Coding Theory

by

Mark Bergman, Jeff Matthes, John Hanson,
Jake Hustad, Thomas Bursey, Brad Teter,
Berit Rollay, Nick Bremer, Tyler Stelzer, Robert Coulson

December 2001 – May 2002
Contents

I. Introduction pg. 3
II. History of a Code pg. 4
III. What is a Code pg. 6
IV. Code Performance pg. 7
V. Abstract Algebra Background pg. 9
VI. Reed – Solomon Codes pg. 11
VII. Algebraic Geometric Background pg. 13
VIII. Conclusion pg. 19
IX. Resources Appendix A
Introduction

Coding theory is the branch of mathematics concerned with transmitting data across noisy channels and recovering the message. Coding theory is about making messages easy to read and finding efficient ways of encoding data.

By looking at the title page you may ask a simple question. What’s with the title? Well, probably the most commonly seen code in day-to-day life is the International Standardized Book Number (ISBN) Code.

This number relates to codes such that the last digit of the ISBN number represents a check digit that checks for errors when the number is transmitted. So, if you imagine a library, every book is assigned an ISBN using the same system for choosing the check digit. If you are working in that library cataloging new books and you make a mistake when typing in this number, the computer can be programmed to catch the typing error.
The History of a code

In 1948, Claude Shannon, working at Bell Laboratories in the USA, inaugurated the whole subject of coding theory by showing that it was possible to encode messages in such a way that the number of extra bits transmitted was as small as possible. Unfortunately his proof did not give any explicit recipes for these optimal codes.

Shannon published A Mathematical Theory of Communication in the Bell System Technical Journal (1948). This paper founded the subject of information theory and he proposed a linear schematic model of a communications system. This was a new idea. Communication was then thought of as requiring electromagnetic waves to be sent down a wire. The idea that one could transmit pictures, words, sounds etc. by sending a stream of 1's and 0's down a wire, something which today seems so obvious as we take the information from a web server sitting in someone's closet and view it on our computer hundreds of miles away, was fundamentally new.

It was two years later that Richard Hamming, also at Bell Labs, began studying explicit error-correcting codes with information transmission rates more efficient than simple repetition. His first attempt produced a code in which four data bits were followed by three check bits which allowed not only the detection but the correction of a single error. (The repetition code would require nine check bits to achieve this.)

It is said that Hamming invented his code after several attempts to punch out a message on paper tape using the parity code. "If it can detect the error," he complained, "why can't it correct it!".

The value of error-correcting codes for information transmission, both on Earth and from space, was immediately apparent, and a wide variety of codes were constructed which achieved both economy of transmission and error-correction capacity. Between 1969 and 1973 the NASA Mariner probes used a powerful Reed--Muller code capable of correcting 7 errors out of 32 bits transmitted, consisting now of 6 data bits and 26 check bits! Over 16,000 bits per second were relayed back to Earth.
A less obvious application of error-correcting codes came with the development of the compact disc. On CDs the signal is encoded digitally. To guard against scratches, cracks and similar damage two "interleaved" codes which can correct up to 4,000 consecutive errors (about 2.5 mm of track) are used. (Audio disc players can recover from even more damage by interpolating the signal.)
**What is a Code?**

First, I would discuss what is a code? Two simple examples of codes are the ISBN codes and the repetition codes. First we will look at ISBN codes, which stands for international standardized book number. The ISBN code system was started in 1968 and it is a standard identification system for books. A sample ISBN code looks like the following 0-444-85193-3. The first part of the code represents a group or country identifier. The second part of the code represents a publisher identifier. The third part of the code represents a title identifier. The last part of the code is a check digit, which is computed from the first nine digits. To compute the check digit you take the first nine digits of the code and then plug it into the given equation \( a = (a + a_2 + \ldots + a_9) \) and mod that answer by 11. Then if your answer is between 0 and 9 then that answer become the check digit, but if the answer is 10 you set the value to the symbol X. Last, ISBN codes are limited to only catching the error and not correcting it.

