Advanced Algorithmics

Strategies for Tackling Hard Problems

Sebastian Wild

Markus Nebel

Lecture 11

2017-05-25

Application 1: Can we trust Quicksort's expectation?

Definition 4.11 (With high probability)

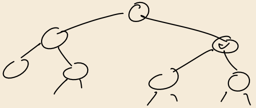
We say

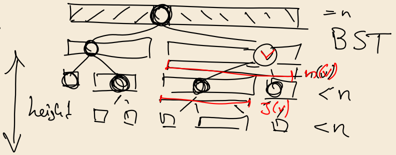
- an event $X = X(n)$ happens *with high probability (w.h.p.)* when $\forall c \ : \ \Pr[X(n)] \ = \ 1 \pm \mathcal{O}(n^{-c})$ as $n \to \infty$.
- a random variable $X = X(n)$ is *in $\mathcal{O}(f(n))$ with high probability (w.h.p.)* when $\forall c \exists d \ : \ \Pr[X \le d f(n)] \ = \ 1 \pm \mathcal{O}(n^{-c})$ as $n \to \infty$.
  (This means, the constant in $\mathcal{O}(f(n))$ may depend on c.)

Theorem 4.12 (Quicksort Concentration)

The height of the recursion tree of (randomized) Quicksort is in $\mathcal{O}(\log n)$ w.h.p.

Hence the number of comparisons are in $\mathcal{O}(n \log n)$ w.h.p.

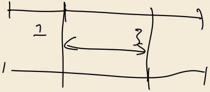




Proof: v : node in recursion tree

$n(v)$: #elems in the subtree of v

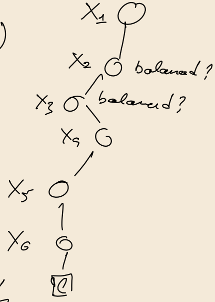
$J(v)$: size of the left child



v balanced $\iff n(v) \leq 1 \;\lor\; \frac{1}{4} \leq \frac{J(v)}{n(v)} \leq \frac{3}{4}$

↳ reduces subtree size of its child to $\leq \frac{3}{4} n(v)$

(*) Any recursion tree for n elements can contain at most $\log_{3/4}(1/n) = \log_{4/3}(n) \leq 3.5 \ln(n)$ balanced nodes. :

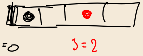


$$n \cdot \left(\frac{3}{4}\right)^{\log_{3/4}(1/n)} = 1 \quad \therefore (*)$$

Problem: to apply Chernoff to $X = X_1 + \dots + X_n$

we need that $X_1, \dots, X_n$ mutually independent ↯

$\Pr[X_i = 1]$ depends on $n(v_i)$



$n(v) = 4$ $\quad \frac{J(v)}{n(v)} \in \left[\frac{1}{4}, \frac{3}{4}\right] \iff J(v) \in [1..3]$ $\quad \Pr[X_i = 1] = \frac{3}{4}$

$n(v) = 5$ $\quad$ " $\quad \iff J(v) \in [2,3]$ $\quad \Pr[X_i = 1] = \frac{2}{5}$

$n(v) = 8$ $\quad J(v) \in [2,6]$ $\quad \frac{5}{8}$

$$\Pr[X_1 = 1 \wedge X_2 = 1] = \Pr[X_1 = 1] \cdot \Pr[X_2 = 1]$$

$P_b(n) = \Pr[v \text{ balanced} \mid n(v) = n]$ $\qquad \boxed{P_b(n) \geq \frac{2}{5} \qquad n \geq 1}$

independent and identically distr.

How to get iid indicators?

$$v \text{ good} \iff v \text{ balanced } \underline{\text{and}}\ B\Big(\underbrace{\frac{p}{P_b(n)}}_{\in (0,1]}\Big) = 1 \qquad B(\ldots) = C_i \qquad p = \frac{2}{5}$$

$G_i = [v_i \text{ good}]$

$$\Pr[v_i \text{ good}] = \Pr[C_i = 1 \mid X_i = 1] \cdot \Pr[X_i = 1] = p$$

$$\Pr[\text{tree height} \geq d\ln(n)] = \Pr[\exists \text{ path of length} \geq d\ln(n)]$$

$= \exists$ leaf of depth $\geq$ ..

