1
Introduction

Since the early 2000s we have seen the basic notions of coding theory expand beyond the role of error correction and algebraic coding theory. The purpose of this volume is to provide a brief introduction to a few of the directions that have been taken as a platform for further reading. Although the approach is to be descriptive with few proofs, there are parts which are unavoidably technical and more challenging.

It was mentioned in the Preface that the prerequisite for this work is a basic course on algebraic coding theory and information theory. In fact only a few aspects of finite fields, particularly certain properties of polynomials over finite fields, Reed–Solomon codes and Reed–Muller codes and their generalizations are considered to provide a common basis and establish the notation to be used. The trace function on finite fields makes a few appearances in the chapters and its basic properties are noted. Most of the information will be familiar and stated informally without proof. A few of the chapters use notions of information theory and discrete memoryless channels and the background required for these topics is also briefly reviewed in Section 1.2. The final Section 1.3 gives a brief description of the chapters that follow.

1.1 Notes on Finite Fields and Coding Theory

Elements of Finite Fields

A few basic notions from integers and polynomials will be useful in several of the chapters as well as considering properties of finite fields. The *greatest common divisor* (gcd) of two integers or polynomials over a field will be a staple of many computations needed in several of the chapters. Abstractly, an integral domain is a commutative ring in which the product of two nonzero elements is nonzero, sometimes stated as a commutative ring with identity

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

which has no zero divisors (i.e., two nonzero elements a,b such that $ab = 0$). A *Euclidean domain* is an integral domain which is furnished with a norm function, in which the division of an element by another with a remainder of lower degree can be formulated. Equivalently the Euclidean algorithm (EA) can be formulated in a Euclidean domain.

Recall that the gcd of two integers $a,b \in \mathbb{Z}$ is the largest integer d that divides both a and b. Let $\mathbb{F}$ be a field and denote by $\mathbb{F}[x]$ the ring of polynomials over $\mathbb{F}$ with coefficients from $\mathbb{F}$. The gcd of two polynomials $a(x), b(x) \in \mathbb{F}[x]$ is the monic polynomial (coefficient of the highest power of x is unity) of the greatest degree, $d(x)$, that divides both polynomials. The EA for polynomials is an algorithm that produces the gcd of polynomials $a(x)$ and $b(x)$ (the one for integers is similar) by finding polynomials $u(x)$ and $v(x)$ such that

$$d(x) = u(x)a(x) + v(x)b(x). \tag{1.1}$$

It is briefly described as follows. Suppose without loss of generality that $\deg b(x) < \deg a(x)$ and consider the sequence of polynomial division steps producing quotient and remainder polynomials:

$$\begin{aligned}
a(x) &= q_1(x)b(x) + r_1(x), &&\deg r_1 < \deg b\\
b(x) &= q_2(x)r_1(x) + r_2(x), &&\deg r_2 < \deg r_1\\
r_1(x) &= q_3(x)r_2(x) + r_3(x), &&\deg r_3 < \deg r_2\\
\vdots &\qquad \vdots\\
r_k(x) &= q_{k+2}(x)r_{k+1}(x) + r_{k+2}(x), &&\deg r_{k+2} < \deg r_{k+1}\\
r_{k+1}(x) &= q_{k+3}(x)r_{k+2}(x), &&d(x) = r_{k+2}(x).
\end{aligned}$$

That $d(x)$, the last nonzero remainder, is the required gcd is established by tracing back divisibility conditions. Furthermore, tracing back shows how two polynomials $u(x)$ and $v(x)$ are found so that Equation 1.1 holds.

A similar argument holds for integers. The gcd is denoted (a,b) or $(a(x),b(x))$ for integers and polynomials, respectively. If the gcd of two integers or polynomials is unity, they are referred to as being *relatively prime* and denoted $(a,b) = 1$ or $(a(x),b(x)) = 1$.

If the prime factorization of n is

$$n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}, \quad p_1, p_2, \ldots, p_k \text{ distinct primes,}$$

then the number of integers less than n that are relatively prime to n is given by the *Euler Totient function* $\phi(n)$ where

$$\phi(n) = \prod_{i=1}^{k} p_i^{e_i - 1}(p_i - 1). \tag{1.2}$$

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

A *field* is a commutative ring with identity in which elements have additive inverses (0 denotes the additive identity) and nonzero elements have multiplicative inverses (1 denotes the multiplicative identity). It may also be viewed as an integral domain in which the nonzero elements form a multiplicative group.

A finite field is a field with a finite number of elements. For a finite field, there is a smallest integer c such that each nonzero element of the field added to itself a total of c times yields 0. Such an integer is called the *characteristic of the field*. If c is not finite, the field is said to have characteristic 0. Notice that in a finite field of characteristic 2, addition and subtraction are identical in that $1 + 1 = 0$. Denote the set of nonzero elements of the field $\mathbb{F}$ by $\mathbb{F}^*$.

Denote by $\mathbb{Z}_n$ the set of integers modulo n, $\mathbb{Z}_n = \{0, 1, 2, \ldots, n-1\}$. It is a finite field iff n is a prime p, since if $n = ab$, $a, b \in \mathbb{Z}$ is composite, then it has zero divisors and hence is not a field. Thus the characteristic of any finite field is a prime and the symbol p is reserved for some arbitrary prime integer. In a finite field $\mathbb{Z}_p$, arithmetic is modulo p. If $a \in \mathbb{Z}_p, a \neq 0$, the inverse of a can be found by applying the EA to $a < p$ and p which yields two integers u, $v \in \mathbb{Z}$ such that

$$ua + vp = 1 \quad \text{in } \mathbb{Z}$$

and so $ua + vp \pmod{p} \equiv ua \equiv 1 \pmod{p}$ and $a^{-1} \equiv u \pmod{p}$. The field will be denoted $\mathbb{F}_p$. In any finite field there is a smallest subfield, a set of elements containing and generated by the unit element 1, referred to as the *prime subfield*, which will be $\mathbb{F}_p$ for some prime p.

Central to the notion of finite fields and their applications is the role of polynomials over the field. Denote the ring of polynomials in the indeterminate x over a field $\mathbb{F}$ by $\mathbb{F}[x]$ and note that it is a Euclidean domain (although the ring of polynomials with two variables $\mathbb{F}[x, y]$ is not). A polynomial $f(x) = f_n x^n + f_{n-1}x^{n-1} + \cdots + f_1 x + f_0 \in \mathbb{F}[x], f_i \in \mathbb{F}$ is monic if the leading coefficient f_n is unity.

A polynomial $f(x) \in \mathbb{F}[x]$ is called reducible if it can be expressed as the product of two nonconstant polynomials and irreducible if it is not the product of two nonconstant polynomials, i.e., there do not exist two nonconstant polynomials $a(x), b(x) \in \mathbb{F}[x]$ such that $f(x) = a(x)b(x)$. Let $f(x)$ be a monic irreducible polynomial over the finite field $\mathbb{F}_p$ and consider the set of p^n polynomials taken modulo $f(x)$ which will be denoted

$$\mathbb{F}_p[x]/\langle f(x)\rangle = \left\{a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1 x + a_0, a_i \in \mathbb{F}_p\right\}$$

where $\langle f(x)\rangle$ is the ideal in $\mathbb{F}_p$ generated by $f(x)$. Addition of two polynomials is obvious and multiplication of two polynomials is taken modulo the

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

irreducible polynomial $f(x)$, i.e., the remainder after division by $f(x)$. The inverse of a nonzero polynomial $a(x) \in \mathbb{F}_p[x]/\langle f(x)\rangle$ is found via the EA as before. That is since by definition $(a(x), f(x)) = 1$ there exist polynomials $u(x), v(x)$ such that

$$u(x)a(x) + v(x)f(x) = 1$$

and the inverse of $a(x) \in \mathbb{F}_p[x]/\langle f(x)\rangle$ is $u(x)$. Algebraically this structure might be described as the factor field of the ring $\mathbb{F}_p[x]$ modulo the maximal ideal $\langle f(x)\rangle$.

It follows the set $\mathbb{F}_p[x]/\langle f(x)\rangle$ forms a finite field with p^n elements. It is conventional to denote $q = p^n$ and the field of p^n elements as either $\mathbb{F}_{p^n}$ or $\mathbb{F}_q$. Every finite field can be shown to have a number of elements of the form $q = p^n$ for some prime p and positive integer n and that any two finite fields of the same order are isomorphic. It will be noted that an irreducible polynomial of degree n will always exist (see Equation 1.4) and so all finite fields can be constructed in this manner.

