

Using lattice packings to create extended codebooks

DANIEL SIKORA

PROFESSOR ANNA GILBERT

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF MICHIGAN, 2074 EAST HALL, ANN ARBOR, MI 48109-1109

Email: dsikora@umich.edu

Introduction

In fields such as communications, the mathematical theory of codes is useful to quickly and accurately transmit information. One continuing goal of coding theory is to create "optimal" codes. Specifically, optimizing codes entails improving such properties of the codes as rate of transmission and error-correcting ability, as well as minimizing the average power required to send messages. One method of finding improved codes is that of extending existing codes. Extending a code can often provide a new codebook which improves upon the properties of the old code.

Gilbert and Tropp [1] mention a particularly interesting method of extending existing codebooks. Let Λ be the coordinates of the points in a m -dimensional sphere-packing; that is, if $\lambda \in \Lambda$, then $\lambda = (a_1, \dots, a_m)$. For the purposes of this paper, we are mostly interested in the case in which Λ is the set of points of a lattice, but this is not necessary. Let C_B be a d -dimensional pre-existing code (which will now be referred to as the "base codebook") consisting of at least m codewords, $C_B = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_s : s \geq m\}$. Our extended codebook, C , is created by using the coordinates of each lattice point as coefficients of all possible m -term subsets of C_B , forming all possible linear combinations. In other words, create a $m \times n$ matrix M which has exactly m distinct words from C_B as its rows. The corresponding new codeword, \mathbf{c} , in our extended codebook is the product of a lattice point λ with M , so

$$\mathbf{c} = a_1 \mathbf{b}_{i_1} + a_2 \mathbf{b}_{i_2} + \dots + a_m \mathbf{b}_{i_m}$$

Using all lattice points and all possible m -term permutations of the codewords in C_B in this way yields the new extended codebook C .

After creating such an extended codebook, one will wish to calculate and analyze several properties of the code:

- \bar{P} , the average power of C (where the power of an individual codeword $\mathbf{c} \in C$ is $\|\mathbf{c}\|_2$).
- P_{max} , the maximum power of any word in C .
- R , the binary rate of C .

$$R = \frac{\log_2 |C|}{d},$$

where d is the dimension of C (and C_B) and $|C|$ is the number of codewords in C .

- ρ , the packing radius of C .

The main interest of this paper is to discuss theoretical results of the study of these codes, as well as the methods created and used both to construct the codes and to calculate the above-mentioned quantities.

Theoretical Results

Theorem. *If a $n \times n \times \dots \times n$ lattice Λ (where n is an odd, positive integer) is constructed from m m -dimensional basis vectors, and Λ is used to extend a d -dimensional canonical basis, then the resulting extended codebook C will have average power*

$$\bar{P} = \frac{1}{12} m n^m (n^2 - 1) \frac{d!}{(d-m)!} \left(-d + \sum_{i=0}^{m-1} \binom{m-1}{i} \frac{d!}{(d-m+i)!} n(n-1)^{m-(i+1)} \right)^{-1}$$

PROOF.

Lemma 1. *If a $n \times n \times \dots \times n$ lattice Λ (where n is an odd, positive integer) is constructed from m m -dimensional basis vectors, and Λ is used to extend a d -dimensional canonical basis, then the resulting extended codebook C will have size*

$$|C| = -d + \sum_{i=0}^{m-1} \binom{m-1}{i} \frac{d!}{(d-m+i)!} n(n-1)^{m-(i+1)}$$

PROOF OF LEMMA 1. The m -dimensional lattice Λ can be described by the basis vectors

$$\begin{aligned} v_1 &= (1, 0, \dots, 0) \\ v_2 &= (\cos \theta_1, \sin \theta_1, 0, \dots, 0) \\ v_3 &= (\cos \theta_2, 0, \sin \theta_2, 0, \dots, 0) \\ &\vdots \\ v_m &= (\cos \theta_{m-1}, 0, \dots, 0, \sin \theta_{m-1}) \end{aligned}$$

Then if Λ is $n \times n \times \dots \times n$, the coordinates of Λ are

$$\Lambda = \{a_1 v_1 + a_2 v_2 + \dots + a_m v_m : |a_i| \leq \frac{n-1}{2} \forall i = 1, \dots, m\}$$

or stated otherwise,

$$\begin{aligned} \Lambda &= \{(a_1 + a_2 \cos \theta_1 + a_3 \cos \theta_2 + \dots + a_m \cos \theta_{m-1}, a_2 \sin \theta_1, a_3 \sin \theta_2, \dots, a_m \sin \theta_{m-1}) : \\ &\quad |a_i| \leq \frac{n-1}{2} \forall i = 1, \dots, m\} \end{aligned}$$