Next, I would like to discuss repetition codes. A repetition code transmits data in repeated blocks of code, so for example a four bit code word like 1010 could be transmitted in three blocks like this: 1010 1010 1010. The idea of repetition codes is that if an error has occurred in one of the blocks the other two blocks would still agree and the error could then be corrected. Now, if you wanted to be able to correct two errors you would have to transmit the given data block five times. In general to correct \( t \) errors you would have to transmit the given data block \( 2t+1 \) times. Last, the formal definition of a code is: a code \( C \) over an alphabet \( A \) is simply a subset of \( A \) to the power \( n \) of \( n \) copies where \( A \) is usually a finite field.
Code Performance

In the search for useful and efficient codes there are several objectives to keep in mind.

1. Detection and correction of errors due to noise
2. Efficient transmission of data
3. Easy encoding and decoding schemes

The first objective is the most important. Ideally we’d like a code that is capable of correcting all errors due to noise. In general, the more errors that a code needs to correct per message digit, the less efficient the transmission and also the more complicated the encoding and decoding schemes.

A code has 3 parameters that affect its performance: n, k, and d.

“n” is the total number of available symbols for a code word. This is the size of the alphabet. An alphabet is the set of all available symbols for a code word. Generally, it is chosen to be a finite field. In the case of binary, the alphabet is just the set (0, 1).

“k” is the number of information symbols in a given code word. Or, more precisely, it is the length of the code. The larger k is, the better error detecting ability it has, but also the more complicated it becomes. So it’s a tradeoff between accuracy and simplicity.

“d” is this concept of a Hamming distance. The Hamming distance between two words is the number of places where the digits differ between the two words. For example, let’s consider a binary alphabet, and the transmitted word is “111” and the received word is “110”, they differ only in the last digit, so the Hamming distance between them is 1. This distance is significant because it gives an idea of how many errors can be detected. The larger this distance is, the more errors can be detected.
Which brings up another important concept known as the minimum distance. The minimum distance is the smallest Hamming distance between any two possible code words. Suppose that the minimum distance for the coding function is 3. Then, given any codeword, at least 3 places in it have to be changed before it gets converted into another codeword. In other words, if up to 2 errors occur, the resulting word will not be a codeword, and occurrence of errors is detected. Thus, we have the following fact: If the minimum distance between code words is \( d \), up to \( d - 1 \) errors can be detected.

It can also be proven that the Hamming distance function is a metric. A metric is a function that associates any two objects, in this case code words, in a set with a number and that also satisfies three axioms that insure that the function preserves some of the notions about distance with which we are familiar.

So these are the three parameters, “n” being the alphabet size, k being the number of information symbols, and “d” being the distance between.
Abstract Algebra

The Abstract Algebra background is needed for explaining the concepts, which will be used in the case study of the Reed-Solomon code. The following concepts that will be used are: a Ring, a Field and a Vector Space.

A Ring is a set $R$, which is equipped with two binary operations: $(+)$ ~ addition, and $(*)$ ~ multiplication. For which the ring $R$ satisfies the following properties:

i. Additive identity, for which every element $a \in R$, such that $a+0 = 0+a = a$

ii. Addition is commutative, for elements $a, b \in R$, such that $a+b = b+a$

iii. Additive inverses, for every element $a \in R$, there exits $b \in R$, such that $a+b = b+a = 0$

iv. Addition is associative, for elements $a, b, c \in R$, such that $(a+b)+c = a+(b+c)$

v. Multiplication has to be associative, for elements $a, b, c \in R$, such that $(a*b)*c = a*(b*c)$

vi. The Distributive law holds, that is for elements $a, b, c \in R$, such that $a*(b+c) = a*b+a*c$

vii. And we assume an identity element exists that acts like a “1”

Some examples of rings are:

i. $\mathbb{Z}$ – The Integers

ii. $\mathbb{Z}/n$ – The Integers mod n, which have the general form of $\{0, 1, 2, \ldots, (n-1)\}$. An example is $\mathbb{Z}_4 = \{0, 1, 2, 3\}$

iii. $\mathbb{Q}$ – The Rational numbers

iv. $M_n(\mathbb{Q})$ – The $n \times n$ square matrices with rational entries

A Field is a ring $F$, in which multiplication works as nicely as addition. In which the following properties are satisfied:

i. Multiplication is commutative, that is for elements $a, b \in F$, such that $a*b = b*a$

ii. Multiplicative inverses, that is for every element $a \in F$ except for 0, there exists $b \in F$, such that $a*b = b*a = 1$

