THE MINIMUM HAMMING DISTANCE OF A LINEAR CODE

TUDOR MICU

We denote by $\mathbb{F}_2 = \{0, 1\}$ the field with two elements. The operations therein follow the same rules as the operations in the ring $\mathbb{Z}_2 = \{\bar{0}, \bar{1}\}$ of residue classes modulo 2.

We consider an (n, k)-linear code given by the generator matrix G and the parity matrix H. The encoding function is the injective linear map $\gamma : \mathbb{F}_2^k \to \mathbb{F}_2^n$ and $\mathcal{C} \leq_{\mathbb{F}_2} \mathbb{F}_2^n$ is the subspace of code words.

Definition. *The* ***minimum Hamming distance*** *of a code $\mathcal{C}$ is*

$$d(\mathcal{C}) = \min_{\substack{x,x' \in \mathcal{C} \\ x \neq x'}} card\{i \in \{1, \ldots, n\} |\ x_i \neq x'_i\} = \min_{\substack{x,x' \in \mathcal{C} \\ x \neq x'}} d_H(x, x')$$

It is the minimal number of positions where two code words disagree.

We aim to compute this minimum distance for our code. We will use the following notion:

Definition. *The* ***weight*** *of a word $x \in \mathbb{F}_2^n$ is*

$$w(x) = card\{i \in \{1, \ldots, n\} | x_i = 1\}$$

It represents the number of positions where the word x has a 1.

Lemma. *For every $x, x' \in \mathbb{F}_2^n$ we have $d_H(x, x') = d_H(x - x', 0) = w(x - x')$.*

Proof. Quite easy, so it's left to the reader. □

Lemma. $d(\mathcal{C}) = \min_{\substack{x \in \mathcal{C} \\ x \neq 0}} w(x)$

Proof. What we have to prove is that $\min_{\substack{x,x' \in \mathcal{C} \\ x \neq x'}} d_H(x, x') = \min_{\substack{x \in \mathcal{C} \\ x \neq 0}} w(x)$.

Let $a := \min_{\substack{x,x' \in \mathcal{C} \\ x \neq x'}} d_H(x, x')$ and $b := \min_{\substack{x \in \mathcal{C} \\ x \neq 0}} w(x)$. We will prove that $a \leq b$ and $b \leq a$.

Date: December 15, 2020.

Let $x, x' \in \mathbb{F}_2^n$ be distinct vectors for which we attain the minimum Hamming distance, that is, $d_H(x, x') = a$. Then by the lemma above we have $w(x - x') = a$. Because $b = \min\limits_{\substack{x \in \mathcal{C} \\ x \neq 0}} w(x)$, we get that $b \leq a$.

Let $y \in \mathbb{F}_2^n$ be a nonzero vector for which we attain the minimal weight, that is, $w(y) = b$. But then $d_H(y, 0) = b$. Because $a = \min\limits_{\substack{x, x' \in \mathcal{C} \\ x \neq x'}} d_H(x, x')$, we get that $a \leq b$.

Because $b \leq a$ and $a \leq b$, we have $a = b$ and $d(\mathcal{C}) = \min\limits_{\substack{x \in \mathcal{C} \\ x \neq 0}} w(x)$.

□

Proposition. *A word $x \in \mathbb{F}_2^n$ is a code word, if and only if it belongs to the kernel of the parity check matrix. In other words:*

$$x \in \mathcal{C} \Leftrightarrow H \cdot [x]_E = 0$$

Proof. See Lecture. □

Lemma. *If $A = (C_1|\ldots|C_n) = (a_{i,j})_{\substack{i=\overline{1,m} \\ j=\overline{1,n}}} \in \mathcal{M}_{m,n}$ and $v = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathcal{M}_{n,1}$, then*

$$A \cdot v = x_1 C_1 + \ldots x_n C_n$$

Proof. $A \cdot v = \begin{bmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,n} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \ldots & a_{m,n} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{1,1}x_1 + a_{1,2}x_2 + \ldots a_{1,n}x_n \\ a_{2,1}x_1 + a_{2,2}x_2 + \ldots a_{2,n}x_n \\ \vdots \\ a_{m,1}x_1 + a_{m,2}x_2 + \ldots a_{m,n}x_n \end{bmatrix} =$

$x_1 C_1 + \ldots x_n C_n$ □

And now we will see the actual way to compute the minimum Hamming distance of a linear code:

Proposition. *Let $\mathcal{C}$ be a linear (n, k)-code, just like above. Then $d(\mathcal{C}) = d$ if and only if the minimal number of columns in H that sum up to the zero vector is d. In other words, $d(\mathcal{C}) = d$ if and only if the sum of every set of $d - 1$ columns from H is nonzero and there is a set of d columns from H whose sum is the zero vector.*

Proof. We note $H = (C_1|C_2|\ldots|C_n)$ with $C_i \in \mathcal{M}_{n-k,1}$ the column vectors of H.

