Beyond Worst Case Analysis **Fall 2018**

Lecture 2: Largest Clique in a Random Graph

Scribe: William Yang, Jiazheng Zhao, James Hulett, Antares Chen 9/17/2018

2.1 Introduction

Worst-case analysis predicates on analyzing problems on classes of inputs that trigger "worst" case behavior. However, this analytical regime is sometimes too pessimistic and fails to capture what what usually happens when solving the problem on many inputs. Consider these examples:

- Quicksort and merge sort both require $O(n^2)$ time to sort a list of n elements in the worst-case, but in practice quicksort often out performs merge sort in arbitrary sorting tasks.
- Most clustering tasks that arise in topics such as unsupervised learning are formulated as NP-Hard optimization problems. However, there are a number of polynomial time algorithms that can still find a meaningful clustering if the pattern exists.
- The simplex algorithm for solving linear programs runs in worst-case exponential time, but in practice it rarely requires that long to solve an instance.

This reading group discusses analyzing algorithms beyond worst-case analysis, but to do so first requires us to define a model for inputs we wish to analyze the problem with. Over the past 30 years, there have been many models proposed and analyzed, varying widely in expressiveness and complexity. As a first step into this domain, we will look at the Erdős-Rényi model for generating random graphs. We will demonstrate that the size of the largest clique in G is with high probability $(2 \pm o(1)) \log(n)$ and then show a simple greedy algorithm that will recover a $\log(n)$ sized clique if it exists in the random graph.

2.2 The Problem with Max-Clique

Consider the max-clique problem. Given a graph G, we are interested in finding the largest subset of vertices in G such that they form a clique. This problem is well known to be NP-Hard, but it is also known to be extremely difficult to approximate. In fact, a theorem of Zuckerman [1] states

Theorem 2.1. *Let n denote the number of vertices in G, then for any constant $\epsilon > 0$, there does not exist an $O(n^{1-\epsilon})$-approximation algorithm for max-clique unless* $\mathrm{P} = \mathrm{NP}$.

To put this into context, a trivial way to get an n-approximation for max-clique is to output a single vertex. This theorem states that doing any better than this would imply $\mathrm{P} = \mathrm{NP}$! Things certainly look bleak in the land of worst-case analysis, but perhaps this is because worst-case analysis is too pessimistic. What happens when we sample random inputs for the max-clique problem? Enter the Erdős-Rényi graph.

2.3 Introducing the Erdős-Rényi Model

The Erdős-Rényi model, denoted as $\mathcal{G}_{n,p}$, defines a distribution on graphs constructed via the following procedure:

The Erdős-Rényi Model
Given n, p where $0 \leq p \leq 1$, a graph $G = (V, E)$ sampled from $\mathcal{G}_{n,p}$ is constructed via the following:

1. Fix n vertices $V = \{1, \ldots, n\}$
2. For any $i, j \in V$ where $i \neq j$, add the edge (i, j) to E with probability p independently at random.

One thing to note about this model is that it only generates simple graphs that do not contain self loops. Since the edges are sampled independently at random, the chance of sampling a specific $G \sim \mathcal{G}_{n,p}$ is given by

$$\Pr[G] = p^{|E(G)|}(1-p)^{\binom{n}{2}-|E(G)|}$$

where $E(G)$ denotes the edges of G. When $p = \frac{1}{2}$, the distribution $\mathcal{G}_{n,1/2}$ defines a uniform distribution over all simple graphs. There are many interesting properties that arise in graphs sampled from this model (see CS271 if interested), but our goal is to find the largest clique in G and so we will be interested in the size of G's largest clique. The calculations that we present in these notes can be generalized for any probability p, so for simplicity, we will restrict to using $\mathcal{G}_{n,1/2}$.

2.3.1 Largest Clique in $\mathcal{G}_{n,1/2}$

Let's first determine the size of the largest clique in $G \sim G_{n,1/2}$. The graph G is randomly constructed, so our statement of the largest clique size will have to be probabilistic in nature. One way to characterize this is as follows. Let X_k be a random variable denoting the number of k-cliques in G. If we can show that if

$$k_0(n) \leq k \leq k_1(n)$$

where $k_0(n)$ and $k_1(n)$ are two values dependent on n that are really close to each other (think an additive constant away), we have that

$$\Pr[X_{k_0(n)} > 0] \to 1 \text{ as } n \to \infty$$
$$\Pr[X_{k_1(n)} > 0] \to 0 \text{ as } n \to \infty$$

then we will know that G almost always has a clique of size about k as n becomes large. We will now show the following theorem

