TUTORIAL IX

1 Error correcting VS error detecting

Show that the following statements are equivalent for a code linear code C:

1. C has minimum distance $\geq d \geq 2$.
2. If d is odd, C can correct $(d-1)/2$ errors.
3. If d is even, C can correct $d/2-1$ errors.
4. C can detect $d-1$ errors.
5. C can correct $d-1$ erasures (in the erasure model, the receiver know where the errors have occurred).

A: *We will show that* $1 \Leftrightarrow 2$, $1 \Leftrightarrow 3$ *and* $1 \Leftrightarrow 4$.

$1 \Leftrightarrow 2$**:** *Suppose* $d = 3$. *Define the sphere of radius* r *around a codeword* $\mathbf{x}$ *as the set of codewords* $\mathbf{y}$ *such that* $\Delta(\mathbf{x},\mathbf{y}) \leq r$. *Spheres of radius 1 drawn around all codewords do not overlap. If* $\mathbf{x}$ *is transmitted with one error and a codeword* $\mathbf{u}$ *is received, then* $\mathbf{u}$ *is in the sphere around* $\mathbf{x}$ *can be easily corrected by choosing the closest codeword. Suppose* $d = 2t+1$ *(odd), then we can draw non-overlapping spheres of radius* t *around the codewords. If* $\mathbf{x}$ *is transmitted with up to* t *errors and* $\mathbf{u}$ *is received, then* $\mathbf{u}$ *is in the sphere around* $\mathbf{x}$ *and the nearest codeword decoding rule can be applied to recover* $\mathbf{x}$. *So, we have* $1 \Rightarrow 2$.

For the other direction, suppose that 1 *doesn't hold, ie* C *has minimum distance* $d' < d$. *Take* $\mathbf{x},\mathbf{y}$ *two codewords of* C *that achieve* d', *and consider a vector* $\mathbf{u}$ *in the middle of* $\mathbf{x},\mathbf{y}$, *ie satisfying* $\Delta(\mathbf{x},\mathbf{u}), \Delta(\mathbf{y},\mathbf{u}) \leq \frac{d'+1}{2}$. *Since* $d' < d$, *we now have two cases. If* $d' <= d-2$, *we have* $\frac{d'+1}{2} \leq \frac{d-1}{2}$ *and so* $\mathbf{u}$ *is reachable from* $\mathbf{x}$ *and* $\mathbf{y}$ *with less than* $\frac{d-1}{2}$ *errors. If* $d' = d-1$, *since* d *is odd,* d' *is even and thus the distance inequalities can be tightened :* $\Delta(\mathbf{x},\mathbf{u}), \Delta(\mathbf{y},\mathbf{u}) \leq \frac{d'}{2} \leq \frac{d-1}{2}$, *and we have the same conclusion. Overall, we have a vector that is reachable from 2 codewords in at most* $\frac{d-1}{2}$ *errors, thus this vector can be corrected, thus* 2 *is false.*

$1 \Leftrightarrow 3$**:** *We are given a codeword* $\mathbf{u}$, *and we want to detect if there has been an error. We simply check whether* $\mathbf{u} \in C$ *or not. Indeed, if no error occured, we have* $\mathbf{u} \in C$. *And if at least one error occured, since the distance is at least* d, *we now that if at most* $d-1$ *errors occured, we must have* $\mathbf{u} \notin C$.

Again, assume that the distance of C *is* $d' < d$, *and take* $\mathbf{x},\mathbf{y}$ *two codewords achieving this distance. In first scenario, suppose we send* $\mathbf{x}$ *and get* $\mathbf{y}$, *with* $d' \leq d-1$ *errors. In second scenario, suppose we send* $\mathbf{y}$ *and get* $\mathbf{y}$ *with no error. Any decoder should answer deterministically for the received message* $\mathbf{y}$, *so it must fail on at least one the two above scenari.*