Some examples of fields are:
i. $\mathbb{R}$ – The Real numbers  
ii. $\mathbb{Q}$ – The Rational numbers 
iii. $\mathbb{C}$ – The Complex numbers  
iv. $\mathbb{Z}/p$ – The Integers mod $p$, where $p$ is prime, which have the general form of $\{0, 1, 2, \ldots, (p-1)\}$. An example is $\mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$ 
v. $\mathbb{Q}[i] = \{a+bi \mid a, b \in \mathbb{Q}\}$ – The elements of this field are complex numbers with the form $a+bi$, were $a$, and $b$ are rational numbers 
vi. $\mathbb{F}[x]$ – Is Polynomial Ring having coefficients from $\mathbb{F}$, where $\mathbb{F}$ is a field

From linear algebra a Vector Space $V$ over a field $F$, is defined as a set of vectors that has a scalar addition, and scalar multiplication. For which the usual vector properties apply, such as with $\alpha, \beta \in F$ and $v, w \in V$.

i. Scalar addition, such that $\alpha (v+w) = \alpha v + \alpha w$ 
ii. Scalar multiplication, such that $\alpha (\beta w) = (\alpha \beta) w$

Plus the rest of the vector space properties, that can be looked up in any linear algebra text book.
Reed Solomon

$F_q$ ~ field with $q$ elements

$L_r := \{ f \in F_q[x] \mid \deg(f) \leq r \} \cup \{0\}$

$r$ is non-negative

note: this is a vector subspace over $F_q$

with dim = $r+1$

basis = [ 1 x $x^2$ … $x^r$ ]

Label $q-1$ nonzero elements of $F_q$ as:

$a_1, a_2, \ldots, a_{q-1}$

Pick a $k \in \mathbb{Z}$ such that $1 \leq k \leq q-1$

Then we have:

$RS(k,q) := \{ (f(a_1), f(a_2), \ldots, f(a_{q-1})) \mid f \in L_{k-1} \}$

What is $d$?

Because of this we know $d$ is the number of nonzero components of each code word.

Thus $(n-d)$ components are zero. This tells us that $f$ will have at least $(n-d)$ roots.

So $\deg(f) \geq (n-d)$.

What is $d$?

So we have: $\deg(f) \geq (n-d)$

but since $f \in L_{k-1}$

We also know: $k-1 \geq \deg(f)$

Thus

$k-1 \geq n-d$

$\Rightarrow d \geq n-k+1$

Singleton Bound says for a Linear code with length $n$, dimension $k$ and minimum distance $d$,

then:

$d \leq n-k+1$

If this is true:

$d = n-k+1$

Why is Singleton Bound Correct?

First we define a subspace $W$ of

$W = \{ a = (a_1, a_2, \ldots, a_n) \mid \text{at least } n-d \text{ of } a_n \text{'s are } 0 \}$
So we can order them as follows:

\[a_1, \ldots, a_{d-1}\] as possible nonzero and \(a_d, \ldots, a_n\) as the elements we are sure are zero

From this we note:

\[\text{wt}(a) \leq d - 1\]

Why is Singleton Bound Correct?

But our code has weight \(d\).

From this we know:

\[W \cap C = \{0\}\]

Where \(C\) is our code.

Why is Singleton Bound Correct?

Since these sets intersect at zero we can say:

\[\dim(W + C) = \dim(W) + \dim(C)\]

It's obvious this must be less than \(n\), so:

\[n \geq \dim(W + C) = \dim(W) + \dim(C)\]

Why is Singleton Bound Correct?

\[n \geq \dim(W + C) = \dim(W) + \dim(C)\]

\[= d - 1 + k\]

From this we have:

\[n \geq d - 1 + k\]

\[= d \leq n - k + 1\]

Which Proves the Singleton Bound Theorem

We now know for sure:

\[d = n - k + 1\]

To summarize:

\[n = q - 1\]
\[k = k \text{ (which was chosen)}\]
\[d = n - k + 1\]
Algebraic Geometric Background

Algebraically closed fields can be defined as a field $k$ that every non-constant polynomial contained in $k[x]$ has at least one root. An example would be $f(x) = x^2 + 1$. $x$ is equal to plus or minus $i$. This is not closed under the real numbers because $i$ is not an element of the real numbers, but it is closed under the complex numbers.

