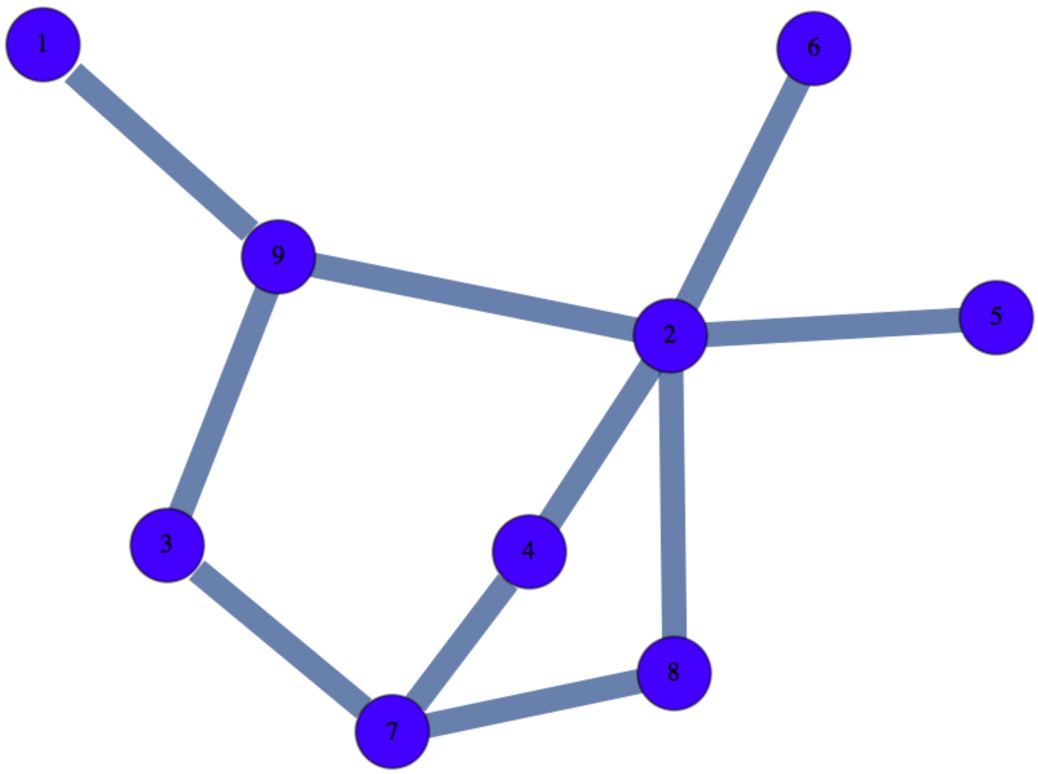
Lecture 28: Vector spaces over finite fields

Thursday, December 3, 2015 9:30 AM

LET'S PLAY A GAME!

Tell me which vertices to click.

Goal: Color the graph red.



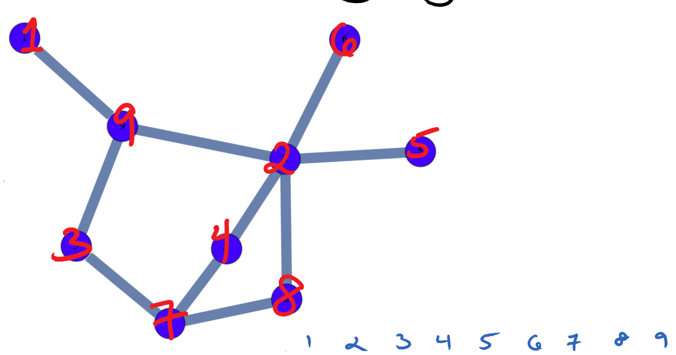
Formalizing the game:

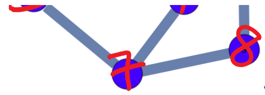
Observe:

1. Clicking a vertex twice cancels it out.
2. The order in which you click the vertices doesn't matter.

$\Rightarrow$ All that matters is the subset of vertices you click.

$\Rightarrow$ **Problem**: Is there a subset of vertices to click that will make everything red?





