Today’s exercises

- 5.12: Lower Bound for Binomial Coefficient
- 5.13: Volume versus Boundary
- Inclass: PPZ on the formula $F^*$
- 6.2: Many $j$-isolated Satisfying Assignments
- 6.4: Make it Hard for PPZ
- 7.1: Covering Radius Example
- 7.3: Exact Radius
5.12: Lower Bound for Binomial Coefficient

First observe that for $i < k \leq n$, we have

$$\frac{n-i}{k-i} = \frac{(n-i)k}{(k-i)n} \cdot \frac{n}{k} = \frac{kn-ik}{kn-in} \cdot \frac{n}{k} \geq \frac{n}{k}.$$

The desired inequality follows directly, as

$$\binom{n}{k} = \frac{n(n-1) \cdots (n-k+1)}{k(k-1) \cdots (k-k+1)} \geq \left(\frac{n}{k}\right)^k.$$
5.13: Volume versus Boundary

Trivial: \( \binom{n}{k} \leq \sum_{l=0}^{k} \binom{n}{l} \). For the other direction, observe that for \( i \geq 1 \)

\[
\frac{n(n-1)\cdots(n-k+i+1)}{(k-i)!} \cdot \frac{n(n-1)\cdots(n-k+1)}{k!}
=
\frac{k(k-1)\cdots(k-i+1)}{(n-k+i)(n-k+i-1)\cdots(n-k+1)}.
\]

We assumed \( k \leq \frac{n}{2} \) and hence \( n-k+i \geq n-k \geq k \), so for \( j \geq 0 \) (similar to 5.12)

\[
\frac{k-j}{n-k+i-j} \leq \frac{k}{n-k+i} \leq \frac{k}{n-k+1}.
\]
5.13: Volume versus Boundary (2)

It follows that for $i \geq 1$ (and also $i = 0$)

$$\frac{\binom{n}{k-i}}{\binom{n}{k}} \leq \left(\frac{k}{n-k+1}\right)^i.$$

Hence

$$\sum_{l=0}^{k} \binom{n}{l} \leq \binom{n}{k} \cdot \sum_{i=0}^{k} \left(\frac{k}{n-k+1}\right)^i$$

$$\leq \binom{n}{k} \cdot \frac{1}{1 - \frac{k}{n-k+1}} = \binom{n}{k} \frac{n-k+1}{n-2k+1} = \binom{n}{k} \left(1 + \frac{k}{n-2k+1}\right).$$
Inclass: PPZ on the formula $F^*$

Consider the formula $F^*_3$ on variables $\{x_1, x_2, \ldots, x_n\}$ containing

- all clauses on $x_1, x_2, x_3$ with at least one positive literal

- all clauses $\{\overline{x}_1, \overline{x}_2, x_i\}$ for $4 \leq i \leq n$

**Question:** What is the success probability of PPZ on $F^*$?
Inclass: PPZ on the formula $F^*$ (2)

Claim. The success probability is $1/\text{poly}(n)$.

Proof: With probability $\frac{1}{n(n-1)}$, $x_1$ and $x_2$ are the first two variables in the random permutation. Supposing that $x_1$ and $x_2$ are set correctly (which happens with probability 1/4), all other variables are forced. $\square$
6.2: Many $j$-isolated Satisfying Assignments

One possible way is to use the example from the lecture notes over $j$ variables (which will be critical in every assignment) and augment it with $n - j$ dummy variables not appearing in the formula (which will be non-critical in every assignment).
6.4: Make it Hard for PPZ

Since we, if possible, do not want to sum over multiple satisfying assignments, let us go for a formula with a unique isolated solution. In a unique solution, each variable has a critical clause. To minimize the success probability, we want to have exactly one critical clause per variable.

Moreover, for simplicity, we would like not to have long-range influences and dependencies between the variables, so the formula should consist of components as independent as possible.
6.4: Make it Hard for PPZ (2)

A good example is the formula $F$ consisting of $n/k$ independent components, where each component is a maximal satisfiable formula over $k$ variables.

This way, there is a unique solution: each component has one satisfying assignment and composing these is the only way to get a satisfying assignment for $F$. 
6.4: Make it Hard for PPZ (3)

And in each component, as long as less than $k-1$ variables are set, there cannot arise any unit clauses. Once $k-1$ variables are set in a component, there is always a unit clause forcing the last variable. Therefore exactly $n - n/k$ variables are being guessed, yielding exactly the success probability desired.
7.1: Covering Radius Example

Map any word $w \in \{0, 1\}^{3m}$ to its signature $\varphi(w) := (a, b, c) \in \{0..m\}^3$ where $a$, $b$ and $c$ are the number of ones in $w$ within the first, second and third $m$ bits.

We now have, because of the various words in the code $C$,

$$d(w, C) \leq \begin{cases} 
  a + b + c \\
  a + (m - b) + (m - c) \\
  (m - a) + b + (m - c) \\
  (m - a) + (m - b) + c 
\end{cases}$$

and thus, by adding the four inequalities,

$$4d(w, C) \leq 6m.$$
7.3: Exact Radius

Let $C$ be the code of radius $r' < r$.

Pick a point $w \in \{0, 1\}^n$ which maximizes the distance $d(w, C)$.

We clearly have $d(w, C) = r'$.

Now consider all codewords $v \in C$ at distance exactly $d(v, w) = r'$ and move all of them “away” from $w$ (e.g. by changing the first coordinate where $v$ and $w$ are the same), producing a code $C'$. 
7.3: Exact Radius

Clearly, $C'$ still has at most $M$ codewords. (why at most?)

The covering radius is now at least $r' + 1$ because the codeword closest to $w$ has distance $r' + 1$.

On the other hand, the covering radius is at most $r' + 1$ as well, as before it was $r'$ and we have moved codewords by only one position, so no distance can have increased by more than one.

Repeat the process until we have covering radius $r$. 