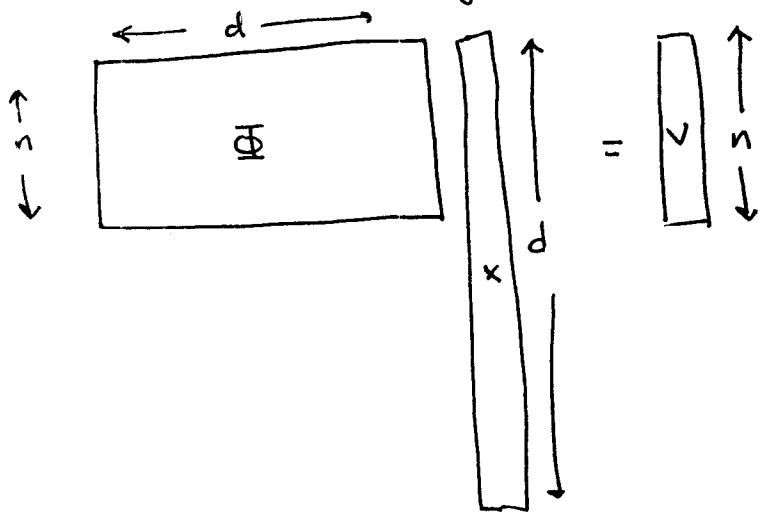
What makes sublinear algs. so fast?

(1) statistical signal recovery ← WHAT problem



observe signal $x \in \mathbb{R}^d$, linear measurements Φ

record observations $v \in \mathbb{R}^n$. Retain Φ, v.

Output: info. about x

ℓ_p norm, median (order statistics), x itself,

rep^n of x in basis/dictionary, 0/1 decision about x,

$\langle x, q \rangle$ with arbitrary q

Example of sublinear alg:

[these types of results have a lot of moving parts — let's try to understand pieces.]

Thm: [GSTV '07]

With high prob., a random matrix Φ with $m \text{ polylog}(d/\varepsilon)$ rows is a good meas. matrix. $\exists$ alg. s.t. given $\Phi x = v$ and m = sparsity, alg. outputs $\hat{x}$ (m non-zero entries) s.t. ($\hat{x}$ has optimal error guarantees.)

$$\|x - \hat{x}\|_2 \leq \|x - \hat{x}\|_1 + \frac{\varepsilon}{\sqrt{m}} \|x - \hat{x}\|_2$$

in time $O(m^2 \varepsilon^{-4} \text{polylog } d)$.

(2) In these lectures, we'll walk through 2 canonical stat. signal recovery algs. SLOWLY!

① Combinatorial Group Testing / Hashing

input: x binary vector, length d

m non-zeros

output: m pos'ns of spikes (or reasonable fract'n)

② 2nd freq. moment, energy of stream

input: $x \in \mathbb{R}^d$ in a data stream

output: $\|x\|_2^2 = \sum_{i=1}^{d} |x(i)|^2$ (or close approx'n)

(3) Models / Applications ← WHERE do these probs. arise?

(0) "Compressed Sensing"

(i) Data streams (IP networks, phone call records, transact'n streams)

items arrive sequentially

define a signal $x: [1, \ldots, d] \to \mathbb{R}$ implicitly

rep. transactions

$\langle i, u(i) \rangle$

$i \in [1, \ldots, d]$ index

$u(i)$ = update to $x(i) \Rightarrow x(i) \mathrel{+}= u(i)$.

domain vals presented in arbitrary order

updates are positive OR negative

d = domain size HUGE!

Alg'ic challenges in data streams

size, volume enormous + most hardware simple, cheap!

ex: OC192 backbone link 9.9 Gigabits/sec

IP pkt hdrs = 30 million items/sec

avg. flow = 10 pkts ⇒ lasts 3.3×10^{-7} sec.

(4) Computational resources — WHAT are we charged for? i.e., What could be sublinear?

(1) SPACE – have to store Φ, V

need auxiliary, working memory

bits, precision

all require RAM, physical space, power

ex: all pkt hdrs on OC192 for 1 month = 1000s PBytes

(2) TIME – have to output answer

operate on Φ, V

post-processing: apply Φ

CS / data stream models.

