Notes on the Chernoff Bound

1 The Bound

Suppose an unknown fraction $\alpha$ of some population satisfy property $P$. We would like to estimate $\alpha$ by sampling. How many samples do we need to get a good estimate? This is a fundamental question for all of science, and the Chernoff bound tells us an answer. We present a simplified bound\(^1\) that will be sufficient for our purposes.

**Theorem 1.** Let $X_1, \ldots, X_n$ be independent random variables taking values in $\{0, 1\}$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = \mathbb{E}[X]$. Then for any $A > 0$

$$\Pr[X - \mu \geq A] \leq e^{-\frac{A^2}{2n}}.$$  

In other words, the probability that $X$ lies away from its mean drops off very quickly, namely exponentially fast in the square of the distance from the mean.

Why does the above tell us an answer to our question? Well, suppose we want our estimate to be within $[\alpha - \epsilon, \alpha + \epsilon]$ with probability at least $1 - \delta$ over our samples. And suppose we sample $n$ times from the population independently at random. For $i = 1$ to $n$ take $X_i$ to be the random variable with value 1 if the $i$-th sample satisfies $P$ and 0 otherwise. Then applying Theorem 1 we can bound the difference between our estimated fraction and $\alpha = \mu/n$ as follows:

$$\Pr \left[ \frac{X}{n} - \frac{\mu}{n} \leq \epsilon \right] = \Pr \left[ X - \mu \leq n\epsilon \right] \leq e^{-\frac{(n\epsilon)^2}{2n}} = e^{-n\epsilon^2/2}$$

If we impose $e^{-n\epsilon^2/2} \leq \delta$, then solving for $n$ gives us $n \geq 2\log(1/\delta)/\epsilon^2$.

2 Application to Error Reduction for BPP

Let $L$ be a language for we have a probabilistic Turing Machine $M_L$ such that $M_L(x) = L(x)$ for all $x \in \{0, 1\}^*$ with some probability $\alpha$ over the coins of $M_L$.\(^2\) Our goal is to build a probabilistic Turing Machine $M'_L$ such that $M'_L = L(x)$ with probability $1 - \delta$ for $\delta$ exponentially small $\delta$ in $|x|$. In other words we want to reduce the error of $M_L$, or, equivalently, amplify its correctness.

Following the above sampling paradigm, given some $x \in \{0, 1\}^*$, we can regard all possible coin sequences for a run of $M_L(x)$ as the population in question. Our machine $M'_L$ on input $x$ will sample to estimate the fraction of the coin sequences that satisfy the property that $M_L(x)$ outputs 1 when run with this coin sequence and output if its estimate is closer to $\alpha$ than $1 - \alpha$. How many


\(^2\)We abuse notation and use $L$ to denote both the language and its characteristic function.
samples does it need? Suppose the true fraction is $\beta$. By the above, if we take $\delta = 2^{-|x|}$ and $\epsilon = |x|^{-c}$ for some constant $c$, it only needs a polynomial number of samples, namely

$$2\log(1/2^{-|x|})/(|x|^{-c+1})^2 = 2|x|^{2c+3}$$

for its estimate of the fraction of coin sequences on which $M_L(x)$ outputs 1 to be within $[\beta - |x|^{-(c+1)}, \beta + |x|^{-(c+1)}]$. If we are guaranteed that the gap between $\alpha$ and $1 - \alpha$ is bounded by $|x|^{-c}$ (i.e., $\alpha$ is not very close to 1/2), then $M'_L(x) = L(x)$ with probability at least $1 - 2^{-|x|}$ as desired.

Assuming $\alpha$ is a constant, this assumption is equivalent to assuming $\alpha \geq 1/2 + |x|^{-c}$ (and an equivalent way of viewing such $M'_L$ is as taking majority vote). This leads to the following theorem.

**Theorem 2.** Let $L$ be a language for we have a probabilistic Turing Machine $M_L$ such that $M_L(x) = L(x)$ for all $x \in \{0, 1\}^*$ with constant probability $\alpha \geq 1/2 + |x|^{-c}$. Then there is a probabilistic Turing Machine $M'_L$ such that $M'_L(x) = L(x)$ with probability at least $1 - 2^{-|x|}$. The running-time of $M'_L$ is $2|x|^{2c+3}$ times the running time of $M_L$ and it uses $2|x|^{2c+3}$ times more random coins than $M_L$.

In particular, if $M_L$ runs in polynomial-time so does $M'_L$, which shows that the definition of the complexity class BPP is equivalent for any of error at most $1/2 - |x|^{-c}$ for a constant $c$.

Can we design a different $M'_L$ use less random coins? This is an interesting direction discussed in the textbook. Some approaches for this are: (1) use weaker forms of the Chernoff bound that do not assume full independence among the $X_i$, e.g., where $X_i$ are only pairwise independent. In this case, one may be able to save on random coins since the coin sequences used for the runs of $M_L$ need not be fully independent. (2) Sample an initial vertex in an expander graph and obtain the coin sequences used for the runs of $M_L$ by representing in binary the vertices visited in a random walk from the initial vertex.

---

3If we work in the “non-uniform” model of computation where a Turing Machine takes an advice string depending on the input length (equivalently, using circuit families), we can even handle the case that $\alpha$ depends on $|x|$ since it can be given as advice. This should give you further appreciation for the distinction between the uniform and non-uniform models of computation.