$\Rightarrow$ Solve

$$\begin{array}{c|ccccccccc|}
 & 1&2&3&4&5&6&7&8&9 \\ \hline
1 & 1&0&0&0&0&0&0&0&1 \\
2 & 0&1&0&1&1&1&0&1&1 \\
3 & 0&0&1&0&0&0&1&0&1 \\
4 & 0&1&0&1&0&0&1&0&0 \\
5 & 0&1&0&0&1&0&0&0&0 \\
6 & 0&1&0&0&0&1&0&0&0 \\
7 & 0&0&1&1&0&0&1&1&0 \\
8 & 0&1&0&0&0&0&1&1&0 \\
9 & 1&1&1&0&0&0&0&0&1
\end{array} \vec{x} = \begin{pmatrix}1\\1\\1\\1\\1\\1\\1\\1\\1\end{pmatrix} \bmod 2$$

for $x_i \in \{0,1\}$

How? Use Gaussian elimination (mod 2)!

$$\begin{array}{c|ccccccccc|c|}
 & 1&2&3&4&5&6&7&8&9 & \\ \hline
1 & 1&0&0&0&0&0&0&0&1 & 1 \\
2 & 0&1&0&1&1&1&0&1&1 & 1 \\
3 & 0&0&1&0&0&0&1&0&1 & 1 \\
4 & 0&1&0&1&0&0&1&0&0 & 1 \\
5 & 0&1&0&0&1&0&0&0&0 & 1 \\
6 & 0&1&0&0&0&1&0&0&0 & 1 \\
7 & 0&0&1&1&0&0&1&1&0 & 1 \\
8 & 0&1&0&0&0&0&1&1&0 & 1 \\
9 & 1&1&1&0&0&0&0&0&1 & 1
\end{array} \xrightarrow{\text{add row 1 to row 9}}
\begin{array}{c|ccccccccc|c|}
 & 1&2&3&4&5&6&7&8&9 & \\ \hline
1 & 1&0&0&0&0&0&0&0&1 & 1 \\
2 & 0&1&0&1&1&1&0&1&1 & 1 \\
3 & 0&0&1&0&0&0&1&0&1 & 1 \\
4 & 0&1&0&1&0&0&1&0&0 & 1 \\
5 & 0&1&0&0&1&0&0&0&0 & 1 \\
6 & 0&1&0&0&0&1&0&0&0 & 1 \\
7 & 0&0&1&1&0&0&1&1&0 & 1 \\
8 & 0&1&0&0&0&0&1&1&0 & 1 \\
9 & 0&1&1&0&0&0&0&0&0 & 0
\end{array}$$

add row 9 to rows 2, 4, 5, 6, 8

$$\begin{array}{c|ccccccccc|c|}
 & 1&2&3&4&5&6&7&8&9 & \\ \hline
1 & 1&0&0&0&0&0&0&0&1 & 1 \\
2 & 0&0&1&1&1&1&0&1&1 & 1 \\
3 & 0&0&1&0&0&0&1&0&1 & 1 \\
4 & 0&0&1&1&0&0&1&0&0 & 1 \\
5 & 0&0&1&0&1&0&0&0&0 & 1 \\
6 & 0&0&1&0&0&1&0&0&0 & 1 \\
7 & 0&0&1&1&0&0&1&1&0 & 1 \\
8 & 0&0&1&0&0&0&1&1&0 & 1 \\
9 & 0&1&1&0&0&0&0&0&0 & 0
\end{array} \xrightarrow{\text{add row 5 to rows 2,3,4,6,7,8,9}}
\begin{array}{c|ccccccccc|c|}
 & 1&2&3&4&5&6&7&8&9 & \\ \hline
1 & 1&0&0&0&0&0&0&0&1 & 1 \\
2 & 0&0&0&1&0&1&0&1&1 & 0 \\
3 & 0&0&0&0&1&0&1&0&1 & 0 \\
4 & 0&0&0&1&1&0&1&0&0 & 0 \\
5 & 0&0&1&0&1&0&0&0&0 & 1 \\
6 & 0&0&0&0&1&1&0&0&0 & 0 \\
7 & 0&0&0&1&1&0&1&1&0 & 0 \\
8 & 0&0&0&0&1&0&1&1&0 & 0 \\
9 & 0&1&0&0&1&0&0&0&0 & 1
\end{array}$$