Let $k$ be a field. A definition of the algebraic closure of $k$ is a field $K$ with $k$ being a subset of $K$ satisfying both that $K$ is algebraically closed, and that if $L$ is a field such that $k$ is a subset of $L$ which is a subset of $K$ and $L$ is algebraically closed, then $L$ must equal $K$.

Algebraic closures are unique. Every field has an unique algebraic closure, up to isomorphism. An example would be the algebraic closure of the real numbers is equal to the complex numbers.

Let $k$ be an algebraically closed field and let $f(x)$ elements in $k[x]$ be a polynomial of degree $n$. Then there exists a non-zero $u$ and alpha one to alpha $n$ elements in $k$ (not necessarily distinct) such that $f$ of $x$ is equal to $u$ times $x$ minus alpha one to $x$ minus alpha $n$. In particular, counting multiplicity, $f$ has exactly $n$ roots in $k$. 
Diophantine Equations

A Diophantine Equation is a polynomial with integer or rational coefficients, such as

\[ x^2 - 2y^2 = 1 \]

. A useful problem to solve is how many rational solutions does this equation have? In order to answer this question, we need to define points at “infinity.”

Let \( k \) be a field. The projective plane \( P^2(k) \) is defined as:

\[ P^2(k) := (k^3 \setminus \{(0,0,0)\}) / \sim \]

where \((X_0, Y_0, Z_0) \sim (X_1, Y_1, Z_1)\) if and only if there is some non-zero \( \alpha \) with \( X_1 = \alpha X_0, \)
\( Y_1 = \alpha Y_0 \) and \( Z_1 = \alpha Z_0 \).

To further understand the projective plane, consider the following illustration:

The projective plane can be described as all of the lines in \( \mathbb{R}^3 \) that pass through the origin. Further, we can say that lines that intersect the plane \( P \) shown above represent “real” points, and lines that do not intersect the plane represent points at “infinity.” These terms will be defined in more depth momentarily.

Let \( k \) be a field, and \( f(x, y) \in k[x, y] \) a polynomial of degree \( d \), and \( C_f \) the curve associated to \( f(f(x,y) = 0) \). The projective closure of the curve \( C_f \) is:

\[ \hat{C}_f := \{(X_0 : Y_0 : Z_0) \in P^2 \mid F(X_0, Y_0, Z_0) = 0\} \]

Where the homogenization of \( f \) is:

\[ F(X, Y, Z) := Z^d \left( \frac{X}{Z}, \frac{Y}{Z} \right) \in k[X, Y, Z] \]

Now we can find all solutions to a Diophantine Equation, including solutions that occur at “infinity.” Any point in the homogenization that is of the form \((X_0 : Y_0 : Z_0)\) with \( Z_0 = 0 \) is called a point at infinity. All other points are called affine points.
Bezout’s Theorem

If \( f, g \in k[x, y] \) are polynomials of degrees \( d \) and \( e \) respectively, then \( C_f \) and \( C_g \) intersect in at most \( de \) points. Further, \( \hat{C_f} \) and \( \hat{C_g} \) intersect in exactly \( de \) points of \( P^2(k) \), when points are counted with multiplicity. This is used in a classical proof of algebraic geometry, but we will not go into that at this time.

**Frobenius Maps**

Suppose \( F_q \) is a finite field (recall this means that \( q \) must be a prime power) and that \( n \geq 1 \). *The Frobenius Automorphism* is the map

\[
\sigma_{q,n} : F_q^n \rightarrow F_q^n
\]

defined by

\[
\sigma_{q,n}(\alpha) = \alpha^q, \quad \text{for any } \alpha \in F_q^n
\]

**Relative and Absolute Frobenius**

If \( q = p^r \) where \( p \) is prime and \( r \geq 2 \), then

- the map \( \sigma_{q,n} \) is often called the *relative Frobenius*
- the function \( \sigma_{p,n} \) if often called the *absolute Frobenius*

**Composing Frobenius with Itself**

The symbol \( \sigma_{q,n}^j \) represents the map obtained by composing \( \sigma_{q,n} \) with itself \( j \) times. For example:

\[
\sigma_{q,n}^2(\alpha) = \sigma_{q,n}(\sigma_{q,n}(\alpha)).
\]

**Nonsingularity**

*When constructing a code, one of the elements needed is a nonsingular projective plane curve*. A projective plane curve, \( C_f \), is *nonsingular when no singular points exist on it*.