$\leq n$ leaves in tree

$$\underset{\text{union bound}}{\leq} n \Pr[\underbrace{\text{leaf has depth}}_{d(\ell)} \geq d\ln(n)]$$

ℓ is a leaf $\qquad v_1, \dots, v_{d(\ell)} = \ell$ path from root to ℓ

$d(\ell)$ ← depth of ℓ

$G_1, \dots, G_{d(\ell)}$ are iid. $\mathcal{B}(p)$

we extend $G_1, \dots, G_{d(\ell)}$ to infinite iid sequence $G_1, \dots, G_{d(\ell)}, \underset{\overset{\mathcal{D}}{=}\ \mathcal{B}(p)}{G_{d(\ell)+1}}, \dots$

$\forall h: \quad G_1 + \dots + G_h \overset{\mathcal{D}}{=} Bin(h, p) \overset{\mathcal{D}}{=} X_h$

$$\Pr[d(\ell) \geq h] \leq \Pr[\underbrace{G_1 + \dots + G_h}_{X_h} \leq 3.5\ln(n)]$$

$$= \Pr\left[p - \frac{X_h}{h} \geq \overbrace{p - 3.5\frac{\ln(n)}{h}}^{=\delta}\right]$$

$$\leq \Pr\left[\left|\frac{X_h}{h} - p\right| \geq \delta\right]$$ assume $\delta \geq 0$

$$\leq 2\exp(-2\delta^2 h)$$

Chernoff

$h = d\ln(n)$ so $\delta = \frac{2}{5} - \frac{3.5}{d} > 0$ for $d \geq 8.25$

$$\Pr[\text{tree height} \geq d\ln(n)] \leq n\Pr[d(\ell) \geq d\ln(n)]$$

$$\leq 2n\exp(-2d\delta^2\ln(n))$$

$$= 2n^{1-2d\delta^2}$$

Given c, we can make $\Pr[\text{tree height} \geq d\ln(n)] = O(n^{-c})$

by picking d s.t. $1 - 2d\delta^2 \leq -c$ $\square$

e.g. height $\geq 42\ln(n)$ with prob $O(n^{-7.4})$

$\Rightarrow$ We can rely on randomized Quicksort.

(In practice: Introsort keep count of recursion depth)

Application 2: Majority Voting for Monte Carlo

Monte Carlo algorithms are allowed to err half the time.
That sound unusable in practice . . . can we improve upon that?

bounded error
success prob. $\geq \frac{1}{2} + \varepsilon$

unbounded error
$> \frac{1}{2}$

Idea: Use t *independent* repetitions of A on x.
If at least $\lceil t/2 \rceil$ runs (i.e., an absolute majority) yield result y, return y, otherwise return ?

Theorem 4.13 (Majority Voting)
Let A be a Monte Carlo algorithm for f with *bounded* error. Then, a *majority vote* of $t = \omega(\log n)$ repetitions of A is correct *with high probability*.

$\log^2 n = (\log(n))^2 \neq \log\log n = \log(\log(n))$

Proof: $X_1, \ldots, X_t$ $\quad X_i = [i\text{th run correct}] \overset{D}{=} B(p)$ $\quad p = \frac{1}{2} + \varepsilon$

$X = X_1 + \cdots + X_t \overset{D}{=} \mathrm{Bin}(t, p)$

majority vote fails $\iff X < \lceil \frac{t}{2} \rceil \iff X \leq \lceil \frac{t}{2} \rceil - 1$

$$\Pr\left[X < \left\lceil \frac{t}{2} \right\rceil\right] \leq \Pr\left[X \leq \frac{t}{2}\right] = \Pr\left[p - \frac{X}{t} \geq p - \frac{1}{2}\right]$$

$$\leq \Pr\left[\left|\frac{X}{t} - p\right| \geq \underbrace{p - \frac{1}{2}}_{\varepsilon > 0}\right]$$

$$\leq 2e^{-\varepsilon^2 t}$$

c arbitrary Chernoff

$t = \omega(\log n)$

$$n^c \cdot \Pr[\text{MV fails}] \leq 2\exp\Big(\underbrace{c\ln(n) - \varepsilon^2 t}_{\to -\infty}\Big)$$

$$\to 0 \qquad (n \to \infty)$$

$$\Rightarrow \Pr[\text{MV fails}] = o(n^{-c})$$

$\square$

Theorem 4.14 (Majority Voting with unbounded error)

There are Monte Carlo algorithms A with *unbounded* error that use only a linear number of random bits ($Random_A(n) = \Theta(n)$ as $n \to \infty$), so that a guarantee for successful *majority votes* with fixed probability $\delta \in (\frac{1}{2}, 1)$ requires the number of repetitions t to satisfy $t = \omega(n^c)$ for *every* constant c as $n \to \infty$.

That means, probability amplification for ***unbounded*** error Monte Carlo methods requires a *superpolynomial* number of repetitions and is thus not feasible.

