数理解析研究所講究録
第 494 巻 1983 年 160-171

The Relation between Time and Accepting Probability on Probabilistic Simple Decision Trees

(extended abstract)

Osamu WATANABE
Dept. of Information Sciences
Tokyo Inst. of Technology

1. Introduction

There are several notions of "a probabilistic algorithm T accepts a set L". Among them is the following: if an input x is in L then T accepts x with a nonzero probability, and otherwise T never accepts x. This is a natural extension of the acceptance by nondeterministic algorithms.

The famous prime test algorithms by Rabin [Ra] and Solovay-Strassen [SS] are probabilistic algorithms of this sense if they are regarded as acceptors of the set COMP of all composite numbers.

Their algorithms are fast and moreover accept x with a high probability if x is in COMP. However we expect intuitively that in many problems every probabilistic algorithm whose accepting probability is high requires more time than one that has lower accepting probability. We have two examples of this phenomenon for some computation models ([MT], [W1]).

In these examples we see the following type of relation between time and accepting probability: In recognition of some set by some computation model (1) we can construct a machine which accepts it within $O(f(n))$ time if we do not mind the accepting probability, but (2) every machine which accepts it

with the high accepting probability needs $\Omega(g(n))$ time (where $f(n) << g(n)$). So these examples show rather rough relation between time and accepting probability. Although we expect more close relation in some problem ([Ad], [Mo]), we have obtained no such example before. In this paper we will give one example for a close relation.

We use probabilistic simple decision trees as the computation model ([MT]). For such a tree, we use the height of the tree as the measure of computation time. It corresponds to the worst case run time.

On this computation model we will obtain the following results: Let p be any number such that $0 < p \leq 1$, then to test element non-uniqueness of n elements with accepting probability p, $O(\max(\log n, pn\log pn, \sqrt{pn^2 \log pn^2}))$ time is sufficient and $\Omega(\max(\log n, pn\log pn, \sqrt{pn^2 \log pn^2}))$ time is necessary.

2. Preliminaries

In this section we define probabilistic simple decision trees and the element non-uniqueness problem. After that we describe the main theorem formally.

Definition 1

A probabilistic simple decision tree is a finite binary tree whose nodes are either query nodes, coin-tossing nodes or leaf nodes. A query node is labeled by "i:j" and has two emanating edges labeled by "<" and ">" respectively. A coin-tossing node is unlabeled and has two unlabeled emanating edges A leaf node has no emanating edges and is either an "accepting node" or a "rejectiong node".

This paper deals only with probabilistic simple decision trees. So we sometimes omit "probabilistic simple" in the following.

Since every decision tree is finite and acyclic, it is natural to assume that each tree works only for particular size of inputs. Let T be any decision tree whose input size is n. Then every input to T is an n-tuple of real numbers. We use $\bar{x}$ to denote n-tuple $(x_1, x_2, \ldots, x_n)$. Suppose that an input $\bar{x}$ is given to a decision tree T. The execution of T on input $\bar{x}$ is defined as follows.

Definition 2

The execution of T on input $\bar{x}$ starts from the root node and continues to proceed to the next node until it reaches a leaf node. On each node, in order to determine the next node, one emanating edge is chosen from two possible ones according to the following rules:

(1) on a query node labeled by "i:j", $\leq$-labeled (or $>$-labeled) edge is chosen if $x_i \leq x_j$ (or $x_i > x_j$), and

(2) on a coin-tossing node, one of two emanating edges is chosen with the same probability.

We say that T accepts $\bar{x}$ if an accepting node is reached after an execution. Since an execution is not deterministic but probabilistic, the event "T accepts $\bar{x}$" occurs with a certain probability. By $\Pr\{$ T accepts $\bar{x}$ $\}$ or $p_T(\bar{x})$, we mean this probability.

Because input size of T is n and every query in T is a simple query (a straight comparison between a pair of input elements), we can assume, without loss of generality, that each element of an input is an integer between 1 and n. So we

define D_n by $\{(x_1, \ldots, x_n) \mid 1 \leq x_i \leq n$, for all i$\}$ and regard D_n as an input domain of T. Then the language recognized by T and accepting probability of T are defined as follows.

Definition 3

The language recognized by T is

$$L(T) = \{\bar{x} \mid \bar{x} \in D_n \text{ and } p_T(\bar{x}) > 0\}.$$

The accepting probability of T is

$$p_T = \min_{\bar{x} \in L(T)} p_T(\bar{x}).$$

Let h_T denote the height of T, that is, the maximum length of paths from root to leaves in T. So h_T is the maximum number of comparisons and coin-tossings, and it corresponds to the worst case run time.

The element non-uniqueness problem is the recognition of the set

$$L_n = \{(x_1, \ldots, x_n) \in D_n \mid x_i = x_j \text{ for some i and j}\},$$

where $n \geq 2$ so that L_n may not be empty.

