1 Probabilistic Method

Consider that we are given some ‘bad’ events $A_1, \ldots, A_m$. Let $p_{A_i}$ be the probability that event $A_i$ occurs. The goal of the probabilistic method is to show there exists some way to avoid all bad events. This is done by running a random experiment and showing that $\Pr[\bigwedge \neg A_i] > 0$. The experiment has a non-zero probability of producing an outcome which avoids all the bad events. Therefore there must be some way to avoid all the bad events.

1.1 Examples

Example 1 (Independence) Consider that the bad events $A_1, \ldots, A_m$ are all independent. In order to apply the probabilistic method, we need that $p_{A_i} < 1$ for all $i$. This gives

$$\Pr[\bigwedge \neg A_i] = \prod_i (1 - p_{A_i}) > 0$$

since every term in the product is positive.

Example 2 (Union Bound) If we have no knowledge at all of the dependencies, then we can still use the union bound to apply the probabilistic method. We can do that if it is the case that $\sum_i p_{A_i} < 1$.

Example 3 (Lovasz Local Lemma) The Lovasz Local Lemma gives us something in between the above two examples. If the bad events are not dependent on too many other events and the probabilities that the bad events occur are not too large, then we know there exists some way of avoiding all the bad events.

2 Hypergraph Coloring

Here we will use Lovasz Local Lemma to show a hypergraph coloring exists.

Definition 4 A hypergraph is a pair of two sets $(V, E)$ referred to as vertices and edges (or hyperedges). The edges are subsets of the vertices. In the case when all the sets in $E$ have size 2 this is a graph.

Definition 5 A valid hypergraph coloring is a coloring of the vertices of a hypergraph so that no edge is monochromatic.
This means that an edge in a hypergraph, could have it’s vertices colored red, red, red, red, and blue and the edge would be colored correctly as not all the vertices are the same color.

Now let us define what the bad events are for this problem. We will let $A_i$ be the event that edge $e_i$ is monochromatic. Therefore we know if no bad event occurs ($\cap_i \neg A_i$) then we have a valid coloring.

Let’s compute $p_{A_i}$. We will use $c$ colors and say that $e_i$ has size $k_i$. An edge’s size is the number of vertices it is adjacent to. So, we get

$$p_{A_i} = \left(\frac{1}{c}\right)^{k_i} c = \left(\frac{1}{c}\right)^{k_i - 1}$$

Since there are $c$ colors, and the probability that $e_i$ is monochromatic in a given color is $\frac{1}{c}$. 

**Definition 6** We call a hypergraph $k$-uniform if every edge contains exactly $k$ vertices.

Now if we consider that our hypergraph is $k$-uniform with $m$ edges. We get that $p_{A_i} = \frac{1}{c^{k-1}}$. So, now if we apply the union bound then we want that:

$$\sum_i p_{A_i} = \sum_i \frac{1}{c^{k-1}} = \frac{m}{c^{k-1}} < 1$$

So, when $m < c^{k-1}$ we can color a $k$-uniform hypergraph with $m$ edges using $c$ colors.

To try and use independence, then we would need the edges to be independent of each other. This could happen if all the edges in the hypergraph were disjoint from each other. We would now also need to guarantee that $\frac{1}{c^{k-1}}$ is less than 1. So, we would need $c \geq 2$ and $k_i \geq 2$ for all $i$. This case is rather uninteresting though as most graphs have edges which intersect.

We can improve on both of these by using Lovasz Local Lemma. Now to use the Lovasz Local Lemma we will need to understand dependencies between the different events.

**Definition 7** Events $B_1, \ldots, B_\ell$ are mutually independent if

$$\Pr[B_i | \beta] = \Pr[B_i]$$

for all choices of $B_i$ and $\beta$ is a set containing $B_j$’s and $\neg B_j$’s but not $B_i$ or $\neg B_i$.

**Definition 8** A graph, $G = (\{A_1, \ldots, A_m\}, E)$, is a dependency graph if for every $A_i$, then $A_i$ is mutually independent of $A_i \setminus \Gamma(A_i)$. In other words, $A_i$ is independent of all the events it is not adjacent to in the graph.

For the hypergraph coloring problem, we get that $A_i$ is adjacent to $A_j$ in the dependency graph if $A_i$ and $A_j$ have a vertex in common.
So, how could the coloring fail locally? If it failed to satisfy the union bound locally:

\[ \sum_{B \in \Gamma^+(A)} p_B \geq 1 \]

Here \( \Gamma^+(A) \) is all the neighbors of \( A \) in the dependency graph including \( A \) itself. In other words, it’s possible that every coloring fails at \( A \) or one of it’s neighbors.

The Lovasz Local Lemma is really the idea that if things work out locally, then it will work out everywhere.