In general, suppose $q = p^m$ and let $f(x)$ be a monic irreducible polynomial over $\mathbb{F}_q$ of degree m (which will be shown to always exist). The set of q^m polynomials over $\mathbb{F}_q$ of degree less than m with multiplication modulo $f(x)$ will then be a finite field with q^m elements and designated $\mathbb{F}_{q^m}$. For future reference denote the set of polynomials of degree less than m by ${\mathbb{F}_q}^{<m}[x]$ and those less than or equal by ${\mathbb{F}_q}^{\leq m}[x]$. Since it involves no more effort, this general finite field $\mathbb{F}_{q^m}$ will be examined for basic properties. The subset $\mathbb{F}_q \subseteq \mathbb{F}_{q^m}$ is a field, i.e., a subset that has all the properties of a field, a subfield of $\mathbb{F}_{q^m}$.

The remainder of the subsection contains a brief discussion of the structure of finite fields and polynomials usually found in a first course of coding theory.

It is straightforward to show that over any field $\mathbb{F}$ $(x^m - 1)$ divides $(x^n - 1)$ iff m divides n, written as

$$(x^m - 1)\Big|(x^n - 1) \text{ iff } m \mid n.$$

Further, for any prime p,

$$(p^m - 1)\Big|(p^n - 1) \text{ iff } m \mid n.$$

The multiplicative group of a finite field, $\mathbb{F}_{q^m}^*$, can be shown to be cyclic (generated by a single element). The order of a nonzero element α in a field $\mathbb{F}$ is the smallest positive integer ℓ such that $\alpha^\ell = 1$, denoted as ord $(\alpha) = \ell$ and referred to as the order of α. Similarly if ℓ is the smallest integer such that the polynomial $f(x) \mid \big(x^\ell - 1\big)$, the polynomial is said to have order ℓ over the

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

understood field. The order of an irreducible polynomial is also the order of its zeros.

If β has order ℓ, then β^i has order $\ell/(i,\ell)$. Similarly if β has order ℓ and γ has order κ and $(\ell,\kappa)=1$, then the order of $\beta\gamma$ is $\ell\kappa$.

An element $\alpha \in \mathbb{F}_{q^m}$ of maximum order q^m-1 is called a *primitive element*. If α is primitive, then α^i is also primitive iff $(i,q^m-1)=1$ and there are $\phi(q^m-1)$ primitive elements in $\mathbb{F}_{q^m}$.

A note on the representation of finite fields is in order. The order of an irreducible polynomial $f(x)$ over $\mathbb{F}_q$ of degree k can be determined by successively dividing the polynomial (x^n-1) by $f(x)$ over $\mathbb{F}_q$ as n increases. If the smallest such n is q^k-1, the polynomial is primitive. To effect the division, arithmetic in the field $\mathbb{F}_q$ is needed. If $f(x)$ is primitive of degree k over $\mathbb{F}_q$, one could then take the field as the elements

$$\mathbb{F}_{q^k}=\left\{0,1,x,x^2,\ldots,x^{q^k-2}\right\}.$$

By definition the elements are distinct. Each of these elements could be taken modulo $f(x)$ (which is zero in the field) which would result in the field elements being all polynomials over $\mathbb{F}_q$ of degree less than k. Multiplication in this field would be polynomials taken modulo $f(x)$. The field element x is a primitive element. While this is a valid presentation, it is also common to identify the element x by an element α with the statement "let α be a zero of the primitive polynomial $f(x)$ of degree k over $\mathbb{F}_q$." The two views are equivalent.

There are $\phi(q^k-1)$ primitive elements in $\mathbb{F}_{q^k}$ and since the degree of an irreducible polynomial with one of these primitive elements as a zero is necessarily k, there are exactly $\phi(q^k-1)/k$ primitive polynomials of degree k over $\mathbb{F}_q$.

Suppose $f(x)$ is an irreducible nonprimitive polynomial of degree k over $\mathbb{F}_q$. Suppose it is of order $n<q^k-1$, i.e., $f(x)\mid(x^n-1)$. One can define the field $\mathbb{F}_{q^k}$ as the set of polynomials of degree less than k

$$\mathbb{F}_{q^k}=\left\{a_{k-1}x^{k-1}+a_{k-2}x^{k-2}+\cdots+a_1x+a_0,\ a_i\mathbb{F}_q\right\}$$

with multiplication modulo $f(x)$. The element x is not primitive if $n<(q^k-1)$ but is an element of order $n, n\mid q^k-1$ (although there are still $\phi(q^k-1)$ primitive elements in the field).

Let $\alpha\in\mathbb{F}_{q^m}$ be an element of maximum order (q^m-1) (i.e., primitive) and denote the multiplicative group of nonzero elements as

$$\mathbb{F}_{q^m}^*=\langle\alpha\rangle=\left\{1,\alpha,\alpha^2,\ldots,\alpha^{q^m-2}\right\}.$$

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

Let $\beta \in \mathbb{F}_{q^m}^*$ be an element of order ℓ which generates a cyclic multiplicative subgroup of $\mathbb{F}_{q^m}^*$ of order ℓ and for such a subgroup $\ell \mid (q^m - 1)$. The order of any nonzero element in $\mathbb{F}_{q^m}$ divides $(q^m - 1)$. Thus

$$x^{q^m} - x = \prod_{\beta \in \mathbb{F}_{q^m}} (x - \beta), \quad x^{q^m - 1} - 1 = \prod_{\beta \in \mathbb{F}_{q^m}^*} (x - \beta) \tag{1.3}$$

is a convenient factorization (over $\mathbb{F}_{q^m}$).

Suppose $\mathbb{F}_{q^m}$ has a subfield $\mathbb{F}_{q^k}$ – a subset of elements which is itself a field. The number of nonzero elements in $\mathbb{F}_{q^k}$ is $(q^k - 1)$ and this set must form a multiplicative subgroup of $\mathbb{F}_{q^m}^*$ and hence $(q^k - 1) \mid (q^m - 1)$ and this implies that $k \mid m$ and that $\mathbb{F}_{q^k}$ is a subfield of $\mathbb{F}_{q^m}$ iff $k \mid m$. Suppose $\mathbb{F}_{q^k}$ is a subfield of $\mathbb{F}_{q^m}$. Then

$$\beta \in \mathbb{F}_{q^m} \text{ is in } \mathbb{F}_{q^k} \text{ iff } \beta^{q^k} = \beta$$

and $\beta = \alpha^j \in \mathbb{F}_{q^m}$ is a zero of the monic irreducible polynomial $f(x)$ of degree k over $\mathbb{F}_q$. Thus

$$f(x) = x^k + f_{k-1}x^{k-1} + \cdots + f_1 x + f_0, \quad f_i \in \mathbb{F}_q,\ i = 0, 1, 2, \ldots, k-1$$

and $f(\alpha^j) = 0$. Notice that

$$\begin{aligned} f(x)^q &= \left(x^k + f_{k-1}x^{k-1} + \cdots + f_1 x + f_0\right)^q \\ &= x^{kq} + f_{k-1}^q x^{q(k-1)} + \cdots + f_1^q x^q + f_0^q \\ &= x^{kq} + f_{k-1}x^{q(k-1)} + \cdots + f_1 x^q + f_0, \text{ as } f_i^q = f_I \text{ for } f_i \in \mathbb{F}_q \\ &= f(x^q) \end{aligned}$$

and since $\beta = \alpha^j$ is a zero of $f(x)$ so is β^q. Suppose ℓ is the smallest integer such that $\beta^{q^\ell} = \beta$ (since the field is finite there must be such an ℓ) and let

$$C_j = \left\{\alpha^j = \beta, \beta^q, \beta^{q^2}, \ldots, \beta^{q^{\ell-1}}\right\}$$

referred to as the conjugacy class of β. Consider the polynomial

$$g(x) = \prod_{i=0}^{\ell-1} \left(x - \beta^{q^j}\right)$$

and note that

$$g(x)^q = \prod_{i=0}^{\ell-1} \left(x - \beta^{q^j}\right)^q = \prod_{i=0}^{\ell-1} \left(x^q - \beta^{q^{j+1}}\right) = \prod_{i=0}^{\ell-1} \left(x^q - \beta^{q^j}\right) = g(x^q)$$

and, as above, $g(x)$ has coefficients in $\mathbb{F}_q$, i.e., $g(x) \in \mathbb{F}_q[x]$. It follows that $g(x)$ must divide $f(x)$ and since $f(x)$ was assumed monic and irreducible it must be that $g(x) = f(x)$. Thus if one zero of the irreducible $f(x)$ is in $\mathbb{F}_{q^m}$,

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

all are. Each conjugacy class of the finite field corresponds to an irreducible polynomial over $\mathbb{F}_q$.