Let $x = (x_1, \ldots, x_n)$ be a word in $\mathbb{F}_2^n$. Then x is a code word if and only if $H \cdot [x]_E = 0$, which is the same as saying that $x_1C_1 + \ldots + x_nC_n = 0$.

By the lemma above we have that $d(\mathcal{C}) = \min\limits_{\substack{x \in \mathcal{C} \\ x \neq 0}} w(x)$.

We will now prove the direct implication. We have $d(\mathcal{C}) = d$. Let $C_{l_1}, \ldots, C_{l_{d-1}}$ be $d-1$ columns of H. Suppose $C_{l_1} + \ldots + C_{l_{d-1}} = 0$. Let y be the word in $\mathbb{F}_2^n$ so that $y_i = \begin{cases} 1, \text{ if } i = l_1, \ldots, l_{d-1} \\ 0, \text{ otherwise} \end{cases}$. Then $H \cdot [y]_E = 0$, so y is a code word. But $w(y) = d - 1$, which contradicts $d(\mathcal{C}) = d$. Thus, the sum of every set of $d-1$ columns from H is nonzero.

$d(\mathcal{C}) = d$, let $x \in \mathcal{C}$ so that $w(x) = d$ with $x_i = \begin{cases} 1, \text{ if } i = r_1, \ldots, r_d \\ 0, \text{ otherwise} \end{cases}$. x is a code word, so $H \cdot [x]_E = 0$, from whence we have $C_{r_1} + \ldots + C_{r_d} = 0$, which proves that there exists a set of d columns from H whose sum is zero.

Proving the converse is easy. Assume we have $d(\mathcal{C}) = d'$ and the minimal number of column vectors in H that sum up to 0 is d. On the other hand, from the argument above, $d(\mathcal{C}) = d'$ implies that the minimal number of column vectors in H that sum up to 0 is d'. This gives $d = d'$ and proves the converse. □

Conclusion. *The algorithm for computing the minimum Hamming distance of a linear (n, k)-code is the following:*

(1) *Write the parity check matrix H.*
(2) *Compute all the possible sums of 2 column vectors of H. If any of these sums is the zero vector, we stop and $d(\mathcal{C}) = 2$. Else, we go further.*
(3) *Compute all the possible sums of 3 column vectors of H. If any of these sums is the zero vector, we stop and $d(\mathcal{C}) = 3$. Else, we go further.*
(4) *We repeat until we reach a sum of d column vectors of H that is zero. We conclude that $d(\mathcal{C}) = d$.*

Example. *To see how this works, let's look at the example from class (exercise 12.5).*

12.5. *Determine the minimum Hamming distance between the code words of the code with generator matrix* $G = \begin{bmatrix} P \\ I_4 \end{bmatrix} \in \mathcal{M}_{9,4}(\mathbb{F}_2)$, *where:* $P = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$

Proof. The parity check matrix is

$$H = [C_1|\dots|C_9] = \left[\begin{array}{ccccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

We see that however we add two column vectors of this matrix we can't get the zero vector. On the other hand, we can find three column vectors that sum up to zero, see for instance, the first, fourth and ninth column. □

Remark. *As we see in the example, a good way to find this Hamming distance is to look at the columns of* P. *What interests us is the column with the minimal number of* 1*'s. Say this happens for a column in* P *in which exactly* d *elements are* 1*'s. This means that there exist* d *columns from* I_{n-k} *that we can add to this column so that we get a zero vector. Therefore* $d(\mathcal{C}) \leq d+1$. *We can't be certain that* $d(\mathcal{C}) = d+1$ *because it may happen that we can obtain the zero vector in other ways, with fewer vectors, by adding two or more columns from* P *and some columns from* I_{n-k}. *This achieves an upper bound for* $d(\mathcal{C})$, *albeit not a very precise one. If this is confusing, then disregard it and just follow the algorithm above.*