**Theorem 9 (Lovasz Local Lemma)** If \( G \) is the dependency graph and

\[ \sum_{B \in \Gamma^+_G(A)} p_B < \frac{1}{e} \]

for all \( i \), then \( \Pr[\land_i \bar{A}_i] > 0 \). More specifically we know that:

\[ \Pr[\land_i \bar{A}_i] > \prod_{A_i}(1 - e p_{A_i}) \]

Note that Lovasz Local Lemma is like a generalization of Union Bound and Independent events with some slack required. If we apply Lovasz Local Lemma in the case of the union bound, we have the complete graph for our dependency graph. The Lovasz Local Lemma would require that \( \sum p_{A_i} < \frac{1}{e} \). In the case of independent events, then our dependency graph is the empty graph. So, the Lovasz Local Lemma would require \( p_{A_i} < \frac{e}{2} \) for all \( i \).

Also, the Lovasz Local Lemma is tight. If we loosen the constraint to be

\[ \sum_{B \in \Gamma^+_G(A)} p_B < \frac{1}{e} + \epsilon \]

then we can no longer guarantee there is a non-zero probability we avoid all bad events.

Back to hyperedge coloring, we will constrain ourselves to \( k \)-uniform graphs were each edge intersects at most \( \Delta \) other edges. Then we now want to guarantee that

\[ (1 + \Delta) \left( \frac{1}{e} \right)^{k-1} < \frac{1}{e} \]

So, we just need that \( \Delta + 1 < \frac{e^{k-1}}{e} \). This is nice because you don’t need to bound the total number of edges, just the number of edges locally.

### 3 \( k \)-SAT

The problem of \( k \)-SAT is to find a satisfying assignment to a boolean expression which has clauses which all have \( k \) literals and these literals are OR’ed together, and then all the clauses are AND’ed together. In other words, we must find a satisfying assignment so that each clause of \( k \) literals has some literal set to true.
Theorem 10 If a $k$-SAT instance has every variable occurring in less than $\frac{2^k}{ek}$ clauses, then the instance is satisfiable.

Proof We will set each literal to TRUE with probability $1/2$ and to false with probability $1/2$. Let $A_i$ be the event that clause $i$ is unsatisfied. So, we have $p_{A_i} = \frac{1}{2^k}$ for all $i$. Now let us consider the dependency graph. $A_i$ is dependent only on those clauses it shares a variable with. The $i$th clause has less than $k \cdot \frac{2^k}{ek} = \frac{2^k}{e}$ clauses it shares a variable with. Therefore we get

$$\sum_{B \in \Gamma^+(A)} p_B = \sum_{B \in \Gamma^+(A)} \frac{1}{2^k} < \frac{2^k}{e} \cdot \frac{1}{2^k} = \frac{1}{e}$$

and satisfy the conditions of the Lovasz Local Lemma. So, some satisfying assignment for this instance exists. ■

We can’t do much better than $\frac{2^k}{ek}$. Consider we have the case where we have $k$ literals and all $2^k$ clauses containing the $k$ literals. For every assignment, some clause is not satisfied. So, if literals appear in $2^k$ clauses then we can no longer guarantee satisfiability.

4 Routing

We will consider that we are given $m$ directed paths of length $D$ (the dilation) and we want to transport a packet along each path. The paths might intersect and each edge can only send one packet in a round.

Clearly, we can transport the packets in $Dm$ rounds simply by using the first $D$ rounds to send the first packet, the second $D$ rounds to send the second packet, and so on.

Now consider that we also know that each edge is used at most $C$(the congestion) times. Now, clearly we can do this in $CD$ time, simply send a packet along an edge if there is a packet to send. Each packet has to travel $D$ edges, and may have to wait for the edge to be used by the other $C - 1$ packets. So, each packet takes at most $C$ rounds to cross each edge.

What is the best possible we could hope for?

- $D$, we need at least $D$ rounds for each packet to travel along it’s path.
- $C$, we need at least $C$ rounds so that each edge is able to handle all the packets associated with it.

The best we could hope for is $O(C + D)$.

Theorem 11 We can route all the packets along their paths in time $(C + D) \log mD$. 
**Proof** We will route the packets randomly by holding each packet at its source and then when we release it, we will hope that it just goes through the whole path without interruption.

Notice, if we delay each packet by 1 with probability 1/2 or let it go immediately with probability 1/2 then we expect each edge to have to handle C/2 packets at a time.

If we instead delay the packet by some amount from 0 to c — 1 chosen uniformly at random, then we expect to get congestion 1 at each edge. With high probability though, we get some edge with congestion log c though.

The expected congestion on an edge is at most 1. Now if we apply the Chernoff bound, then we get that:

\[
\Pr[\text{congestion on one edge} \geq 4 \log mD] \leq e^{-16 \log^2 (mD)/3} < \left( \frac{1}{mD} \right)^5
\]

So, we get with high probability the congestion on an edge is log mD. To deal with this, we break each of the original rounds into log mD rounds so each edge can handle its congestion. So, we only need \( (D + C) \log mD \) rounds to handle everything.

Next time, we will use Lovasz Local Lemma to achieve \( O(D + C) \), which doesn’t depend on \( m \).