(a) NOT charged to collect V
except 1 entry = unit cost

(b) charged PER UPDATE to x

ex: router has to process each pkt quickly

(3) COMMUNICATION – send $\underline{\Phi}x, v$ to central location

(4) RANDOMNESS – $\underline{\Phi}$ is a random matrix
generate AND store $\underline{\Phi}$, + random bits

More practically, hardware = SWEPT metrics.
size, weight, energy, power, time

(5) Sublinear Algs.

Informally, resource usage grows slower than size of input. If double input size, use LESS than 2x more resources to compute answer.

ex: input size = d

space/time $O(d^\alpha)$ for $\alpha < 1$

$O(\log d)$

polylog(d) = $\log^c(d)$ for some c.

Consequences of sublinear resources (or HUGE data!)

- can't output x itself
- can't take too many meas.
- can't store $\underline{\Phi}$ explicitly
- can't do bit stuffing
- can't use lots of indep. bits
- post process fast
- one pass

HARD:
data stream of active ph. #s
1 million phone #s
see 100K records
only 4 active at time
you look
where are they?!

CGT

- Goes back to WWII, 2 economists in OPA wanted an efficient method for testing draftees for syphilis

  (1) draw blood from each person

  (2) pool blood samples into groups of 5

  (3) use a single test on each group

  OBSERVE

  (1) If no one in pool has antigen, then test negative

  (2) If ≥ 1 people have antigen, then pool does and test is positive

  ↳ can then test each individually

∴ If no one has antigen, save 4 tests (over all 5)

If ≥ 1 —— ", waste 1 test (over all 5)

Def^n (CGT) Have a universe of d items, m are "defective"

GOAL: Construct a collection of tests (a design) to minimize the # tests needed to find defective set for worst case input.

Not specified: (1) type of tests

binary / not

linear / not

adaptive / not

det. / prob.

(2) algorithm

WANT: linear, binary, non-adaptive tests

good & fast algs.

prob. ok.

WARM UP EXAMPLE: x = binary signal, length d, 1 sparse

Defn: Let B_1 be a binary matrix with $n = \lg d$ rows and d cols. The i^{th} col. of B_1 is i written in binary. Bit tester matrix

$\overline{B}_1$ = bit tester with add'l top row == 1.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

Algorithm: ① Compute $B_1 x = V$

② output V = pos^n of defect (in binary)

Analysis: ① space for $V = \lg d$

② alg. is trivial, takes time $\lg d$ to output

③ don't have to store B_1 explicitly

receive update in pos^n i,

generate i in binary,

for each entry in i, multiply update

accumulate meas.

PROBLEM #1: let x be m-sparse binary vector length d

GOAL: pos^{ns} of m non-zero entries of x.

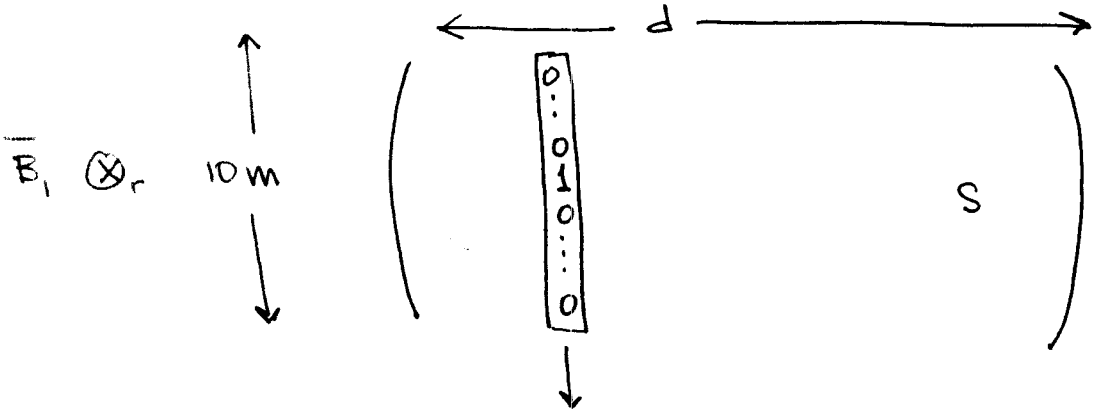
Lemma: There is a random measurement matrix Φ and an ~~[illegible]~~ alg that, for each x of length d with m defects, returns a list of at most $10m$ items that contains at least $\frac{m}{10}$ defects with prob. $\geq \frac{8}{9}$. The alg. takes time $O(m \lg d)$