add row 4 to rows 2 & 7

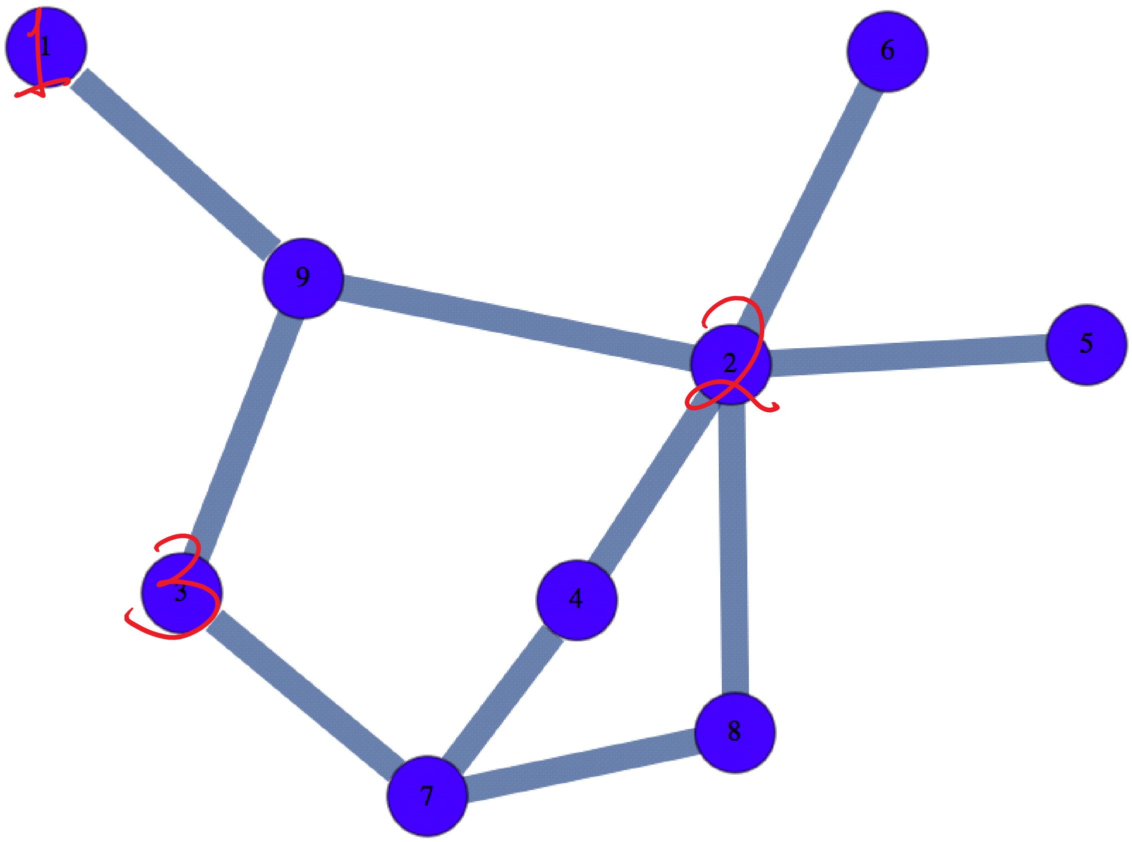
$$\begin{array}{c|ccccccccc|c|}
 & 1&2&3&4&5&6&7&8&9 & \\ \hline
1 & 1&0&0&0&0&0&0&0&1 & 1 \\
2 & 0&0&0&0&1&1&1&1&1 & 0 \\
3 & 0&0&0&0&1&0&1&0&1 & 0 \\
4 & 0&0&0&1&1&0&1&0&0 & 0 \\
5 & 0&0&1&0&1&0&0&0&0 & 1 \\
6 & 0&0&0&0&1&1&0&0&0 & 0 \\
7 & 0&0&0&0&0&0&0&1&0 & 0 \\
8 & 0&0&0&0&1&0&1&1&0 & 0 \\
9 & 0&1&0&0&1&0&0&0&0 & 1
\end{array} \xrightarrow[\substack{\text{add row 6 to rows}\\ 2,3,4,5,8,9}]{\substack{\text{row 7} \Rightarrow x_8 = 0\\ \text{add row 7 to rows 2, 8}}}
\begin{array}{c|ccccccccc|c|}
 & 1&2&3&4&5&6&7&8&9 & \\ \hline
1 & 1&0&0&0&0&0&0&0&1 & 1 \\
2 & 0&0&0&0&0&0&1&0&1 & 0 \\
3 & 0&0&0&0&0&1&1&0&1 & 0 \\
4 & 0&0&0&1&0&1&1&0&0 & 0 \\
5 & 0&0&1&0&0&1&0&0&0 & 1 \\
6 & 0&0&0&0&1&1&0&0&0 & 0 \\
7 & 0&0&0&0&0&0&0&1&0 & 0 \\
8 & 0&0&0&0&0&1&1&0&0 & 0 \\
9 & 0&1&0&0&0&1&0&0&0 & 1
\end{array}$$

