The Eigen distribution for multi-branching Boolean trees on ID

Weiguang Peng

Southwest University, China

CACML, 2020

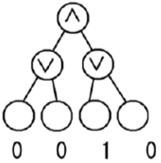
Outline

1. Introduction
   - Terminology
   - Backgrounds
2. The Eigen-distribution for multi-branching trees under ID
   - The Eigen-distribution for balanced multi-branching trees
   - The Eigen-distribution for weighted trees
3. Optimal algorithms for multi-branching trees under ID
   - DIR_d is optimal among all depth-first algorithms
   - DIR_d is optimal among all non-depth-first algorithms
4. Future Research

AND-OR trees

An AND-OR tree is a tree whose root is labeled by AND and nodes are level-by-level labeled by OR or AND alternatively except for leaves.

Each leaf is assigned Boolean value 1 or 0, where 1 denotes true and 0 denotes false.



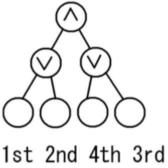
$$f(0,0,1,0) = (0 \vee 0) \wedge (1 \vee 0) = 0$$

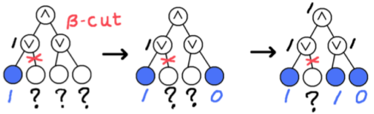
α-β pruning algorithms

An α-β pruning algorithm satisfies the following conditions.

- When an algorithm knows a child of an AND-node has value 0, it recognizes that the value of AND-node is 0 without probing other children (α-cut).
- When an algorithm know a child of an OR-node has value 1, it recognizes that the value of OR-node is 1 without probing other children (β-cut).

α-β pruning algorithm $A = 1243$.





$C(A, \omega)$: the number of leaves checked by A under assignment ω.

Distributions on assignments

Let d be a (probability) distribution on the set Ω of assignments, the expected cost of algorithm A under the distribution d is defined by

$$C(A,d) := \sum_{\omega \in \Omega} C(A,\omega) d(\omega).$$

Let D be the set of distributions and $\mathcal{A}$ the set of deterministic algorithms computing tree T.

The *distributional complexity* computing tree T is defined by

$$\max_{d \in D} \min_{A \in \mathcal{A}} C(A,d).$$

A distribution d is said to be an *eigen-distribution* if

$$\min_{A \in \mathcal{A}} C(A,d) = \max_{d' \in D} \min_{A \in \mathcal{A}} C(A,d').$$

Background

- Saks and Wigderson (1986) showed that the randomized complexity of n-branching trees with height h is $\Theta\left(\left(\frac{n-1+\sqrt{n^2+14n+1}}{4}\right)^h\right)$.
- Yao's Principle (1977) implies that the **randomized complexity** equals to the **distributional complexity**.

$$\underbrace{\min_{A_R \in \mathcal{A}_R} \max_{\omega} C(A_R, \omega)}_{\text{Randomized complexity}} = \underbrace{\max_{d} \min_{A \in \mathcal{A}} C(A, d)}_{\text{Distributional complexity}}.$$

where $\mathcal{A}_R$ denotes the class of probability distribution over the family of deterministic algorithms.

IID

- D:= the set of all distributions
- ID:= the set of all independent distributions.
- IID:= the set of all independent identical distributions.

Let Ω be the set of assignments for a given tree. We say $d : \Omega \to [0, 1]$ is an *independent distribution* (denoted by $d \in$ ID) if there exist p_i's (the probability of the i-th leaf that has value 0) such that for any $\omega \in \Omega$,

$$d(\omega) = \prod_{\{i:\ \omega(i)=0\}} p_i \prod_{\{i:\ \omega(i)=1\}} (1 - p_i).$$

We say $d \in$ IID if d is an ID satisfying $p_1 = p_2 = \cdots = p_n$.

T_2^h

	eigen-distribution is unique (D)	$d \in ID \to d \in IID$
T_2^h	• Liu-Tanaka (2007) • False w.r.t. only directional-alg by Suzuki and Nakamura (2013)	• Claimed by Liu-Tanaka (2007) • Justified by Suzuki and Niida (2015)

Question

How about multi-branching trees, especially T_n^h, Balanced Multibranching trees?

A tree is *balanced* if each nonterminal node at the same level has the same number of children.

Note that we do not require that nodes from different levels have the same number of children

The n-branching tree with height h is denoted by T_n^h.

