Unions, Linearity, and Satisfiability

1 Unions of Events

1.1 Union Lemmas

Lemma 1. *Union Bound: For any events $\mathcal{E}_1, \dots \mathcal{E}_n$,*

$$\Pr\left[\bigcup_i \mathcal{E}_i\right] \le \sum_i \Pr[\mathcal{E}_i].$$

Lemma 2. *Inclusion-Exclusion Principle: For any events $\mathcal{E}_1, \dots, \mathcal{E}_n$,*

$$\Pr\left[\bigcup_i \mathcal{E}_i\right] = \sum_i \Pr[\mathcal{E}_i] - \sum_{i<j} \Pr[\mathcal{E}_i \cap \mathcal{E}_j] + \sum_{i<j<k} \Pr[\mathcal{E}_i \cap \mathcal{E}_j \cap \mathcal{E}_k] - \dots$$

If the $\mathcal{E}_i$ are independent,

$$\begin{aligned}\Pr\left[\bigcup_i \mathcal{E}_i\right] &= 1 - \Pr[\bigcap_i \bar{\mathcal{E}_i}] \\ &= 1 - \prod_i \Pr[\bar{\mathcal{E}_i}] \\ &= 1 - \prod_i (1 - \Pr[\mathcal{E}_i]).\end{aligned}$$

Of course, if the $\mathcal{E}_i$ are disjoint,

$$\Pr\left[\bigcup_i \mathcal{E}_i\right] = \sum_i \Pr[\mathcal{E}_i].$$

1.2 Team Pairs Problem

Let's say we have n players and we form two teams by independently flipping a coin for each player. We do this k times. What is the probability that there are two players who are always on the same team (all k times)?

Let $\mathcal{E}$ be the event that such a pair exists. Let $\mathcal{E}_{ij}$ be the event that players i and j are always on the same team. Then $\mathcal{E} = \bigcup_{i,j} \mathcal{E}_{ij}$. Since $\Pr[\mathcal{E}_{ij}] = 2^{-k}$, by the union bound we have

$$\Pr[\mathcal{E}] \le \binom{n}{2} 2^{-k}.$$

Then by the inclusion-exclusion principle,

$$\begin{aligned}\Pr[\mathcal{E}] &\ge \sum_{i,j} \Pr[\mathcal{E}_{ij}] - \sum_{(i,j)\neq(i',j')} \Pr[\mathcal{E}_{ij} \cap \mathcal{E}_{i'j'}] \\ &= \binom{n}{2} 2^{-k} - \binom{\binom{n}{2}}{2} 2^{-2k} \\ &\ge \frac{n^2}{2} 2^{-k} - \frac{n^4}{8} 2^{-2k}\end{aligned}$$

Where the last line follows from $\binom{n}{2} \leq \frac{n^2}{2}$. Let's check a few examples to see how tight our bounds are.

$$\begin{aligned} k = 3\log n : \frac{1}{2n} - \frac{1}{8n^2} &\leq \Pr[\mathcal{E}] \leq \frac{1}{2n} \\ k = 2\log n : \frac{1}{2} - \frac{1}{8} &\leq \Pr[\mathcal{E}] \leq \frac{1}{2} \\ k = \log n : \frac{n}{2} - \frac{n^2}{8} &\leq \Pr[\mathcal{E}] \leq \frac{n}{2} \end{aligned}$$

Note the last bound is trivial! It only tells us our probability is between 0 and 1. So our bound is really only useful when $k \geq 2\log n$. In fact, when $k < \log n$, there is always such a pair (by a counting argument).

2 Linearity of Expectations

2.1 Definitions

For any random variables $X_1, \dots, X_n$ and any $\alpha_1, \dots, \alpha_n \in \mathbb{R}$,

$$\mathbb{E}\Big[\sum_i \alpha_i X_i\Big] = \sum_i \alpha_i \mathbb{E}[X_i].$$

Special cases: all $\alpha_i = 1$ or all $X_i \in \{0, 1\}$ implies $\mathbb{E}[X_i] = \Pr[X_i = 1]$.

Proof. By definition,

$$\begin{aligned} \mathbb{E}[X] &= \sum_j j \cdot \Pr[X = j] \\ &= \sum_{\omega \in \Omega} X(\omega) \cdot \Pr[\omega] \end{aligned}$$

So,

$$\begin{aligned} \sum_i \alpha_i X_i &= \sum_{\omega \in \Omega} \Big(\sum_i \alpha_i X_i(\omega)\Big) \Pr[\omega] \\ &= \sum_i \alpha_i \sum_{\omega \in \Omega} X_i(\omega) \Pr[\omega] \\ &= \sum_i \alpha_i \mathbb{E}[X_i]. \end{aligned}$$

□

2.2 Coat Check Problem

We have n people who each check a coat. At the end of the evening, each coat is returned to a random person. Each person gets one coat. What is the expected number of people who get their own coat back?