Applying these coordinates to a d -dimensional canonical basis,

$$C_B = \{(1, 0, \dots, 0), (0, 1, 0, \dots, 0), \dots, (0, \dots, 0, 1)\},$$

yields vectors of length d with coordinates which are all possible permutations of the m terms in the points of Λ and $(d-m)$ zeros. To count all possible words in the new extended codebook, count them in cases:

Case 1: $|a_i| \leq \frac{n-1}{2} \forall i = 1, \dots, m; a_2, \dots, a_m, \neq 0$.

There are n possible values of a_1 , and $(n-1)$ values each of a_2, \dots, a_m . Permuting all terms in the words with the $(d-m)$ zeros, there are $\frac{d!}{(d-m)!}$ permutations. Then there are $\frac{d!}{(d-m)!} n(n-1)^{m-1}$ such codewords.

Case 2: $|a_i| \leq \frac{n-1}{2} \forall i = 1, \dots, m$; one of the values a_2, \dots, a_m equals zero, and the rest of these values are nonzero.

There are, for reasons similar to Case 1, $n(n-1)^{m-2}$ possible combinations of values for a_1, \dots, a_m . $(d-m+1)$ of the coordinate values are zero, so there are $\frac{d!}{(d-m+1)!}$ permutations of the coordinate values.

There are $(m-1)$ choices for which of the values a_2, \dots, a_m is equal to zero (per the conditions of Case 2), so there are $(m-1) \frac{d!}{(d-m+1)!} n(n-1)^{m-2}$ such codewords.

⋮

Continuing in this manner, if $|a_i| \leq \frac{n-1}{2} \forall i = 1, \dots, m$, i of the values a_2, \dots, a_m are equal to zero and the remaining values are nonzero, then there are $n(n-1)^{m-(i+1)}$ possible combinations of values of a_1, \dots, a_m , $\frac{d!}{(d-m+i)!}$ permutations of the codeword coordinates, and $\binom{m-1}{i}$ selections of the zero coordinates from a_1, \dots, a_m , and thus $\binom{m-1}{i} \frac{d!}{(d-m+i)!} n(n-1)^{m-(i+1)}$ such codewords.

Now note that there are d permutations of the coordinates $(a_1, 0, \dots, 0)$. If $a_1 = 0$, then the zero codeword has been counted d times. The zero codeword is of no interest to the extended code, so this count is subtracted from our number of words.

Therefore, the size of the extended codebook C is

$$|C| = -d + \sum_{i=0}^{m-1} \binom{m-1}{i} \frac{d!}{(d-m+i)!} n(n-1)^{m-(i+1)}$$

□

Lemma 2. *If a $n \times n \times \dots \times n$ lattice Λ (where n is an odd, positive integer) is constructed from m m -dimensional basis vectors, and Λ is used to extend a d -dimensional canonical basis, then the total power of all codewords in the resulting extended codebook C is*

$$P_{total} = \frac{1}{12} m n^m (n^2 - 1) \frac{d!}{(d-m)!}$$

PROOF OF LEMMA 2. Any word $\mathbf{c} \in C$ is a permutation of the coordinates $(a_1 + a_2 \cos \theta_1 + \dots + a_m \cos \theta_{m-1}, a_2 \sin \theta_1, \dots, a_m \sin \theta_{m-1})$ and $(d-m)$ zeros. The power of an individual codeword, then, is

$$P(\mathbf{c}) = (a_1 + a_2 \cos \theta_1 + \dots + a_m \cos \theta_{m-1})^2 + \sum_{i=1}^m a_i^2 \sin^2 \theta_{i-1}$$

Now, define δ such that

$$\delta := (a_1 + a_2 \cos \theta_1 + \dots + a_m \cos \theta_{m-1})^2 - (a_1^2 + a_2^2 \cos^2 \theta_1 + \dots + a_m^2 \cos^2 \theta_{m-1})$$

Now

$$P(\mathbf{c}) = \delta + a_1^2 + \sum_{i=2}^m (a_i^2 \cos^2 \theta_{i-1}) + \sum_{i=2}^m (a_i^2 \sin^2 \theta_{i-1}),$$

and therefore the power of an individual word $\mathbf{c} \in C$ is $P(\mathbf{c}) = \delta + \sum_{i=1}^m a_i^2$.

