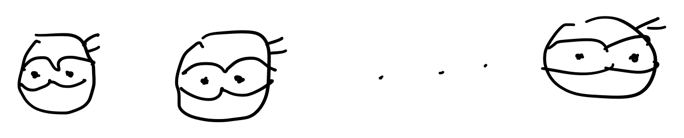
Lecture 1: Random Testing with Theoretical Guarantees

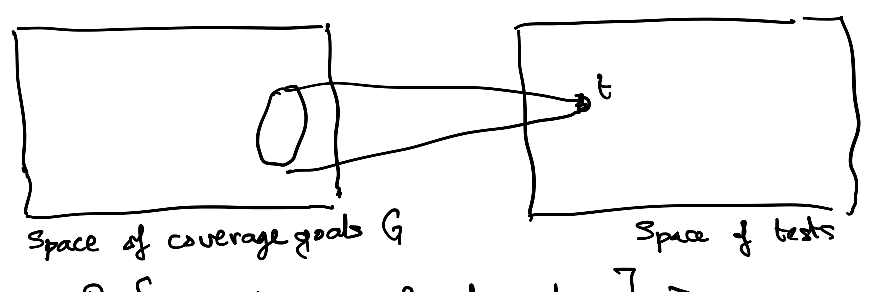
Thursday, 1. August 2019 17:46

The Power of Random Sampling

You can follow live on:
http://tiny.cc/byssaz
https://1drv.ms/o/s!AufrpyzkMX1pcZcVPUGorRH7xMI



What we have so far:



$\Pr_t[\text{Test } t \text{ covers fixed goal } g] \geq p$

Claim: There is a covering family of size $\frac{1}{p}\log|G|$

$\Pr[\text{Test } t \text{ does not cover } g] \leq 1-p$

$\Pr[K \text{ tests do not cover } g] \leq (1-p)^K$

$\Pr[K \text{ tests do not form a covering family}] \leq |G|(1-p)^K$

Pick $K = \frac{1}{p}\lg|G|$

$\Pr(\_\_\_\_\_\_\_\_) \leq |G|(1-p)^K < 1$

$\mathcal{F} \quad |\mathcal{F}| \geq \frac{1}{p}(\lg|G| - \lg\varepsilon)$

Then $\mathcal{F}$ is a covering family w. prob $\geq 1-\varepsilon$

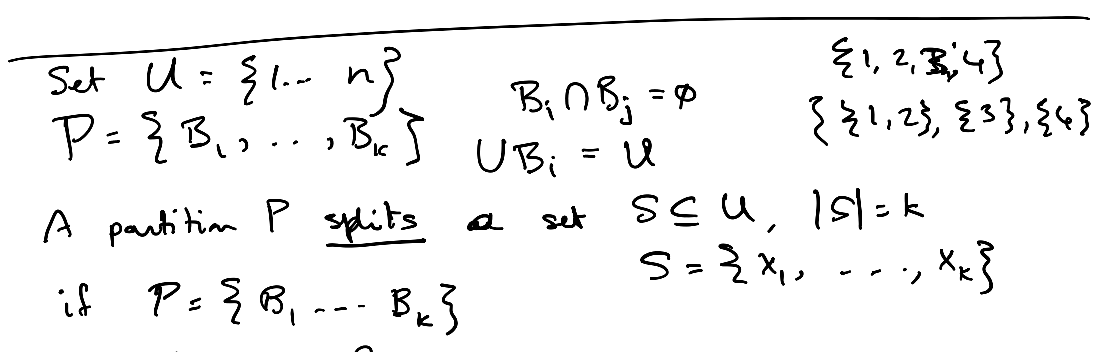
---

k-splitting coverage goal

Given a set of k nodes in a system with n nodes $(k \ll n)$ the goal is to partition the system so every one of these k nodes are in a different block.

$|G| = \binom{n}{k}$

Puzzle: Ninjas training



$\lfloor\lg n\rfloor + 1$ rounds are enough for training to be complete

---

Set $\mathcal{U} = \{1 \ldots n\}$ $\quad B_i \cap B_j = \emptyset$ $\quad \{1, 2, 3, 4\}$

$P = \{B_1, \ldots, B_k\}$ $\quad \cup B_i = \mathcal{U}$ $\quad \{\{1,2\}, \{3\}, \{4\}\}$

A partition P **splits** a set $S \subseteq \mathcal{U}$, $|S| = k$

$S = \{x_1, \ldots, x_k\}$

if $P = \{B_1 \ldots B_k\}$

and $x_1 \in B_1$
$x_2 \in B_2$
⋮
$x_k \in B_k$

The number of partitions of n elements into k blocks

$\left\{ {n \atop k} \right\}$ "Stirling # of the 2nd kind" $S(n,k)$

$\left\{ {n \atop 1} \right\} = 1 \quad \left\{ {n \atop n} \right\} = 1 \qquad$ Ex: $\left\{ {n \atop n-1} \right\} = \binom{n}{2}$

$\left\{ {n \atop 2} \right\} = 2^{n-1} - 1$

$\left\{ {n \atop k} \right\} = \left\{ {n-1 \atop k-1} \right\} + k\left\{ {n-1 \atop k} \right\}$

Lemma: $\left\{ {n \atop k} \right\} k! \leq k^n$

(# of surjective fn from $[n]$ to $[k]$) $\leq$ (# of fn from $[n]$ to $[k]$)

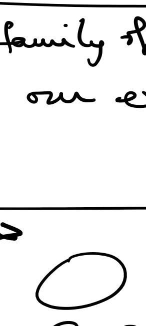
Theorem Let $S \subseteq \mathcal{U}$, $|S| = k$