Proof: success prob $> \frac{1}{2}$ $\quad$ $Random_A(n) \overset{= n}{<} \infty$

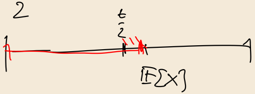
- A has up to $2^{Random_A(x)}$ different runs on x $\quad$ "2^n"
- each run has probability $2^{-Random_A(x)}$ $\quad$ "2^{-n}"

$p = \frac{1}{2} + \underbrace{2^{-n}}_{\varepsilon}$

MV fails $\iff X \leq \frac{t}{2}$ $\quad$ $X \overset{\mathcal{D}}{=} Bin(t, p)$

We show $\Pr\left[X \in \left(\frac{t}{2}, \mathbb{E}[X]\right]\right] \leq \varepsilon t \, 3^{\varepsilon t}$

Since Bin is symmetric : $\Pr[X \leq \mathbb{E}[X]] \geq \frac{1}{2}$



$$\Rightarrow \Pr[\text{MV fails}] = \Pr\left[X \leq \frac{t}{2}\right] \geq \Pr\left[X < \frac{t}{2}\right]$$

$$= \Pr[X \leq \mathbb{E}[X]] - \Pr\left[X \in \left(\frac{t}{2}, \mathbb{E}[X]\right]\right]$$

$$\geq \frac{1}{2} - \varepsilon t \, 3^{\varepsilon t} \longrightarrow \frac{1}{2}$$

$$\boxed{t = n^c \qquad \varepsilon = 2^{-n}}$$

$$\varepsilon t \to 0$$

$$\varepsilon t \, 3^{\varepsilon t} \to 0$$

$$\Rightarrow \Pr[\text{MV fail}] \to \frac{1}{2}$$

to show: $\Pr\left[\frac{t}{2} < X \leq t\left(\frac{1}{2} + \varepsilon\right)\right] \leq \varepsilon t \, 3^{\varepsilon t}$

$$\Pr[n] = \sum_{i=\frac{t}{2}+1}^{\frac{t}{2}+\varepsilon t} \underbrace{\binom{t}{i}}_{\leq 2^t} \left(\frac{1}{2}+\varepsilon\right)^{i} \left(\frac{1}{2}-\varepsilon\right)^{t-i}$$

$$\leq \sum_{i=\frac{t}{2}+1}^{\frac{t}{2}+\varepsilon t} \underbrace{2^t}_{= 4^{t/2}} \left(\frac{1}{2}+\varepsilon\right)^{t/2} \left(\frac{1}{2}-\varepsilon\right)^{t/2-\varepsilon t}$$

$$= \varepsilon t \cdot \Big(\underbrace{4\left(\tfrac{1}{2}+\varepsilon\right)\left(\tfrac{1}{2}-\varepsilon\right)}_{1-4\varepsilon^2 \leq 1}\Big)^{t/2} \Big(\underbrace{\tfrac{1}{2}-\overset{2^{-n}}{\overset{\|}{\varepsilon}}}_{> \frac{1}{3} \text{ for } n \geq 3}\Big)^{-\varepsilon t}$$

$$\leq \varepsilon t \cdot 3^{\varepsilon t} \qquad \square$$

$\Rightarrow$ Unbounded error is not usable in practice.

4.4 Randomized Complexity Classes

Does randomization extend the range of problems solvable by poly-time algorithms?
$\rightsquigarrow$ back to *decision* problems.

Some simplifications:

- Only 3 sensible output values: $0, 1, ?$.
- To allow full power of randomization, always allow $Random_A(c) = time_A(c)$, i.e., every step may use a random bit.

Definition 4.15 (ZPP)

$\mathcal{ZPP}$ (zero-error probabilistic poly-time) is the class of all languages L with a poly-time *Las Vegas* algorithm A, i.e., $\Pr\big[A(x) = [x \in L]\big] \geq \frac{1}{2}$ (and $A(x) \neq [x \in L]$ implies $A(x) = ?$), and $time_A(n) = \mathcal{O}(n^c)$ as $n \to \infty$ for some fixed c. ◀

Definition 4.16 (BPP and PP)

$\mathcal{BPP}$ (bounded-error probabilistic poly-time) and $\mathcal{PP}$ (probabilistic poly-time) is the class of languages with a poly-time *bounded-error resp. unbounded-error Monte Carlo* algorithm. ◀

Error Bounds Matter

Remark 4.17 (Success Probability)

From the point of view of complexities, the success probability bounds are flexible:

- $\mathcal{BPP}$ only requires success probability $\frac{1}{2} + \varepsilon$, but using *Majority Voting*, we can also obtain any fixed success probability $\delta \in (\frac{1}{2}, 1)$, so we could also define $\mathcal{BPP}$ to require, say, $\Pr\big[A(x) = [x \in L]\big] \geq \frac{2}{3}$.
- Similarly for $\mathcal{ZPP}$, we can use probability amplification on Las Vegas algorithms to obtain any success probability $\delta \in (\frac{1}{2}, 1)$.

But recall: this is *not* true for unbounded errors and class $\mathcal{PP}$.
In fact, we have the following result.

Theorem 4.18 (PP can simulate nondeterminism)

$\mathcal{NP} \cup \text{co-}\mathcal{NP} \ \subseteq \ \mathcal{PP}$. ◀

$\rightsquigarrow$ Useful algorithms must avoid unbounded errors.