Decoding Linear Codes

Decoding

The usefulness of an error-correcting code would be greatly diminished if the decoding procedure was very time consuming. While the concept of decoding, i.e., finding the nearest codeword to the received vector, is simple enough, the algorithms for carrying this out can vary tremendously in terms of time and memory requirements.

Usually, the best (i.e., fastest) decoding algorithms are those designed for specific types of codes. We shall examine some algorithms which deal with the general class of linear codes and so, will not necessarily be the best for any particular code in this class.

Syndromes

When a codeword **c** is transmitted and vector **r** is received, the difference between the two is called the *error vector* **e**,
i.e. $\mathbf{r} = \mathbf{c} + \mathbf{e}$.

If H is a parity-check matrix for the linear code C, then

$$H\mathbf{r}^T = H(\mathbf{c} + \mathbf{e})^T = H\mathbf{c}^T + H\mathbf{e}^T = H\mathbf{e}^T$$

since $H\mathbf{c}^T = \mathbf{0}$ for any codeword.

$H\mathbf{r}^T$ is called the ***syndrome*** of **r**.

Syndromes

If wt(**e**) ≤ 1 then the syndrome of $\mathbf{r} = H\mathbf{e}^T$ is just a scalar multiple of a column of H. This observation leads to a simple decoding algorithm for 1-error correcting linear codes.

First, calculate the syndrome of **r**. If the syndrome is **0**, no error has occurred and **r** is a codeword.

Otherwise, find the column of H which is a scalar multiple of the syndrome.

If no such column exists then more than one error has occurred and the code can not correct it.

If however, the syndrome is α times column j, say, then add the vector with $-\alpha$ in position j and 0's elsewhere to **r**. This corrects the error.

Example

Let $H = \begin{pmatrix} 1 & 0 & 0 & 1 & 2 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$ be the parity check matrix of a code C in V[5,3]. Now **x** = (1 0 1 2 0) is a code word since,

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 2 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \\ 3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

If **r** = (1 0 1 1 0) is received when **x** is transmitted, then the syndrome of **r** is:

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 2 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 2 \end{pmatrix} = 2 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}.$$

Correction: **r** + (0 0 0 -2 0) = (1 0 1 -1 0) = (1 0 1 2 0).

Hamming Decoding

This method can be improved for binary Hamming codes. The very simple decoding algorithm that results is called ***Hamming Decoding***.

Rearranging the columns of the parity check matrix of a linear code gives the parity check matrix of an equivalent code. In the binary Hamming code of order r, the columns are all the non-zero binary vectors of length r. Each such column represents the binary form of an integer between 1 and $n = 2^r - 1$. We can arrange the columns of the parity check matrix so that the column in position i represents the integer i. Let H be the parity check matrix formed in this way.

Hamming Decoding

For example, the parity check matrix for the [7,4]-Hamming code would be written as:

$$H_3' = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Here, the column $(x,y,z)^T$ represents the number

$$x(2^0) + y(2^1) + z(2^2).$$

If v is the received vector, calculate the syndrome Hv^T. If at most one error has been made, the result is a vector of length r, either the zero vector or a column of H. When one error has been made the number represented by this column is the position in v which is in error – and since this is a binary code, we can correct it.

Hamming Decoding

0101010 is a codeword in the [7,4]-Hamming code. Suppose that we received the vector v = 0101110. We calculate:

$$H_3' v^T = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}.$$

The result represents the number 5, which is the position of the error.

Code Cosets

This procedure does not work for codes which are more than single error correcting. To deal with the more general situation, we introduce an equivalence relation on the vectors of V[n,q] with respect to the code C.

We say that vectors **x** and **y** are ***equivalent with respect to C*** if and only if, $\mathbf{x} - \mathbf{y} \in C$. This is an equivalence relation since,

1) $\mathbf{x} - \mathbf{x} = \mathbf{0} \in C$ for all **x**,

2) if $\mathbf{x} - \mathbf{y} \in C$, then $\mathbf{y} - \mathbf{x} = -(\mathbf{x}-\mathbf{y}) \in C$ since it is a scalar multiple of an element of C, and

3) if $\mathbf{x}-\mathbf{y} \in C$ and $\mathbf{y}-\mathbf{z} \in C$ then $\mathbf{x}-\mathbf{z} = (\mathbf{x}-\mathbf{y}) + (\mathbf{y}-\mathbf{z}) \in C$.

and the equivalence classes (which form a partition of V[n,q]) are called the ***cosets*** of C.