Theorem 2.2. *Given $G \sim \mathcal{G}_{n,1/2}$, then the largest clique of G has size $2(1 \pm o(1)) \log_2(n)$ with high probability. Specifically with $k_0(n) = 2(1 + o(1)) \log(n)$ and $k_1(n) = 2(1 - o(1)) \log(n)$, we have*

$$\Pr[X_{k_0(n)} > 0] \to 1 \text{ as } n \to \infty$$
$$\Pr[X_{k_1(n)} > 0] \to 0 \text{ as } n \to \infty$$

We will prove this result in three steps. First, we will calculate the expected number of k-cliques in $G \sim \mathcal{G}_{n,1/2}$. We will then prove the upper-bound statement for the largest clique of G having size at most $2(1+o(1))\log(n)$ w.h.p. using Markov's inequality. Finally, we demonstrate the lower-bound statement that the largest clique has size at least $2(1-o(1))\log n$ w.h.p. using what is sometimes called the second-moment method.

2.3.1.1 Number of k-Cliques in $\mathcal{G}_{n,1/2}$

Let's first quantify how many k-cliques G can have on expectation.

Claim 2.3. *Suppose $G \sim \mathcal{G}_{n,1/2}$, then $\mathbb{E}[X_k] = \binom{n}{k}2^{-\binom{k}{2}}$.*

Proof. For $S \subseteq V$ the vertices of G, let Y_S be the indicator random variable for if G contains a clique on all vertices in S. The number of k-cliques is then

$$X_k = \sum_{S \subseteq V : |S|=k} Y_S$$

which by linearity of expectations is the following.

$$\mathbb{E}[X_k] = \sum_{S \subseteq V : |S|=k} \mathbb{E}[Y_S] = \sum_{S \subseteq V : |S|=k} \Pr[Y_S = 1]$$

For $|S| = k$, the chance that every edge is added to G between every pair of distinct vertices in S is given by

$$\Pr[Y_S = 1] = \frac{1}{2^{\binom{k}{2}}}$$

Finally, note that there are $\binom{n}{k}$ ways to choose k vertices from n. This means that

$$\mathbb{E}[X_k] = \sum_{S \subseteq V : |S|=k} \Pr[Y_S = 1] = \sum_{S \subseteq V : |S|=k} 2^{-\binom{k}{2}} = \binom{n}{k}2^{-\binom{k}{2}} \qquad \square$$

2.3.1.2 The Upper-bound

Now that we know the expected number of k-cliques in G, we can calculate the probability that $\mathbb{E}[X_k] > 0$. First observe that

$$\mathbb{E}[X_k] = \binom{n}{k}2^{-\binom{k}{2}} \leq n^k 2^{-\frac{k(k-1)}{2}} = 2^{k\log(n) - \frac{k(k-1)}{2}} = 2^{\frac{k}{2}(2\log n - k + 1)}$$

This means that $\mathbb{E}[X_{k_0(n)}]$ for $k_0(n) = 2\log(n) + 2$ is given by the following.

$$\mathbb{E}[X_{k_0(n)}] \leq 2^{-(\log(n)+1)} = n^{-\Omega(1)}$$

Because $X_{k_0(n)}$ is an integer, we have by Markov's inequality

$$\Pr[X_{k_0(n)} > 0] = 1 - \Pr[X_{k_0(n)} \leq 1] \geq 1 - n^{-\Omega(1)}$$

Indeed as $n \to \infty$, we have that $\Pr[X_{k_0(n)} > 0] \to 1$ satisfying the first part of theorem 2.2.

2.3.1.3 The Lower-bound

Markov's inequality is usually sufficient to demonstrate an upper-bound, but to prove the lower-bound requires us to control the variance of the random variable we are interested in. For example, one might be inclined to demonstrate that $\mathbb{E}[X_{k_1(n)}] \to 0$ in order to prove $\Pr[X_{k_1(n)} > 0] \to 0$ as $n \to \infty$. While limit is accurate, it is not sufficient to show $\Pr[X_{k_1(n)} > 0] \to 0$ because the variance of $X_{k_1(n)}$ could so large such that for any n, there is some amount of probability mass greater than 0.

