数理解析研究所講究録
第2083巻 2018年 1-7

高さ2の多分岐AND-OR木に対する探索アルゴリズム

重水美香* 宇佐美紘貴

首都大学東京大学院理工学研究科数理情報科学専攻

Abstract

Tarsiは, ある性質を持つAND-OR木の葉に確率分布が独立同分布(IID)で与えられているならば, 最適かつ深さ優先なアルゴリズムが存在することを示した. では, 確率分布が独立分布(ID)で与えられている場合はどうだろうか. 鈴木登志雄により, 高さ3の場合には反例が存在することが分かっている. そこで, 我々は高さ2の多分岐AND-OR木に対して, 確率分布が独立分布(ID)で与えられているならば, 最適かつ深さ優先なアルゴリズムが存在することを示した.

1 序

ブール関数の一種であるAND-OR関数を木として表したものをAND-OR木という. 根に$\wedge$がラベル付けされ, その子である内部節には$\vee$, それ以下は$\wedge$と$\vee$が交互にラベル付けされている. 各葉には0または1の値が与えられ, 各内部節に付けられたラベル($\wedge$または$\vee$)に応じて, 根が0または1を出力する. アルゴリズムが葉にクエリ(葉の値が0なのか1なのか質問すること)を行うことによって, 根の値を明らかにする. 木に対して, アルゴリズムの個数は有限個であり, クエリする順番や葉の値によって, 行わなければならないクエリの回数に差が生まれる.

様々な形の木に対して, どのようなアルゴリズムを用いれば, 効率よく(クエリの回数を少なく)根の値を知ることができるのだろうか. この問題は計算量理論の観点から, 非常に興味深いものである.

Tarsiはある性質を持つAND-OR木で葉に確率分布が独立同分布(IID)で与えられている場合, 最適かつ深さ優先なアルゴリズムであることを示した. また, 鈴木登志雄は葉に確率分布が独立分布(ID)で与えられている場合, 高さ2の2分木に関しては最適かつ深さ優先なアルゴリズムが存在することを示した. 一方, 高さ3の2分木に関してはある確率分布が存在し, いかなる深さ優先アルゴリズムよりもコスト期待値が小さい非深さ優先アルゴリズムが存在することも示した.

*Coresponding author. lovegirl.mika@hotmail.com

我々は, 確率分布が独立分布(ID)で与えられている場合について, 高さ2で多分岐の場合にも最適かつ深さ優先なアルゴリズムが存在することを示した.

2 定義

2.1 AND-OR木

ブール関数の一種であるAND-OR関数を木で表したものをAND-OR木という．木を構成する要素には節と枝がある．最上部の節を根といい，子を持たない節を葉という．根でも葉でもない節を内部節という．

節の深さとは，根を深さ0として，深さ$n(n \geq 0)$の節の子を深さ$n+1$の節と定義する．

本論文で考えるAND-OR木とは，根に$\wedge$がラベル付けされており，その下の内部節からは$\vee$と$\wedge$が交互にラベル付けされているものをいう．つまり，根と深さ$m(m \geq 1)$の内部節に対して，以下のようなラベル付けがされている．

- 根(深さ0)には$\wedge$がラベル付けされている．
- mが奇数の内部節には，$\vee$がラベル付けされている．
- mが偶数の内部節には，$\wedge$がラベル付けされている．

葉の深さの中で最大のものを木の深さ(高さ)という．

葉に0または1の値が与えられると，内部節と根のラベルに基づいて根が0または1を出力する．高さ2のAND-OR木は，以下のような形のブール関数で表せる．

2.2 深さ優先アルゴリズム

深さ優先アルゴリズム(depth-first algorithm)とは，ある内部節v下の葉(vの子孫でかつ葉になっているもの)にクエリを行ったら，vの値が分かるまで，v下の葉以外の葉にクエリを行わないアルゴリズムである．深さ優先アルゴリズムではないアルゴリズムを非深さ優先アルゴリズムという．また，v下の葉を少なくとも一つクエリした後に，vの値が分かってないにもかかわらず，v下の葉以外の葉にクエリを行うことをnon-depth-first moveという．

また，ある内部節vの値が探索のある時点で決まった場合，その時点以降はvの子孫にクエリは行わないものとする．

2.3 ディレクショナルアルゴリズム

葉の集合上に全順序が探索に先立って定義され，その順序に反しないようにクエリを行うアルゴリズムをdirectional algorithmという．また，ディレクショナルではないアルゴリズムをnon-directional algorithmという．

本論文でのdirectional algorithmは深さ優先アルゴリズムのみではなく，非深さ優先アルゴリズムの場合も含むとする．

2.4 確率分布

葉に与えられた確率分布が独立同分布(independent and identical distribution:IID)であるとは，葉の値が0になる確率が葉によらず，各葉に等しく与えられていることである．

葉に与えられた確率分布が独立分布(independent distribution:ID)であるとは，葉の値が0を取る確率が葉によらず，各葉にそれぞれ与えられていることである．