T_n^h(Balanced Multi-branching trees)

	eigen-distribution is unique (D)	$d \in ID \to d \in IID$
T_2^h	• Liu-Tanaka (2007) • False w.r.t. only directional-alg by Suzuki and Nakamura (2013)	• Claimed by Liu-Tanaka (2007) • Justified by Suzuki and Niida (2015)
T_n^h	• Holds w.r.t. deterministic-alg • by Peng et al. (2016)	• Holds if we restrict $0 < r < 1$ by Peng et al (2017)

Remark: r is the probability of root being 0.

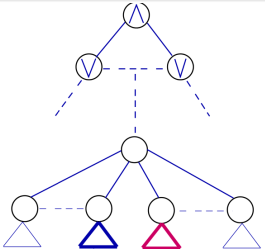
Let ID(r) denote the set of independent distributions which induce that the probability of the root having value 0 is r.

Theorem (Peng *et al.* (2017))

For any balanced multi-branching AND-OR tree $\mathcal{T}$, we fix $\delta \in \mathrm{ID}(r)$ and $0 < r < 1$. If the following equation holds,

$$\min_{A:\mathrm{depth}} C(A, \delta) = \max_{d \in \mathrm{ID}(r)} \min_{A:\mathrm{depth}} C(A, d),$$

then $\delta \in \mathrm{IID}$.



We can show the following conclusion which is a generalization of Suzuki-NIida's result in Suzuki (2015).

Theorem (1)

Suppose that T is an n-branching AND-OR tree (OR-AND tree). Let $r \in \{0, 1\}$, d_0 be the IID such that each leaf has the probability $1 - r$. Then,

- *in the case where the height of T is even, denoted by $h = 2k$,* $\min_A C(A, d_0) = n^k$.
- *In the case where h is odd, denoted by $2k + 1$,*
  - $\min_A C(A, d_0) = n^k$ *if we consider AND-OR tree and $r = 0$, or we consider OR-AND tree and $r = 1$.*
  - $\min_A C(A, d_0) = n^{k+1}$, *otherwise.*

ID implies IID

Theorem (2019)

For any n-branching tree T, suppose that d_1 is an ID such that the following holds.

$$\min_{A} C(A, \delta) = \max_{d} \min_{A} C(A, d),$$

where d over all IDs and A over all depth-first algorithms. Then δ is an IID.

Sketch of proof: This theorem holds in the case $0 < r < 1$ by Peng *et al.* (2017), the left work is to investigate the case $r = 0$ and $r = 1$. It is enough to show the following claim.

Claim: When $r = 0$ or 1, there exists r_0 such that

$$\min_{A} C(A, \delta) < \max_{d \in ID_{r_0}} \min_{A} C(A, d),$$

where $\delta \in IID$ such that probability of the root is r, and d over all IDs such that the probability of the root is r_0.

Proof of Claim: Let x be the probability of each leaf having value 0, r_x be the probability of the root having value 0 with respect to x. Given an $d \in IID$, for any depth-first algorithm A, we get the same expected cost. i.e., $\min_{A_0} C(A_0, d) = C(A, d)$.

Let $x = 1/2$, it is clear that for any depth-first algorithm A,

$$C(A, d_{1/2}) \leq \max_{d \in ID_{r_{1/2}}} \min_{A} C(A, d). \qquad (*)$$

We also can show the following conclusions:

- In the case $h = 2k$, $C(A, d_{r_{1/2}}) > n^k$,
- In the case where $h = 2k + 1$,
  - $C(A, d_{r_{1/2}}) > n^k$ if we consider AND-OR tree and $i = 0$, or we consider OR-AND tree and $i = 1$.
  - $C(A, d_{r_{1/2}}) > n^{k+1}$, otherwise.

By Theorem 1, $\min_A C(A, \delta) < C(A, d_{r_{1/2}})$ holds. we complete the proof of the claim. □

Non-depth-first-algorithms

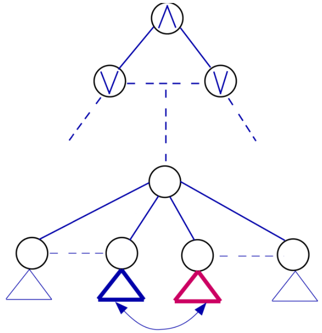
With the condition $0 < r < 1$, Suzuki(2018) extended our results to the case where non-depth-first algorithms are taken into consideration.

