Two Proofs and Interesting applications of Chernoff-Hoeffding Theorem

Xiao Yunxuan

March 28^{th}, 2018

1 Chernoff-Hoeffding Theorem

Suppose $X_1, ..., X_n$ are random variables, taking values in $\{0, 1\}$. Let $p = E[X_i]$ and $t > 0$. Then

$$P\left(\left|\frac{S_n}{n} - p\right| \geq t\right) \leq e^{-nh_+(t)} + e^{-nh_-(t)}$$

where:

$$h_+(t) = D_{KL}(p + t\|p)$$
$$h_-(t) = D_{KL}(p - t\|p)$$

D_{KL} is the Kullback–Leibler divergence between Bernoulli distributed random variables with parameters x and y respectively.

$$D_{KL}(x\|y) = x \ln \frac{x}{y} + (1 - x) \ln \left(\frac{1 - x}{1 - y}\right)$$

Proof1: Suppose S_n is a random variable that denotes the total number of 1s in Bernoulli distributed random variables$\{X_1, X_2, \cdots, X_n\}$.For any $\lambda > 0$ and through Chebyshev's Inequality, we have

$$\begin{aligned} P(S_n \geq (p + t)n) =& P(\lambda S_n \geq \lambda(p + t)n) = P(e^{\lambda S_n} \geq e^{\lambda(p+t)n}) \\ \leq& \frac{E[e^n]}{e^{\lambda(p+t)n}} = \frac{pe^\lambda + 1 - p}{e^{\lambda(p+t)}}^n = g(\lambda) \end{aligned}$$

The lower bound of the inequality can be determined by calculating the zero point λ' of the derivative of $g(\lambda)$

$$\frac{d(g(\lambda))}{d(\lambda)} = n\left[\frac{pe^\lambda + 1 - p}{e^{\lambda(p+t)}}\right]^{n-1} \frac{p(1 - p - t)e^\lambda - (p + t)(1 - p)}{e^{2\lambda(p+t)}} = 0$$

$$p(1 - p - t)e^{\lambda'} - (p + t)(1 - p) = 0$$
$$e^{\lambda'} = \left(\frac{1 - p}{p}\right)\left(\frac{p + t}{1 - p - t}\right)$$

Then we have the lower bound $g(\lambda^{'})$

$$\begin{aligned} g(\lambda^{'}) &= \left(\frac{pe^{\lambda^{'}} + 1 - p}{e^{\lambda^{'}(p+t)}} \right)^n \\ &= \left[\frac{p\left(\frac{1-p}{p}\right)\left(\frac{p+t}{1-p-t}\right) + 1 - p}{\left[\left(\frac{1-p}{p}\right)\left(\frac{p+t}{1-p-t}\right)\right]^{(p+t)}} \right]^n \\ &= \left[\left(\frac{p}{p+\varepsilon} \right)^{p+\varepsilon} \left(\frac{1-p}{1-p-\varepsilon} \right)^{1-p-\varepsilon} \right]^n \quad = e^{-nh_+(t)} \end{aligned}$$

hence we get

$$P\left(\frac{S_n}{n} - p \geq t \right) \leq e^{-nh_+(t)}$$

Similarly, by setting t to $-t$ and applying the same derivation process, it can also be proved that

$$P\left(\frac{S_n}{n} - p \leq -t \right) \leq e^{-nh_-(t)}$$

Therefore,

$$\begin{aligned} P\left(\left| \frac{S_n}{n} - p \right| \geq t \right) =& P\left(\frac{S_n}{n} - p \geq t \right) + P\left(\frac{S_n}{n} - p \leq -t \right) \\ \leq& e^{-nh_+(t)} + e^{-nh_-(t)} \end{aligned}$$

Proof 2(Encoding Arguments): Let D be a probability distribution on $\{0,1\}^n$ that assign to each element x a probabilityP_x, let ω be a non-negative weight function such that $\sum_{x\in\{0,1\}^n} \omega(x) \leq 1$ First, we prove a Lemma:

$$For\ any\ s \leq 1, P_{x\sim D}[\omega(x) \geq sP_x] \leq \frac{1}{s} \tag{$*$}$$

Let $Z_s = x|\omega(x) \geq sP_x$, in terms of Markov Inequality, we have

$$\begin{aligned} P_{x\sim D}[\omega(x) \geq sP_x] \leq & \frac{E[\omega(x)]}{sP_x} \\ \leq & \frac{\sum_{x\in Z_s} P_x\omega(x)}{sP_x} = \frac{1}{s}\sum_{x\in Z_s}\omega(x) \\ \leq & \frac{1}{s} \end{aligned}$$

Now consider $X = X_1X_2\cdots X_n$ is an encoded 0/1 string. Suppose that p denotes the probability of $X_i = 1$ ($p \geq 1/2$), k_x is the total number of 1s in string X. So

$$P_x = p^{k_x}(1-p)^{n-k_x}$$

Then we construct a weight function $\omega_{k_x}(x)$

$$\omega_{k_x}(x) = (p+t)^{k_x}(1-p-t)^{n-k_x}$$

Notice that $\omega(x)$ is a monotonically increasing function:

$$\frac{d\omega_{k_x}(x)}{dk_x} = \left(ln\frac{p+t}{1-p-t}\right)\omega(x) > 0$$

