0368-4159: First Course in Derandomization 29/10/2018 – Lecture 3

The Chernoff bound & Maximal Independent Set

Lecturer: Amnon Ta-Shma

Scribe: Roy Nadler

1 Chernoff Bound

Let $X_1, ..., X_n$ be boolean random variables with $\Pr(X_i = 1) = \mu_i$. Defining $X = \sum_i X_i$, we get $\mathbb{E}[X] = \sum_i \mu_i := \mu$. Our goal is to bound the following probabilities:

1. $Pr(|X - \mu| \geqslant A)$
2. $Pr(X \leqslant (1-\delta)\mu)$
3. $Pr(X \geqslant (1+\delta)\mu)$

In previous lecture we have seen:

- If X is positive, we can use Markov inequality: $\Pr(X \geqslant A) \leqslant \frac{\mathbb{E}[X]}{A}$. (If $A = (1+\delta)\,\mathbb{E}[X]$ this gives constant error).
- If $\{X_i\}$ are k-wise independent, we can use Chebyshev. (If $\mu_1 = \ldots = \mu_n$ is a constant, and the relative deviation δ is a constant, the error is $\Theta(n^{k/2})$.
- (This lecture) If $\{X_i\}_1^n$ are n-wise independent, we can use Chernoff.

Theorem 1. Suppose $X_1, \ldots, X_n$ are independent boolean random variables and $\Pr[X_i = 1] = \mu_i$, $\mu = \sum_i \mu_i$. Then $\Pr(|X - \mu| \geqslant \epsilon\mu) \leqslant 2^{-\Omega(\epsilon^2\mu)}$

Example 1.1. In the special case of $x_i \propto Bin(1/2)$, we get

$$Pr(|X - \frac{n}{2}| \geqslant \frac{n}{2} \cdot \frac{k}{\sqrt{n}}) = 2^{-\Omega(\frac{k^2}{n}\frac{n}{2})} = 2^{-\Omega(k)},$$

which restates the fact that asymptotically the error goes down exponentially with the number of standard deviations. This also means that section of width $\sqrt{n}$ around the mean gets constant probability. This, in turn, implies that the peak $X = \frac{n}{2}$ gets at least $\Omega(\frac{1}{\sqrt{n}})$ probability mass. You can also derive this directly (and get a much better bound) by noticing that $\Pr(X = \frac{n}{2}) = \binom{n}{n/2}$ and then using Stirling (do that!).

Proof. (Of Chernoff's bound) In Chebyshev we exploited pair-wise independence by utilizing the second moment $\mathbb{E}(X^2)$. For k-wise independence we utilized k moments, $\mathbb{E}(X^k)$. Now that we have full independence we should utilize all moments. One way to do this is by using exponentiation, as $e^x = \sum \frac{x^i}{i!}$ and $\mathbb{E}(e^X) = \sum_i \frac{\mathbb{E}(X^i)}{i!}$ takes advantage of all moments $\mathbb{E}(X^i)$. Indeed:

$$\Pr(X \geq (1+\delta)\mu) \quad = \quad \Pr(e^X \geq A) =_{\forall s>0} Pr(e^{sX} \geqslant e^{sA}) \leq_{(Markov)} \frac{\mathbb{E}[e^{sX}]}{e^{sA}}.$$

We can proceed from here. However, we can slightly simplify this: instead of writing e^{sy} (with $s > 0$) we can write t^y where $t = e^s > 1$. Redoing the previous line with this notation we get:

$$\begin{aligned}
\Pr(X \geq A) \quad &=_{\forall t>1} \quad \Pr(t^X \geq t^A) \leq_{(Markov)} \frac{\mathbb{E}[t^X]}{t^A} = \frac{\mathbb{E}[t^{\sum_i X_i}]}{t^A} = \frac{\mathbb{E}[\prod t^{X_i}]}{t^A} \\
&=_{(indpen)} \quad \frac{\prod \mathbb{E}[t^{X_i}]}{t^A} = \frac{\prod(\mu_i \cdot t + (1-\mu_i)\cdot 1)}{t^A} = \frac{\prod(\mu_i \cdot (t-1)+1)}{t^A} \\
&=_{[(1+x)\leqslant e^x]} \quad \frac{\prod e^{\mu_i \cdot (t-1)})}{t^A} = \frac{e^{\sum \mu_i \cdot (t-1)}}{t^A} = \frac{e^{(t-1)\cdot \mu}}{t^A}.
\end{aligned}$$