$1 \Leftrightarrow 4$**:** *Let* $\mathbf{u} \in (\Sigma \cup ?)^n$ *be the received codeword. First we claim that there is a unique* $c \in C$ *that agrees with* $\mathbf{u}$ *(ie* $\mathbf{u}_i = c_i$ *for indices* i *st* $\mathbf{u}_i \neq ?$*). Indeed, if both* c_1, c_2 *agrees with* $\mathbf{u}$, *we would have* $\Delta(c_1,c_2) \leq |\{i : \mathbf{u}_i = ?\}| \leq d-1$, *thus the distance would be* $< d$. *The algorithm is quite simple: go through codewords and pick the only one agreeing with* $\mathbf{u}$.

Again, assume that the distance of C *is* $d' < d$, *and take* $\mathbf{x},\mathbf{y}$ *two codewords achieving this distance. Define the vector* $\mathbf{u}$ *as* $\mathbf{u}_i = \mathbf{x}_i$ *if* $\mathbf{x}_i = \mathbf{y}_i$, *and* $\mathbf{u}_i = ?$ *otherwise. Starting from codeword* $\mathbf{x}$, *we can reach vector* $\mathbf{u}$ *in* $d' \leq d-1$ *erasures, since* $\Delta(\mathbf{x},\mathbf{y}) = d'$. *Idem from codeword* $\mathbf{y}$. *Thus the vector* $\mathbf{u}$ *cannot be correctly corrected.*

2 Generalized Hamming bound

Prove the following bound: for any (n,k,d) code $C \subseteq (\mathbb{F}_q)^n$,

$$k \leq n - \log_q \left(\sum_{i=0}^{\left\lfloor \frac{(d-1)}{2} \right\rfloor} \binom{n}{i} (q-1)^i \right)$$

A: *Since minimum distance is d, the code allows correction of $t = \lfloor \frac{d-1}{2} \rfloor$ errors and hence spheres of radius t drawn around codewords do not overlap. Consider a codeword $\mathbf{x}$. For $0 \leq i \leq t$ the sphere around $\mathbf{x}$ contains exactly $\binom{n}{i}(q-1)^i$ vectors of distance i from $\mathbf{x}$. The reason is that $\binom{n}{i}$ is the number of choices for the i positions in which a vector $\mathbf{u}$ can differ from $\mathbf{x}$ and for each position j, there are $q-1$ possible values that u_j can take.*

The code has dimension k which means there are a total of q^k codewords or spheres. Their union is no doubt a subset of ${\mathbb{F}_q}^n$ containing q^n elements since the spheres are disjoint. We have

$$q^k \left(\sum_{i=0}^{t} \binom{n}{i} (q-1)^i \right) \leq q^n$$

Taking logarithm on both sides to base q, we get the desired inequality.

3 Hamming riddle

There are n people in a room, each of whom is given a black/white hat chosen uniformly at random (and independent of the choices of all other people). Each person can see the hat colour of all other people, but not their own. Each person is asked if (s)he wishes to guess their own hat colour. They can either guess, or abstain. Each person makes their choice without knowledge of what the other people are doing. They either win collectively, or lose collectively. They win if all the people who don't abstain guess their hat colour correctly and at least one person does not abstain. They lose if all people abstain, or if some person guesses their colour incorrectly. The goal below is to come up with a strategy that will allow the n people to win with pretty high probability

1. Argue that the n people can win with probability at least $\frac{1}{2}$

   A: *There is a strategy where one person guesses and the rest abstain. With probability $1/2$, they win.*

2. Lets say that a directed graph G is a subgraph of the n-dimensional hypercube if its vertex set is $\{0,1\}^n$ and if $u \to v$ is an edge in G, then u and v differ in at most one coordinate. Let $K(G)$ be the number of vertices of G with in-degree at least one, and out-degree zero. Show that the probability of winning the hat problem equals the maximum, over directed subgraphs G of the n-dimensional hypercube, of $K(G)/2^n$