By similar reasoning it can be shown that if $f(x)$ is irreducible of degree k over $\mathbb{F}_q$, then $f(x) \mid (x^{q^m} - x)$ iff $k \mid m$. It follows that the polynomial $x^{q^m} - x$ is the product of all monic irreducible polynomials whose degrees divide m. Thus

$$x^{q^m} - x = \prod_{\substack{f(x) \text{ irreducible} \\ \text{over } \mathbb{F}_q \\ \text{degree} f(x)=k|m}} f(x).$$

This allows a convenient enumeration of the polynomials. If $N_q(m)$ is the number of monic irreducible polynomials of degree m over $\mathbb{F}_q$, then by the above equation

$$q^m = \sum_{k|m} k N_q(k)$$

which can be inverted using standard combinatorial techniques as

$$N_q(m) = \frac{1}{m} \sum_{k|m} \mu\left(\frac{m}{k}\right) q^k \tag{1.4}$$

where $\mu(n)$ is the Möbius function equal to 1 if $n = 1$, $(-1)^s$ if n is the product of s distinct primes and zero otherwise. It can be shown that $N_q(k)$ is at least one for all prime powers q and all positive integers k. Thus irreducible polynomials of degree k over a field of order q exist for all allowable parameters and hence finite fields exist for all allowable parameter sets.

Consider the following example.

Example 1.1 Consider the field extension $\mathbb{F}_{2^6}$ over the base field $\mathbb{F}_2$. The polynomial $x^{2^6} - x$ factors into all irreducible polynomials of degree dividing 6, i.e., those of degrees $1, 2, 3$ and 6. From the previous formula

$$N_2(1) = 2, \; N_2(2) = 1, \; N_2(3) = 2, \; N_2(6) = 9.$$

For a primitive element α the conjugacy classes of $\mathbb{F}_{2^6}$ over $\mathbb{F}_2$ are (obtained by raising elements by successive powers of 2 mod 63, with tentative polynomials associated with the classes designated):

$$\begin{array}{l}
\alpha^1, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}, \alpha^{32} \approx f_1(x) \\
\alpha^3, \alpha^6, \alpha^{12}, \alpha^{24}, \alpha^{48}, \alpha^{33} \approx f_2(x) \\
\alpha^5, \alpha^{10}, \alpha^{20}, \alpha^{40}, \alpha^{17}, \alpha^{34} \approx f_3(x) \\
\alpha^7, \alpha^{14}, \alpha^{28}, \alpha^{56}, \alpha^{49}, \alpha^{35} \approx f_4(x) \\
\alpha^9, \alpha^{18}, \alpha^{36} \approx f_5(x) \\
\alpha^{11}, \alpha^{22}, \alpha^{44}, \alpha^{25}, \alpha^{50}, \alpha^{37} \approx f_6(x)
\end{array}$$

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

$$\begin{aligned}
&\alpha^{13},\alpha^{26},\alpha^{52},\alpha^{41},\alpha^{19},\alpha^{38} \approx f_7(x)\\
&\alpha^{15},\alpha^{30},\alpha^{60},\alpha^{57},\alpha^{51},\alpha^{39} \approx f_8(x)\\
&\alpha^{21},\alpha^{42} \approx f_9(x)\\
&\alpha^{23},\alpha^{46},\alpha^{29},\alpha^{58},\alpha^{53},\alpha^{43} \approx f_{10}(x)\\
&\alpha^{27},\alpha^{54},\alpha^{45} \approx f_{11}(x)\\
&\alpha^{31},\alpha^{62},\alpha^{61},\alpha^{59},\alpha^{55},\alpha^{47} \approx f_{12}(x).
\end{aligned}$$

By the above discussion a set with ℓ integers corresponds to an irreducible polynomial over $\mathbb{F}_2$ of degree ℓ. Further, the order of the polynomial is the order of the conjugates in the corresponding conjugacy class.

Notice there are $\phi(63) = \phi(9 \cdot 7) = 3 \cdot 2 \cdot 6 = 36$ primitive elements in $\mathbb{F}_{2^6}$ and hence there are $36/6 = 6$ primitive polynomials of degree 6 over $\mathbb{F}_2$. If α is chosen as a zero of the primitive polynomial $f_1(x) = x^6 + x + 1$, then the correspondence of the above conjugacy classes with irreducible polynomials is

Poly. No.	Polynomial	Order
$f_1(x)$	x^6+x+1	63
$f_2(x)$	$x^6+x^4+x^3+x^2+x+1$	21
$f_3(x)$	$x^6+x^5+x^2+x+1$	63
$f_4(x)$	x^6+x^3+1	9
$f_5(x)$	x^3+x^2+1	7
$f_6(x)$	$x^6+x^5+x^3+x^2+1$	63
$f_7(x)$	$x^6+x^4+x^3+x+1$	63
$f_8(x)$	$x^6+x^5+x^4+x^2+1$	21
$f_9(x)$	x^2+x+1	3
$f_{10}(x)$	$x^6+x^5+x^4+x+1$	63
$f_{11}(x)$	x^3+x+1	7
$f_{12}(x)$	x^6+x^5+1	63

The other three irreducible polynomials of degree 6 are of orders 21 (two of them, $f_2(x)$ and $f_8(x)$) and nine ($f_4(x)$). The primitive element α in the above conjugacy classes could have been chosen as a zero of any of the primitive polynomials. The choice determines arithmetic in $\mathbb{F}_{2^6}$ but all choices will lead to isomorphic representations. Different choices would have resulted in different associations between conjugacy classes and polynomials.

Not included in the above table is the conjugacy class $\{\alpha^{63}\}$ which corresponds to the polynomial $x + 1$ and the class $\{0\}$ which corresponds to the polynomial x. The product of all these polynomials is $x^{2^6} - x$.

The notion of a *minimal polynomial* of a field element is of importance for coding. The *minimal polynomial* $m_\beta(x)$ of an element $\beta \in \mathbb{F}_{q^n}$ over $\mathbb{F}_q$ is that

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

the monic irreducible polynomial of least degree that has β as a zero. From the above discussion, every element in a conjugacy class has the same minimal polynomial.

Further notions of finite fields that will be required include that of a *polynomial basis* of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$ which is one of a form $\{1,\alpha,\alpha^2,\ldots,\alpha^{n-1}\}$ for some $\alpha \in \mathbb{F}_{q^n}$ for which the elements are linearly independent over $\mathbb{F}_q$. A basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$ of the form $\{\alpha,\alpha^q,\alpha^{q^2},\ldots,\alpha^{q^{n-1}}\}$ is called a *normal basis* and such bases always exist. In the case that $\alpha \in \mathbb{F}_{q^n}$ is primitive (of order $q^n - 1$) it is called a primitive normal basis.

The Trace Function of Finite Fields

Further properties of the trace function that are usually discussed in a first course on coding will prove useful at several points in the chapters. Let $\mathbb{F}_{q^n}$ be an extension field of order n over $\mathbb{F}_q$. For an element $\alpha \in \mathbb{F}_{q^n}$ the *trace function* of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$ is defined as

$$\mathrm{Tr}_{q^n|q}(\alpha) = \sum_{i=0}^{n-1} \alpha^{q^i}.$$

The function enjoys many properties, most notably that [8]

(i) $\mathrm{Tr}_{q^n|q}(\alpha+\beta) = \mathrm{Tr}_{q^n|q}(\alpha) + \mathrm{Tr}_{q^n|q}(\beta), \quad \alpha,\beta \in \mathbb{F}_{q^n}$
(ii) $\mathrm{Tr}_{q^n|q}(a\alpha) = a\mathrm{Tr}_{q^n|q}(\alpha),\ a \in \mathbb{F}_q, \alpha \in \mathbb{F}_{q^n}$
(iii) $\mathrm{Tr}_{q^n|q}(a) = na,\ a \in \mathbb{F}_q$
(iv) $\mathrm{Tr}_{q^n|q}$ is an onto map.