Finally, taking $t = \frac{A}{\mu}$ (which you'll find to be the best choice for t if you the calculation) and $\delta = \frac{A}{\mu} - 1$ (which is the relative error) we get:

$$\Pr(X \geq (1+\delta)\mu) \quad \leq \quad (\frac{e^\delta}{(1+\delta)^{(1+\delta)}})^\mu.$$

A similar calculation gives $\Pr(X \leq (1-\delta)\cdot\mu) \leq (\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}})^\mu$. The bound in the theorem follows by bounding the expressions we got for the error probabilities. □

2 Chernoff Hoeffding

The error estimates above $(\frac{e^\delta}{(1+\delta)^{(1+\delta)}})^\mu$ and $(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}})^\mu$ are not that meaningful functions. They can be simplified (and further degraded) to give useful error estimates, but they are not natural by themselves. Also, these bounds are not sharp, and, in particular don't appear in inverse-Chernoff bound (that state that with independent random variables deviations appear with some probability). We now give an alternative (arguably simpler) and tight bound that is also more meaningful. We do it for a special choice of parameters, but it hols in large generality (see [1]). We also mention that you can adapt the proof we gave in the previous section to get the better bound (see, e.g., [3]), but we think the simpler analysis in this section is nice and intuitive. For that we need a definition:

Definition 2. The KL-divergence between two numbers p and q is $KL : (0,1)^2 \to \mathbb{R}$ defined by

$$KL(p||q) = p \cdot ln(\frac{p}{q}) + (1-p)\cdot ln(\frac{1-p}{1-q}).$$

It can be proved that it always gives a non-negative real number. Furthermore, $KL(p||q) = 0$ iff $p = q$. Thus, it is a measure of *closeness* (or distance) between p and q. It can be naturally extended to distributions P, Q but we do need that and will not define it. We remark, however, that if we view the number $p \in (0,1)$ as a boolean distribution P that gets 1 with probability p and

0 otherwise, and we similarly view q as a boolean distribution Q with expectation q, then we can give $KL(p||q)$ the following interpretation. Let $Surpise_D(b)$ be the amount of surprise in a value $b \in \{0,1\}$ according to distribution D, namely, $Surprise_D(b) = \log \frac{1}{\Pr[D=b]}$ (and it better be that b is possible in the distribution D). Then:

$$\begin{aligned} KL(p||q) &= p \cdot ln(\frac{1}{q}) + (1-p) \cdot ln(\frac{1}{1-q}) - [p \cdot ln(\frac{1}{p}) + (1-p) \cdot ln(\frac{1}{1-p})] \\ &=_{trivially} \mathop{\mathbb{E}}_{b \in P}[ln(\frac{1}{\Pr(Q=b)})] - \mathop{\mathbb{E}}_{b \in P}[ln(\frac{1}{\Pr(P=b)})] \\ &= \mathop{\mathbb{E}}_{P}(Surprise_Q) - \mathop{\mathbb{E}}_{P}(Surprise_P). \end{aligned}$$

In words, we get samples from a distribution P. While P is the true distribution, we might not know it and think we actually get samples from a distribution Q. $KL(P||Q)$ measures the difference in expected surprise between doing the calculation with the right surprise function ($Surprise_P$) and our conjectured surprise function $Surprise_Q$).

It is now also easy to see that KL is not symmetric. For example if the support of P is strictly contained in the support of Q, when we sample according to P, all surprises are finite, but when we sample according to Q, if we happen to sample an element not in the support of P, the the surprise according to P is infinite.

Theorem 3. Suppose X_i are i.i.d., $\mathbb{E}(X_1) = \ldots = \mathbb{E}(X_n) = p$. Let $q \in (0,1)$ be arbitrary (larger or smaller than p). Then: if $q \geq p$, $\Pr(X \geqslant qn) \leqslant e^{-KL(q||p)n}$ and if $q \leq p$, $\Pr(X \leq qn) \leqslant e^{-KL(q||p)n}$.