Adding this up over all possible values of each a_i and counting the $\frac{d!}{(d-m)!}$ permutations of the values,

$$P_{total} = \frac{d!}{(d-m)!} \sum_{|a_i| \leq \frac{n-1}{2} \forall i} \left(\delta + \sum_{j=1}^m a_j^2 \right) = \sum_{|a_i| \leq \frac{n-1}{2} \forall i} \delta + \sum_{|a_i| \leq \frac{n-1}{2} \forall i} \left(\sum_{j=1}^m a_j^2 \right)$$

By its definition, one can see that all terms in δ have a first-degree a_i term as a coefficient. Therefore, as the sum of these terms from $a_i = -(\frac{n-1}{2})$ to $a_i = \frac{n-1}{2}$ is taken over all $i = 1, \dots, m$, it can be seen that $\sum_{|a_i| \leq \frac{n-1}{2} \forall i} \delta = 0$, so

$$P_{total} = \frac{d!}{(d-m)!} \sum_{|a_i| \leq \frac{n-1}{2} \forall i} \left(\sum_{j=1}^m a_j^2 \right) = \frac{d!}{(d-m)!} \sum_{|a_1| \leq \frac{n-1}{2}} \sum_{|a_2| \leq \frac{n-1}{2}} \dots \sum_{|a_m| \leq \frac{n-1}{2}} \left(\sum_{j=1}^m a_j^2 \right)$$

Define $x := \frac{n-1}{2}$, so

$$P_{total} = \frac{d!}{(d-m)!} \sum_{a_1=-x}^x \sum_{a_2=-x}^x \dots \sum_{a_m=-x}^x (a_1^2 + a_2^2 + \dots + a_m^2)$$

$$\begin{aligned}
&= \frac{d!}{(d-m)!} (2x+1)^{m-1} \left(\sum_{a_1=-x}^x a_1^2 + \cdots + \sum_{a_m=-x}^x a_m^2 \right) \\
&= \frac{d!}{(d-m)!} m(2x+1)^{m-1} \sum_{i=-x}^x i^2 = 2m(2x+1)^{m-1} \frac{d!}{(d-m)!} \sum_{i=1}^x i^2 \\
&= 2m(2x+1)^{m-1} \frac{d!}{(d-m)!} \frac{x(x+1)(2x+1)}{6}
\end{aligned}$$

Substituting back in $x = \frac{n-1}{2}$ results in

$$P_{total} = \frac{1}{12} mn^m (n^2 - 1) \frac{d!}{(d-m)!}$$

□

Since $\bar{P} = P_{total}/|C|$, by Lemma 1 and Lemma 2,

$$\bar{P} = \frac{1}{12} mn^m (n^2 - 1) \frac{d!}{(d-m)!} \left(-d + \sum_{i=0}^{m-1} \binom{m-1}{i} \frac{d!}{(d-m+i)!} n(n-1)^{m-(i+1)} \right)^{-1}$$

□

Corollary (Corollary of Lemma 1). *If a $n \times n \times \dots \times n$ lattice Λ (where n is an odd, positive integer) is constructed from m m -dimensional basis vectors, and Λ is used to extend a d -dimensional canonical basis, then the binary rate of the resulting extended codebook C is*

$$R = \frac{\log_2 \left(-d + \sum_{i=0}^{m-1} \binom{m-1}{i} \frac{d!}{(d-m+i)!} n(n-1)^{m-(i+1)} \right)}{d}$$

Observations

An interesting application of the above theorem comes in the analysis of its asymptotics; that is, an analysis of how \bar{P} behaves as d (the dimension of the base codebook), m (the dimension of the lattice packing), or n (the side length of the lattice packing) changes. Picking constant values for two of the variables d , m , and n , MATLAB was used to calculate and plot increasing values of the third variable against the corresponding values of \bar{P} .

