The Uniqueness of Eigen-distribution for Multi-branching Trees

Shohei Okisaka
Joint work with
Weiguang Peng, Wenjuan Li and Kazuyuki Tanaka

Mathematical Institute, Tohoku University

January 16, 2016

Outline

1 Introduction
- Abstract
- Terminology
- History

2 Main Theorem
- i-set and E^1-distribution
- The equivalence of two distributions
- The uniqueness of eigen-distribution

Abstract

This study is an extension of Suzuki and Nakamura's results[4] which states for any uniform binary trees, the E^1-distribution with respect to all deterministic algorithm is equal to the unique eigen-distribution to any balanced multi-branching trees.

Game tree

In this talk, AND-OR tree (OR-AND tree) is a tree whose root is labeled AND (OR) and the internal nodes are level-by-level labeled OR or AND alternatively except for leaves.

A tree is *balanced* if all nodes at the same level have the same number of children.
Especially, if all nonterminal nodes (not leaves) have the same number of children, the tree is called *uniform* (or *n-branching*).
$\mathcal{T}_n^h$ denote n-branching tree with height h.

We mainly consider n-branching trees, and we will often identify $\mathcal{T}_n^h$ with $\{0, \cdots, n-1\}^{\leq h}$

An assignment ω for $\mathcal{T}_n^h$ is a function from the set of all leaves to Boolean value (i.e. $\omega : \{0, \cdots, n-1\}^h \to \{0, 1\}$).

Given AND-OR tree $\mathcal{T}_n^h$, and assignment ω, we can compute the value of the root according to the label of each internal nodes.

Our interest is that how many leaves should be probed (opened) to determine the value of the root.

An algorithm A tells how to proceed to evaluate a tree.
In this talk, ”algorithm” means *deterministic, depth-first, alpha-beta pruning* algorithm.

- deterministic : the choice of leaves in each step is unique.
- depth-first : if algorithm evaluates the value of some subtree, it never evaluate another subtree until it finishes to evaluates the current one.
- alpha-beta pruning : if we know the value of some node, it starts to evaluate another subtree.

An algorithm is *directional* if there is some linear ordering on the leaves such that the computation follow this ordering.

Given an assignment ω, algorithm $\mathbb{A}$ for $\mathcal{T}_n^h$, $C(\mathbb{A}, \omega)$ denote the number of leaves probed by $\mathbb{A}$ under the assignment ω.

Let d be a (probability) distribution on the set of assignments, the cost of $\mathbb{A}$ under the distribution d is defined by

$$C(\mathbb{A}, d) := \sum_{\omega} d(\omega) C(\mathbb{A}, \omega)$$

Then the value $\max_d \min_{\mathbb{A}} C(\mathbb{A}, d)$ is called *the distributional complexity of* $\mathcal{T}_n^h$ where d runs over all distributions and $\mathbb{A}$ runs over all deterministic algorithms.

A distribution d_0 is the *eigen-distribution* if $\min_{\mathbb{A}} C(\mathbb{A}, d_0) = \max_d \min_{\mathbb{A}} C(\mathbb{A}, d)$

History

- Yao (1977, [1]) showed that the randomized complexity equals to the distributional complexity.

$$\min_{\mathbb{A}_R} \max_{\omega} C(\mathbb{A}_R, \omega) = \max_{d} \min_{\mathbb{A}_D} C(\mathbb{A}_D, d)$$

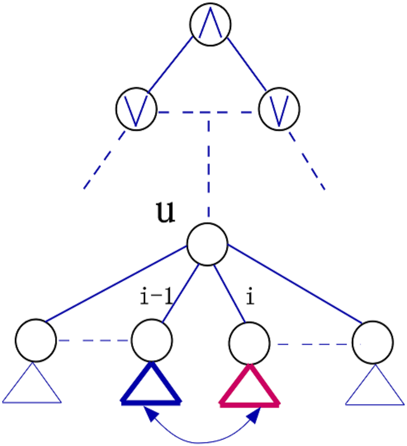
- Saks and Wigderson (1986, [2]) showed that the randomized complexity of $\mathcal{T}_n^h$ is $\Theta\left(\left(\frac{n-1+\sqrt{n^2+14n+1}}{4}\right)^h\right)$

- Liu and Tanaka (2009, [3]) defined the concepts of *i-set* and *E^i-distribution* ($i = 0, 1$), then they showed E^i-distribution is the unique eigen-distribution for uniform binary tree.
- Suzuki and Nakamura (2012, [4]) extended this results and showed that uniqueness fails if we consider only directional algorithm.