Let p be the Pr random k-partition splits S

$$p = \frac{k^{n-k}}{\left\{ {n \atop k} \right\}} \geq \frac{k!}{k^k}$$

There is a covering family $\frac{k^k}{k!}\log n^k = \frac{k^{k+1}}{k!}\lg n$

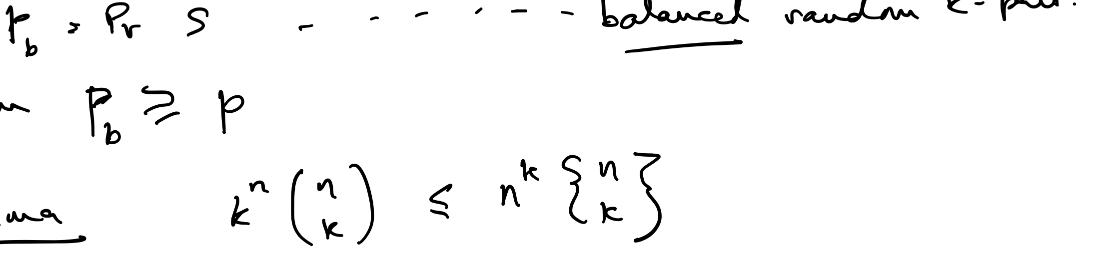
A k partition that splits S is uniquely determined by a map $\mathcal{U}\setminus S \to S$



Pr random k-partition splits S

$$\frac{k^{n-k}}{\left\{ {n \atop k} \right\}} \underset{\text{Lemma}}{\geq} k^{-k}\, k!$$

Ex $k=2$ we have a k splitting family of size $\lceil 2\lg n - 1\rceil$ (which is slightly worse than our explicit bound of $\lfloor\lg n\rfloor + 1$



Theorem Let $|S| = k$, $S \subseteq \mathcal{U}$.

If $p = \Pr$ S is split by a random k-partition

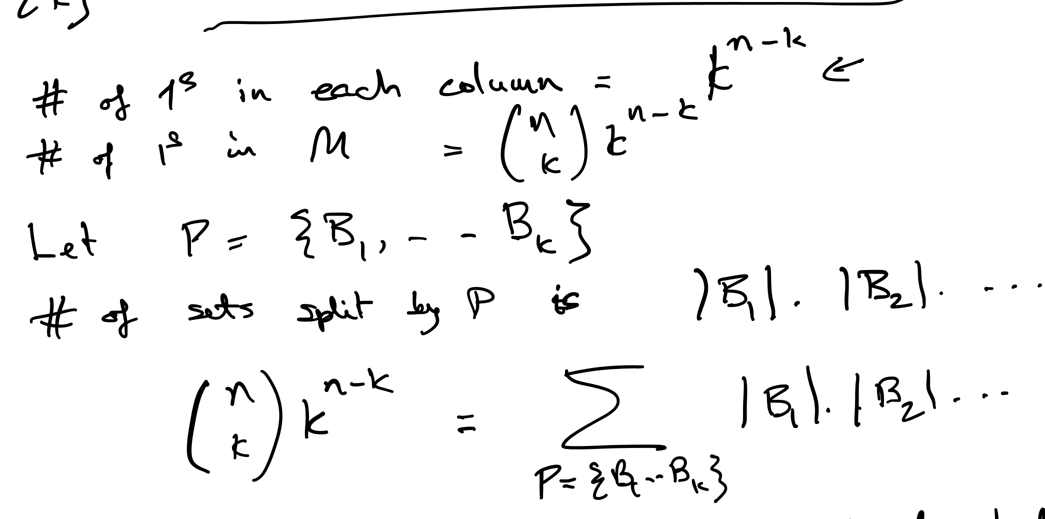
$p_b = \Pr$ S - - - - - - - **balanced** random k-part.

Then $p_b \geq p$

Lemma $k^n \binom{n}{k} \leq n^k \left\{ {n \atop k} \right\}$

Let's rewrite this as:

$$k^{n-k}\binom{n}{k} \leq \left(\frac{n}{k}\right)^k \left\{ {n \atop k} \right\}$$



# of 1s in each column $= k^{n-k}$

# of 1s in M $= \binom{n}{k}k^{n-k}$

Let $P = \{B_1, \ldots B_k\}$

# of sets split by P is $|B_1| \cdot |B_2| \cdot \ldots |B_k|$

$$\binom{n}{k}k^{n-k} = \sum_{P=\{B_1 \ldots B_k\}} |B_1| \cdot |B_2| \ldots |B_k|$$

$B = |B_1| \ldots |B_k|$ is maximized for balanced partitions.

Suppose: $|B_1| - |B_2| \geq 2$

$B = |B_1| \cdot |B_2| \cdot B'$

New product: $(|B_1| - 1)(|B_2| + 1) \cdot B'$

$(|B_1||B_2| + \underbrace{|B_1| - |B_2|}_{\geq 2} - 1) \cdot B'$

$> |B_1| \cdot |B_2|$

$$\sum_P |B_1| \ldots |B_k| \leq \left(\frac{n}{k}\right)^k \cdot \left\{ {n \atop k} \right\}$$

$$\binom{n}{k}k^{n-k} = \sum_P |B_1| \ldots |B_k| \leq \left(\frac{n}{k}\right)^k \left\{ {n \atop k} \right\}$$

Covering Array

Given n config. parameters, each taking v values, and a parameter k.

Test: Vector of n values, in v^n.

Find $N(n,k,v)$ of tests s.t for every subset of k config parameters and every vector ω in v^k, there is a test in $N(n,k,v)$ whose projection to the k parameters is ω.

What is the size of $N(n,k,v)$?
(Prob method!)