[and uses $O(m \lg d)$ measurements.]

Def^n: $B \otimes_r S = M$ = row tensor product of 2 matrices.

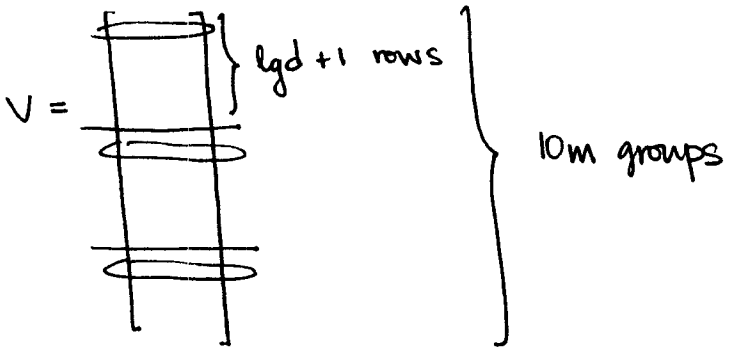
Each row of M = elt. wise prod. of 1 row in B and 1 row in S.
[loop over all rows of B, S to get elt. wise prod. over all pairs of rows from B, S.]

Measurement Φ: $\Phi = \overline{B}_1 \otimes_r S$ = binary matrix with $10m \lg d$ rows, d cols.



1 non-zero entry per col.
pos^n chosen indep'ly, unif'ly at random
i.e., choose row with prob = $\frac{1}{10m}$

Algorithm:



List = $\emptyset$
for each of $10m$ blocks
  check 1st entry, if non zero
    then lgd rows give pos^n of spike
    add to list.
output list.

Analysis:

- observe that each row of S selects entries of x at random
  there are really only m entries we care about
  $\Rightarrow$ S is placing m balls into $10m$ bkts randomly
  $\therefore$ need to analyze balls/bins process
  [enemy = collisions = balls in same bin]

- Assuming that a row of S isolates a spike, then $\overline{B}_1$ finds it with the right label.
  Note top row just sums all entries in bkt.

- Analyze balls/bins and collisions
  Consider a particular spike x_j
  it's in some group k; $S(x_j) = k$ by abuse of notation

- Conditional prob. that another spike x_ℓ is assigned to k by S

$$\Pr\left(S(x_\ell) = k \mid S(x_j) = k\right) = \frac{\Pr\left(S(x_\ell) = k \wedge S(x_j) = k\right)}{\Pr\left(S(x_j) = k\right)}$$

independence of balls/bins assign.

$$= \Pr\left(S(x_\ell) = k\right) = \frac{1}{10m}$$

- Let $Y_\ell = X_k(S(x_\ell)) = \begin{cases} 1, & \text{prob } \frac{1}{10m} \quad [\text{if } x_\ell \text{ lands in } k] \\ 0, & \text{prob. } 1 - \frac{1}{10m} \quad [\text{o.w.}] \end{cases}$

Then let Y = # other spikes in group k

$$= \sum_{l=1}^{m-1} Y_l$$

and $\mathbb{E}(Y)$ = expected # other spikes in k

$$= \mathbb{E}\left(\sum_{l=1}^{m-1} Y_l\right) = \sum_{l=1}^{m-1} \mathbb{E}(Y_l)$$

linearity of expectation.