**Singular Points**
Definition 1: A singular point of $C_f$ is a point $(x_0, y_0) \in k \times k$ such that $f(x_0, y_0) = 0$ and $f_x(x_0, y_0) = 0$ and $f_y(x_0, y_0) = 0$.

Definition 2: If $F(X,Y,Z)$ is the homogenization of $f(x,y)$, then $(X_0:Y_0:Z_0)$ is a singular point of $C_f$ if the point is on the curve and:

- $F(X_0:Y_0:Z_0) = F_x(X_0:Y_0:Z_0) = F_y(X_0:Y_0:Z_0) = F_z(X_0:Y_0:Z_0) = 0$.  

**What are $f(x,y)$, $f_x(x,y)$, $f_y(x,y)$?**

- $f_x(x,y)$?
  
  $f_x(x,y)$ is the partial derivative of $f(x,y)$ with respect to $x$.

  Example: Let $f(x,y) = 2x^2y + xy^3 + x^2 + 2y$
  
  Then $f_x(x,y) = 4xy + y^3 + 2x$.

- $f_y(x,y)$?
  
  $f_x(x,y)$ is the partial derivative of $f(x,y)$ with respect to $x$.

  Example: Let $f(x,y) = 2x^2y + xy^3 + x^2 + 2y$
  
  Then $f_x(x,y) = 4xy + y^3 + 2x$.

- $f_y(x,y)$?
  
  $f_y(x,y)$ is the partial derivative of $f(x,y)$ with respect to $y$.

  Example: Let $f(x,y) = 2x^2y + xy^3 + x^2 + 2y$
  
  Then $f_y(x,y) = 2x^2 + 3xy^2 + 2$. 
Genus and the Plücker Formula

A nonsingular curve can be realized as a torus-like object with one or more holes in $\mathbb{R}^3$. This torus has a certain number of holes which is called the topological genus ($g$). The genus is given by the formula $g = (d-1)(d-2)/2$ where $d$ is the degree of the polynomial which makes the curve nonsingular. This formula is called the Plücker Formula.

Points, Functions, and Divisors on Curves

Definition: Let $k$ be a field, and let $C$ be the projective plane curve defined by $F = 0$, where $F = F(X,Y,Z) \in k[X,Y,Z]$ is a homogenous polynomial. For any field $K$ containing $k$, we define a $K$-rational point on $C$ to be a point $(X_0 : Y_0 : Z_0) \in \mathbb{P}^2(K)$ such that $F(X_0, Y_0, Z_0) = 0$. The set of all $K$-rational points on $C$ is denoted $C(K)$. Elements of $C(k)$ are called points of degree one or simply rational points.

Definition: Let $C$ be a nonsingular projective plane curve. A point of degree $n$ on $C$ over $F_q$ is a set $P = \{P_0, \ldots, P_{n-1}\}$ of $n$ distinct points in $C(F_q^n)$ such that $P_i = \mathcal{O}_{q^n}(P_0)$ for $i = 1, \ldots, n-1$.

Definition: A divisor $D$ on $X$, a nonsingular projective plane curve, over $F_q$ is an element of the free abelian group on the set of points on $X$

$$D = \sum n_QQ$$

over $F_q$. Every divisor is of the form:

where the $n_Q$ are integers and each $Q$ is a point on $X$. If $n_Q \geq 0$ for all $Q$, $D$ is effective and $D \geq 0$.

$$D = \sum n_{Q} \deg Q$$

The degree of $D$ is:

The support of $D$ is $\text{supp} D = \{Q \mid n_Q \neq 0\}$.