Proof. We will prove the case $q > p$. The case $q < p$ is left as an exercise. Let $S = \{a = (a_1, \ldots, a_n) \in \{0,1\}^n | w(a) := sum_i(a_i) \geq qn\}$. Then, for every $a \in S$ with $w = w(a) \geq qn$ we have:

$$\begin{aligned} \frac{Pr(Q=a)}{Pr(P=a)} &= \frac{q^w \cdot (1-q)^{(n-w)}}{p^w \cdot (1-p)^{(n-w)}} = (\frac{q \cdot (1-p)}{p \cdot (1-q)})^w \cdot (\frac{(1-q)}{(1-p)})^n \\ &\geq_* (\frac{q \cdot (1-p)}{p \cdot (1-q)})^{qn} \cdot (\frac{(1-q)}{(1-p)})^n \\ &= (\frac{q^q (1-q)^{1-q}}{p^q (1-p)^{(1-q)}})^n = e^{KL(q||p)n}. \end{aligned}$$

Summing over $a \in S$:

$$\Pr(X \geq qn) = \sum_{a \in S} \Pr(P=a) \leqslant \sum_{a \in S} \Pr(Q=a) \cdot e^{-KL(q||p)n} = Q(S) \cdot e^{-KL(q||p)n} \leq e^{-KL(q||p)n}.$$

[*] - We have $w(a) \geqslant qn$, therefore this inequality is true if we show that $base \geqslant 1$:
$\frac{q(1-p)}{p(1-q)} \geqslant 1 \iff q(1-p) \geqslant p(1-q) \iff q \geqslant p$ which is true for this case.

□

3 Maximal IS and Luby's Algorithm [2]

Definition 4. Maximal IS

Input: $G = (V, E)$ an undirected graph.

Output: $A \subset V$

such that:

1. A is an independent set. I.e., for all $a, b \in A$, $(a, b) \notin E$.
2. Any extension of A is dependent: For all $s \in (V/A)$ there exists $a \in A$ such that $(s, a) \in E$.

There is a simple *sequential* algorithm for the problem (running in polynomial time):

Algorithm 1 Naive Polynomial solution

```
A = ∅
while S ≠ ∅ do
    take s ∈ S
    A ← s
    S.remove({s} ∪ Γ(s))
return A
```

where $\Gamma(S)$ is the set of neighbours of S in the graph. The process is very sequential. A natural question is whether it can be parallelized. Luby's algorithm does that (first with a randomized algorithm) in a very elegant way.

The basic idea is to have few (poly-logarithmically many) rounds, and each round to choose in parallel many independent vertices.

Algorithm 2 Parallel Probabilistic Algorithem (Michael Luby)

```
A = ∅
for O(log(|G|)) cycles do
    T = ∅
    In parallel ∀s ∈ S : T ← s with Prob = 1/(2·d(s))
    for (a, b) ∈ E do In parallel
        if a ∈ T ∧ b ∈ T then
            T.remove(argmin_{s∈{a,b}} d(s))
    In parallel ∀t ∈ T : S.remove({t} ∪ Γ(t))
if S = ∅ then
    return A
else
    return False
```

In words: at each round each vertex v chooses itself as a *candidate* with $\frac{1}{2d(v)}$ and isolated vertices choose themselves as candidates with probability 1. Thus, low degree vertices are more likely to belong to the candidate list (which makes sense, because, e.g., isolated vertices have to be chosen

into the list of candidates otherwise they will never be eliminated, and low-degree vertices that do not take care of themselves are likely to remain alive for very long).

Once the candidate list is chosen we check for collisions (two candidates that are connected by an edge), and, in parallel, for each collision remove the low-degree vertex from the candidate list. Thus, for the candidate list we give priority for low-degree vertices, but when resolving conflicts we give priority for high-degree vertices (that, if chosen, will eliminate more edges).

A word about conflicts between two vertices of the same degree: we resolve this arbitrarily. More precisely, we fix any total order on the vertices that respects the condition that $v < w$ when $d(v) < d(w)$ (e.g., first look at degree, then sort lexicographically on vertex name). We then determine conflicts by this total order.