$$= \frac{m-1}{10m} = \frac{1}{10} - \frac{1}{10m} < \frac{1}{10}$$

i.e., on avg. # other spikes assigned is $\frac{1}{10}$

$$\Rightarrow \Pr(Y \geq 10 \cdot \mathbb{E}(Y)) = \Pr(Y \geq 1)$$

$$\leq \frac{\mathbb{E}(Y)}{10 \cdot \mathbb{E}(Y)} = \frac{1}{10}$$

$$\therefore \Pr((Y \geq 1)^c) = \Pr(\text{no other spike in } k) \geq \frac{9}{10}$$

$\Rightarrow$ each defect is isolated in its own group with prob. $\geq \frac{9}{10}$ (and it fails to be isolated prob $\leq \frac{1}{10}$)

- Let's look at all the balls now, not just one.

Let $Z_j = \begin{cases} 1, & \text{if } x_j \text{ NOT isolated, prob } \frac{1}{10} \\ 0, & \text{if } x_j \text{ isolated, prob } \frac{9}{10} \end{cases}$

and set $Z = \sum_{j=1}^{m} Z_j$ = # spikes NOT isolated

$$E(Z) = \sum_{j=1}^{m} E(Z_j) \leq \frac{m}{10} = \text{expected \# of spikes NOT isolated}$$

again, $$\Pr\left(Z \geq \frac{9m}{10}\right) \leq \frac{\frac{m}{10}}{\frac{9m}{10}} = \frac{1}{9}$$

i.e., # of balls we fail to isolate exceeds $\frac{9m}{10}$

with prob. $\frac{1}{9}$ and

$$\Pr\left(\left(Z \geq \frac{9m}{10}\right)^c\right) \geq \frac{8}{9}$$

with prob. $\frac{8}{9}$ we isolate at least $\frac{m}{10}$ defects.

Note that total # groups = $10m$, so the # false positives is under control. Actually, in this simple example, we KNOW when we have a false positive BUT... you can generalize this problem and still use same framework. //

Extensions

① iterate to get more (all) spikes

② increase success prob. with repeated trials, high enough prob. to get ALL m-sparse sig.

STILL NEED

① small-space

② limited randomness

Pairwise indep. and hash functions

Defn: A set of rvars $X_1, X_2, \ldots, X_n$ is **k-wise indep.** if, for any subset $I \subseteq [1, \ldots, n]$ with $|I| \leq k$ and for any vals. x_i, $i \in I$,

$$\Pr\left(\bigcap_{i \in I} X_i = x_i\right) = \prod_{i \in I} \Pr\left(X_i = x_i\right).$$

Example: pairwise indep. mod prime p

Construct $Y_0, Y_1, \ldots, Y_{p-1}$ pairwise indep. rvars that are uniform over $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$.

Use 2 indep. uniform vals. $X_1, X_2 \in \mathbb{Z}_p$.

Let $Y_i = (X_1 + i X_2) \bmod p$.

Lemma: $Y_0, \ldots, Y_{p-1}$ are 2-wise indep. unif. rvars over $\mathbb{Z}_p$.

proof: (1) Each Y_i unif'ly distributed over $\mathbb{Z}_p$.

Given X_2, possibilities for $X_1 \Rightarrow p$ distinct vals. for $Y_i \bmod p$. unif. too. ✓

(2) Consider Y_i, Y_j and $a, b \in \mathbb{Z}_p$. Want

$$\Pr(Y_i = a \cap Y_j = b) = \frac{1}{p^2}.$$

$$\left.\begin{aligned} Y_i &= a \\ Y_j &= b \end{aligned}\right\} \iff \left\{\begin{aligned} X_1 + iX_2 &= a \bmod p \\ X_1 + jX_2 &= b \bmod p \end{aligned}\right.$$

↑ syst. of eq^{n}s
2 unknowns, 2 eq^{n}s
∴ 1 sol^{n}!

$$X_2 = \frac{(b-a)}{(j-i)} \bmod p \qquad X_1 = a - \frac{i(b-a)}{(j-i)} \bmod p$$

X_1, X_2 are unif iid over $\mathbb{Z}_p$ ∴ done. ✓

Hash functions

We've defined $Y_0, \ldots, Y_{p-1}$ as a coll^{n} of rvars but we can also view them as a function $Y: \mathbb{Z}_p \to \mathbb{Z}_p$ which takes elts. or indices in $\mathbb{Z}_p$ and maps them randomly (pwise indep'ly) to other elts. in $\mathbb{Z}_p$.