2.5 アルゴリズムのコスト期待値

コスト期待値とは，AND-OR木と確率分布に対して，アルゴリズムの効率を測る指標である．

葉に与える値の集合をSとする．$s \in S$に対して，アルゴリズムAを用いた時のクエリの回数を$V_A(s)$と表記する．また，$s \in S$が葉に起こる確率を$p(s)$とする．アルゴリズムAのコスト期待値$V(A)$を以下のように定義する．

定義(アルゴリズムAのコスト期待値$V(A)$)

$$V(A) = \sum_{s \in S} p(s) V_A(s)$$

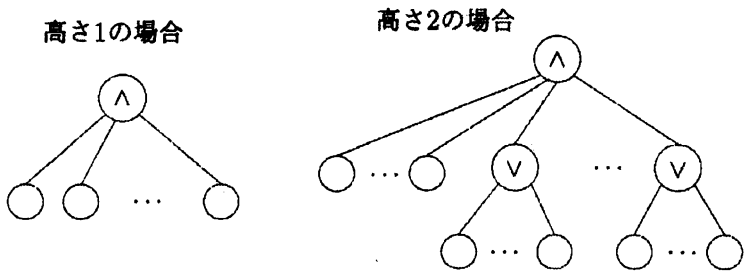
与えられたAND-OR木と確率分布に対して，$V(A)$が最小になるようなアルゴリズムAを最適であるという．

3 結果

本論文では, p_{ij}を葉x_{ij}の値が0になる確率とする.

まず, 考えるべきアルゴリズムを少なくするための重要な補題を以下で与える.

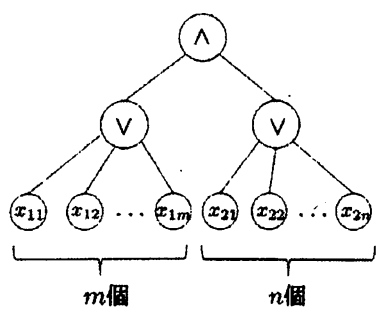
Lemma 3.1. 高さが1または2の多分岐*AND-OR*木(*OR-AND*木)を考える. ただし, *root* 直下に葉と*OR*ゲートが混在しているものも含むとする. 確率分布は*ID*で与えられるとする. (ただし, 確率は0,1ではないものとする.) アルゴリズムは*depth-first* かつ*directional* とする.
このとき, *AND*ゲート直下の葉は*0*になる確率が大きい順, *OR*ゲート直下の葉は*0*になる確率が小さい順に探索をした方がコスト期待値が小さくなる.
また, 途中で何回か*non-depth-first move* をはさんで*OR*ゲート(*AND*ゲート)に戻ってくる場合も同様である.



Theorem 3.2. 高さ2の*AND-OR*木で以下の形のものを考える. *root* の子が2つの*OR*ゲートで, 一方の*OR*ゲートの子はm個($m \geqq 2$)かつ他方の子はn個($n \geqq 2$)とする. 確率分布は*ID*で与えられるとする. (ただし, 確率は0,1ではないものとする.) アルゴリズムは*directional* の場合(*non-depth-first algorithm*も含む)に限定する.

$$p_{11}\cdots p_{1m}(1+p_{21}+p_{21}p_{22}+\cdots+p_{21}\cdots p_{2\,n-1})$$
$$\geqq p_{21}\cdots p_{2n}(1+p_{11}+p_{11}p_{12}+\cdots+p_{11}\cdots p_{1\,m-1})$$

のとき, 左*subtree* から探索を始める*depth-first algorithm* が最適となる.
(不等号が逆の場合は, 左*subtree* と右*subtree* を入れ換えた木を考えればよい.)



Proof. 補題3.1 より, $p_{11} \leqq \cdots \leqq p_{1m}, p_{21} \leqq \cdots \leqq p_{2n}$とし, 各subtreeの葉は0になる確率が小さい順に探索をするものとしてよい.$\cdots(*)$
mに関する帰納法で示す.
(I)$m=2$のとき
(case1) x_{11}から探索を始めるとき
depth-first algorithmは,以下のA_L^{dep}のみである.

$$A_L^{dep} : x_{11} \to x_{12} \to x_{21} \to \cdots \to x_{2n}$$

また,non-depth-first algorithmは, 以下のA_k $(1 \leqq k \leqq n)$のみである.

$$A_k : x_{11} \to x_{21} \to \cdots \to x_{2k} \to x_{12} \to x_{2\,k+1} \to \cdots \to x_{2n}$$

ここで, コスト期待値を計算する.
$1 \leqq k \leqq n-2$の場合

$$V(A_{k+1}) > V(A_k)$$

よって,$V(A_1), \cdots, V(A_{n-1})$の中では$V(A_1)$が最小値をとる.
また,

$$V(A_1) > V(A_L^{dep})$$

ゆえに,x_{11}から見るものとしたとき最適なアルゴリズムの候補は$V(A_L^{dep}), V(A_n)$となる.
(case2) x_{21}から探索を始めるとき
depth-first algorithmは, 以下のA_R^{dep}のみである.