- Liu and Tanaka (2009, [3]) defined the concepts of *i-set* and *E^i-distribution* ($i = 0, 1$), then they showed E^i-distribution is the unique eigen-distribution for uniform binary tree.
- Suzuki and Nakamura (2012, [4]) extended this results and showed that uniqueness fails if we consider only directional algorithm.

We showed that the same statements hold for balanced multi-branching tree.

Transposition



i-th u-transposition

of node/ of assignment/ of algorithm

transposition

i-th *u*-transposition induces a *transposition* of assignments and algorithms.

Let ω be an assignment, *i*-th *u*-transposition of ω (denoted by $tr_i^u(\omega)$) is defined by $tr_i^u(\omega)(v) = \omega(tr_i^u(v))$ for any leaf v.

transposition of algorithm $tr_i^u(\mathbb{A})$ is defined by the same way. Note that $C(tr_i^u(\mathbb{A}),\ \omega) = C(\mathbb{A},\ tr_i^u(\omega))$

closed and connected

The set of assignments (algorithms) is *closed* if it is closed under transposition.
The set of assignments Ω is *connected* if for any $\omega, \omega' \in \Omega$, there exists a path of transpositions in Ω from ω to ω'.

Definition (i-set)

For i=0,1, *i-set* consists of assignments such that

- the root has value i.
- if AND-node has value 0, just one of its children has value 0, and other children have value 1, and similar condition hold for OR-node.

Note that i-set is closed connected.

Definition (i'-set)

A closed set of assignments Ω is called *i'-set* if it is not i-set but for any assignment $\omega \in \Omega$, the root has value i under ω.

Let $\mathcal{A}$ be a nonempty set of algorithms.

A distribution d on i-set is called an *E^i-distribution* with respect to $\mathcal{A}$ if there exists $c \in \mathbb{R}$ such that for any $\mathbb{A} \in \mathcal{A}$, $C(\mathbb{A}, d) = c$

Theorem

Let $\mathcal{T}_n^h$ be an AND-OR tree, d distribution, $\mathcal{A}$ closed set of algorithms. Then the following conditions are equivalent.

1. *d is an eigen-distribution w.r.t. $\mathcal{A}$*
2. *d is an E^1-distribution w.r.t. $\mathcal{A}$*

Note that (1) mean that $\min_{\mathbb{A} \in \mathcal{A}} C(\mathbb{A}, d) = \max_d \min_{\mathbb{A} \in \mathcal{A}} C(\mathbb{A}, d)$

$d_{uni}(\Omega)$ denote the uniform distribution on Ω.

Let $\Omega_1, \cdots, \Omega_m$ be disjoint nonempty subsets of assignments, $p_1, \cdots, p_m$ positive real numbers such that their sum is 1. The distribution $p_1 d_{uni}(\Omega_1) + \cdots + p_m d_{uni}(\Omega_m)$ is denoted by $d_{uni}(p_1\Omega_1 + \cdots + p_m\Omega_m)$.

We say d is a distribution on $p_1\Omega_1 + \cdots + p_m\Omega_m$ if there are distributions d_j on Ω_j such that $d = p_1 d_1 + \cdots p_m d_m$

Lemma (1)

If each Ω_j is closed, then there exists $c \in \mathbb{R}$ such that for any algorithm $\mathbb{A}$, $C(\mathbb{A}, d_{uni}(p_1\Omega_1 + \cdots + p_m\Omega_m)) = c$

By lemma (1), it is enough to consider one special algorithm $\mathbb{A}_0$ which evaluates leaves from left to right.

For simplicity, $C_i^{\wedge,h} := C(\mathbb{A}_0, d_{uni}(i\text{-}set))$ for AND-OR tree $\mathcal{T}_n^h$.
Similarly, for any i'-set Ω,
$C_\Omega^{\wedge,h} := C(\mathbb{A}_0, d_{uni}(\Omega))$ for AND-OR tree. $\mathcal{T}_n^h$.

$C_i^{\vee,h}, C_\Omega^{\vee,h}$ is defined by the same way.