Critically, the second moment method uses Chebyshev's inequality to show the probability tends toward 0:

$$\Pr[|X - \mathbb{E}[X]| \geq \mathbb{E}[X]] \leq \frac{\mathbf{Var}[X]}{\mathbb{E}[X]^2}$$

The form that will be most helpful for us is the following.

$$\Pr[X \leq 0] \leq \Pr[X \leq 0 \text{ or } X \geq 2\mathbb{E}[X]] = \Pr[|X - \mathbb{E}[X]| \geq \mathbb{E}[X]] \leq \frac{\mathbf{Var}[X]}{\mathbb{E}[X]^2}$$

Thus computing $\Pr[X_{k_1(n)} > 0]$ will require us to calculate the right most ratio. Finally, before we proceed with the proof of the lower-bound, observe if $Y = Y_1 + \ldots + Y_n$ is a sum of non-negative random variables

$$\mathbf{Var}[Y] = \sum_{i=1}^{n} \mathbf{Var}[Y_i] + \sum_{i \neq j} \mathbf{Cov}[Y_i, Y_j] \leq \sum_{i=1}^{n} \mathbb{E}[Y_i^2] + \sum_{i \neq j} \mathbb{E}[Y_i Y_j]$$

Let's now show that $\Pr[X_{k_1(n)} > 0] \to 0$ by first calculating the variance of $X_{k_1(n)}$. Again let Y_S be the indicator random variable for the case $S \subseteq V$ forms a clique and fix $|S| = k_1(n)$. Now observe that if $S \cap T = \emptyset$ then Y_S and Y_T are independent meaning $\mathbf{Cov}[Y_S, Y_T] = 0$. Let $S \sim T$ denote S and T are not independent (i.e. when $S \neq T$ and $|S \cap T| \geq 2$ as this is the case where S and T share an edge). The variance of $X_{k_1(n)}$ is

$$\begin{aligned}
\mathbf{Var}[X_{k_1(n)}] &\leq \sum_{S \subseteq V}^{n} \mathbb{E}[Y_S^2] + \sum_{S \sim T} \mathbb{E}[Y_S Y_T] \\
&= \sum_{S \subseteq V}^{n} \mathbb{E}[Y_S] + \sum_{S \sim T} \mathbb{E}[Y_S Y_T] \\
&= \mathbb{E}[X_{k_1(n)}] + \sum_{S \sim T} \mathbb{E}[Y_S Y_T]
\end{aligned}$$

with the second equality following as Y_S is a 0-1 random variable. But notice that $\mathbb{E}[Y_S Y_T] = 1$ if and only if $Y_S = Y_T = 1$. We have

$$\begin{aligned}
\mathbf{Var}[X_{k_1(n)}] &\leq \mathbb{E}[X_{k_1(n)}] + \sum_{S \sim T} \mathbb{E}[Y_S Y_T] \\
&= \mathbb{E}[X_{k_1(n)}] + \sum_{S \sim T} \Pr[Y_S = 1 \text{ and } Y_T = 1] \\
&= \mathbb{E}[X_{k_1(n)}] + \sum_{S \sim T} \Pr[Y_S = 1] \cdot \Pr[Y_T = 1 \mid Y_S = 1] \\
&= \mathbb{E}[X_{k_1(n)}] + \sum_{S \in V} \Pr[Y_S = 1] \cdot \sum_{T : S \sim T} \Pr[Y_T = 1 \mid Y_S = 1]
\end{aligned}$$

Now look at the right-most sum. It really does not matter where we choose S to condition T on because the edges are all sampled independently at random! By symmetry

$$\Pr[Y_T = 1 \mid Y_S = 1] = \Pr[Y_T = 1 \mid Y_{S_0} = 1]$$

for a fixed S_0 where $|S_0| = k_1(n)$, thus the variance becomes

$$\begin{aligned}
\mathbf{Var}[X_{k_1(n)}] &\le \mathbb{E}[X_{k_1(n)}] + \sum_{S\in V} \Pr[Y_S = 1] \cdot \sum_{T:S\sim T} \Pr[Y_T = 1 \mid Y_S = 1] \\
&= \mathbb{E}[X_{k_1(n)}] + \left(\sum_{S\in V} \Pr[Y_S = 1]\right) \cdot \left(\sum_{T:S_0\sim T} \Pr[Y_T = 1 \mid Y_{S_0} = 1]\right) \\
&= \mathbb{E}[X_{k_1(n)}] + \mathbb{E}[X_{k_1(n)}] \sum_{T:S_0\sim T} \Pr[Y_T = 1 \mid Y_{S_0} = 1]
\end{aligned}$$