To show property (iv), which the trace map is onto (i.e., codomain is $\mathbb{F}_q$), it is sufficient to show that there exists an element α of $\mathbb{F}_{q^n}$ for which $\mathrm{Tr}_{q^n|q}(\alpha) \neq 0$ since if $\mathrm{Tr}_{q^n|q}(\alpha) = b \neq 0, b \in \mathbb{F}_q$, then (property ii) $\mathrm{Tr}_{q^n|q}(b^{-1}\alpha) = 1$ and hence all elements of $\mathbb{F}_q$ are mapped onto. Consider the polynomial equation

$$x^{q^{n-1}} + x^{q^{n-2}} + \cdots + x = 0$$

that can have at most q^{n-1} solutions in $\mathbb{F}_{q^n}$. Hence there must exist elements of $\beta \in \mathbb{F}_{q^n}$ for which $\mathrm{Tr}_{q^n|q}(\beta) \neq 0$. An easy argument shows that in fact exactly q^{n-1} elements of $\mathbb{F}_{q^n}$ have a trace of $a \in \mathbb{F}_q$ for each element of $\mathbb{F}_q$.

Notice that it also follows from these observations that

$$x^{q^n} - x = \prod_{a \in \mathbb{F}_q} \left(x^{q^{n-1}} + x^{q^{n-2}} + \cdots + x - a\right)$$

since each element of $\mathbb{F}_{q^n}$ is a zero of the LHS and exactly one term of the RHS.

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

Also, suppose [8] $L(\cdot)$ is a linear function from $\mathbb{F}_{q^n}$ to $\mathbb{F}_q$ in the sense that for all $a_1, a_2 \in \mathbb{F}_q$ and all $\alpha_1, \alpha_2 \in \mathbb{F}_{q^n}$

$$L(a_1\alpha_1 + a_2\alpha_2) = a_1 L(\alpha_1) + a_2 L(\alpha_2).$$

Then $L(\cdot)$ must be of the form

$$L(\alpha) = \mathrm{Tr}_{q^n|q}(\beta\alpha) \overset{\Delta}{=} L_\beta(\alpha)$$

for some β. Thus the set of such linear functions is precisely the set

$$L_\beta(\cdot), \ \ \beta \in \mathbb{F}_{q^n}$$

and these are distinct functions for distinct β.

A useful property of the trace function ([8], lemma 3.51, [11], lemma 9.3) is that if $u_1, u_2, \ldots, u_n$ is a basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$ and if

$$\mathrm{Tr}_{q^n|q}(\alpha u_i) = 0 \text{ for } i = 1, 2, \ldots, n, \quad \alpha \in \mathbb{F}_{q^n},$$

then $\alpha = 0$. Equivalently if for $\alpha \in \mathbb{F}_{q^n}$

$$\mathrm{Tr}_{q^n|q}(\alpha u) = 0 \quad \forall u \in \mathbb{F}_{q^n}, \tag{1.5}$$

then $\alpha = 0$. This follows from the trace map being onto. It will prove a useful property in the sequel. It also follows from the fact that

$$\begin{bmatrix} u_1 & u_1^q & \cdots & u_1^{q^{n-1}} \\ u_2 & u_2^q & \cdots & u_2^{q^{n-1}} \\ \vdots & \vdots & \vdots & \vdots \\ u_n & u_n^q & \cdots & u_n^{q^{n-1}} \end{bmatrix}$$

is nonsingular iff $u_1, u_2, \ldots, u_n \in \mathbb{F}_{q^n}$ are linearly independent over $\mathbb{F}_q$. A formula for the determinant of this matrix is given in [8].

If $\mu = \{\mu_1, \mu_2, \ldots, \mu_n\}$ is a basis of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$, then a basis $\nu = \{\nu_1, \nu_2, \ldots, \nu_n\}$ is called a *trace dual basis* if

$$\mathrm{Tr}_{q^n|q}(\mu_i \nu_j) = \delta_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

and for a given basis a unique dual basis exists. It is noted that if $\mu = \{\mu_1, \ldots, \mu_n\}$ is a dual basis for the basis $\{\nu_1, \ldots, \nu_n\}$, then given

$$y = \sum_{i=1}^{n} a_i \, \mu_i \quad \text{then} \quad y = \sum_{i=1}^{n} \mathrm{Tr}_{q^n|q}(y\nu_i) \, \mu_i, \quad a_i \in \mathbb{F}_q. \tag{1.6}$$

Thus an element $y \in \mathbb{F}_{q^n}$ can be represented in the basis μ, by the traces $\mathrm{Tr}_{q^n|q}(y\nu_j), \ j = 1, 2, \ldots, n$.

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

It can be shown that for a given normal basis, the dual basis is also normal. A convenient reference for such material is [8, 12].

Elements of Coding Theory

A few comments on BCH, Reed–Solomon (RS), Generalized Reed–Solomon (GRS), Reed–Muller (RM) and Generalized Reed–Muller (GRM) codes are noted. Recall that a cyclic code of length n and dimension k and minimum distance d over $\mathbb{F}_q$, designated as an $(n,k,d)_q$ code, is defined by a polynomial $g(x) \in \mathbb{F}_q[x]$ of degree $(n-k)$, $g(x) \mid (x^n - 1)$ or alternatively as a principal ideal $\langle g(x) \rangle$ in the factor ring $\mathcal{R} = \mathbb{F}_q[x]/\langle x^n - 1\rangle$.

Consider a BCH code of length $n \mid (q^m - 1)$ over $\mathbb{F}_q$. Let β be a primitive n-th root of unity (an element of order exactly n). Let

$$g(x) = \operatorname{lcm}\left\{m_\beta(x), m_{\beta^2}(x), \ldots, m_{\beta^{2t}}(x)\right\}$$

be the minimum degree monic polynomial with the sequence $\beta, \beta^2, \ldots, \beta^{2t}$ of $2t$ elements as zeros (among other elements as zeros). Define the BCH code with length n designed distance $2t+1$ over $\mathbb{F}_q$ as the cyclic code $C = \langle g(x) \rangle$ or equivalently as the code with null space over $\mathbb{F}_q$ of the parity-check matrix

$$H = \begin{bmatrix} 1 & \beta & \beta^2 & \cdots & \beta^{(n-1)} \\ 1 & \beta^2 & \beta^4 & \cdots & \beta^{2(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \beta^{2t} & \beta^{2(2t)} & \cdots & \beta^{2t(n-1)} \end{bmatrix}.$$

That the minimum distance bound of this code, $d = 2t+1$, follows since any $2t \times 2t$ submatrix of H is a Vandermonde matrix and is nonsingular since the elements of the first row are distinct.

A cyclic Reed–Solomon $(n,k,d = n-k+1)_q$ code can be generated by choosing a generator polynomial of the form

$$g(x) = \prod_{i=1}^{n-k} (x - \alpha^i), \ \alpha \in \mathbb{F}_q, \ \alpha \text{ primitive of order } n.$$

That the code has a minimum distance $d = n-k+1$ follows easily from the above discussion.

A standard simple construction of Reed–Solomon codes over a finite field $\mathbb{F}_q$ of length n that will be of use in this volume is as follows. Let $\boldsymbol{u} = \{u_1, u_2, \ldots, u_n\}$ be a set, referred to as the *evaluation set* (and viewed as a set rather than a vector – we use boldface lowercase letters for both sets and

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

vectors) of $n \leq q$ distinct evaluation elements of $\mathbb{F}_q$. As noted, $\mathbb{F}_q^{<k}[x]$ is the set of polynomials over $\mathbb{F}_q$ of degree less than k. Then another incarnation of a Reed–Solomon code can be taken as

$$RS_{n,k}(\boldsymbol{u},q) = \left\{ \boldsymbol{c}_f = (f(u_1), f(u_2), \ldots, f(u_n)), f \in \mathbb{F}_q^{<k}[x] \right\}$$

where $\boldsymbol{c_f}$ is the codeword associated with the polynomial $\boldsymbol{f}$. That this is an $(n,k,d = n-k+1)_q$ code follows readily from the fact that a polynomial of degree less than k over $\mathbb{F}_q$ can have at most $k-1$ zeros. As the code satisfies the Singleton bound $d \leq n-k+1$ with equality it is referred to as *maximum distance separable* (MDS) code and the dual of such a code is also MDS. Of course the construction is valid for any finite field, e.g., $\mathbb{F}_{q^\ell}$.

The dual of an RS code is generally not an RS code.