Now that we have defined the computation model and the problem, we describe the main theorem formally. Our main theorem shows the upper bound and the lower bound of the height of simple decision trees which recognizes L_n with $p_T \geq p$. So it is convenient to define the optimal height H(n, p) of such trees.

$$H(n, p) = \min\{ h_T \mid T \text{ is a decision tree such that } L(T) = L_n \text{ and } p_T \geq p \}.$$

Also define $g(n, p) = \max(\log n, pn\log pn, \sqrt{pn^2 \log pn^2})$ (in this paper the base of logarithms is 2).

The Main Theorem

There exist positive integers c, c' and $n_0 \geq 2$ such that, for all $n \geq n_0$, and all p, $0 < p \leq 1$,

$$c \cdot g(n, p) < H(n, p) < c' \cdot g(n, p).$$

2. The Upper Bound

In this section we will prove that $H(n, p) = O(g(n, p))$.

We state the main theorem with the function g. But here we introduce a function f and use it instead.

Definition 4

The function t(y) is defined on $y \geq 0$, and

$t(y) = x$ such that $x^2 \log x = y$.

The function f(n, p) is defined on n, $p > 0$, and

$f(n, p) = \max(pn\log pn, t(pn^2)\log t(pn^2))$.

We show some important properties of f and t, which are deduced from more general ones ([W2]).

Proposition 1

Let n and p be any number such that $n \geq 2$ and $0 < p \leq 1$ respectively.

(a) The values of $t(pn^2)$ and f(n, p) are well defined.

(b) $pn \geq t(pn) \Rightarrow f(n, p) = pn\log pn \geq n$,

$pn \leq t(pn) \Rightarrow f(n, p) = t(pn^2)\log t(pn^2) \leq n$.

(c) There exist n'_0, c'_1 and $c'_2 > 0$ such that if $n \geq n'_0$ then

$$c'_1 \cdot g(n, p) < \max(\log n, f(n, p)) < c'_2 \cdot g(n, p).$$

Prop. 1 (c) ensures us using max(log n, f(n, p)) in place of g. So we need to prove the following theorem.

Theorem 2

There exist $c_1 > 0$ and $n_1 \geq 2$ such that for all $n \geq n_1$ and all p, $0 < p \leq 1$, there exists a probabilistic simple decision tree T which recognizes L_n with $p_T \geq p$ and $h_T < c_1 \max(\log n, f(n, p))$.

Proof. First we note that we can construct a deterministic decision tree for L_n with height $O(n\log n)$ (such a tree first sorts all input elememnts and then checks equality between every pair of elements neighboring in this ordering).

If $p \geq 1/16$, we use this deterministic decision tree. So $p_T = 1 \geq p$, and there exists d_1 such that for a sufficiently large n,

$$h_T \leq c \cdot n\log n \leq d_1 pn\log pn \leq d_1 \max(\log n, f(n, p)).$$

So assume that $p < 1/16$.

Let any n and p be fixed. Then the description of T which recognizes L_n with $p_T \geq p$ and $h_T = O(\max(\log n, f(n, p)))$ is as follows:

Let

$p' = p \times 16 < 1$,

$m = \max(2, p'n, t(p'n))$,

$n' = \lceil m\log m \rceil$, and

$k = \lceil n / n' \rceil$, then

begin

choose k_1 randomly from $\{0, \ldots, k-1\}$;

choose k_2 randomly from $\{0, \ldots, k-1\}$;

$S_1 \leftarrow \{x_{n'k_1+1}, \cdots, x_{n'(k_1+1)}\}$
$\{x_{n'k_2+1}, \cdots, x_{n'(k_2+1)}\}$;

(if $k_i = k-1$ then $\{x_{n'k+1}, x_{n'k+2}, \ldots, x_n\}$ is used)

$n_0 \leftarrow |S_1|$;

($|S_1|$ means the number of elements in S_1)

choose i_0 randomly from $\{1, \ldots, n_0\}$;

$i_1 \leftarrow \max(1, i_0 - (m - 1))$;

$i_2 \leftarrow \min(n_0, i_0 + (m - 1))$;

$y_1 \leftarrow i_1$ th best in S_1;

(by the fast selection algorithm [B1])

$y_2 \leftarrow i_2$ th best in S_1;

$S_2 \leftarrow \{x \in S_1 \mid y_1 \leq x \leq y_2\}$;

$m_0 \leftarrow |S_2|$;

if $m_0 > i_2 - i_1 + 1$ then accept (and halt);

test element non-uniqueness of S_2

by the deterministic decision tree;

end.

It is not so difficult to prove that this T satisfies the theorem (see [W2] for the detail).

3. The Lower Bound

In this section we will prove that $H(n, p) = \Omega(g(n, p))$.