Y is called a hash function – it takes a collection of items $\{0,1,..,p-1\}$ and hashes them into $\mathbb{Z}_p$.

$$Y(0) = Y_0$$
$$Y(1) = Y_1$$
$$\vdots$$
$$Y(p-1) = Y_{p-1}.$$

Defn: Let U be a universe with $|U| \geq n$ and let $V = \{0,1,..,n-1\}$. A family of hash functions $\mathcal{H}: U \to V$ is k-universal if for any elts. $x_1, x_2, .., x_k$ and for a hash function h chosen unif'ly at random from $\mathcal{H}$, we have

$$\Pr\left(h(x_1) = h(x_2) = \ldots = h(x_k)\right) \leq \frac{1}{n^{k-1}}.$$

We want $|U| = d \gg n$; i.e., hash a large # items into a smaller # bins.

and, frequently, just 2-universal hash functions, as we care about collisions.

Example: 2-universal hash function

Let $U = \{0, 1, \ldots, d-1\}$ universe

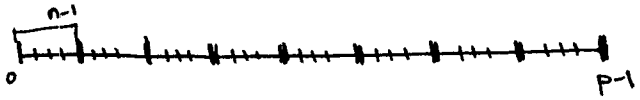
$V = \{0, 1, \ldots, n-1\}$ $n < d$ (= range of hash)

choose prime $p \geq d$.

Consider family $\mathcal{H}: U \to V$ given by

$$\mathcal{H} = \{ h_{a,b} \mid a \in \mathbb{Z}_p^*, b \in \mathbb{Z}_p \} \qquad a \neq 0!$$

$$h_{a,b}(x) = ((ax + b) \bmod p) \bmod n$$



Lemma: $\mathcal{H}$ is 2-universal.

proof: Need to look at how many of the functs $h_{a,b}$ have collisions; i.e., as a function of a,b for $x_1 \neq x_2$ pairs x_1, x_2, how many have

$$h_{a,b}(x_1) = h_{a,b}(x_2) \; ?$$

omit ...

To apply this to our CGT problem —

recall that $n = 10m$ (hash into $10m$ bkts)

and that we needed

$$\Pr\left(S(x_\ell) = k \mid S(x_j) = k \right) = \frac{1}{10m} = \frac{1}{n}$$

- If S is one of our hash functions, then we're done.
- $\therefore$ pick $a \in \mathbb{Z}_p^*$, $b \in \mathbb{Z}_p$ unif'ly indep'ly at random
  ↳ $\Theta(\lg p) = \Theta(\lg d)$ bits of randomness

- to compute

$$S(x_\ell) = h_{a,b}(x_\ell) = \left[(a x_\ell + b) \bmod p \right] \bmod n .$$

ℓ_2 norm of data stream

Alon, Matias, Szegedy '96.

input: data stream $\langle i, u(i) \rangle$ defines $x \in \mathbb{R}^d$ implicitly

GOAL: compute $\sum_{i=1}^{d} |x(i)|^2 = \|x\|_2^2$ highly eff'ly.

Thm: (AMS '96)