A slight but useful generalization of this code is the *Generalized Reed–Solomon* (GRS) code denoted as $GRS_{n,k}(\boldsymbol{u},\boldsymbol{v},q)$, where $\boldsymbol{u}$ is the evaluation set of distinct field nonzero elements as above and $\boldsymbol{v} = \{v_1, v_2, \ldots, v_n\}$, $v_i \in \mathbb{F}_q^*$ (referred to as the *multiplier set*) is a set of not necessarily distinct nonzero elements of $\mathbb{F}_q$. Then $GRS_{n,k}(\boldsymbol{u},\boldsymbol{v},q)$ is the (linear) set of codewords:

$$GRS_{n,k}(\boldsymbol{u},\boldsymbol{v},q) = \left\{ \boldsymbol{c}_f = (v_1 f(u_1), v_2 f(u_2), \ldots, v_n f(u_n)),\ f \in \mathbb{F}_q^{<k}[x] \right\}.$$

Since the minimum distance of this linear set of codewords is $n-k+1$ the code is MDS, for the same reason noted above. Clearly an RS code is a $GRS_{n,k}(\boldsymbol{u},\boldsymbol{v},q)$ code with $\boldsymbol{v} = (1,1,\ldots,1)$.

The dual of any MDS code is MDS. It is also true [7, 9, 10] that the dual of a GRS code is also a GRS code. In particular, given $GRS_{n,k}(\boldsymbol{u},\boldsymbol{v},q)$ there exists a set $\boldsymbol{w} \in (\mathbb{F}_q^*)^n$ such that

$$\begin{aligned} GRS_{n,k}^{\perp}(\boldsymbol{u},\boldsymbol{v},q) &= GRS_{n,n-k}(\boldsymbol{u},\boldsymbol{w},q) \\ &= \left\{ w_1 g(u_1), w_2 g(u_2), \ldots, w_n g(u_n),\ g \in \mathbb{F}_q^{<n-k}[x] \right\}. \end{aligned} \tag{1.7}$$

In other words, for any $f(x) \in \mathbb{F}_q^{<k}[x]$ and $g(x) \in \mathbb{F}_q^{<n-k}[x]$ for a given evaluation set $\boldsymbol{u} = \{u_1, \ldots, u_n\}$ and multiplier set $\boldsymbol{v} = \{v_1, \ldots, v_n\}$ there is a multiplier set $\boldsymbol{w} = \{w_1, \ldots, w_n\}$ such that the associated codewords $\boldsymbol{c}_f \in GRS_{n,k}(\boldsymbol{u},\boldsymbol{v},q)$ and $\boldsymbol{c}_g \in GRS_{n,n-k}(\boldsymbol{u},\boldsymbol{w},q)$ are such that

$$(\boldsymbol{c}_f, \boldsymbol{c}_g) = v_1 f(u_1) w_1 g(u_1) + \cdots + v_n f(u_n) w_n g(u_n) = 0.$$

Indeed the multiplier set vector $\boldsymbol{w}$ can be computed as

$$w_i = \left(v_i \prod_{j \neq i} (u_i - u_j) \right)^{-1}. \tag{1.8}$$

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

To see this, for a given evaluation set $\boldsymbol{u}$ (distinct elements), denote

$$e(x) = \prod_{i=1}^{n}(x - u_i) \quad \text{and} \quad e_i(x) = e(x)/(x - u_i) = \prod_{k \neq i}(x - u_k),$$

a monic polynomial of degree $(n-1)$. It is clear that

$$\frac{e_i(u_j)}{e_i(u_i)} = \begin{cases} 1 \text{ if } & j = i \\ 0 \text{ if } & j \neq i. \end{cases}$$

It follows that for any polynomial $h(x) \in \mathbb{F}_q[x]$ of degree less than n that takes on values $h(u_i)$ on the evaluation set $\boldsymbol{u} = \{u_1, u_2, \ldots, u_n\}$ can be expressed as

$$h(x) = \sum_{i=1}^{n} h(u_i) \frac{e_i(x)}{e_i(u_i)}.$$

To verify Equation 1.7 consider applying this interpolation formula to $f(x)g(x)$ where $f(x)$ is a codeword polynomial $f(x) \in \mathbb{F}_q^{<k}[x]$ (in $GRS_{n,k}(\boldsymbol{u},\boldsymbol{v},q)$) and $g(x) \in \mathbb{F}_q^{<n-k}[x]$ (in $GRS_{n,k}(\boldsymbol{u},\boldsymbol{v},q)^{\perp} = GRS_{n,n-k}(\boldsymbol{u},\boldsymbol{w},q)$) where it is claimed that the two multiplier sets $v = \{v_1, v_2, \ldots, v_n\}$ and $w = \{w_1, w_2, \ldots, w_n\}$ are related as in Equation 1.8.

Using the above interpolation formula on the product $f(x)g(x)$ (of degree at most $(n-2)$) gives

$$f(x)g(x) = \sum_{k=1}^{n} f(u_k)g(u_k) \frac{e_k(x)}{e_k(u_k)}.$$

The coefficient of x^{n-1} on the left side is 0 while on the right side is 1 (as $e_k(x)$ is monic of degree $(n-1)$) and hence

$$\begin{aligned} 0 &= \sum_{k=1}^{n} \frac{1}{e_k(u_k)} f(u_k)g(u_k) = \sum_{k=1}^{n} (v_k f(u_k)) \left(\frac{v_k^{-1}}{e_k(u_k)} g(u_k) \right) \\ &= \sum_{k=1}^{n} (v_k f(u_k))(w_k g(u_k)) \quad \text{(by Equation 1.8)} \\ &= (\boldsymbol{c_f}, \boldsymbol{c_g}) = 0. \end{aligned}$$

It is noted in particular that

$$RS_{n,k}^{\perp}(\boldsymbol{u},q) = GRS_{n,n-k}(\boldsymbol{u},\boldsymbol{w},q)$$

for the multiplier set $w_i = \prod_{j \neq i}(u_i - u_j)$.

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

Reed–Muller Codes

Reed–Muller (RM) codes are discussed in some depth in most books on coding (e.g., [3, 4]) with perhaps the most comprehensive being [2] which considers their relationship to Euclidean geometries and combinatorial designs. The properties of RM codes are most easily developed for the binary field but the general case will be considered here – the *Generalized Reed–Muller* (GRM) codes (generalized in a different sense than the GRS codes). The codes are of most interest in this work for the construction of locally decodable codes (Chapter 8) and their relationship to multiplicity codes introduced there.

Consider m variables $x_1, x_2, \ldots, x_m$ and the ring $\mathbb{F}_q[x_1, x_2, \ldots, x_m] = \mathbb{F}_q[\boldsymbol{x}]$ of multivariate polynomials over $\mathbb{F}_q$ (see also Appendix B). The set of all monomials of the m variables and their degree is of the form

$$\left\{ \boldsymbol{x}^{\boldsymbol{i}} = x_1^{i_1} x_2^{i_2} \cdot x_m^{i_m},\ \boldsymbol{i} \sim (i_1, i_2, \ldots, i_m),\ \text{degree } = \sum_j i_j \right\}. \tag{1.9}$$

A multivariate polynomial $f(\boldsymbol{x}) \in \mathbb{F}_q[\boldsymbol{x}]$ is the sum of monomials over $\mathbb{F}_q$ and the degree of f is largest of the degrees of any of its monomials. Notice that over the finite field $\mathbb{F}_q, x_i^q = x_i$ and so only degrees of any variable less than q are of interest.

In the discussion of these codes we will have the need for two simple enumerations: (i) the number of monomials on m variables of degree *exactly* d and (ii) the number of monomials of degree *at most* d. These problems are equivalent to the problems of the number of partitions of the integer d into at most m parts and the number of partitions of all integers *at most* d into at most m parts. These problems are easily addressed as "balls in cells" problems as follows.