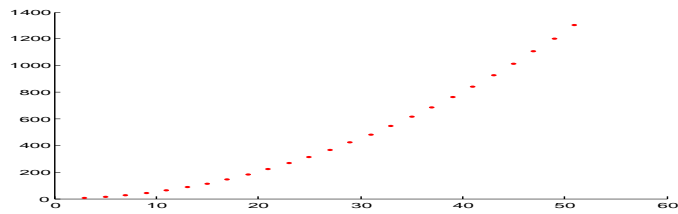


Figure 1: \bar{P} vs. n , $d = m = 6$

If d and m are held constant while n is changed, \bar{P} behaves predictably. As n increases (that is, the size of the lattice increases), the average power increases, and appears to do so exponentially.

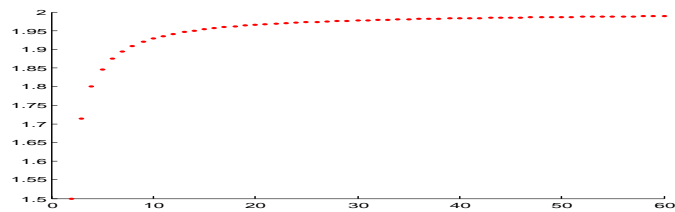


Figure 2: \bar{P} vs. d , $m = 2$, $n = 3$

If m and n are held at constant values (in the above figure, $m = 2$ and $n = 3$), \bar{P} appears to increase asymptotically to a certain value, dependent on the values of m and n , as d increases (in this case, \bar{P} is asymptotic to 2).

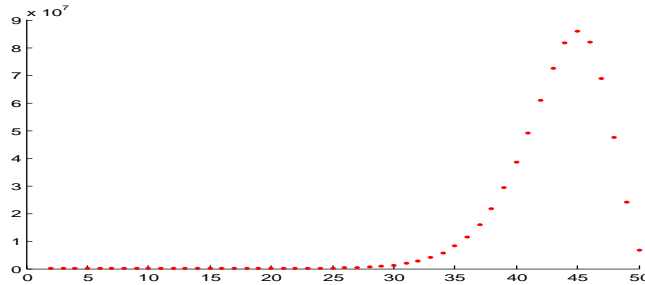


Figure 3: \bar{P} vs. m , $d = 50$, $n = 3$

Due to the nature of the construction of these extended codebooks, \bar{P} is undefined for any value of m greater than d . Therefore, the above figure displays the behavior of \bar{P} as $m \rightarrow d$. An interesting observation of these curves is that \bar{P} does not behave "predictably" by increasing indefinitely; rather, it has a point of maximum \bar{P} at some value $m < d$, and then decreases substantially as m increases further. This lends itself to the idea that there is an "optimal" choice of values d , m , and n to produce a code with the most desirable properties, particularly \bar{P} .

Examples

All of these results were preceded by working by hand with multiple examples in small dimensions. Specifically, one of the simplest cases was studied with $d = 2$, $m = 2$, and $n = 3$, using the canonical basis in \mathbf{R}^2 as our base codebook C_B . The basis vectors of the lattice Λ were chosen to represent all possible 2-dimensional lattices; one vector is fixed at $(1, 0)$, and the other is $(\cos \alpha, \sin \alpha)$, a unit-norm vector created with an arbitrary angle α , with $\alpha \in [0, \frac{\pi}{2}]$ (all other values of α produce lattices which can be rotated or reflected to yield the above-mentioned lattices; as mentioned in Chapter 1 of [2], these lattices are equivalent, and therefore values of α other than $[0, \frac{\pi}{2}]$ need not be considered).

The extended codebook is created with $C_B = \{(1, 0), (0, 1)\}$ and $\Lambda = \{(-1 - \cos \alpha, -\sin \alpha), (-\cos \alpha, -\sin \alpha), (1 - \cos \alpha, -\sin \alpha), (-1, 0), (0, 0), (1, 0), (-1 + \cos \alpha, \sin \alpha), (\cos \alpha, \sin \alpha), (1 + \cos \alpha, \sin \alpha)\}$.

As previously seen in the proof of Lemma 2, using the canonical basis in \mathbf{R}^d as C_B yields an extended code C containing the original lattice points and all permutations thereof. Then, if the resulting extended code from the base codebook and lattice packing which we have specified is plotted in the xy -plane, the new code is simply the original lattice Λ and all of its points reflected over the line $x = y$.

