AIC(T)で、ノード j の親ノードの 1 つの組み合わせ(集合 T の 1 要素)についてスコアを算出する。9~10 行目を T の要素ごとに繰り返すことでノード j のスコアを算出する。12 行目でノード j のスコアを score[i]に足し込む。この操作をモデル i 中の他ノードにも行うことで、モデル i のスコアを算出する。生成したすべてのモデルに対して同様の操作を並列に行うことで、モデルのスコア算出を高速化する。

#### 4. おわりに

本研究では、GPU を用いてベイジアンネットワークの学習の負荷分散を行い、より高速に最適なモデルを求める方法を提案した。モデル学習手順のスコア算出に着目し、膨大な複数モデルのスコア算出の手順を並列化することで学習の高速化する。

今後は性能評価を行い、またベイジアンネットワーク学習において、さらに並列処理が可能な部分をみつけ適応することである。

#### 参考文献

- 1) D. M. Chickering, D. Heckerman, C. Meek, “Large-Sample Learning of Bayesian Networks is NP-Hard,” Journal of Machine Learning Research, Vol. 5, pp. 1287-1330, 2004.
- 2) G. F. Cooper, and E. Herskovits, “A Bayesian method for the induction of probabilistic networks from data.” Machine Learning, Vol. 9, pp. 309-347, 1992.
- 3) 福田 翔, 吉廣 卓哉, “遺伝的アルゴリズムによる確率学習に基づいたベイジアンネットワークモデル学習手法” 和歌山大学大学院コミュニケーション科学クラス修士論文 2014.