add row 2 to row 3 $\Rightarrow x_6 = 0$

add row 3 to rows 4, 5, 6, 8, 9

$$\begin{array}{c|ccccccccc|c|}
 & 1&2&3&4&5&6&7&8&9 & \\ \hline
1 & 1&0&0&0&0&0&0&0&1 & 1 \\
2 & 0&0&0&0&0&0&1&0&1 & 0 \\
3 & 0&0&0&0&0&1&0&0&0 & 0 \\
4 & 0&0&0&1&0&0&1&0&0 & 0 \\
5 & 0&0&1&0&0&0&0&0&0 & 1 \\
6 & 0&0&0&0&1&0&0&0&0 & 0 \\
7 & 0&0&0&0&0&0&0&1&0 & 0 \\
8 & 0&0&0&0&0&0&1&0&0 & 0 \\
9 & 0&1&0&0&0&0&0&0&0 & 1
\end{array} \xrightarrow[\text{then row 2 to row 1}]{\text{add row 8 to rows 2, 4}}
\begin{array}{c|ccccccccc|c|}
 & 1&2&3&4&5&6&7&8&9 & \\ \hline
1 & 1&0&0&0&0&0&0&0&0 & 1 \\
2 & 0&0&0&0&0&0&0&0&1 & 0 \\
3 & 0&0&0&0&0&1&0&0&0 & 0 \\
4 & 0&0&0&1&0&0&0&0&0 & 0 \\
5 & 0&0&1&0&0&0&0&0&0 & 1 \\
6 & 0&0&0&0&1&0&0&0&0 & 0 \\
7 & 0&0&0&0&0&0&0&1&0 & 0 \\
8 & 0&0&0&0&0&0&1&0&0 & 0 \\
9 & 0&1&0&0&0&0&0&0&0 & 1
\end{array}$$

Done! $\vec{x} = (\overset{1}{1}, \overset{2}{1}, \overset{3}{1}, \overset{4}{0}, \overset{5}{0}, \overset{6}{0}, \overset{7}{0}, \overset{8}{0}, \overset{9}{0})$ works!



Conclusion: As well as real and complex numbers, $\mathbb{R}$ and $\mathbb{C}$, linear algebra works over the integers mod 2.

Concept:	Works over finite fields?
Vector space	✓
Subspace	✓
Linear independence, bases, dimension	✓
Gaussian elimination	✓
LU decomposition	✓
Orthogonality	✓
Rank-Nullity Theorem	✓
Angles	✗
Norms	✗
Isometries, unitaries, ...	✗
Singular-value decomposition	✗

Isometries, unitaries, ... $\times$
Singular-value decomposition $\times$
Spectral decomposition $\times$

In general, a **field** supports addition and subtraction, multiplication and division (except by 0)

multiplication's inverse operation

Examples:

$\mathbb{R}, \mathbb{C}, \{0,1\}, \{0,1,2\}$

how to divide by 2?
it should be the inverse of multiplying by 2

Observe: $x \cdot 2 \cdot 2 = x \cdot 4$
$= x \mod 3$
$\Rightarrow 2^{-1} = 2$

$\{0,1,2,3\}$ No! For no x is $2x = 1 \mod 4$
$\Rightarrow 2^{-1}$ does not exist mod 4

(Alternatively, $1 \cdot 2 = 2 \mod 4$
$3 \cdot 2 = 2 \mod 4$
so multiplication by 2 cannot be undone/inverted.)

Fact:
- For any prime p, the integers mod p form a field.
- For any prime p and $k = 1, 2, 3, \ldots$, there is a field of size p^k.
  (But for $k \geq 2$, the field is **not** the integers mod p^k.)
- These are the only finite fields.

For today: Let's focus on the field $\{0,1\}$.