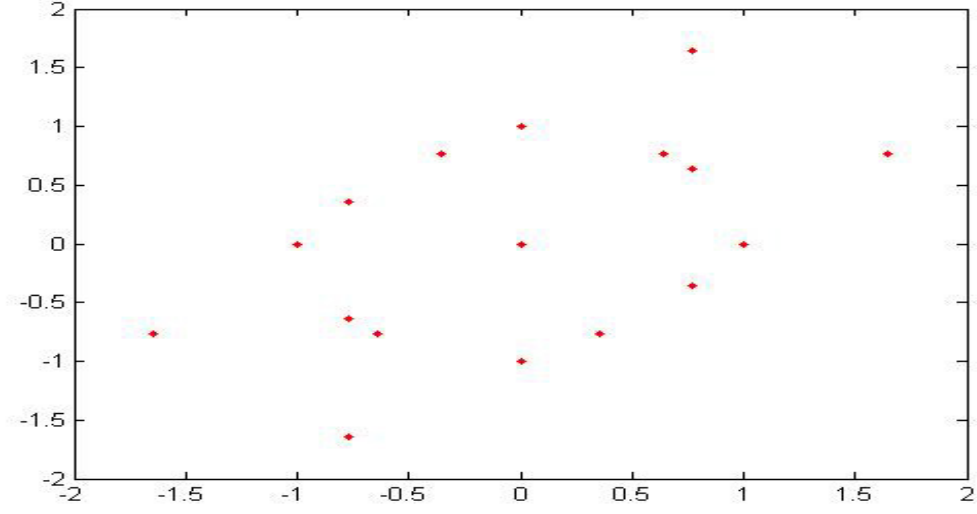


Figure 4: Extended code with $m = 2$, $n = 3$; C_B is the canonical basis in 2-D.

This code has 16 codewords, $R = 2$, $\bar{P} = \frac{3}{2}$, $P_{max} = 2 + 2 \cos \alpha$, and packing radius, as a function of the chosen angle α ,

$$\rho(\alpha) = \begin{cases} \frac{1}{2}(3 - 2(\sin \alpha + \cos \alpha))^{1/2} & \text{if } \alpha \in [0, \frac{\pi}{6}] \text{ or } \alpha \in [\frac{\pi}{3}, \frac{\pi}{2}] \\ \frac{1}{2}(2 - 4 \cos \alpha \sin \alpha)^{1/2} & \text{if } \alpha \in [\frac{\pi}{6}, \frac{\pi}{4}] \text{ or } \alpha \in (\frac{\pi}{4}, \frac{\pi}{3}] \\ \frac{1}{2}(3 - 2\sqrt{2})^{1/2} & \text{if } \alpha = \frac{\pi}{4} \end{cases}$$

This construction was also repeated, with d and m held constant at 2, for $n = 5, 7, 9$.

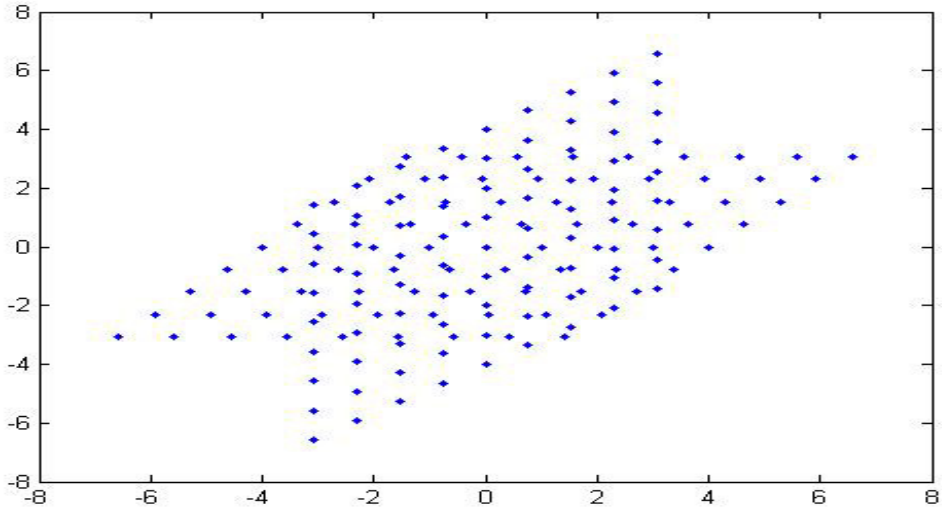


Figure 5: Extended code with $m = 2$, $n = 9$; C_B is the canonical basis in 2-D.

Software

Even the most basic examples of creating extended codes and calculating their physical properties can be rather difficult and time-consuming if done by hand. For this reason, I have written a number of programs in MATLAB to perform these tasks.