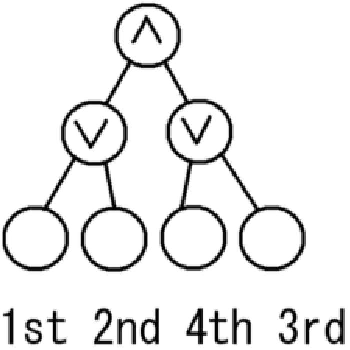
Definition (Depth-first-algorithms)

if an algorithm evaluates the value of one subtree, it will never evaluate the others until it completes the current one.

Otherwise, it is called **non-depth-first-algorithm**.
Thus, the above theorem still holds with respect to non-depth-first algorithms.

Non-depth-first-algorithms





Weighted trees

Definition (Okisaka *et al.* 2017)

Let A be an algorithm, ω an assignment, $\sharp_1(A, \omega)$ (resp., $\sharp_0(A, \omega)$) denote the number of leaves probed by A and assigned 1 (resp., 0) on ω. For any positive real numbers a, b,

$$C(A, \omega; a, b) := a \cdot \sharp_1(A, \omega) + b \cdot \sharp_0(A, \omega),$$

is called a *generalized cost* weighted with (a, b). Obviously, $C(A, \omega) = C(A, \omega; 1, 1)$.
For a distribution d on Ω, the *expected generalized cost* $C(A, d; a, b) := \sum_{\omega \in \Omega} d(\omega) \cdot C(A, \omega; a, b)$. We may say that $\mathcal{T}$ is a tree weighted by (a, b) if we consider the generalized cost.

Note that: for weighted trees, the weight not dependent on the assigned value.

We consider an IID on $\mathcal{T}_n^1(a, b)$ such that each leaf is assigned 0 with probability x. The expected cost is denoted by $C(x, a, b)$.

Lemma (Technical Lemma)

Suppose that the distribution on $\mathcal{T}_n^1$ weighted with (a, b) is an IID with all leaves assigned probability x. Then

(1) $p(x)$ *is a strictly increasing function of* x.

(2) $\dfrac{C(x, a, b)}{p(x)}$ *is strictly decreasing.*

(3) $\dfrac{C'(x, a, b)}{p'(x)}$ *is strictly decreasing.*

Theorem (Peng *et al.* 2019)

For any balanced multi-branching AND-OR tree $\mathcal{T}$ weighted by (a, b), we fix $\delta \in \mathrm{ID}(r)$ and $0 < r < 1$. If the following equation holds,

$$\min_{A:\mathrm{depth}} C(A, \delta, a, b) = \max_{d \in \mathrm{ID}(r)} \min_{A:\mathrm{depth}} C(A, d, a, b),$$

then $\delta \in \mathrm{IID}$.

Notations

For any non-terminal node σ in a given T, T_σ denotes the subtree of T rooted from σ.

For a node σ with n children, we denote $T_{\sigma * i}$ $(1 \leq i \leq n)$ the i-th subtree under the node of σ from left to right, and particularly the subtrees under the root λ is simplified as T_i instead of $T_{\lambda * i}$.

Notations

For any non-terminal node σ in a given T, T_σ denotes the subtree of T rooted from σ.
For a node σ with n children, we denote $T_{\sigma * i}$ $(1 \leq i \leq n)$ the i-th subtree under the node of σ from left to right, and particularly the subtrees under the root λ is simplified as T_i instead of $T_{\lambda * i}$.

By q_i, we denote the probability of the root of T_i being 0 with respect to a given distribution d.

An algorithm A is called optimal with respect to d if for any algorithm A', we have $C(A, d) \leq C(A', d)$.

DIR_d for uniform binary trees

Definition (1)

For any uniform binary tree T and $d \in \mathrm{ID}$ on T, the depth-first directional algorithm DIR_d is defined inductively as follows. If σ is a leaf then it is trivial. At the induction case, for any non-terminal node σ, let $\sigma * 1$ and $\sigma * 2$ be the children of σ, and assume $\mathrm{DIR}_{d_{\sigma*i}}$ $(i = 1, 2)$ for subtrees $T_{\sigma*i}$ are defined.

(1) In the case that σ is labeled $\wedge$, DIR_{d_σ} is a concatenation $\mathrm{DIR}_{d_{\sigma*1}}$ and $\mathrm{DIR}_{d_{\sigma*2}}$ i.e., $\mathrm{DIR}_{d_\sigma} := \mathrm{DIR}_{d_{\sigma*1}} \cdot \mathrm{DIR}_{d_{\sigma*2}}$ if $\frac{C_{\sigma*1}(\mathrm{DIR}_{d_{\sigma*1}},\ d_{\sigma*1})}{q_{\sigma*1}} \leq \frac{C_{\sigma*2}(\mathrm{DIR}_{d_{\sigma*2}},\ d_{\sigma*2})}{q_{\sigma*2}}$, otherwise $\mathrm{DIR}_{d_\sigma} := \mathrm{DIR}_{d_{\sigma*2}} \cdot \mathrm{DIR}_{d_{\sigma*1}}$.