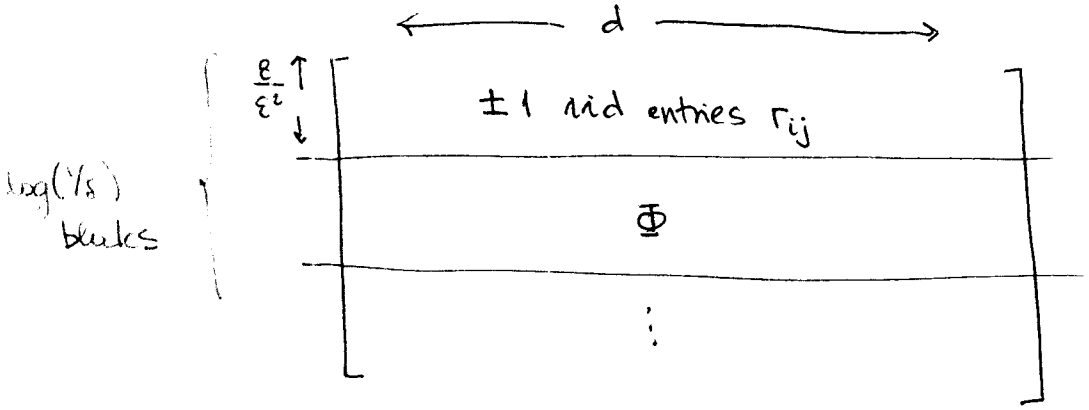
There is a streaming alg. which takes as input a stream of records (defining x), and for each x, with prob. $1-\delta$, produces an estimate Z of $\|x\|_2^2$ which sat.

$$(1-\varepsilon)\|x\|_2^2 \le Z \le (1+\varepsilon)\|x\|_2^2 .$$

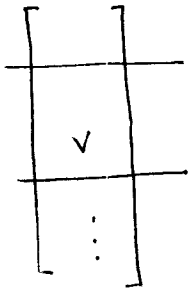
The alg. uses $O(\frac{1}{\varepsilon^2}\log(\frac{1}{\delta}))$ space and both returns the estimate and process each stream record in time $O(\frac{1}{\varepsilon^2}\log(\frac{1}{\delta}))$. The alg. uses $O(\log d)$ random bits.

proof:

Measurements:



Algorithm:



each row j of v is an estimator $X_j = \sum_{i=1}^{d} r_{ij} x(i)$

Let $Z_j = X_j^2 = \left(\sum_{i=1}^{d} r_{ij} x(i)\right)^2$

for each block k of $\frac{8}{\varepsilon^2}$ estimators Z_j, compute

$Y_k = \frac{\varepsilon^2}{8}\sum_j Z_j$ average Z_j on each block.

- output $W = \underset{k}{\text{Median}}\ Y_k$ = median over all blocks.

Analysis:

(1) Z_j is unbiased est.

$$\mathbb{E}(Z_j) = \mathbb{E}(x_j^2) = \sum_i \mathbb{E}(r_{ij}^2)\, x(i)^2 + \sum_i \sum_{i \neq l} \mathbb{E}(r_{ij})\,\mathbb{E}(r_{lj})\, x(i)\, x(l)$$

$$= \|x\|_2^2$$

(2) 2^{nd} moment of Z_j

$$\mathbb{E}(Z_j^2) = \mathbb{E}(x_j^4) = \sum x(i)^4 + 6 \sum_{i \neq l} x(i)^2\, x(l)^2$$

$$\Rightarrow \text{Var}(Z_j) = \mathbb{E}(Z_j^2) - \mathbb{E}^2(Z_j)$$

$$= 4 \sum_{i \neq l} x(i)^2\, x(l)^2$$

$$\leq 2 \|x\|_2^4$$

(3) Y_k = avg. of $\frac{8}{\varepsilon^2}$ unbiased est. $\Rightarrow$ Y_k also unbiased

$$\mathbb{E}(Y_k) = \|x\|_2^2$$

$$\text{Var}(Y_k) = \frac{\varepsilon^2}{8} \text{Var}(Z_j) \leq \frac{\varepsilon^2 \|x\|_2^4}{4}.$$

(4) Chebyshev / Markov

$$\Pr\left(|Y_k - \mathbb{E}(Y_k)| \geq \varepsilon \|x\|_2^2\right) \leq \frac{\text{Var}(Y_k)}{\varepsilon^2 \|x\|_2^4} \leq \frac{1}{4}.$$

5) W = median of $\log(1/\delta)$ vars Y_k

If $|W - \|x\|_2^2| > \varepsilon \|x\|_2^2$, then half of Y_ks

are too large, but we expect this to happen

for only 1/4 of Y_ks.