   A: *First, given a graph $G = (V, E)$, we describe a strategy that achieves a probability $K(G)/2^n$. We will label the people in the room $1, \ldots, n$. A possible situation will be denoted by $p \in \{0,1\}^n$, and can be associated with the vertices of the n-dimensional hypercube. The person labeled i sees a projection of the situation, denoted by $p[i = ?]$. The strategy for person i is the following: abstain unless $p[i = 1] \in K(G)$ and $(p[i = 0], p[i = 1]) \in E$, in this case guess 1 (0 in the opposite/symmetric case).*
   Let's now show that if the situation $p \in K(G)$, all the people who didn't abstain will guess the correct color of his hat. If $p \in K(G)$, there must be $p' \in G$ such that $(p', p) \in E$. But, since G is a subgraph of the hypercube, $(p', p) \in E \Rightarrow \exists i_0$ such that p and p' differs only on their i_0^{th} coordinate. Thus, the person labeled i_0 would have guessed correctly his hat. The same reasoning applies to show that all the people who guessed have guessed correctly.
   Hence, all position in $K(G)$ are winning, thus the probability of winning for this strategy is at least $K(G)/2^n$. This proves the first part of the inequality.

   For the other direction of the inequality, we have to show that any strategy infers some graph G with $K(G)$ bigger than 2^n times the probability of winning of the strategy. If we denote by S the set of winning positions, we have that the probability of success is $|S|/2^n$. We thus want to construct a graph with $K(G) \geq |S|$.
   For any winning position p, and any person i making a correct guess a in this situation, we add the edge $p[i = \overline{a}] \to p$. For sure, the winning positions have in-degree at least one, since at least one person should guess correctly in order to win. And the winning positions can't have an out going edge, because if $p \to p[i = \overline{a}]$ was an edge, this means that when person labeled i sees projection $p[i = ?]$, he guesses $\overline{a}$. But that means that in the situation p, the person labeled i will guess $\overline{a}$, which will be wrong, and hence p is not a winning position. Overall, we have $K(G) \geq |S|$, establishing the second part of the inequality.

3. Using the fact that the out-degree of any vertex is at most n,show that $K(G)/2^n$ is at most $\frac{n}{n+1}$ for any directed subgraph G of the n-dimensional hypercube.

A: *We have $K(G) + (2^n - K(G)) = 2^n$. But any vertex not in $K(G)$ sees at most n vertices in $K(G)$, we must have: $(2^n - K(G)) \geq \frac{K(G)}{n}$. Thus, we have:*

$$\begin{aligned} 2^n &\geq K(G) + \frac{K(G)}{n} \\ 2^n &\geq \frac{n+1}{n} K(G) \\ \frac{K(G)}{2^n} &\leq \frac{n}{n+1} \end{aligned}$$

4. Show that if $n = 2^r - 1$, then there exists a directed subgraph G of the n-dimensional hypercube with $K(G)/2^n = \frac{n}{n+1}$.
Hint:This is where the Hamming code comes in.

A: *For any r, we know that there exists a Hamming code of parameters $[2^r - 1, 2^r - r - 1, 3]$. We take this Hamming code C, and define the following subgraph G of the n-dimensional hypercube: for any codeword $p \in C$, we add an edge from p to all the codewords at distance 1 of p. Then, any position $p \notin C$ corresponds to a vertex with out-degree 0, and with in-degree exactly 1, since balls of radius 1 cover the space of the Hamming code. Thus $K(G) = 2^n - |C| = 2^n - 2^{2^r - r - 1}$. In our case, $r = \log_2(n+1)$, thus:*

$$\begin{aligned} K(G) &= 2^n - 2^{(n+1) - \log_2(n+1) - 1} \\ &= 2^n - 2^{n+1} \frac{1}{n+1} \frac{1}{2} \\ &= 2^n \left(1 - \frac{1}{n+1}\right) \\ &= \frac{n}{n+1} 2^n \end{aligned}$$

(or one can simply conclude by saying that each vertex not in $K(G)$ sees exactly n vertices in $K(G)$, which is the case of equality in previous question)