When $k_x = n(p+t)$:

$$\begin{aligned} P_x \cdot e^{nh_+(t)} &= p^{n(p+t)}(1-p)^{n(1-p-t)}\left[\left(\frac{p+t}{p}\right)^{(p+t)}\left(\frac{1-p-t}{1-p}\right)^{(1-p-t)}\right]^n \\ &= (p+t)^{(p+t)n}(1-p-t)^{(1-p-t)n} \\ &= \omega_{(p+t)n}(x) \end{aligned} \tag{$**$}$$

Thus we obtain the probability:

$$\begin{aligned} P(\frac{S_n}{n} - p \geq t) =& P(S_n \geq (p+t)n) = P_{x\sim D}[k_x \geq (p+t)n] \\ =& P_{x\sim D}[\omega_{k_x}(x) \geq \omega_{(p+t)n}(x)] \end{aligned}$$

And from lemma (*) and equation (**) we obtain:

$$P(\frac{S_n}{n} - p \geq t) = P_{x\sim D}[\omega(x) \geq P_x \cdot e^{nh_+(t)}] \leq e^{-nh_+(t)}$$

Similarly

$$P(\frac{S_n}{n} - p \leq -t) = P_{x\sim D}[\omega(x) \leq P_x \cdot e^{nh_-(t)}] \leq e^{-nh_-(t)}$$

We now have our desired result, that

$$P\left(\left|\frac{S_n}{n} - p\right| \geq t\right) \leq e^{-nh_+(t)} + e^{-nh_-(t)}$$

2 Interesting Application

Biased Coins A interesting application of Chernoff bound is tossing a biased coin. Consider a biased coin with probability $p = \frac{1}{3}$ of landing on one side(head) and probability $\frac{2}{3}$ of landing on the other(tail). Two sides of the coin seems to be identical and you want to determine which side is more likely to appear. Without precision instruments to measure the physical structure of the coin, there is still a simple solution ——we can flip it many times and choose the one that comes up the most.

But, there is a question left: how many times you need to flip so that you will be confident and make your final decision? With the help of Chernoff bound, we can determine a value n so that the probability that more than half of the coin flips come out heads is less that 0.001.(i.e. the probability that the head is more likely to appear than the tail).

Solution Let X_i be a random variable denoting the i^{th} flip, where $X_i = 1$ means heads, and $X_i = 0$ means tails. Random variable X is the sum of $\{X_i\}$. Let A represent the event that $\frac{1}{n}\sum_{n=1}^{n} X_i \geq \frac{1}{2}$, i.e. the event that more than half of the coin flips come out heads.

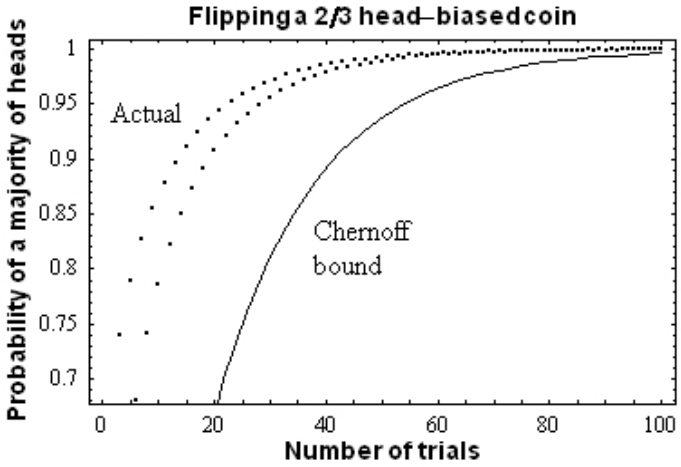
Notice that $E[X_i] = \frac{1}{3}$, and thus $E[S] = \frac{n}{3}$:

$$P(A) = P(\frac{1}{n}\sum_{n=1}^{n} X_i \geq \frac{1}{2}) = P(X \geq (1+\frac{1}{2})\frac{n}{3})$$

According to a looser version of Chernoff bound:

$$\Pr(X \geq (1+\delta)\mu) \leq e^{-\frac{\delta^2\mu}{3}}, \qquad 0 \leq \delta \leq 1$$

Thus $P(A) \leq e^{-n/36} \leq 0.001$, and we get $n \geq 32ln(1000) \approx 221$.



Bounded Probabilistic Polynomial Time Algorithms

In computational complexity theory, BPP, which stands for bounded-error probabilistic polynomial time is the class of decision problems solvable by a probabilistic Turing machine in polynomial time.

BPP is the set of decision problems that have the following kind of algorithm:

- If input $\in$ No $\Rightarrow$ with probability $\geq \frac{3}{4}$ say No.
- If input $\in$ Yes $\Rightarrow$ with probability $\geq \frac{3}{4}$ say Yes.

Denote p is the probability that input $\in$ Yes and hence $E[X_i] = p$. We can boost the confidence of BPP algorithms by replacing $\frac{3}{4}$ with $1-\delta$. The confidence of a BPP algorithm can be boosted by running it k times and calculating $\hat{p}$ = fraction of times the algorithm said Yes:

- If $\hat{p} \geq \frac{1}{2}$ say Yes.
- Otherwise say No.

Since the output would be wrong only when $|p-p^i| > \frac{1}{4}$, this can be used to calculate the number of trials needed to achieve the desired confidence:

$$Pr(|p - p^i| < \frac{1}{4}) < 2e^{-2n\frac{1}{16}} < \delta$$

Thus we get $n > 8ln(2/\delta)$.