Chernoff $\Rightarrow$ bad events happen for $> \frac{1}{2} Y_k$s

with prob. exponi'ly low in # vars $= \Theta(\log 1/\delta)$. ✓

Discussion:

(1) Did we need full independence? NO!

just need 4-wise to bound $\mathbb{E}(Z_j^2)$

(2) How do we construct 4-wise indep. ± 1 (or 0/1)

vars quickly and compute eff'ly?

5-wise indep. rvars and error correcting codes

Thm: Suppose $d = 2^k - 1$ and $5 = 2t+1$ (so $t = 2$).

Then there exists a symmetric prob. space Ω of size $2(d+1)^2$ and 5-wise indep. rvars $y_1, \ldots, y_d$ over Ω each of which takes the vals 0,1 with prob. $1/2$.

The space and vars. are explicitly constructed, given α a primitive elt. of field $F = GF(2^k)$ a k-dim'l algebra over $GF(2)$.

proof: [Everything instantiated for 5, nothing special about 5!].

(MacWilliams and Sloane)

Let $\alpha, \alpha^2, \ldots, \alpha^d$ be the d non-zero elts of $GF(2^k)$ rep'ed as col vectors of length k over $GF(2)$.

Let H be the parity check matrix of the extended BCH code of length d and distance $2t+2 = 6$.

$$\underbrace{1 + 2\cdot k}_{\text{rows}} \quad \overbrace{\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha & \alpha^2 & \cdots & \alpha^d \\ \alpha^3 & \alpha^6 & \cdots & \alpha^{3d} \end{pmatrix}}^{d}$$

H is a matrix over GF(2).

Lemma: Any set of $5 = 2\cdot 2+1$ cols of H is linearly indep. over $GF(2)$.

proof: see MacWilliams & Sloane.

have a homog. system of $2t+1$ eq^{n}s in $2t+1$ unknowns

and matrix of coeffs. is a vandermonde matrix

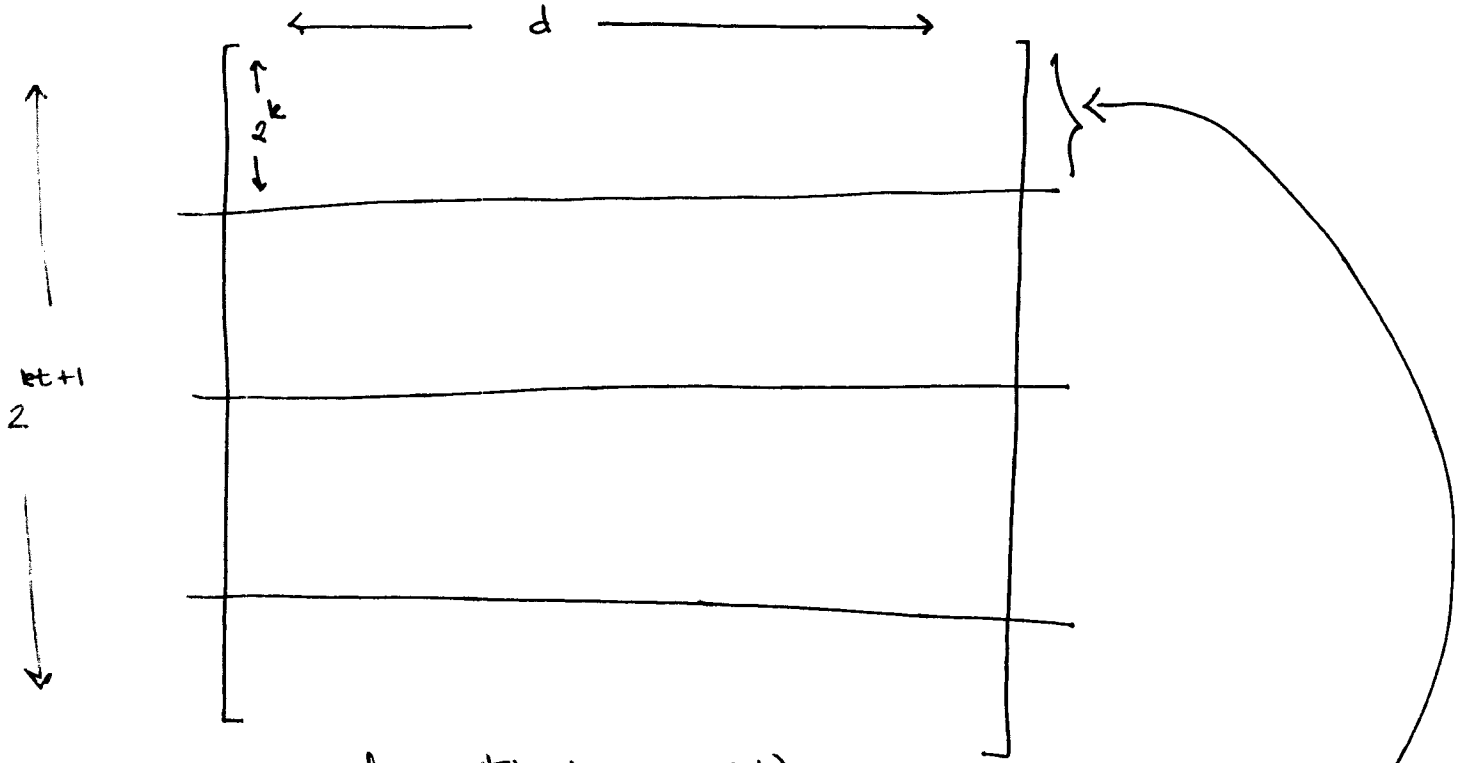
(which is nonsingular).