For the first problem, place d balls in a row and add a further m balls. There are $d + m - 1$ spaces between the $d + m$ balls. Choose $m - 1$ of these spaces in which to place markers (in $\binom{d+m-1}{m-1}$ ways). Add markers to the left of the row and to the right of the row. Place the balls between two markers into a "bin" – there are m such bins. Subtract a ball from each bin. If the number of balls in bin j is i_j, then the process determines a partition of d in the sense that $i_1 + i_2 + \cdots + i_m = d$ and all such partitions arise in this manner. Thus the number of monomials on m variables of degree equal to d is given by

$$\binom{d+m-1}{m-1} = \left|\{i_1 + i_2 + \cdots + i_m = d,\ i_j \in \mathbb{Z}_{\geq 0}\}\right|. \tag{1.10}$$

To determine the number of monomials on (at most) m variables of total degree *at most* d, consider the setup as above except now add another ball to the row to have $d + m + 1$ balls and choose m of the spaces between the balls in $\binom{d+m}{m}$ ways in which to place markers corresponding to $m+1$ bins. As before subtract a ball from each bin. The contents of the last cell are regarded as superfluous and discarded to take into account the “at most” part of the enumeration. The contents of the first m cells correspond to a partition and the number of monomials on m variables of total degree at most d is

$$\binom{d+m}{m} = \left|\{i_1 + i_2 + \cdots + i_m \leq d,\, i_j \in \mathbb{Z}_{\geq 0}\}\right|. \tag{1.11}$$

Note that it follows that

$$\sum_{j=1}^{d} \binom{j+m-1}{m-1} = \binom{d+m}{m},$$

(i.e., the number of monomials of degree at most d is the number of monomials of degree exactly j for $j = 1, 2, \ldots, d$) as is easily shown by induction.

Consider the code of length q^m denoted $GRM_q(d,m)$ generated by monomials of degree at most d on m variables for $d < q-1$, i.e., let $f(\boldsymbol{x}) \in \mathbb{F}_q[\boldsymbol{x}]$ be an m-variate polynomial of degree at most $d \leq q-1$ (the degree of polynomial is the largest degree of its monomials and no variable is of degree greater than $q-1$). The corresponding codeword is denoted

$$\boldsymbol{c}_f = \Big(f(\boldsymbol{a}),\ \boldsymbol{a} \in \mathbb{F}_q^m,\ f \in \mathbb{F}_q[\boldsymbol{x}],\ f \text{ of degree at most } d\Big),$$

i.e., a codeword of length q^m with coordinate positions labeled with all elements of $\mathbb{F}_q^m$ and coordinate labeled $\boldsymbol{a} \in \mathbb{F}_q^m$ with a value of $f(\boldsymbol{a})$. It is straightforward to show that the codewords corresponding to the monomials are linearly independent over $\mathbb{F}_q$ and hence the code has length and dimension

$$\text{code length } n = q^m \quad \text{and} \quad \text{code dimension } k = \binom{m+d}{d}.$$

To determine a bound on the minimum distance of the code the theorem ([8], theorem 6.13) is used that states the maximum number of zeros of a multivariate polynomial of m variables of degree d over $\mathbb{F}_q$ is at most dq^{m-1}. Thus the maximum fraction of a codeword that can have zero coordinates is d/q and hence the normalized minimum distance of the code (code distance divided by length) is bounded by

$$1 - d/q, \quad d < q - 1.$$

(Recall d here is the maximum degree of the monomials used, not code distance.) The normalized (sometimes referred to as fractional or relative) distance of a code will be designated as $\Delta = 1 - d/q$. (Many works use δ to denote this, used for the erasure probability on the BEC here.) Thus, e.g., for

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

$m = 2$ (bivariate polynomials) this subclass of GRM codes has the parameters $\left(q^2, \binom{d+2}{d}, q^2 - dq\right)_q$. Note that the rate of the code is

$$\binom{d+2}{2}\Big/ q^2 \approx d^2/2q^2 = (1-\Delta)^2/2.$$

Thus the code can have rate at most $1/2$.

For a more complete analysis of the GRM codes the reader should consult ([2], section 5.4). Properties of GRS and GRM codes will be of interest in several of the chapters.

1.2 Notes on Information Theory

The probability distribution of the discrete random variable X, $Pr(X = x)$, will be denoted $P_X(x)$ or as $P(x)$ when the random variable is understood. Similarly a joint discrete random variable $X \times Y$ (or XY) is denoted $Pr(X = x, Y = y) = P_{XY}(x,y)$. The conditional probability distribution is denoted $Pr(Y = y \mid X = x) = P(y \mid x)$. A probability density function (pdf) for a continuous random variable will be designated similarly as $p_X(x)$ or a similar lowercase function.

Certain notions from information theory are required. The entropy of a discrete ensemble $\{P(x_i), i = 1, 2, \ldots\}$ is given by

$$H(X) = -\sum_i P(x_i) \log P(x_i)$$

and unless otherwise specified all logs will be to the base 2. It represents the amount of uncertainty in the outcome of a realization of the random variable.

A special case will be important for later use, that of a binary ensemble $\{p, (1-p)\}$ which has an entropy of

$$H_2(p) = -p \log_2 p - (1-p)\log_2(1-p) \tag{1.12}$$

referred to as the binary entropy function. It is convenient to introduce the q-ary entropy function here, defined as

$$H_q(x) = \begin{cases} x\log_q(q-1) - x\log_q x - (1-x)\log_q(1-x), & 0 < x \le \theta = (q-1)/q \\ 0, & x = 0 \end{cases} \tag{1.13}$$

an extension of the binary entropy function. Notice that $H_q(p)$ is the entropy associated with the q-ary discrete symmetric channel and also the entropy of the probability ensemble $\{1-p, p/(q-1), \ldots, p/(q-1)\}$ (total of q values). The binary entropy function of Equation 1.12 is obtained with $q = 2$.

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

Similarly the entropy of a joint ensemble $\{P(x_i,y_i), i=1,2,\ldots\}$ is given by

$$H(X,Y) = -\sum_{x_i,y_i} P(x_i,y_i)\log P(x_i,y_i)$$

and the conditional entropy of X given Y is given by

$$H(X \mid Y) = -\sum_{x_i,y_i} P(x_i,y_i)\log P(x_i \mid y_i) = H(X,Y) - H(Y)$$

which has the interpretation of being the expected amount of uncertainty remaining about X after observing Y, sometimes referred to as equivocation.

The mutual information between ensembles X and Y is given by

$$I(X;Y) = \sum_{x_i,y_i} P(x_i,y_j)\log \frac{P(x_i,y_j)}{P(x_i)P(y_j)} \tag{1.14}$$

and measures the amount of information knowledge that one of the variables gives on the other. The notation $\{X;Y\}$ is viewed as a joint ensemble. It will often be the case that X will represent the input to a discrete memoryless channel (to be discussed) and Y the output of the channel and this notion of mutual information has played a pivotal role in the development of communication systems over the past several decades.

Similarly for three ensembles it follows that

$$I(X;Y,Z) = \sum_{i,j,k} P(x_i,y_j,z_k)\log \frac{P(x_i,y_j,z_k)}{P(x_i)P(y_j,z_k)}.$$

The conditional information of the ensemble $\{X;Y\}$ given Z is

$$I(X;Y \mid Z) = \sum_{i,j,k} P(x_i,y_j,z_k)\log \frac{P(x_i,y_j \mid z_k)}{P(x_i \mid z_k)P(y_j \mid z_k)}$$

or alternatively

$$I(X;Y \mid Z) = \sum_{i,j,k} P(x_i,y_j,z_k)\log \frac{P(x_i,y_j,z_k)P(z_k)}{P(x_i,z_k)P(y_j,z_k)}.$$

The process of conditioning observations of X,Y on a third random variable Z may increase or decrease the mutual information between X and Z but it can be shown the conditional information is always positive. There are numerous relationships between these information-theoretic quantities. Thus

$$I(X;Y) = H(X) - H(X \mid Y) = H(X) + H(Y) - H(X,Y).$$

Our interest in these notions is to define the notion of capacity of certain channels.

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

A *discrete memoryless channel* (DMC) is a set of finite inputs $\mathcal{X}$ and a discrete set of outputs $\mathcal{Y}$ such that at each instance of time, the channel accepts an input $x \in \mathcal{X}$ and with probability $W(y \mid x)$ outputs $y \in \mathcal{Y}$ and each use of the channel is independent of other uses and

$$\sum_{y \in \mathcal{Y}} W(y \mid x) = 1 \text{ for each } x \in \mathcal{X}.$$

Thus if a vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, $x_i \in \mathcal{X}$ is transmitted in n uses of the channel, the probability of receiving the vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ is given by

$$P(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{i=1}^{n} W(y_i \mid x_i).$$

At times the DMC might be designated simply by the set of transition probabilities $W = \{W(y_i \mid x_i)\}$.