Now what is the conditional probability on the right? Given $i = |T \cap S_0|$, we have that

$$\Pr[Y_T = 1 \mid Y_{S_0} = 1] = \binom{k_1}{i}\binom{n-k_1}{k_1-i} 2^{-\left(\binom{k_1}{2}-\binom{i}{2}\right)}$$

Here $\binom{k_1}{i}$ is the number of ways T can choose vertices to share with S_0, factor $\binom{n-k_1}{k_1-i}$ counts all the other ways that T can choose its vertices, and $2^{-\left(\binom{k_1}{2}-\binom{i}{2}\right)}$ is the probability that edges in T but not in S_0 are added to G. This means that

$$\sum_{T:S_0\sim T} \Pr[Y_T = 1 \mid Y_{S_0} = 1] = \sum_{i=2}^{k_1(n)} \binom{k_1}{i}\binom{n-k_1}{k_1-i} 2^{-\left(\binom{k_1}{2}-\binom{i}{2}\right)}$$

Our variance is thus

$$\mathbf{Var}[X_{k_1(n)}] \le \mathbb{E}[X_{k_1(n)}] + \mathbb{E}[X_{k_1(n)}] \sum_{i=2}^{k_1(n)} \binom{k_1}{i}\binom{n-k_1}{k_1-i} 2^{-\left(\binom{k_1}{2}-\binom{i}{2}\right)}$$

Using the form of Chebyshev's inequality above

$$\Pr[X_{k_1(n)} \le 0] \le \frac{1}{\mathbb{E}[X_{k_1(n)}]} + \frac{\sum_{i=2}^{k_1(n)} \binom{k_1}{i}\binom{n-k_1}{k_1-i} 2^{-\left(\binom{k_1}{2}-\binom{i}{2}\right)}}{\mathbb{E}[X_{k_1(n)}]}$$

Because $\mathbb{E}[X_{k_1(n)}] = \binom{n}{k_1}2^{-\binom{k_1}{2}}$, the left summand tends toward 0 as $n \to \infty$. We need only show that the right summand does the same! Observe

$$\begin{aligned}
\frac{\sum_{i=2}^{k_1(n)} \binom{k_1}{i}\binom{n-k_1}{k_1-i} 2^{-\left(\binom{k_1}{2}-\binom{i}{2}\right)}}{\mathbb{E}[X_{k_1(n)}]} &= \frac{\sum_{i=2}^{k_1(n)} \binom{k_1}{i}\binom{n-k_1}{k_1-i} 2^{-\left(\binom{k_1}{2}-\binom{i}{2}\right)}}{\binom{n}{k_1}2^{-\binom{k_1}{2}}} \\
&= \frac{\sum_{i=2}^{k_1(n)} \binom{k_1}{i}\binom{n-k_1}{k_1-i} 2^{\binom{i}{2}}}{\binom{n}{k_1}} \\
&\le \frac{k_1 \cdot \max_{2\le i\le k_1} \binom{k_1}{i}\binom{n-k_1}{k_1-i} 2^{\binom{i}{2}}}{\binom{n}{k_1}}
\end{aligned}$$

The numerator of the above is maximized at $i = 2$ (the proof is delegated the appendix) thus

$$\begin{aligned}
\frac{k_1 \cdot \max_{2 \le i \le k_1} \binom{k_1}{i}\binom{n-k_1}{k_1-i} 2^{\binom{i}{2}}}{\binom{n}{k_1}} &\le \frac{k_1 \cdot \binom{k_1}{2}\binom{n-k_1}{k_1-2} 2^{\binom{2}{2}}}{\binom{n}{k_1}} \\
&\le k_1^3 (k_1 - 1)^2 \left(\frac{n-k_1}{n}\right) \cdots \left(\frac{n-2k_1+3}{n-k_1+3}\right)\left(\frac{1}{n-k_1+2}\right)\left(\frac{1}{n-k_1+1}\right) \\
&\le \frac{k_1^5}{n-k_1+1}
\end{aligned}$$

With $k_1 = 2(1 + o(1)) \log(n)$, we have that $\lim_{n\to\infty} \frac{\mathbf{Var}[X_{k_1(n)}]}{\mathbb{E}[X_{k_1(n)}]}$. This means that $\Pr[X_{k_1(n)} > 0] \to 0$ as required! This completes the proof that with high probability, the largest clique in $G \sim \mathcal{G}_{n,1/2}$ is $2(1 \pm o(1)) \log(n)$. We will now use this fact to construct an algorithm that recovers a clique close to this size.