Code Cosets

An alternative way of viewing the cosets of C is to note that a coset consists of all the vectors that have the same syndrome. Indeed, if $\mathbf{x} - \mathbf{y} \in$ C, then $\mathbf{x} - \mathbf{y} = \mathbf{c}$ for some $\mathbf{c} \in$ C. Thus, $\mathbf{x} = \mathbf{c} + \mathbf{y}$ and

$$H\mathbf{x}^T = H(\mathbf{c} + \mathbf{y})^T = H\mathbf{c}^T + H\mathbf{y}^T = H\mathbf{y}^T.$$

Furthermore, if two vectors have the same syndrome then they are equivalent. If $H\mathbf{x}^T = H\mathbf{y}^T$,

$$\mathbf{0} = H\mathbf{x}^T - H\mathbf{y}^T = H(\mathbf{x} - \mathbf{y})^T,$$

and so, $\mathbf{x} - \mathbf{y}$ is $\in$ C.

Example

The table below lists the cosets of the [5,2]-code of V[5,2] whose generator matrix is given by,

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

The cosets are the rows in the table, with the first row being the code C (which is also a coset).

00000	10101	01110	11011
00001	10100	01111	11010
00010	10111	01100	11001
00100	10001	01010	11111
01000	11101	00110	10011
10000	00101	11110	01011
11000	01101	10110	00011
10010	00111	11100	01001

When the vectors of V[n,q] are arranged by cosets in this way it is called a ***standard array*** for the code

Example

The parity-check matrix for this code is

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

and the syndromes for each row are given below

00000	10101	01110	11011	**000**
00001	10100	01111	11010	**001**
00010	10111	01100	11001	**010**
00100	10001	01010	11111	**100**
01000	11101	00110	10011	**110**
10000	00101	11110	01011	**101**
11000	01101	10110	00011	**011**
10010	00111	11100	01001	**111**

Syndrome Decoding

In each coset we can select a vector of minimum weight (when there is a choice we can make the selection arbitrarily), this vector is called a ***coset leader***. In the previous example, the coset leaders were written in the first column of the standard array. We can use the coset leaders in a decoding algorithm.

When a vector **r** is received, we find the coset that it is contained in (via the syndrome calculation) and then subtract the coset leader for that coset to obtain a codeword. This procedure is called ***syndrome decoding***. That syndrome decoding gives the correct error correction is the content of the next theorem.

Syndrome Decoding

Theorem 4 : *Let C be an* [n,k]-*code in* V[n,q] *with codewords* c_j, $0 \le j \le q^k - 1$. *Let* l_i, $0 \le i \le q^{n-k} - 1$ *be a set of coset leaders for the cosets of this code. If* $r = l_i + c_h$ *then*

$$d(r,c_h) \le d(r,c_j) \text{ for all } j,\ 0 \le j \le q^k - 1.$$

Proof: Expressing these distances in terms of weights we have,

$$d(r,c_h) = d(l_i + c_h, c_h) = wt(l_i + c_h - c_h) = wt(l_i) \text{ and}$$

$$d(r,c_j) = d(l_i + c_h, c_j) = wt(l_i + c_h - c_j).$$

But $l_i + c_h - c_j$ is in the same coset as l_i, and the coset leader has minimal weight. ■

Syndrome Decoding

Note that if the coset leader is unique then c_h is the closest codeword (if it is not unique, then c_h is still as close as any other codeword). We have the following result about uniqueness.

Theorem 5 : *Let C be a linear code with minimum distance d. If* ***x*** *is any vector such that*

$$wt(x) \leqslant \left[\frac{d-1}{2}\right],$$

then ***x*** *is a* ***unique*** *element of minimum weight in its coset of C and hence is always a coset leader.*

Syndrome Decoding

Proof: Suppose that there exists another vector **y** with the same weight as **x** in **x**'s coset.

Since x and y are in the same coset, **x - y** $\in$ C, but

$$wt(x-y) \leqslant wt(x)+wt(y) \leqslant \left[\frac{d-1}{2}\right]+\left[\frac{d-1}{2}\right] \leqslant d-1.$$

This contradicts the minimum distance of the code, unless **x - y** is the zero vector, i.e. **x = y**. ■

Example

In the previous example, the [5,2]-code had minimum distance 3 (the minimum non-zero weight of a codeword) and so the above theorem states that the zero vector and all vectors of weight 1 will be unique coset leaders as can easily be verified from the standard array.

00000	10101	01110	11011
00001	10100	01111	11010
00010	10111	01100	11001
00100	10001	01010	11111
01000	11101	00110	10011
10000	00101	11110	01011
11000	01101	10110	00011
10010	00111	11100	01001