For the remainder of this chapter it will be assumed the channel input is binary and referred to as a binary-input DMC or BDMC where $\mathcal{X} = \{0,1\}$ and that $\mathcal{Y}$ is finite. Important examples of such channels include the *binary symmetric channel* (BSC), the *binary erasure channel* (BEC) and a general BDMC, as shown in Figure 1.1 (a) and (b) while (c) represents the more general case.

Often, rather than a general BDMC, the additional constraint of symmetry is imposed, i.e., a *binary-input discrete memoryless symmetric channel* by which is meant a binary-input $\mathcal{X} = \{0,1\}$ channel with a channel transition probability $\{W(y \mid x), x \in \mathcal{X}, y \in \mathcal{Y}\}$ which satisfies a symmetry condition, noted later.

The notion of mutual information introduced above is applied to a DMC with the $\mathcal{X}$ ensemble representing the channel input and the output ensemble $\mathcal{Y}$ to the output. The function $I(X;Y)$ then represents the amount of information the output gives about the input. In the communication context it would be desirable to maximize this function. Since the channel, represented by the channel transition matrix $W(y \mid x)$, is fixed, the only variable that can be adjusted is the set of input probabilities $P(x_i), x_i \in \mathcal{X}$. Thus the maximum

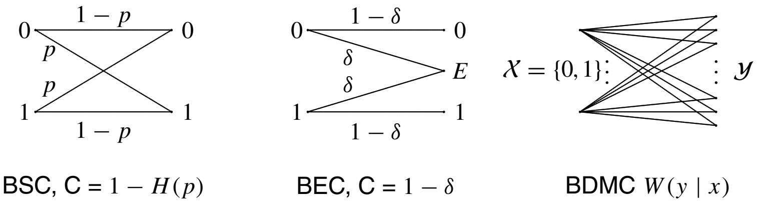

Figure 1.1 Binary-input DMCs

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

amount of information that on average can be transmitted through the channel in each channel use is found by determining the set of input probabilities that maximizes the mutual information between the channel input and output.

It is intuitive to define the *channel capacity* of a DMC as the maximum rate at which it is possible to transmit information through the channel, per channel use, with an arbitrarily low error probability:

$$\begin{aligned}\text{channel capacity } &= C \overset{\Delta}{=} \max_{P(x),\, x\in\mathcal{X}} I(X,Y) = I(W) \\ &= \max_{P(x),\, x\in\mathcal{X}} \textstyle\sum_{x\in\mathcal{X},\, y\in\mathcal{Y}} W(y\mid x)P(x)\log \dfrac{W(y\mid x)P(x)}{P(x)(P(y)}\end{aligned}$$

where $P(y) = \sum_{x\in\mathcal{X}} W(y|x)P(x)$. For general channels, determining channel capacity can be a challenging optimization problem. When the channels exhibit a certain symmetry, however, the optimization is achieved with equally likely inputs:

Definition 1.2 ([6]) A DMC is *symmetric* if the set of outputs can be partitioned into subsets in such a way that for each subset, the matrix of transition probabilities (with rows as inputs and columns as outputs) has the property each row is a permutation of each other row and each column of a partition (if more than one) is a permutation of each other column in the partition.

A consequence of this definition is that for a symmetric DMC, it can be shown that the channel capacity is achieved with equally probable inputs ([6], theorem 4.5.2). It is simple to show that both the BSC and BEC channels are symmetric by this definition. The capacities of the BSC (with crossover probability p) and BEC (with erasure probability δ) are

$$C_{BSC} = 1 + p\log_2 p + (1-p)\log_2(1-p) \quad \text{and} \quad C_{BEC} = 1 - \delta. \tag{1.15}$$

This first relation is often written

$$C_{BSC} = 1 - H_2(p)$$

and $H_2(p)$ is the binary entropy function of Equation 1.12.

For channels with continuous inputs and/or outputs the mutual information between channel input and output is given by

$$I(X,Y) = \int_x \int_y p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right)dxdy$$

for probability density functions $p(\cdot,\cdot)$ and $p(\cdot)$.

Versions of the Gaussian channel where Gaussian-distributed noise is added to the signal in transmission are among the few such channels that offer

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

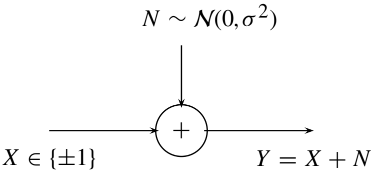

Figure 1.2 The binary-input additive white Gaussian noise channel

tractable solutions and are designated *additive white Gaussian noise* (AWGN) channels. The term "white" here refers to a flat power spectral density function of the noise with frequency. The binary-input AWGN (BIAWGN) channel, where one of two continuous-time signals is chosen for transmission during a given time interval $(0,T)$ and experiences AWGN in transmission, can be represented as in Figure 1.2:

$$Y_i = X_i + N_i, \text{ and } X_i \in \{\pm 1\}, \qquad \text{BIAWGN},$$

where N_i is a Gaussian random variable with zero mean and variance σ^2, denoted $N_i \sim \mathcal{N}(0,\sigma^2)$. The joint distribution of (X,Y) is a mixture of discrete and continuous and with $P(X=+1) = P(X=-1) = 1/2$ (which achieves capacity on this channel) and with $p(x) \sim \mathcal{N}(0,\sigma^2)$. The pdf $p(y)$ of the channel output is

$$\begin{aligned} p(y) &= \frac{1}{2} \cdot \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y+1)^2}{2\sigma^2}} + \frac{1}{2} \cdot \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-1)^2}{2\sigma^2}} \\ &= \frac{1}{\sqrt{8\pi\sigma^2}} \left(\exp\left(-\frac{(y+1)^2}{2\sigma^2} \right) + \exp\left(-\frac{(y-1)^2}{2\sigma^2} \right) \right) \end{aligned} \tag{1.16}$$

and maximizing the expression for mutual information of the channel (equally likely inputs) reduces to

$$C_{BIAWGN} = -\int_y p(y)\log_2 p(y)dy - \frac{1}{2}\log_2(2\pi e\sigma^2). \tag{1.17}$$

The general shape of these capacity functions is shown in Figure 1.3 where SNR denotes signal-to-noise ratio.

Another Gaussian channel of fundamental importance in practice is that of the band-limited AWGN channel (BLAWGN). In this model a band-limited signal $x(t), t \in (0,T)$ with signal power $\leq S$ is transmitted on a channel band-limited to W Hz, i.e., $(-W,W)$ and white Gaussian noise with two-sided power spectral density level $N_o/2$ is added to the signal in transmission. This channel can be discretized via orthogonal basis signals and the celebrated and much-used formula for the capacity of it is

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

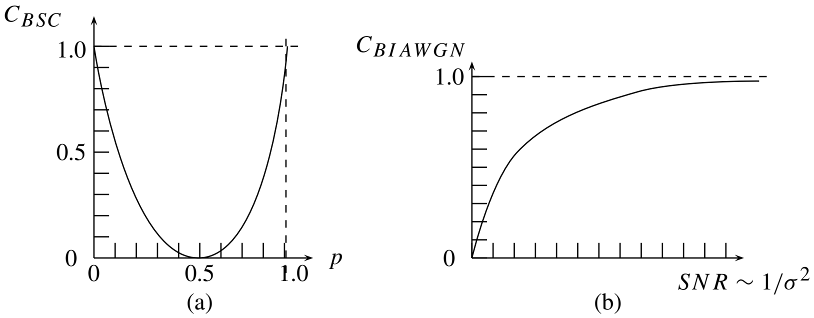

Figure 1.3 Shape of capacity curves for (a) BSC and (b) BIAWGN

$$C_{BLAWGN} = W \log_2(1 + S/N_o W) \text{ bits per second.} \tag{1.18}$$

The importance of the notion of capacity and perhaps the crowning achievement of information theory is the following coding theorem, informally stated, that says k information bits can be encoded into n coded bits, code rate $R = k/n$, such that the bit error probability P_e of the decoded coded bits at the output of a DMC of capacity C can be upper bounded by

$$P_e \leq e^{-nE(R)}, \; R < C \tag{1.19}$$

where $E(R)$, the error rate function, is > 0 for all $R < C$. The result implies that for any code rate $R < C$ there will exist a code of some length n capable of transmitting information with negligible error probability. Thus reliable communication is possible even though the channel is noisy as long as one does not transmit at too high a rate.

The information-theoretic results discussed here have driven research into finding efficient codes, encoding and decoding algorithms for the channels noted over many decades. The references [5, 15] present a more comprehensive discussion of these and related issues.