2.4 Finding Cliques in an Erdős-Rényi Graph

When G is constructed randomly, can we find a clique in polynomial time larger than that guaranteed by theorem 2.1? The fact that the largest clique in $G \sim \mathcal{G}_{n,1/2}$ is close to $2\log(n)$ allows us to say yes, and in fact, a simple greedy algorithm is all we need!

Greedy Algorithm
Given $G \sim \mathcal{G}_{n,1/2}$ do the following:

1. Initialize $S = \{v\}$ where v is an arbitrary vertex of G.
2. While there is still a vertex $i \in V$ connected to every $v \in S$ by an edge, add i to S.
3. Return S.

For the purposes of our analysis, assume that we do not sample an edge until one of its endpoints is added to S (or equivalently that when we add a vertex to S, all the edges incident to it are "revealed" to us); this will make it easier to talk about the probability of an event occurring over the random choices of the graph.

All this algorithm is doing is maintaining a clique S at every iteration of the algorithm thus the iteration that this algorithm terminates determines the size of the clique returned. When does this algorithm terminate? At the start of the algorithm, there are a total of n vertices that could eventually appear in the clique. However, each time we add a vertex v to S, we would expect that about half of the remaining vertices will end up not having an edge to v, so the pool of vertices we can choose from should get cut approximately in half each time we add a vertex to S. Intuitively, we can only add about $\log(n)$ vertices to S before our original pool of n vertices gets cut down to a size of 1. Thus, once the algorithm adds that last vertex to S, we are done.

Formalizing this intuition, we will show that, with high probability, this greedy algorithm returns a clique of size $\log(n) + \log\log(n)$. Notice that this is significantly better than what theorem 2.1 admits. Whereas in the worst-case, we may only find a clique that is roughly a $\frac{1}{n}$ factor the size of the max-clique's actual size, choosing G to be sampled randomly from $\mathcal{G}_{n,1/2}$ allows us to recover a clique that is roughly $\frac{1}{2}$ the size of the expected max size!

2.4.1 Upper-bounding Number of Iterations

We begin by demonstrating the algorithm will halt before step $\log(n) + \log\log(n)$ with high probability. We define R_k to be the number of vertices remaining after k additions to S; that is R_k is the number of vertices that have not been added to S but have edges to every vertex in S and so have the possibility of being added in the next round. We have that

$$\mathbb{E}[R_k|R_{k-1}] = \max\left(\frac{R_{k-1}-1}{2}, 0\right)$$

The first term in the maximum comes into play if $R_{k-1} > 1$, meaning that we lose one vertex from the previous step to being added to S, while in expectation half of the remainder still have an edge to everything in S; the second term comes into play when $R_{k-1} = 0$. We can simplify this to say that

$$\mathbb{E}[R_k|R_{k-1}] \leq \frac{R_{k-1}}{2}$$

Applying that $R_0 := n$ and repeatedly iterating expectations, we get that

$$\mathbb{E}[R_k] \leq \frac{n}{2^k}$$

We can now plug in $k = \log(n) + \log\log(n)$ to get that

$$\mathbb{E}[R_{\log(n)+\log\log(n)}] \leq \frac{n}{2^{\log(n)+\log\log(n)}} = \frac{n}{n\log(n)} = \frac{1}{\log(n)}$$

Applying a simple Markov bound, this gives us

$$\Pr(R_{\log(n)+\log\log(n)} \geq 1) \leq \frac{1}{\log(n)}$$

But saying that at least one vertex survived through $\log(n) + \log\log(n)$ rounds is exactly the same as saying that the algorithm has not yet terminated after that many additions to S. Hence, we get that with probability at least $1 - \frac{1}{\log(n)}$, the algorithm will have terminated by round $\log(n) + \log\log(n)$.