Another comment is in place: in line 4 we use $d(s)$. That degree is the degree of s at the graph that remained after all previous deletions. E.g., if at the beginning G is connected, but eventually s gets isolated, then the degree of s at that stage is 0, and s will be chosen as a candidate and will survive conflict resolution. The same applies for the total order: it is computed from scratch every round based on the graph that survived so far.

Finally, notice that if at some stage we start with a graph G and choose a set T then T is independent. We then add T to the independent set, and delete T and $\Gamma(T)$ from the graph (because we are not allowed to use $\Gamma(T)$ any more) and this also deletes all *edges* that touch $T \cup \Gamma(T)$, so we expect (and hope to have) quite a massive destruction, hopefully shrinking the graph fast. We clearly have the following:

Theorem 5. 1. The set A is independent throughout the run.

2. When we terminate (i.e. $S = \emptyset$) A is a *maximal* independent set.

3. The running time of the algorithm is $O(\log(|G|)^2)$.

Statements Proofs:

1. At every stage we only add to A nodes that are neither neighbors of each other (step 5-7) nor neighbours of A (step 8). Therefor if A is independent, it remains so at the next iteration. At the 0'th iteration A is independent, therefore by induction it is always independent.

2. If A was returned, $s = \emptyset$. Therefore any node that could be appended to A had to be removed in the building process. Since we only remove from S nodes that are neighbouring A, any of these would cause dependence. Therefore A is a maximal independent set.

3. The decision to keep/remove any candidates in T requires $\wedge$ on n bits (or $\vee$) which would take depth log(n) (Check it!). Thus each of the log($|G|$)iteration takes up to log($|G|$) steps for a total of $\log^2(|G|)$ time.

What remains to show is the heart of the analysis:

Lemma 6. The algorithm returns a set with probability $O(\frac{1}{n})$.

For the proof we first define a special set. Let:

$$\begin{aligned} GOOD &= \left\{ v \in V \mid \left| \left\{ v' \in \Gamma(v) \mid d(v') < d(v) \right\} \right| \geqslant \frac{1}{3}|\Gamma(V)| \right\}, and, \\ BAD &= V \setminus Good. \end{aligned}$$

I.e., a vertex is *good*, if at least one third of its neighbours are smaller than it, and *bad* otherwise (most of its neighbours are larger).

Lemma 7. $Pr_{(v,w)\in E}(v \in Good \vee w \in Good) \geqslant 1/2$

Proof. We will define a one-to-one mapping

$$\Phi : BAD \mapsto \binom{E}{2},$$

i.e., each "bad" edge $e = (v, w)$, $(v \in Bad \wedge w \in Bad)$, is mapped to 2 unique edges in E. This is impossible if there are more than $\frac{|E|}{2}$ bad edges, there would be more than $|E|$ unique edges, thus the lemma follows.

For visualization, choose an orientation on the edge E by the total order, and imbue edges with direction from low to high order. Each edge gets a unique orientation. Now, for each bad edge $e = (v, w)$, both v and w are bad. W.l.o.g. assume $v < w$ in the total order, so the edge is oriented from v to w. As w is bad, up to one third of the neighbours of w have lower order, and atleast 2/3 of the neighbours of w have higher order. Therefore for each bad edge (v_i, w) entering w in the orientation, we have atleast twice "outgoing" edges (w, v_j) in our orientation (and these edges might be good or bad). We match the bad edges entering w to two edges leaving w (in our orientation) in an arbitrary way, as long as the pairs are disjoint.

Now, it is a moment thought to convince yourself that this gives the desired mapping. Indeed, if e_1 and e_2 are two bad edges. Assume e_1 in our orientation appears as (v_1, w_1) and e_2 as (v_2, w_2). Then, either $w_1 = w_2$, and then by the construction $\Phi(e_1) \cap \Phi(e_2) = \emptyset$, or $w_1 \neq w_2$, in which case $\Phi(e_1)$ has two edges leaving w_1 in our orientation, while $\Phi(e_1)$ has two edges leaving w_2 in our orientation, hence $\Phi(e_1) \cap \Phi(e_2) = \emptyset$(because if they share an edge, it will appear in both cases with the same orientation). □

In the next lecture we will prove that at every stage, every good node has atleast some constant probability to be removed. Namely,

Lemma 8. There exists a constant $\alpha > 0$ such that $\Pr_{v\in Good}(v \in T \cup \Gamma(T)) \geqslant \alpha$.

