Administrivia

- Webpage: `http://theory.lcs.mit.edu/~madhu/FT02`.
- Send email to `madhu@mit.edu` to be added to course mailing list. Critical!
- Sign up for scribing.
- Pset 1 out today. Due in a week.
- Madhu's office hours for now: Next Tuesday 11am-12pm.
- Course under development! Limited staffing. Patience and constructive criticism appreciated.

Hamming's Problem (1940s)

- Magnetic storage devices are prone to making errors.
- How to store information so that 1 bit flip can be corrected?
- Simple solution:
  - Repeat every bit three times.
  - Works. To correct 1 bit flip error, take majority vote for each bit.
  - But efficiency of storage $1/3$. Do better?

Hamming's Solution - 1

- Break bits into blocks of size 4.
- Represent each block of 4 bits by a 7 bit string, so that any 1 bit flip can be corrected.
- How?

$[7,4,3]$-Hamming code

- Will explain notation later.
- Let

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- Encode $\mathbf{b} = \langle b_0 b_1 b_2 b_3 \rangle$ as $\mathbf{b} \cdot G$.
- Claim: If $\mathbf{a} \neq \mathbf{b}$, then $\mathbf{a} \cdot G$ and $\mathbf{b} \cdot G$ differ in at least 3 coordinates.
- Will defer proof of claim.

Hamming's Notions

- Since codewords (i.e., $\mathbf{b} \cdot G$) differ in at least 3 coordinates, can correct one error.
- Motivates Hamming distance, Hamming weight, Error-correcting codes etc.
- Alphabet Σ of size q. Ambient space, Σ^n: Includes codewords and their corruptions.
- Hamming distance between strings $\mathbf{x}, \mathbf{y} \in \Sigma^n$, denoted $\Delta(\mathbf{x}, \mathbf{y})$, is # of coordinates i s.t. $x_i \neq y_i$. (Converts ambient space into metric space.)
- Hamming weight of $\mathbf{z}$, denoted $\mathrm{wt}(\mathbf{z})$, is # coordinate where $\mathbf{z}$ is non-zero.

Hamming notions (contd.)

Code: Subset $C \subseteq \Sigma^n$.

Min. distance: Denoted $\Delta(C)$, is $\min_{\mathbf{x} \neq \mathbf{y} \in C}\{\Delta(\mathbf{x}, \mathbf{y})\}$.

e **error detecting code** If up to e errors happen, then codeword does not mutate into any other code.

t **error-correcting code** If up to t errors happen, then codeword is uniquely determined (as the unique word within distance t from the received word).

Proposition: C has min. dist. $2t+1 \Leftrightarrow$ it is $2t$ error-detecting $\Leftrightarrow$ it is t error-correcting.

Standard notation/terminology

- q: Alphabet size
- n: Block length
- k: Message length, where $|C| = q^k$.
- d: Min. distance of code.
- Code with above is an $(n, k, d)_q$ code. $[n, k, d]_q$ code if linear. Omit q if $q = 2$.
- k/n: Rate
- d/n: Relative distance.

Back to Hamming code

- So we have an $[7, 4, 3]$ code (modulo proof of claim).
- Can correct 1 bit error.
- Storage efficiency (rate) $4/7$!
- Will do better, by looking at proof of claim.

Proof of Claim

$$\text{Let } \mathrm{H} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

- Sub-Claim 1: $\{\mathbf{x}G|\mathbf{x}\} = \{\mathbf{y}|\mathbf{y} \cdot H = 0\}$. Simple linear algebra (mod 2). You'll prove this as part of Pset 1.
- Sub-claim 2: Exist codewords $\mathbf{z}_1 \neq \mathbf{z}_1$ s.t. $\Delta(\mathbf{z}_1, \mathbf{z}_2) \leq 2$ iff exists $\mathbf{y}$ of weight at most 2 s.t. $\mathbf{y} \cdot H = 0$.

- Let $\mathbf{h}_i$ be ith row of H. Then $\mathbf{y} \cdot H = \sum_{i|y_i=1} \mathbf{h}_i$.
- Let $\mathbf{y}$ have weight 2 and say $y_i = y_j = 1$. Then $\mathbf{y} \cdot H = \mathbf{h}_i + \mathbf{h}_j$. But this is non-zero since $\mathbf{h}_i \neq \mathbf{h}_j$. QED.