- `lattice.m`

- Called with `lattice(Basis, n)`.
- Inputs: *Basis*, a matrix with all of the lattice's basis vectors as its row vectors, and *n*, the side length of the lattice (this is the same *n* as discussed in the Theoretical Results section).
- Output: The function outputs the lattice constructed from the input specifications, in the form of a matrix whose rows are each the coordinates of a lattice point.

- `extend.m`

- Called with `extend(Packing, BaseCode)`.
- Inputs: *Packing*, a matrix with every point in the packing Λ as its row vectors, and *BaseCode*, a matrix with each word in the base codebook C_B as its row vectors.
Note: The output of `lattice.m` can be used as an input into this function if a lattice packing is desired.
- Output: The function outputs the extended codebook, *C*, constructed from the input specifications, in the form of a matrix whose rows are each codewords in *C*.

- `codestats.m`

- Called with `codestats(Code)`
- Input: *Code*, a matrix with every point in the codebook as its row vectors.
Note: The output of `extend.m` can be used as an input in this function (yielding the properties of the created extended code).
- Output: The function outputs the statistics packing radius ρ , average power \bar{P} , maximum power P_{max} , and rate R , respectively, as the four elements of a 1 x 4 matrix.

References

- [1] A.C. Gilbert and J.A. Tropp, "Applications of sparse approximation in communications."
- [2] J.H. Conway and N.J.A. Sloane, *Sphere Packings, Lattices and Groups*, 1988.

Source Code

lattice.m

```
function [Packing] = lattice(LBasis, n) m = length(LBasis(:,1));
coeff = zeros(n^m, m);

if mod(n, 2)~=1
    error('Side length of lattice must be odd')
end

x = (n - 1)/2;

for col = 1:m
    y = (n^(m - col));
    for i = 1:n
        coeff(((i-1)*y)+1:((i-1)*y + y), col) = (i-1-x)*ones(y,1);
    end
end

for col = 2:m
    y = (n^(m - (col-1)));
    X = coeff(1:y, col);
    for i = (y + 1):y:(n^m)
        coeff(i:i+y-1, col) = X;
    end
end

Packing = zeros(length(coeff(:,1)), length(LBasis(1,:)));

for row = 1:(length(coeff(:,1)))
    Packing(row,:) = coeff(row,:)*LBasis;
end
```

extend.m

```
function [Code] = extend(Packing, BaseCode)

d = length(BaseCode(1,:)); B = length(BaseCode(:,1)); m =
length(Packing(1,:)); N = length(Packing(:,1));

if d<m
    error('ERROR: Not enough words in base code')
end

index = zeros(B^m, m);

for col = 1:m
    y = (B^(m - col));
    for i = 1:B
        index(((i-1)*y)+1:((i-1)*y + y), col) = i*ones(y,1);
    end
end

for col = 2:m
    y = (B^(m - (col-1)));
    X = index(1:y, col);
    for i = (y + 1):y:(B^m)
        index(i:i+y-1, col) = X;
    end
end

match2 = 1; while(match2 ~= 0)
    match2 = 0;
    for row = 1:(B^m)
        match = 0;
        for i = 1:m
            for j = 1:m
                if match~=0
                    continue
                end
                if(row<=length(index(:,1)))
                    if ((index(row,i) == index(row,j)) & (i~=j))
                        match = 1;
                        match2 = 1;
                        index(row,:) = [];
                    end
                end
            end
        end
    end
end

count = 1; for p = 1:length(Packing(:,1)) for index_row =
1:length(index(:,1))

    Code(count,:) = Packing(p,:)*BaseCode(index(index_row,:),:);
    count = count + 1;
end
```

```

    end
end

X = length(Code(:,1)); dup = 1; while(dup~=0)
    dup = 0;
    for row1 = 1:X
        for row2 = 1:X
            if ((row1<=length(Code(:,1))) & (row2<=length(Code(:,1))))
                if ((row1~=row2) & (Code(row1,)==Code(row2,)))
                    dup = 1;
                    Code(row2,:) = [];
                end
            end
        end
    end
end

X = length(Code(:,1)); zero = 1; while(zero~=0)
    zero = 0;
    for row = 1:X
        if row<=length(Code(:,1))
            if Code(row,)== 0;
                Code(row,)= [];
                zero = 1;
            end
        end
    end
end
end

```