SUBSPACES OF $\{0,1\}^n$

Examples:

① $\{(0,0,\dots,0), (1,1,\dots,1)\}$

closed under addition? ✓

closed under multiplication by 0 and 1? ✓

② $\text{Span}\left\{\begin{array}{l}(0,0,0,1,1,1,1),\\ (0,1,1,0,0,1,1),\\ (1,0,1,0,1,0,1),\\ (1,1,1,1,1,1,1)\end{array}\right\}$

<u>Claim</u>: This is a four-dimensional subspace, i.e., the four vectors in the spanning set are linearly independent.

Proof:

0 0 0 1 1 1 1
0 1 1 0 0 1 1
1 0 1 0 1 0 1

are linearly indep. because each has a 1 in a position that the other two can't cancel out

The last vector, 1 1 1 1 1 1 1, is linearly indep. of the first three, because it has a 1 in all 3 of the positions marked above, so it could only equal the sum of the first 3, and it isn't.

<u>Alternatively</u>, it has odd weight, while any combination of the first 3 vectors has an even number of 1s.

<u>Alternatively</u>, we want to show that this matrix has a trivial nullspace: (so there are no linear dependencies between its columns, and its rank is 4)

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

Solving $G\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = 0$

$\Rightarrow x_3 + x_4 = 0 \quad \Rightarrow x_3 = x_4$

$x_2 + x_4 = 0 \quad \Rightarrow x_2 = x_4$

$x_1 + x_4 = 0 \quad \Rightarrow x_1 = x_4$

and $\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ doesn't work. ✓

<u>Problem</u>: How big is a k-dimensional subspace of $\{0,1\}^n$?

Answer:

Problem: How big is a k-dimensional subspace of $\{0,1\}^n$?

Answer:

The answer "has to" be 2^k, since $|\{0,1\}^k| = 2^k$.

To prove it, let $v_1, v_2, \dots, v_k$ be a basis for the subspace, and start listing elements:

$$\begin{array}{cccc} 0 = 0^n & v_2 & v_3 & v_2 + v_3 \\ v_1 & v_1 + v_2 & v_1 + v_3 & v_1 + v_2 + v_3 \end{array}$$

component-wise addition mod 2

these are all different!

again, all different, or there would be a linear dependency

... ✓

Problem: How many k-dimensional subspaces of $\{0,1\}^n$ are there?

Answer:

$$\#\,k\text{-dim subspaces of } \{0,1\}^n = \frac{\#\text{ ways to pick } k \text{ linearly independent vectors from } \{0,1\}^n}{\#\text{ ways to pick } k \text{ linearly independent vectors from } \{0,1\}^k}$$

$$\#\text{ ways to pick } k \text{ basis vectors from } \{0,1\}^n = \underbrace{(2^n - 1)}_{v_1 \neq 0^n} \times \underbrace{(2^n - 2)}_{v_2 \notin \{0, v_1\}} \times \underbrace{(2^n - 4)}_{v_3 \notin \operatorname{Span}\{v_1, v_2\}} \times \cdots \times (2^n - 2^{k-1})$$

$$\Rightarrow \#\,k\text{-dim subspaces of } \{0,1\}^n = \boxed{\frac{\prod_{j=0}^{k-1} (2^n - 2^j)}{\prod_{j=0}^{k-1} (2^k - 2^j)}} \quad ✓$$

Linear error correcting codes

Definition: The Hamming distance between x and y in $\{0,1\}^n$ is the number of coordinates in which they differ.

The Hamming weight of $x \in \{0,1\}^n$ is
$|x|$ = the number of nonzero coordinates

Thus

$$d_H(x,y) = |x+y|$$

since $x+y = (x_1+y_1, x_2+y_2, \ldots, x_n+y_n)$

$$x_1+y_1 = \begin{cases} 0 & \text{if } x_1 = y_1 \\ 1 & \text{if } x_1 \neq y_1 \end{cases} \qquad \ldots$$

Definition: A "binary, error-correcting code" is a subset of $\{0,1\}^n$.

The code's distance is the minimum Hamming distance between any two distinct elements.

Example: $C = \{(0,0,\ldots,0), (1,1,\ldots,1)\} \subset \{0,1\}^n$ has distance n.

Definition: A binary, error-correcting, linear code is a subspace of $\{0,1\}^n$.