Generalizing Hamming codes

- Important feature: Parity check matrix should not have identical rows. But then can do this for every ℓ.

$$H_\ell = \begin{bmatrix} 0 & \cdots & 0 & 0 & 1 \\ 0 & \cdots & 0 & 1 & 0 \\ 0 & \cdots & 0 & 1 & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & \cdots & 1 & 1 & 1 \end{bmatrix}$$

- H_ℓ has ℓ columns, and $2^{\ell-1}$ rows.
- H_ℓ : Parity check matrix of ℓth Hamming code.
- Message length of code = exercise. Implies rate $\to 1$.

Applications of error-correcting codes

- Obvious: Communication/Storage.
- Algorithms: Useful data structures.
- Complexity: Pseudorandomness (ϵ-biased spaces, pairwise independent spaces), Hardness amplification, PCPs.
- Cryptography: Secret sharing, Crypto-schemes.
- Central object in extremal combinatorics: relates to extractors, expanders, etc.
- Recreational Math.

Example: Pairwise independent spaces

Motivation:

- Randomized algorithms often work with n random bits.
- But may not need all n bits to be independent.
- Sometime suffices for bits to be pairwise independent.

Formally ...

- $S \subseteq \{0,1\}^n$ is a pairwise independent space if for every i, j distinct in $[n]$, and bits $b_i, b_j \in \{0,1\}$, we have

$$\Pr_{\mathbf{y} \in S}[y_i = b_i \text{ and } y_j = b_j] = \frac{1}{4}.$$

- Informally, when the n bit random vector $\mathbf{y} = \langle y_1, \ldots, y_n \rangle$ is chosen, not uniformly from $\{0,1\}^n$, but uniformly from a small subset $S \subseteq \{0,1\}^n$, any pair of coordinates are still independent.
- Can extend definition to t-wise independence.

Pairwise independence (contd.)

- Fact: There exist pairwise independent spaces S with $|S| = O(n)$.
- You'll prove this as an exercise. Follows almost immediately from properties we've proved about the Hamming code.
- Conclusion: If we have a randomized algorithm that uses n pairwise independent random bits and solves a given problem (with one-sided error), then by running this algorithm on all n vectors of S, can get a *deterministic* algorithm whose running time is only larger by a factor of $O(n)$.

Example: The Hat Problem

Scenario: n people in a room with Black/White hats on their head. Hat colors chosen at random independently. Everybody sees the color of the hat on every one else's head, but not their own. People don't communicate with each other.

Game: Every one gets to guess (by writing on a piece of paper) the color of their hat. They may write Black/White/Abstain.

Win/Loss: The people in the room win together or lose together. Win if at least one person did not abstain, and everyone that did not abstain guessed the color of their hat correctly.

Question: What is the probability of winning?

Deeper question: What does this have to do with Hamming codes? See Pset 1.

Summary of Hamming's paper (1950)

- Defined Hamming metric and codes.
- Gave codes with $d = 1, 2, 3, 4$!
- $d = 2$: Parity check code.
- $d = 3$: We've seen.
- $d = 4$? Take an $[n, k, 3]_2$ code and add a parity check bit to get an $[n + 1, k, 4]_2$ code!
- Gave a tightness result: His codes have maximum number of codewords. Proof in Pset 1! "Lower bound".
- Gave decoding "procedure".

Decoding the Hamming code

- Can recognize codewords? Yes - multiply by H_ℓ and see if 0.
- What happens if we send codeword $\mathbf{c}$ and ith bit gets flipped?
- Received vector $\mathbf{r} = \mathbf{c} + \mathbf{e}_i$.
- $\mathbf{r} \cdot H = \mathbf{c} \cdot H + \mathbf{e}_i \cdot H$
  $= 0 + \mathbf{h}_i$
  $=$ binary representation of i.
- $\mathbf{r} \cdot H$ gives binary rep'n of error coordinate!

Rest of the course

- More history!
- More codes (larger d).
- More lower bounds (will see other methods).
- More algorithms - decode less simple codes.
- More applications: Modern connections to theoretical CS.