Let X be the number of people who get their own coat back. Then,

$$X_i = \begin{cases} 1 & \text{if coat } i \text{ goes to its owner} \\ 0 & \text{otherwise} \end{cases}.$$

Note that $X = \sum_i X_i$. Now,

$$\begin{aligned}
\mathbb{E}[X] &= \sum_i \mathbb{E}[X_i] \\
&= \sum_i \Pr[X_i = 1] \\
&= n \cdot \Pr[X_i = 1] \\
&= n \cdot \frac{1}{n} \\
&= 1.
\end{aligned}$$

So in expectation, one person gets their own coat back.

2.3 Coupon Collector Problem

There are n types of coupons, and in each round you draw an independently uniformly random coupon. How many rounds until you have at least one copy of each type?

Let X be the number of rounds until all coupons have been collected. As a first attempt, let X_i be the number of rounds until coupon i is collected for the first time. However, this does not work because $X \neq \sum_i X_i$. Instead, let X_i be the number of steps between seeing the $i-1^{st}$ distinct coupon and the i^{th} distinct coupon. Then $X = \sum_i X_i$ and $\mathbb{E}[X] = \sum_i \mathbb{E}[X_i]$ as desired.

Note that X_i is a geometric random variable. In each round we succeed (get a new coupon) with probability $p_i = (n+1-i)/n$. So,

$$\begin{aligned}
\mathbb{E}[X_i] &= \frac{1}{p_i} \\
&= \frac{n}{n+1-i}.
\end{aligned}$$

And,

$$\begin{aligned}
\mathbb{E}[X] &= \sum_{i=1}^{n} \frac{n}{n+1-i} \\
&= n \sum_{i=1}^{n} \frac{1}{i} \\
&= n \cdot H_n.
\end{aligned}$$

Since $H_n = \Theta(\log n)$, collecting n coupons takes $\Theta(n \log n)$ rounds in expectation. However, the majority of the time is spent collecting the last few coupons. Suppose we only need any 90% of coupons. Then,

$$\begin{aligned}
\mathbb{E}[x] &= \sum_{i=1}^{0.9n} \frac{n}{n+1-i} \\
&= n \sum_{i=0.1n}^{n} \frac{1}{i} \\
&= \Theta(n \cdot (\log n - \log 0.1n)) \\
&= \Theta(n \cdot \log \frac{n}{0.1n}) \\
&= \Theta(n).
\end{aligned}$$

Thus, we should design codes such that we can decode with any 90% (constant fraction) instead of needing all the “coupons”.

3 Satisfiability

3.1 Johnson's Algorithm

In SAT we are given a CNF formula Φ with clauses $C_1, \ldots, C_m$. Each $C_j = \bigvee_{i=1}^{k_j} \ell_{j,i}$. Each $\ell_{j,i}$ is a literal, either x_k or $\bar{x_k}$ for some k. Clause C_j is satisfied if and only if at least one of its $\ell_{j,i}$ is true. It is ***NP***-hard to decide if there is an assignment to the x_k that satisfies all of the C_j.

In the approximation version called Max-SAT, we want to find an assignment approximately satisfying as many clauses as possible. A simple algorithm is called Johnson's Algorithm, in which for each variable x_i independently, we make true with probability 1/2 and false otherwise.

Let Y be the number of satisfied clauses. Then $Y = \sum_j Y_j$, where

$$Y_j = \begin{cases} 1 & \text{if clause } C_j \text{ is satisfied} \\ 0 & \text{otherwise} \end{cases}.$$

Then,

$$\begin{aligned} \mathbb{E}[Y] &= \sum_j \mathbb{E}[Y_j] \\ &= \sum_j \Pr[C_j \text{ satisfied}] \\ &= \sum_j (1 - \Pr[\text{all } \ell_{j,i} \text{ false}]) \\ &= \sum_j 1 - 2^{-|C_j|}. \end{aligned}$$

In the special case where $|C_j| = k$, we have $\mathbb{E}[Y] = m(1 - 2^{-k})$. Since $OPT \leq m$, we have a $1 - 2^{-k}$ approximation. For 3-SAT, this is a 7/8-approximation.

Theorem 3. *(Hastad, 1997): For any $k \geq 3$, unless **P=NP**, Max-k-SAT cannot be approximated in polynomial time to within $1 - 2^{-k} + \epsilon$ for any $\epsilon > 0$.*

When clauses have different sizes, Johnson's Algorithm isn't as good. In particular, if we have some singleton clauses, then we only get a bound of 1/2. Next time, we'll try an LP rounding algorithm that gives a better guarantee.