We now show that Lemma 8 together with Lemma 7 imply Lemma 6 (i.e., quick termination).

Proof. (Proof of Lemma 6)

The algorithm returns a set if $S = \emptyset$, so we want to show that with probability $O(\frac{1}{n})$ we remove all of the edges from E. We show this by proving that at each iteration there is a constant probability of removing a constant fraction of the remaining edges. Details follow.

First, we compute the *expectation* of the number of deleted edges:

$$\begin{aligned}
\mathbb{E}[\#removed\ edges] &= \sum_{(v,w)\in E} \Pr(v \vee w \text{ is in } T \cup \Gamma(T)) \\
&\geq \sum_{\substack{(v,w)\in E\ :\\ (v\in Good \vee w \in Good)}} \Pr(v \vee w \text{ is in } T \cup \Gamma(T)) \\
&\geq |\{e=(v,w)\in E \mid v \vee w \text{ is in } T \cup \Gamma(T)\}| \cdot \alpha \ \geq\ \frac{|E|}{2}\alpha.
\end{aligned}$$

Now, denote by Y_i the boolean random variable that is 1 iff atleast $\frac{\beta}{2}$ of the remaining edges were removed at step i, where $\beta = \frac{\alpha}{2}$. Let $p_i = \Pr[Y_i = 1]$, i.e., that at step i we removed a β fraction of the edges. We claim:

Claim 9. $p_i = \Pr[Y_i = 1] \geq \gamma = \frac{1-\beta}{1-\frac{\beta}{2}}$ and therefore is a constant $\gamma > 0$.

Proof. We know that $\mathbb{E}[\#removed\ edges] \geq \beta|E|$. On the other hand, it is at most $\frac{\beta}{2}$ with probability p_i, and $|E|$ with probability $1 - p_i$. This gives an equation that you can solve. (As remarked in the class, this argument is in fact a Markov bound on the positive random variables $|E| - \#removed - edges$). □

The random coins at each phase are independent. We would like to say that the random variables $\{Y_i\}$ are independent, but this is not exactly so, because Y_i also depends on the graph it operates on, which depends on the value of the previous Y_j. However, we can say that whatever the previous history is, Y_i is a boolean random variable with expectation at least γ. In this case we can apply the same bound as in the Chernoff bound.[1] Then, if we take the number of rounds to be R such that $\gamma R = 2\log_{(1-\frac{\beta}{2})}|E|)$, then $R = O(\log|E|)$ and:

$$\begin{aligned}
\Pr(s \neq \emptyset) &\leq \Pr(\sum_{i=1}^{R} Y_i \leq \log_{(1-\frac{\beta}{2})}|E|) \\
&\leq \Pr(\sum_{i=1}^{R} Y_i \leq \frac{\mathbb{E}[\sum Y_i]}{2}) \leq 2^{-\Omega(\gamma R)} = O(\frac{1}{n}).
\end{aligned}$$

□

References

[1] Russell Impagliazzo and Valentine Kabanets. Constructive proofs of concentration bounds. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 617–631. Springer, 2010.

[1] One way to see this in our case, is that given an input graph (that gurantees success probability γ or higher) we just change some of the successes to failures so that $\Pr(Y_i) = \gamma$. Now the modified random variables $\widetilde{Y}_1, \ldots, \widetilde{Y}_R$ are truly independent, and clearly the bound we get for the modified random variables $\left\{\widetilde{Y}_i\right\}$ also holds for the original random variables $\{Y_i\}$ because always $\sum \widetilde{Y}_i \leq \sum Y_i$.

[2] Michael Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM journal on computing*, 15(4):1036–1053, 1986.

[3] Rob Schapire. https://www.cs.princeton.edu/courses/archive/spring13/cos511/scribe_notes/0228.pdf.