From Prop. 1 (c), we need to prove the following theorem.

Theorem 3

There exist $c_2 > 0$ and $n_2 \geq 2$ such that for all $n \geq n_2$ and all p, $0 < p \leq 1$, if a probabilistic simple decision tree T recognizes L_n with $p_T \geq p$, then $h_T \geq c_2 \max(\log n, f(n, p))$.

It is easy to show that every nondeterministic simple decision tree which recognizes L_n requires $\Omega(\log n)$ height ([MT]). It is also true in probabilistic model since we can regard a nondeterministic decision tree as a probabilistic one.

Lemma 4

For any $n \geq 2$, if a probabilistic simple decision tree T recognizes L_n then $h_T \geq \log n$.

We will show that $h_T = \Omega(f(n, p))$ if T recognizes L_n with $p_T \geq p$. We first introduce some notations which are useful in the following.

Define X_n by $D_n - L_n$. So X_n is the set of all permutations of (1, 2, ..., n). For any $\bar{x} \in X_n$, $\pi_{\bar{x}}$ denotes a function which maps x_i's value to i (i.e. $x_{\pi_{\bar{x}}(x')} = x'$ for $1 \leq x' \leq n$). We sometimes use π if it does not make any confusion. Let $\bar{x} \in X_n$. Consider the input $\bar{y} = (y_1, \ldots, y_n)$ such that $y_{\pi(i_0+1)} = x_{\pi(i_0)}$ and $y_{\pi(i)} = x_{\pi(i)}$ for all i, $i \neq i_0+1$. Then $\bar{y} \in L_n$ only because $y_{\pi(i_0+1)} = y_{\pi(i_0)}$. We use $Y_{\bar{x}}$ to denote the set of all such inputs. And define Y_n by $\bigcup_{\bar{x} \in X_n} Y_{\bar{x}}$.

Mandor and Tompa proved that $h = (n \log n)$ if T recognizes L_n with $p_T > 1/2$ (Th. 8 in [MT]). We extend it to the following lemma which also plays basic role to get our lower bound.

Lemma 5

There exists h_1, $c_4 > 0$ such that for any $k \geq 2$ and q, $0 < q \leq 1$, if a probabilistic simple decision tree T satisfies

(a) $h_T \geq h_1$ (i.e., h_T is sufficiently large) and $h_T \geq k$,

(b) $L(T) = L_k$, and

(c) $\exp_{\bar{y} \in Y_k} p_T(\bar{y}) \geq q$,

then $h > c_4 qk \log qk$.

Proof. The proof is rather long and we omit it here (please see [W2]).

From this lemma it is easy to show that $h_T = \Omega(pn\log pn)$ for any T which recognizes L_n with $p_T \geq p$ and $h_T \geq n$. We can not apply it, however, to a tree T such that $h_T < n$. To use this lemma in that case, we need the following lemma.

Lemma 6

For any n, $k \geq 2$, if there is a decision tree T such that T recognizes L_n with $p_T \geq p$ and for every path the number of different input elements refered on it is at most k, then we can construct a decision tree T' such that

(a) $h_{T'} = h_T$,

(b) $L(T') = L_k$,

(c) $\exp_{\bar{y} \in Y_k} p_{T'}(y) \geq \frac{pn^2}{2k^2}$.

Proof. The proof is rather long and we omit it here (please see [W2]).

From Lemma 5 and Lemma 6 we get another lower bound for h_T that is, $h_T = \Omega(\ t(pn^2)\log t(pn^2)\ )$.

Lemma 7

There exist h_2, $c_5 > 0$ such that for any p, $0 < p \leq 1$, if a probabilistic simple decision tree T satisfies

(a) $h_T > h_2$ (i.e., h_T is sufficiently large),

(b) $L(T) = L_n$, and

(c) $p_T \geq p$

then $h_T \geq c_5 t(pn^2)\log t(pn^2)$.

Proof. Let $h_2 > h_1$ for h_1 defined in Lemma 5. Let any p, $0 < p \leq 1$, and any decision tree T which recognizes L_n with $p_T \geq p$ and $h_T \geq h_2$ be fixed. And let k denote the maximum number of different input elements refered on each path for any path in T

Then from Lemma 6, we can construct T' such that (a) $h_T = h_{T'}$ ($\geq h_1$), (b) $L(T') = L_k$, and (c) $\exp_{\bar{y} \in Y_k} p_{T'}(\bar{y}) \geq pn^2 / 2k^2$. By Lemma 5 we have

$$h_T = h_{T'} \geq c_4 \frac{pn^2}{2k^2} k \cdot \log\left(\frac{pn^2 \cdot k}{2k^2}\right) = c_4 \frac{pn^2}{2k} \log \frac{pn^2}{2k}. \qquad (1)$$

Assume that $h < \sqrt{c_4/8}\, t(pn^2)\log t(pn^2)$. Note that $k \leq 2h_T$. So we have $k < \sqrt{c_4/2}\, t(pn^2)\log t(pn^2)$. Thus from (1) we have $h_T > \sqrt{c_4/8}\, t(pn^2)\log t(pn^2)$, which contradicts the above assumption. Therefore $h_T \geq c_5 t(pn^2)\log t(pn^2)$ for some $c_5 > 0$.

