Tutorials on Probability Bounds

Kanat Tangwongsan
(ktangwon@cs.cmu.edu)

Computer Science Department
Carnegie Mellon University

September 27, 2007
Main Focus: Chernoff bounds and their applications to proving high-probability bounds.

- Basic bounds: Markov’s inequality, Chebyshev’s inequality
- Chernoff bounds
- Examples
- Tree contraction, line breaking (revisited)
Markov’s inequality

For any random variable $X \geq 0$,

$$\Pr[X > \lambda] < \frac{\mathbb{E}[X]}{\lambda}$$

- If $f : X \rightarrow \mathbb{R}_+$ is a positive function, we have

  $$\Pr[f(X) > f(\lambda)] < \frac{\mathbb{E}[f(X)]}{f(\lambda)}.$$ 

- If $f$ is a non-decreasing function, we get that

  $$\Pr[X > \lambda] = \Pr[f(X) > f(\lambda)].$$

(Blackboard interlude)
Chebyshev’s inequality

For any random variable $X \in \mathbb{R}$,

$$\Pr[|X - \mathbb{E}[X]| \geq \lambda] \leq \frac{\text{Var}(X)}{\lambda^2}$$
Let \( X = \sum_i X_i \), where \( \mathbb{E}[X_i] = p_i \). The \( X_i \)'s are independent.

**Will show:** For \( 0 < \delta < 1 \),

\[
\Pr[X < (1 - \delta)\mu] \leq e^{-\mu\delta^2/2}.
\]

**Useful facts**

1. \( 1 + x \leq e^x \)
2. \( (1 - x) \ln(1 - x) > -\delta + \delta^2/2 \) for \( 0 < \delta < 1 \).

(Blackboard interlude)
The following bounds can be shown similarly:

1. For $\delta > 0$,
   $$\Pr[X < (1 - \delta)\mu] < \left( \frac{e^{-\delta}}{(1 - \delta)^{1-\delta}} \right)^\mu.$$

2. For $0 < \delta < 1$,
   $$\Pr[X < (1 - \delta)\mu] \leq e^{-\mu\delta^2/2}.$$

3. For any $\delta > 0$,
   $$\Pr[X \geq (1 + \delta)\mu] < \left( \frac{e^{\delta}}{(1 + \delta)^{1+\delta}} \right)^\mu.$$

4. For any $0 < \delta \leq 1$,
   $$\Pr[X \geq (1 + \delta)\mu] \leq e^{-\mu\delta^2/3}.$$

5. For $R \geq 6\mu$,
   $$\Pr[X \geq R] \leq 2^{-R}.$$
Rand-QSort(\(A\)) =

0. If \(|A| \leq 1\) return \(A\).
1. Pick \(e \in A\) uniformly at random.
2. Split \(A\) into \(A_<\) and \(A_>\).
3. Go to Step 1 if \(\max\{|A_<|, |A_>|\} > \frac{3}{4}|A|\).
4. return \(\text{Rand-QSort}(A_<) ++ \{e\} ++ \text{Rand-QSort}(A_>)\)
Parallel Tree Contraction (Miller and Reif, 1985)

Each contraction step performs **rake** and **compress** (in the order you wish):

- **Rake.** Contract all leaves into parent nodes.
- **Compress.** Shorten long chains of one-child nodes by employing a standard coin-flipping algorithm. All nodes flip coins; if a node comes up “heads”, and has a single child that comes up “tails”, it contracts that child into itself.

**In Your Homework:** 1) Each contraction removes at least a constant fraction of nodes *w.h.p.* 2) The algorithm terminates in $O(\log n)$ steps *w.h.p.*
Two major components:

- If I start a line, who should start the next line?
- You have a tree, what do you do with it?