DIR_d for uniform binary trees

Definition (1)

For any uniform binary tree T and $d \in \mathrm{ID}$ on T, the depth-first directional algorithm DIR_d is defined inductively as follows. If σ is a leaf then it is trivial. At the induction case, for any non-terminal node σ, let $\sigma * 1$ and $\sigma * 2$ be the children of σ, and assume $\mathrm{DIR}_{d_{\sigma * i}}$ $(i = 1, 2)$ for subtrees $T_{\sigma * i}$ are defined.

(1) In the case that σ is labeled $\wedge$, DIR_{d_σ} is a concatenation $\mathrm{DIR}_{d_{\sigma*1}}$ and $\mathrm{DIR}_{d_{\sigma*2}}$ i.e., $\mathrm{DIR}_{d_\sigma} := \mathrm{DIR}_{d_{\sigma*1}} \cdot \mathrm{DIR}_{d_{\sigma*2}}$ if $\frac{C_{\sigma*1}(\mathrm{DIR}_{d_{\sigma*1}},\ d_{\sigma*1})}{q_{\sigma*1}} \leq \frac{C_{\sigma*2}(\mathrm{DIR}_{d_{\sigma*2}},\ d_{\sigma*2})}{q_{\sigma*2}}$, otherwise $\mathrm{DIR}_{d_\sigma} := \mathrm{DIR}_{d_{\sigma*2}} \cdot \mathrm{DIR}_{d_{\sigma*1}}$.

(2) In the case σ is labeled $\vee$, $\mathrm{DIR}_{d_\sigma} := \mathrm{DIR}_{d_{\sigma*1}} \cdot \mathrm{DIR}_{d_{\sigma*2}}$ if $\frac{C_{\sigma*1}(\mathrm{DIR}_{d_{\sigma*1}},\ d_{\sigma*1})}{1-q_{\sigma*1}} \leq \frac{C_{\sigma*2}(\mathrm{DIR}_{d_{\sigma*2}},\ d_{\sigma*2})}{1-q_{\sigma*2}}$, otherwise $\mathrm{DIR}_{d_\sigma} := \mathrm{DIR}_{d_{\sigma*2}} \cdot \mathrm{DIR}_{d_{\sigma*1}}$.

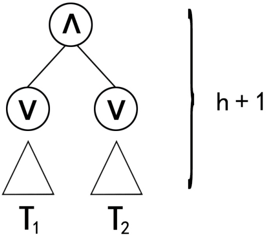
Idea of construction

If

$$C_1(A_1,d_1)+(1-q_1)\cdot C_2(A_2,d_2) \leq C_2(A_2,d_2)+(1-q_2)\cdot C_1(A_1,d_1),$$

$$i.e. \frac{C_1(A_1,d_1)}{q_1} \leq \frac{C_2(A_2,d_2)}{q_2},$$

where q_i is the probability of the root of i-th tree being 0, then, $\text{DIR}_d = A_1 \cdot A_2$

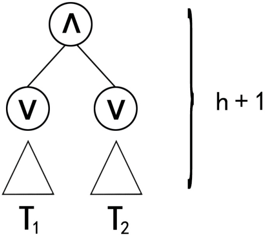


Optimal depth-first algorithms

Theorem (2017)

For any uniform binary tree T and $d \in ID$, if A is any depth-first algorithm, then $C(A, d) \geq C(DIR_d, d)$, i.e., DIR_d is optimal among all the depth-first algorithms.

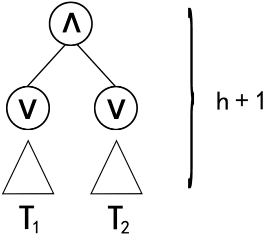
Proof. We prove this by induction on height h. The base case is trivial. At the induction step, let T be a uniform binary tree with height $h + 1$.