2.4.2 Lower-bounding Number of Iterations

We know that the algorithm will likely terminate in at most $\log(n) + \log\log(n)$ and now want to say the algorithm will likely terminate in at least $\log(n) - \log\log(n)$ so that with high probability it will output a clique of size roughly $\log(n)$. Say that a vertex "fails" if it either gets added to S or if it does not have an edge to some other vertex that got added to S. Notice that the algorithm terminates once all vertices have failed. Let F_k be the event that all vertices have failed by round k. If F_k happens, at most k of the vertices failed because they were added to S. This means at least $n-k$ of them failed because they did not have an edge to some vertex in S. The probability of a vertex failing in this latter way is just $(1 - 2^{-k})$, so

$$\Pr(F_k) \leq (1 - 2^{-k})^{n-k}$$

Since we will use $k = \log(n) - \log\log(n) \ll n$, we can safely replace the $n - k$ with a $\frac{n}{2}$ while still maintaining the bound. We now apply the "computer scientist's favorite inequality" $(1 - x \leq e^{-x})$ to rewrite this as

$$\Pr(F_k) \leq e^{-2^{-k}(n/2)}$$

Plugging in $k = \log(n) - \log\log(n)$ to the exponent, we get

$$\begin{aligned} -2^{-k}\binom{n}{2} &= -2^{-\log(n)+\log(\log(n)}\binom{n}{2} \\ &= -\left(\frac{\log(n)}{n}\right)\binom{n}{2} \\ &= -\frac{\log(n)}{2} \end{aligned}$$

Hence,

$$\Pr(F_{\log(n)-\log\log(n)}) \leq e^{-\log(n)/2} = n^{-\Theta(1)}$$

The probability that our algorithm terminates in less than $\log(n) - \log\log(n)$ steps goes to zero as $n \to \infty$, so we can say that our algorithm finds a clique of size at least $\log(n) - \log\log(n)$ with high probability.

2.4.3 Some Refinements

Previously, we treated the algorithm as if it kept a list of all the vertices that could still be added to S, then chose an arbitrary one to add to S at each step. However, we can make the algorithm much simpler by simply fixing an arbitrary ordering of the vertices, then iterating through them and checking if each one can safely be added to S. In order to find the expected runtime of the algorithm given this refinement, we have the following claim.

Claim 2.4. *When considering whether or not we can add a vertex v to S, the expected number of edges we have to check for is at most 2.*

Proof. If v is the first vertex we consider, we know we can always add it to S (as there are no vertices there for it to have a problem with), so we don't have to look at any edges. Otherwise, suppose that there are already $k > 0$ vertices in S when we consider v. We always have to check to see if v is connected to the first vertex in S. However, if it is not, we don't have to check if v is connected to the second vertex; thus, we only need to check this second connection with probability $\frac{1}{2}$. Similarly, we only have to check the third connection with probability $\frac{1}{4}$, and so forth. Thus, the expected number of connections we have to check is

$$\sum_{i=1}^{k} \frac{1}{2^{i-1}} \leq \sum_{i=0}^{\infty} \frac{1}{2^i} = 2$$

□

This tells us that in expectation, we have to check at most $2n$ connections between vertices during the run of our algorithm. Hence, the expected runtime is just $O(n)$.

References

[1] Zuckerman, D. (2006, May). Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing* (pp. 681-690). ACM.

A Additional Proofs

A.1 Bounding the Binomial Coefficient

An inequality that we have used throughout these set of notes to bound the binomial coefficient is given by

Claim 2.5. $\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq n^k$

Proof. First the lower-bound. Observe

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)\dots(n-k+1)}{k!} = \frac{n}{k} \cdot \frac{n-1}{k-1} \cdot \dots \cdot \frac{n-k+1}{1} \geq \left(\frac{n}{k}\right)^k$$

For the upper-bound observe

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \leq n(n-1)\dots(n-k+1) \leq n^k$$

as required. □

A.2 Maximizing the Variance Term

We use the fact that $\max_{2 \leq i \leq k_1} \binom{k_1}{i}\binom{n-k_1}{k_1-i}2^{\binom{i}{2}}$ is maximized when $i = 2$. The following demonstrates this fact.

Claim 2.6. $\operatorname{argmax}_{2 \leq i \leq k_1} \binom{k_1}{i}\binom{n-k_1}{k_1-i}2^{\binom{i}{2}} = 2$

Proof. Observe that $\binom{k}{i}\binom{n-k}{k-i}2^{\binom{i}{2}}$ is a decreasing sequence for $2 \leq i \leq t$. This can be seen as follows:

$$\frac{a_i}{a_{i-1}} = \frac{\binom{k}{i}\binom{n-k}{k-i}2^{\binom{i}{2}}}{\binom{k}{i-1}\binom{n-k}{k-(i-1)}2^{\binom{i-1}{2}}} = \frac{(k-i+1)^2}{i(n-2k+i)}2^{i-1} < 1$$

□