$$A_R^{dep} : x_{21} \to \cdots \to x_{2n} \to x_{11} \to x_{12}$$

次に,non-depth-first algorithmは, 以下のB_k $(1 \leqq k \leqq n-1)$, $C_{k,\ell}(k+\ell = 2, \cdots, n,\ k \geqq 1,\ \ell \geqq 1)$の2通りである.

$$B_k : x_{21} \to \cdots \to x_{2k} \to x_{11} \to x_{12} \to x_{2\,k+1} \to \cdots \to x_{2n}$$
$$C_k : x_{21} \to \cdots \to x_{2k} \to x_{11} \to x_{2\,k+1} \to x_{2\ell} \to x_{12} \to x_{2\,\ell+1} \to \cdots \to x_{2n}$$

コスト期待値を計算すると,

$$V(B_k) = V(A_k)$$
$$V(C_{k,\ell}) = V(A_{k+\ell})$$

結局, $V(A_L^{dep}), V(A_R^{dep}), V(A_n)$の大小関係を比べればよい.

$$V(A_n) > V(A_R^{dep})$$

仮定の条件式より,

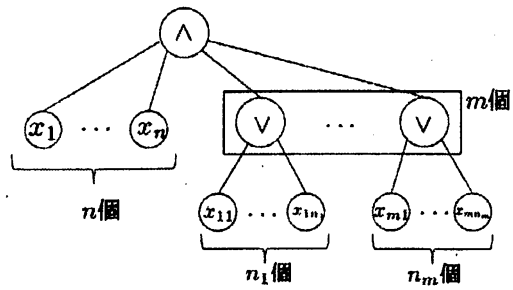
$$V(A_R^{dep}) \geqq V(A_L^{dep})$$

よって, 最適なアルゴリズムはA_L^{dep}となる.
(II)$m-1$のとき成り立つと仮定して、mのときにも成り立つことを示す.
同様の手法により示される.
(I), (II)よりすべてのmについて成り立つ.

□

主結果を証明する過程で必要となるため, ルート直下に葉が存在する場合も証明する.

Theorem 3.3. 高さ2の*AND-OR*木で以下の形のものを考える. *root*の子がn個($n \geqq 1$)の葉とm個($m \geqq 1$)の*OR*ゲートで, *OR*ゲートの子はそれぞれ$n_1, \cdots, n_m$個($n_1, \cdots, n_m \geqq 2$)とする. 確率分布は*ID*で与えられるとする. (ただし, 確率は0, 1ではないものとする.) アルゴリズムは*directional* の場合(*non-depth-first algorithm*も含む)に限定する.
このとき, *depth-first* となる最適なアルゴリズムが存在する.



証明には帰納法を用いる.
手順は以下の通りである.

すべてのnon-depth-first algorithmを考える.
そのnon-depth-first algorithm Aに対して, depth-first algorithm A^{dep}を構成する. 具体的な構成方法としては, non-depth-first moveが起こるタイミングで, その直後に探索する予定だった処理を先に行うようにする方法である.

$$A : B_0 \to x_{i1} \to B_1 \to x_{i2} \to B_2$$

$$A^{dep} : B_0 \to B_1 \to x_{i1} \to x_{i2} \to B_2$$

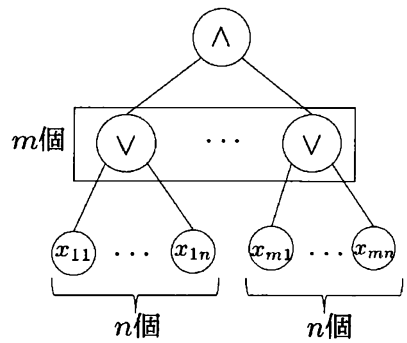
このようなアルゴリズムに対してコスト期待値を計算することにより

$$V(A^{dep}) \leqq V(A)$$

が得られる.

ルート直下に葉がない場合も同様に帰納法を用いることで証明される.

Theorem 3.4. (主結果)
高さ2の*AND-OR*木で以下の形のものを考える. *root*の子がm個($m \geqq 2$)の*OR*ゲートで, それらの*OR*ゲートの子はn個($n \geqq 2$)とする. 確率分布は*ID*で与えられるとする. (ただし, 確率は0,1ではないものとする.) アルゴリズムは*directional* の場合(*non-depth-first algorithm*も含む)に限定する.
このとき, *depth-first* となる最適なアルゴリズムが存在する.



References

[1] S.Arora and B.Barak, Computational complexity : A modern approach, Cambridge university press, NewYork, 2009

[2] M.Saks and A.Wigderson, Probabilistic boolean decision trees and the complexity of evaluating game trees, *ini Proc.27th Annual IEEE Symposium on Foundations of Computer Science*, pp.29-38, 1986

[3] T.Suzuki, Non-Depth-First Search against Independent Distributions on an AND-OR Tree, *preprint*, arXiv:17709.07358v1[cs.DS](2017)

[4] T.Suzuki,Y.Niida, Equilibrium points of AND-OR tree: Under constraints on probability, *Annals of Pure and Appliced Logic 166(2015)*, 1150-1164

[5] M.Tarsi, Optimal search on some game trees,*Journal of the Association for Computing Machmery*, Vol.30, pp.389-396, 1983