Suppose that DIR_{d_i} is an optimal algorithm for each subtree T_i with height h. For any depth-first algorithm A, if A evaluates T_1 first, then

$$C(A,d) = C_1(A_1,d_1) + \Big(\sum_{\omega_1 \in \Omega_h^1} d_1(\omega_1)\Big) \cdot C_2(A_{\omega_1}, d_2)$$

where A_1 is the algorithm of A in T_1 and A_{ω_1} is the algorithm of A in T_2 depending on the assignment $\Omega_h^i := \{\omega_1 \in \Omega_h \mid \omega_1 \textit{ assigns } i \textit{ to } T_1\}$.



By induction hypothesis, we take algorithms DIR_{d_1} and DIR_{d_2}, which satisfy that
$C(A,d) \geq C_{\lambda*1}(\mathrm{DIR}_{d_1}, d_1) + (1-q_1)C_{\lambda*2}(\mathrm{DIR}_{d_2}, d_2)$.
Let $A'_1 = \mathrm{DIR}_{d_1} \cdot \mathrm{DIR}_{d_2}$, then clearly $C(A,d) \geq C(A'_1, d)$.

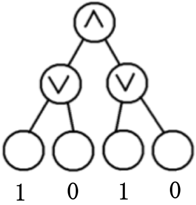
Using the similar arguments as above, if A evaluates T_2 first, we can get algorithm $A'_2 = \mathrm{DIR}_{d_2} \cdot \mathrm{DIR}_{d_1}$. Thus, $C(A,d) \geq C(A'_2, d)$.
If $\frac{C_{\lambda*1}(\mathrm{DIR}_{d_1}, d_1)}{q_1} \leq \frac{C_{\lambda*2}(\mathrm{DIR}_{d_2}, d_2)}{q_2}$, then
$C(A'_1, d) = C_{\lambda*1}(\mathrm{DIR}_{d_1}, d_1) + (1-q_1)C_{\lambda*2}(\mathrm{DIR}_{d_2}, d_2) =$
$C(A'_2, d) - (q_1 C_{\lambda*2}(\mathrm{DIR}_{d_2}, d_2) - q_2 C_{\lambda*1}(\mathrm{DIR}_{d_1}, d_1)) \leq C(A'_2, d)$.

By Definition 1, DIR_d is A'_1 if $\frac{C_{\lambda*1}(A_{d_1}, d_1)}{q_1} \leq \frac{C_{\lambda*2}(A_{d_2}, d_2)}{q_2}$, otherwise A'_2. Clearly, DIR_d is optimal among all the depth-first algorithms. □

non-directional algorithms

Directional algorithms: there are many E^1 -distributions w.r.t. the set of all directional algorithms.
An algorithm is called non-directional if its order of searching leaves depends on the query history.

Let's consider a directional algorithm A=**1234**. We define a non-directional algorithm $A' = \bar{1}2\widehat{34}$ as follows: If the first leaf is labeled by 1, exchange the order of 3 and 4. Otherwise, it is the same as A.



The above results can generalized to weighted trees:

Theorem (Peng *et al*)

For any multi-branching weighted tree, a directional algorithm w.r.t. ID *is optimal.*

Proof.

We show this theorem by induction on height h. □

Theorem

For any uniform binary tree T *and* $d \in ID$, DIR_d *is optimal among all the non-depth-first algorithms.*

Future research

Algorithm	T_2^h	ID	IID
Depth-First	h ≥ 2	**Directional Algorithm**[4]	Directional Algorithm is optimal[2]
Non-Depth-First	h = 2	Directional Algorithm[5] (Depth-First)	
Non-Depth-First	h ≥ 3	**Directional Algorithm (Depth-First)**	
Non-Depth-First	Generalized Trees (Weighted Trees)	Directional Algorithm	

Thank you for your attention!

A.C.-C. Yao:
Probabilistic computations: toward a unified measure of complexity,
in: Proc. 18th Annual IEEE Symposium on Foundations of Computer Science, pp. 222-227, 1977.

M. Saks and A. Wigderson,
"Probabilistic Boolean decision trees and the complexity of evaluating game trees",
Proc. 27th Annual IEEE Symposium on Foundations of Computer Science, pp. 29-38, 1986.

C. G. Liu and K. Tanaka,
"Eigen-distribution on random assignments for game trees",
Inform. Process. Lett., 104 (2007) 73-77.

W. Peng, N. Peng, K. Ng, K. Tanaka, Y. Yang, Optimal depth-first algorithms and equilibria of independent distributions on multi-branching trees, *Inform. Process. Lett.*, 125 (2017) 41-45.

T. Suzuki,
Non-depth-first search against independent distributions on an AND-OR tree, *Inform. Process. Lett.*, 139 (2018) 13-17.