To generate rand vars., define $\Omega = \{1,2,\ldots, 2(d+1)^2\}$

and let $A = (a_{ij})$ $i \in \Omega$. $j = 1,\ldots,d$ be a binary

matrix with $2(d+1)^2 = 2^{kt+1}$ rows and d cols.

rows are all linear comb^{n}s of the rows of H (over $GF(2)$).



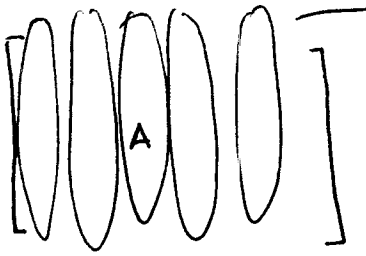
observe that for $b_i \in GF(2^k)$ $i=0,1,2$ we get one block of A

$$(b_0\ b_1\ b_2)\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha & \alpha^2 & \cdots & \alpha^d \\ \alpha^3 & \alpha^6 & \cdots & \alpha^{3d} \end{pmatrix} = \left(\sum_{i=0}^{2} b_i\alpha^i, \ldots, \sum_{i=0}^{2} b_i\alpha^{di}\right)$$

- We endow the sample space Ω with the unif. prob. measure
  Then y_j is defined by

$$y_j(i) = a_{ij} \quad \text{for} \quad i \in \Omega \text{ and } j = 1, \ldots, d.$$

- We need to show that y_j are 5-wise indep. and that each takes on 0,1 with equal prob.



Need to show that for every set J of up to 5 cols of A, the rows of submatrix

$$A_J = (a_{ij}) \quad i \in \Omega, \; j \in J$$

take on each of the 2^5 binary vectors equally often.

Lemma tells us that cols of submatrix H_J are linearly indep.

$\Rightarrow$ #(rows of A_J = any given vector of length 2^5)

= #(linear comb^{n}s of rows of H_J = vector)

= #(sol^{n}s of system of 5 lin. indep. eq^{n}s in $kt+1$ vars)

$$= 2^{kt+1-5}$$

//

- To generate $y_j(i)$ quickly upon seeing j, pick i uniformly from Ω $\Longleftrightarrow$ pick setting of bit vectors $(b_0\ b_1\ b_2)$ unifly at random. There are $2 \cdot k+1$ random bits here.

$$= 2 \cdot \log d + 1$$

Then compute

$$\sum_{i=0}^{2} b_i \alpha^{j \cdot i} \quad \in GF(2^k)$$

each of these are vectors of length k

and α is stored ahead of time.

$\Rightarrow$ Not storing A AND using a few random bits.