Definition: Let $F(X,Y,Z)$ be the polynomial which defines the nonsingular projective plane curve $C$ over the field $F_q$. The field of rational functions on $C$ is
\[ F_q(C) := \left\{ \frac{g(X,Y,Z)}{h(X,Y,Z)} : g,h \in \mathbb{F}_q[X,Y,Z] \text{ are homogeneous of the same degree} \right\} / \sim \]

where \( g/h \sim g'/h' \) if and only if \( gh' - g'h \in \mathbb{F}_q \) \( F_q[X,Y,Z] \).

Definition: Let \( C \) be a curve defined over \( \mathbb{F}_q \) and let \( f := g/h \in \mathbb{F}_q(C) \). The divisor of \( f \) is defined to be \( \text{div}(f) := \Sigma P - \Sigma Q \), where \( \Sigma P \) is the intersection divisor \( C \cap C_g \) and \( \Sigma Q \) is the intersection divisor \( C \cap C_h \).

Definition: Let \( D \) be a divisor on the nonsingular projective plane curve \( C \) defined over the field \( \mathbb{F}_q \). Then the space of rational functions associated to \( D \) is

\[ L(D) := \{ f \in \mathbb{F}_q(C) \mid \text{div}(f) + D \succeq 0 \} \cup \{0\} \]

**Riemann–Roch Theorem**

Let \( C \) be a nonsingular projective plane curve of genus \( g \) defined over the field \( \mathbb{F}_q \) and let \( D \) be a divisor on \( X \). Then

\[ \text{dim } L(D) \geq \deg D + 1 - g. \]

Further, if \( \deg D > 2g - 2 \), then

\[ \text{dim } L(D) = \deg D + 1 - g. \]
Algebraic Geometric Reed–Solomon Code

Goppa’s Construction

Goppa’s construction is extended from the Reed–Solomon Code formula:

\[ \text{RS}(k,q) := \{ ( f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_{q-1}) ) | f \in L_{k-1} \} \]

Goppa’s idea behind this construction was to generalize the formula. This new formula is:

\[ \text{C}(X,P,D) := \{ ( f (P_1), \ldots, (P_n) ) | f \in L(D) \} \]

Definitions for: \( \text{C}(X,P,D) := \{ ( f (P_1), \ldots, (P_n) ) | f \in L(D) \} \)

X: X is a projective nonsingular plane curve over \( F_q \), a finite field with \( q \) number of elements.
P: \( P = \{P_1, \ldots, P_n\} \subset X(F_q) \)
i.e. \( P \) is the set of \( n \) distinct \( F_q \)-rational points on \( X \).
D: \( D \) is a divisor on \( X \).
L(D): \( L(D) \) is the space of rational functions associated with the defined divisor \( D \) over \( X \).

Goppa’s Construction – History

In 1981, Goppa derived a class of linear codes from algebraic curves over finite fields which

- are quite general as codes
- have parameters circumscribed by the Riemann–Roch theorem

The discovery of these codes also gave renewed stimulus to investigations on the number of rational points on an algebraic curve for a particular genus as well as to asymptotic values of the ratio of the number of points to the genus.
Goppa – The Problem

The three most important parameters of a linear code over the finite field are:

1. the length $n$ which gives the speed of transmission
2. the dimension $k$ which gives the number of words in the code
3. and the minimum distance $d$ which gives the number of errors that can be corrected.

“Good” Codes

“Good” codes have the following properties:

• a large information rate $R = k/n$
• and a large relative distance $\delta = d/n$

The relation between $R$ and $\delta$ as $n$ gets large is given by the Gilbert-Varshamov bound. Good codes can be constructed from an algebraic curve of genus $g$, and particular examples show that the G-V bound is not best possible. This brings to the fore the problem of determining the limit $\beta$ of $n / g$ for a sequence of curves.

Conclusion

In the past two years the goal of finding explicit codes that reach the limits predicted by Shannon’s original work has been achieved. The constructions require techniques from a surprisingly wide range of pure mathematics: linear algebra, the theory of fields and algebraic geometry all play a vital role. Not only has coding theory helped to solve problems of vital importance in the world outside mathematics, it has enriched other branches of mathematics, with new problems as well as new solutions.
Appendix A

Codes and Curves by Judy L. Walker
- American Mathematical Society, 2000

Coding theory: the first 50 years