Now we summarize the previous lemmas and prove Th. 3.

Proof of Th. 3.

Lemma 4 says that for any $n \geq 2$, if a decision tree T recognizes L_n with $p_T > p$, then

$$h_T \geq \log n. \qquad (1)$$

This is one lower bound for h_T. And it ensures the existence of n_0 such that $h_T > h_2$ for any $n \geq n_0$ and any T. Let any $n \geq n_0$ and any p, $0 < p \leq 1$ be fixed in the following. Also let any tree T which recognizes L_n with $p_T \geq p$ be fixed.

First consider the case that $h_T \geq n$. From the definition of p_T, we have

$$p \leq p_T = \min_{\bar{y} \in L_n} p_T(\bar{y}) \leq \exp_{\bar{y} \in Y_n} p_T(\bar{y}).$$

So using Lemma 5 we have

$$h_T \geq c_4 pn\log pn.$$

By Lemma 7 we also have

$$h_T \geq c_5 t(pn^2)\log t(pn^2).$$

Thus, for some $d_1 > 0$,

$$h_T \geq d_1 \max(pn\log pn,\ t(pn^2)\log t(pn^2)) = d_1 f(n, p).$$

Next consider the case that $h_T < n$. Lemma 7 works here and we have

$h_T \geq c_5 t(pn^2)\log t(pn^2)$.

This implies $n > c_5 t(pn^2)\log t(pn^2)$, and it is not so difficult to show $t(pn^2)\log t(pn^2) > d_2 pn\log pn$ (Prop. 1 (c) in [W2]). Thus, for some $d_3 > 0$,

$h_T \geq d_3 \max(pn\log pn,\ t(pn^2)\log t(pn^2)) = d_3 f(n, p)$.

Hence, for $d_4 = \min(d_1, d_3)$, we have another lower bound for h_T,

$h_T \geq d_4 f(n, p)$. (2)

From (1) and (2) we can conclude that for some $c_2 > 0$

$h_T > c_2 \max(\log n, f(n, p))$.

Remark

In order to describe the lower bound simply, we put three functions, log n, pnlog pn and $t(pn^2)\log t(pn^2)$ together. But for the case $h_T \geq n$, only the lower bound $\Omega(pn\log pn)$ has the essential meaning and $\Omega(\max(\log n, t(pn^2)\log t(pn^2)))$ does for the case $h_T < n$.

4. Conclusion

In this paper we showed some relation between time and accepting probability in solving the element non-uniqueness problem by probabilistic simple decision trees.

Here we considered the probabilistic computation model where a tree does not make a mistake for any input to be rejected. It is also possibe, however, to have the same type of result for the Gill's type computation model where a tree T may accept an input to be rejected and the language recognized by T is

$L(T) = \{\bar{x} \quad Pr\{ T \text{ accepts } \bar{x} \} > \frac{1}{2}\}$.

In this model we have the tradeoff relation between time and error probability as follows: To recognize L by probabilistic

simple decision trees with the error probability less than $1/2 - \delta$, order of $\max(\log n, \delta n \log \delta n, \sqrt{\delta n^2 \log \delta n^2})$ time is necessary and sufficient.

Acknowledgment

I wish to thank to Prof. Kojiro Kobayashi for his valuable comments on the first draft of this paper.

References

[Ad] L. Adleman, Two theorems on random polynomial time, 19th FOCS (1978) 75-83.

[Bl] M. Blum, et al., Time bounds for selection, J.C.S.S. 7 (1973) 448-461.

[Mo] S. Moran, On the accepting dencity hierarchy in NP, SIAM J. Comput. 11 (2) (1982) 344-349.

[MT] U. Manber and M. Toampa, Probabilistic, nondeterministic, and alternating decision trees, 14th STOC (1982) 234-244.

[Ra] M. Rabin, Probabilistic algorithms, in "Algrithms and Complexity: New Disrections and Recent Results (J. Traub, Ed.)", Academic Press, New York (1979) 21-39.

[SS] R. Solovay and V. Strassen, A fast Monte-Carlo test for primarity, SIAM J. Comput. 6 (1977) 84-85.

[W1] O. Watanabe, The time-precision tradeoff problems on on-line probabilistic Turing machines, to appear in Theoret. Comput. Sci.

[W2] O. Watanabe, The relation between time and accepting probability on probabilistic simple decision trees, in preparation.