Lemma (2)

1. $\forall h \in \mathbb{N},\ C_1^{\wedge,h} > C_0^{\wedge,h}$
2. *For any i′-set* Ω, *and* $h \in \mathbb{N}$, $C_i^{\wedge,h} > C_\Omega^{\wedge,h}$

Lemma (3)

If each Ω_j *is closed and connected,* $\mathcal{A}$ *is a closed, and d is a distribution on* $p_1\Omega_1 + \cdots +_m \Omega_m$*, then the following are equivalent:*

1. $\min_{\mathbb{A}\in\mathcal{A}} C(\mathbb{A}, d) = \max_{d'} \min_{\mathbb{A}\in\mathcal{A}} C(\mathbb{A}, d')$
   where d' *runs over all distribution on* $p_1\Omega_1 + \cdots +_m \Omega_m$
2. *There exists* $c \in \mathbb{R}$ *such that for any* $\mathbb{A} \in \mathcal{A}$, $C(\mathbb{A}, d) = c$
3. $\min_{\mathbb{A}\in\mathcal{A}} C(\mathbb{A}, d) = \sum_{1\le j\le m} p_j C(\mathbb{A}, d_{uni}(\Omega_j))$

(proof of theorem) "eigen $\Rightarrow E^1$"
Let d be an eigen-distribution.

The set of all assignments can be divided into disjoint closed connected subset $\Omega_1, \cdots, \Omega_m$. We can assume Ω_1 is 1-set.

Moreover, we can choose $p_1, \cdots, p_m$ such that d is a distribution on $p_1\Omega_1 + \cdots + p_m\Omega_m$.

Then, lemma(2), (3) implies $p_1 = 1$ and $p_j = 0$ for $j > 1$ which means d is a distribution on 1-set.

By lemma (3) again, d is an E^1-distribution.

"E^1 $\Rightarrow$ eigen"
By lemma (3), we can show that for any d on $p_1\Omega_1 + \cdots + p_m\Omega_m$,

$$\min_{\mathbb{A}} C(\mathbb{A}, d_{uni}(p_1\Omega_1 + \cdots + p_m\Omega_m)) \geq \min_{\mathbb{A}} C(\mathbb{A}, d)$$

If d_0 is E^1-distribution,

$$\min_{\mathcal{A}} C(\mathbb{A}, d_0) = C(\mathbb{A}, d_{uni}(1\text{-}set)) \geq C(\mathbb{A}, d_{uni}(p_1\Omega_1 + \cdots + p_m\Omega_m))$$

These two inequalities imply

$$\min_{\mathbb{A}} C(\mathbb{A}, d_0) \geq \min_{\mathbb{A}} C(\mathbb{A}, d)$$

for any d on $p_1\Omega_1 + \cdots + p_m\Omega_m$.

We can show the eigen-distribution is unique.

Theorem

For any AND-OR tree $\mathcal{T}_n^h$, E^1-distribution is uniform on 1-set.

We can show the eigen-distribution is unique.

Theorem

For any AND-OR tree $\mathcal{T}_n^h$, E^1-distribution is uniform on 1-set.

(Sketch of proof) For simplicity, we consider $\mathcal{T}_3^2$. Let d be an E^1-distribution for $\mathcal{T}_3^2$.
Suppose $\omega_1 = 001001001$, $\omega_2 = 001001010$ with probability $p_1 = d(\omega_1)$ and $p_2 = d(\omega_2)$.
Let $\mathbb{A}$ be an algorithm from left to right, and $\mathbb{A}'$ is defined as follows:
$\mathbb{A}'$ evaluates from left to right for first two subtrees and the leftmost child of the rightmost subtree.
If the result is 0010010, then $\mathbb{A}'$ probe the rightmost child of the rightmost subtree.
otherwise, it is same as $\mathbb{A}$.

Then

$$C(\mathbb{A}, d) = C(\mathbb{A}, \omega_1)p_1 + C(\mathbb{A}, \omega_2)p_2 + \cdots = 9p_1 + 8p_2 + \underbrace{\cdots}_{r_1}$$

$$C(\mathbb{A}', d) = C(\mathbb{A}', \omega_1)p_1 + C(\mathbb{A}', \omega_2)p_2 + \cdots = 8p_1 + 9p_2 + \underbrace{\cdots}_{r_2}$$

By definition of $\mathbb{A}'$, $r_1 = r_2$ and we can conclude $p_1 = p_2$.
Using similar argument, we can show all probabilities are same.

References

1. Yao, A,C.-C. “Probabilistic computations: towards a unified measure of complexity,” Proc. 18th annual IEEE symposium on foundations of computer science, pp .222-227, 1977
2. Saks, M. and Wigderson, A. “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.
3. C. G. Liu and K. Tanaka, “Eigen-distribution on random assignments for game trees,” *Information Processing Letters*, vol. 104, no. 2, pp. 73-77, 2007.
4. T. Suzuki and G. R. Nakamura, “The eigen distribution of an AND-OR tree under directional algorithms,” *IAENG International Journal of Applied Mathematics*, vol. 42, no. 2, pp. 122-128, 2012.