Notation: $[n, k, d]$ code

- n: n bits
- k: dim-k subspace → 2^k codewords → k encoded bits
- d: distance d

Example: 7-bit Hamming code

To encode 4 bits into 7, multiply by

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

generator matrix

columns $= v_1, v_2, v_3, v_4$

$$G\begin{pmatrix} x_1 \\ \vdots \\ x_4 \end{pmatrix} = x_1\vec{v}_1 + x_2\vec{v}_2 + x_3\vec{v}_3 + x_4\vec{v}_4 \in \operatorname{Span}\{\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{v}_4\}$$

Claim: For a linear code,

the code's distance $= \min_{x \in \text{code}} |x|$

Proof: code distance $= \min_{x,y \in \text{code}} d_H(x,y)$
$$d_H(x,y) = |x+y|$$

But the code is linear, so $x+y$ is in the code, too ✓ $\square$

Code duality

For any subspace V,
$$V^\perp = \{ \vec{w} \mid \vec{w} \perp \vec{v} \text{ for all } \vec{v} \in V \}$$
is another subspace.

(and $(V^\perp)^\perp = V$)

For a code C with generator matrix $G_{n \times k}$

ie., $C = R(G)$, $N(G) = \{0\}$

so columns are linearly independent

$C^\perp = N(G^T)$ has dimension $n-k$ (by rank-nullity).

Example: n-bit repetition code

$$G = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}_{n \times 1}$$

$N(G^T)$ has dimension $n-1$,

generated by

$$P = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots \\ 1 & 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

$P =$ parity-check matrix

rows of $P \perp$ columns of G

$(PG = 0)$

Example: 7-bit Hamming code

$G =$ generator matrix
$$G = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

$P =$ parity-check matrix
$$P = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}_{3 \times 7}$$

Claim: This code has distance 3 (it is a $[7,1,3]$ code)

Proof:

① The distance is ≥ 3:

Why?

Let $x \in R(G)$ be a codeword.

$P\vec{x} = 0$

$\Rightarrow P(\vec{x}+\vec{y}) = P\vec{y}$ for any error $\vec{y} \in \{0,1\}^7$

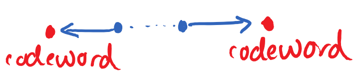
Observe: If $|\vec{y}| \leq 1$, then it is uniquely identified by $P\vec{y}$.

Since $P\vec{0} = \vec{0}$

and $P\vec{e}_i = i$ in binary!

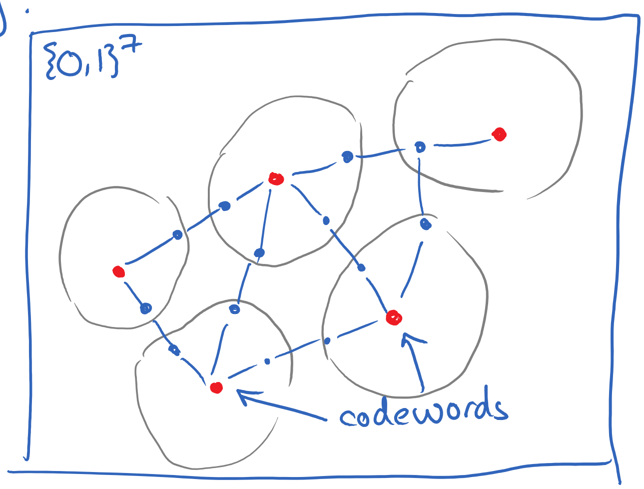
$\Rightarrow$ Given $\vec{z}$, promised to be within distance 1 of some codeword, you can recover the codeword by flipping the $(P\vec{z})$'th bit.

$\Rightarrow$ Any pair of codewords must be ≥ 3 apart ✓



② The distance is ≤ 3:

Why?



If the distance is ≥ 3, then the balls of Hamming radius 1 around the codewords must all be disjoint.

This accounts for $\underbrace{2^k}_{\#\text{ codewords}} \cdot \underbrace{(1 + \binom{n}{1})}_{\text{size of ball}} = 16 \cdot 8 = 2^7$

points in $\{0,1\}^7$ $\Rightarrow$ no room for any more! ✓ □

Remark: This argument holds for any linear code, and is known as the "Hamming bound."

The [7,1,3] Hamming code is called "perfect" because the Hamming balls fill $\{0,1\}^7$ completely, with no gaps between them.

Subspaces
- counting
- codes
    - distance
    - Hamming bound
    - rank-nullity

use repetition & Hamming codes (presented first as subspaces) as running examples