1.3 An Overview of the Chapters

A brief description of the following chapters is given.

The chapter on coding for erasures is focused on the search for erasure-correcting algorithms that achieve linear decoding complexity. It starts with a discussion of Tornado codes. Although these codes did not figure prominently in subsequent work, they led to the notion of codes from random graphs with irregular edge distributions that led to very efficient decoding algorithms. In turn these can be viewed as leading to the notion of fountain codes which

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

are not erasure-correcting codes. Rather they are codes that can efficiently recreate a file from several random combinations of subfiles. Such codes led to the important concept of Raptor codes which have been incorporated into numerous standards for the download of large files from servers in a multicast network while not requiring requests for retransmissions of portions of a file that a receiver may be missing, a crucial feature in practice.

Certain aspects of low-density parity-check (LDPC) codes are then discussed. These codes, which derive from the work of Gallager from the early 1960s, have more recently assumed great importance for applications as diverse as coding for flash memories as well as a wide variety of communication systems. Numerous books have been written on various aspects of the construction and analysis of these codes. This chapter focuses largely on the paper of [13] which proved crucial for more recent progress for the analytical techniques it introduced.

The chapter on polar codes arose out of the seminal paper [1]. In a deep study of information-theoretic and analytical technique it produced the first codes that provably achieved rates approaching capacity. From a binary-input channel with capacity $C \leq 1$, through iterative transformations, it derived a channel with $N = 2^n$ inputs and outputs and produced a series of NC sub-channels that are capable of transmitting data with arbitrarily small error probability, thus achieving capacity. The chapter discusses the central notions to assist with a deeper reading of the paper.

The chapter on network coding is devoted to the somewhat surprising idea that allowing nodes (servers) in a packet network to process and combine packets as they traverse the network can substantially improve throughput of the network. This raises the question of the capacity of such a network and how to code the packets in order to achieve the capacity. This chapter looks at a few of the techniques that have been developed for multicast channels.

With the wide availability of the high-speed internet, access to information stored on widely distributed databases became more commonplace. Huge amounts of information stored in distributed databases made up of standard computing and storage elements became ubiquitous. Such elements fail with some regularity and methods to efficiently restore the contents of a failed server are required. Many of these early distributed storage systems simply replicated data on several servers and this turned out to be an inefficient method of achieving restoration of a failed server, both in terms of storage and transmission costs. Coding the stored information greatly improved the efficiency with which a failed server could be restored and Chapter 6 reviews the coding techniques involved. The concepts involved are closely related to locally repairable codes considered in Chapter 7 where an erased

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

coordinate in a codeword can be restored by contacting a few other coordinate positions.

Chapter 8 considers coding techniques which allow a small amount of information to be recovered from errors in a codeword without decoding the entire codeword, termed locally decodable codes. Such codes might find application where very long codewords are used and users make frequent requests for modest amounts of information. The research led to numerous other variations such as locally testable codes where one examines a small portion of data and is asked to determine if it is a portion of a codeword of some code, with some probability.

Private information retrieval considers techniques for users to access information on servers in such a manner that the servers are unaware of which information is being sought. The most common scenario is one where the servers contain the same information and users query information from individual servers and perform computations on the responses to arrive at the desired information. More recent contributions have shown how coded information stored on the servers can achieve the same ends with greatly improved storage efficiency. Observations on this problem are given in Chapter 9.

The notion of a batch code addresses the problem of storing information on servers in such a way that no matter which information is being sought no single server has to download more than a specified amount of information. It is a technique to ensure load balancing between servers. Some techniques to achieve this are discussed in Chapter 10.

Properties of expander graphs find wide application in several areas of computer science and mathematics and the notion has been applied to the construction of error-correcting codes with efficient decoding algorithms. Chapter 11 introduces this topic of considerable current interest.

Algebraic coding theory is based on the notion of packing spheres in a space of n-tuples over a finite field $\mathbb{F}_q$ according to the Hamming metric. Rank-metric codes consider the vector space of matrices of a given shape over a finite field with a different metric, namely the distance between two such matrices is given by the rank of the difference of the matrices which can be shown to be a metric on such a space. A somewhat related (although quite distinct) notion is to consider a set of subspaces of a vector space over a finite field with a metric defined between such subspaces based on their size and intersection. Such codes of subspaces have been shown to be of value in the network coding problem. The rank-metric codes and subspace codes are introduced in Chapter 12.

A problem that was introduced in the early days of information theory was the notion of list decoding where, rather than the decoding algorithm producing

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

a unique closest codeword to the received word, it was acceptable to produce a list of closest words. Decoding was then viewed as successful if the transmitted codeword was on the list. The work of Sudan [14] introduced innovative techniques for this problem which influenced many aspects of coding theory. This new approach led to numerous other applications and results to achieve capacity on such a channel. These are overviewed in Chapter 13.

Shift register sequences have found important applications in numerous synchronization and ranging systems as well as code-division multiple access (CDMA) communication systems. Chapter 14 discusses their basic properties.

The advent of quantum computing is likely to have a dramatic effect on many storage, computing and transmission technologies. While still in its infancy it has already altered the practice of cryptography in that the US government has mandated that future deployment of crypto algorithms should be quantum-resistant, giving rise to the subject of "postquantum cryptography." A brief discussion of this area is given in Chapter 15. While experts in quantum computing may differ in their estimates of the time frame in which it will become significant, there seems little doubt that it will have a major impact.

An aspect of current quantum computing systems is their inherent instability as the quantum states interact with their environment causing errors in the computation. The systems currently implemented or planned will likely rely on some form of quantum error-correcting codes to achieve sufficient system stability for their efficient operation. The subject is introduced in Chapter 16.

The final chapter considers a variety of other coding scenarios in an effort to display the width of the areas embraced by the term "coding" and to further illustrate the scope of coding research that has been ongoing for the past few decades beyond the few topics covered in the chapters.

The two appendices cover some useful background material on finite geometries and multivariable polynomials over finite fields.

References

[1] Arıkan, E. 2009. Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Trans. Inform. Theory*, **55**(7), 3051–3073.

[2] Assmus, Jr., E.F., and Key, J.D. 1992. *Designs and their codes*. Cambridge Tracts in Mathematics, vol. 103. Cambridge University Press, Cambridge.

[3] Blahut, R.E. 1983. *Theory and practice of error control codes*. Advanced Book Program. Addison-Wesley, Reading, MA.

[4] Blake, I.F., and Mullin, R.C. 1975. *The mathematical theory of coding*. Academic Press, New York/London.

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press

[5] Forney, G.D., and Ungerboeck, G. 1998. Modulation and coding for linear Gaussian channels. *IEEE Trans. Inform. Theory*, **44**(6), 2384–2415.

[6] Gallager, R.G. 1968. *Information theory and reliable communication*. John Wiley & Sons, New York.

[7] Huffman, W.C., and Pless, V. 2003. *Fundamentals of error-correcting codes*. Cambridge University Press, Cambridge.

[8] Lidl, R., and Niederreiter, H. 1997. *Finite fields*, 2nd ed. Encyclopedia of Mathematics and Its Applications, vol. 20. Cambridge University Press, Cambridge.

[9] Ling, S., and Xing, C. 2004. *Coding theory*. Cambridge University Press, Cambridge.

[10] MacWilliams, F.J., and Sloane, N.J.A. 1977. *The theory of error-correcting codes: I and II*. North-Holland Mathematical Library, vol. 16. North-Holland, Amsterdam/New York/Oxford.

[11] McEliece, R.J. 1987. *Finite fields for computer scientists and engineers*. The Kluwer International Series in Engineering and Computer Science, vol. 23. Kluwer Academic, Boston, MA.

[12] Menezes, A.J., Blake, I.F., Gao, X.H., Mullin, R.C., Vanstone, S.A., and Yaghoobian, T. 1993. *Applications of finite fields*. The Kluwer International Series in Engineering and Computer Science, vol. 199. Kluwer Academic, Boston, MA.

[13] Richardson, T.J., and Urbanke, R.L. 2001. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Trans. Inform. Theory*, **47**(2), 599–618.

[14] Sudan, M. 1997. Decoding of Reed Solomon codes beyond the error-correction bound. *J. Complexity*, **13**(1), 180–193.

[15] Ungerboeck, G. 1982. Channel coding with multilevel/phase signals. *IEEE Trans. Inform. Theory*, **28**(1), 55–67.

https://doi.org/10.1017/9781009283403.002 Published online by Cambridge University Press