By Øyvind Ytrehus

Chapter 2

Background

The design and analysis of error correcting codes are based on mathematical foundations, mostly from the areas of linear algebra, finite fields, probability and statistics, and information theory. However, in order to understand how error correcting codes are used or can be used in practical applications, we also need to know some aspects of informatics/computer science.

This chapter provides a review of this theoretical and practical background.

2.1 Context: Communication, storage, and protocol stacks

Informatics is, arguably, all about the design and analysis of complex logical information structures and the actions we can perform on them. The Internet is the most complex logical structure constructed by mankind. The favoured and obvious approach to resolving this complexity, in network applications as well as in programming, is to use a “divide and conquer” strategy: Divide the main program into tasks, divide each task into subtasks, and so on. In network applications, the tasks deal with communication, and the focus will be on protocols (Section 2.1.2). It is common to organize or represent the layers of such protocols in terms of a protocol hierarchy (Section 2.1.1).

2.1.1 Internet protocols and protocol stacks

In the Internet, it is convenient to think of communication takes place along a hierarchical protocol stack. Applications (e.g. e-mail, skype, web) communicate with their peer applications using a dedicated peer-to-peer protocol (e.g. http), but the communication is virtual, - they will actually communicate by the use of transport layer services, which in the classical Internet settings will be either TCP or UDP. TCP and UDP communicate through the network layer service provided by IP. IP datagrams are forwarded on links using technology specific link layer protocols, which in turn communicate on top of a physical communication service. There are also dedicated sublayer protocols dedicated to e.g. security and multicasting.

2.1.2 Protocols for robust transmission

A protocol (in the context of communication) is a set of rules that govern the communication between two and more parties. The protocol specifies the format and semantics of messages sent, the allowable scenarios of message sequences, proper response to messages received from the other party, who sends next, and so on. In a layered, hierarchical model like in the Internet, a protocol in one layer uses services provided by the layer below, and in turn provides services to the layer above.

An important class of protocols are those that robustify transmission. Sometimes, messages may disappear, may arrive late, or may be garbled by the communication service due to noise, interference or malicious manipulation. A protocol for robust transmission will detect and correct this, using a context-unique identifier for each packet in order to detect packet loss and, either, error detecting codes in connection with retransmission or only error correction, or a combination, in order to make sure that the connection remains error free.

We may divide this type of protocol into three categories: Those that use error detection and retransmission, those that rely only on error correction, and hybrid protocols. Below, we briefly describe each.

Protocols for error detection

An ARQ (Automatic Repeat-Request) protocol is designed to detect error situations in the communication channel, and to make sure that lost or damaged data are retransmitted. ARQ protocols may be implemented on several layers in a protocol hierarchy.

In a link layer implementation of an ARQ protocol, damaged data are typically identified by an error detecting code, in which the receiving side will request a retransmission. If an error situation is not resolved following this protocol, the link layer service may report the event to higher layers. However, another and more common approach is to ignore the event, and pretend it did not happen.

Thus a transport layer ARQ protocol need to consider loss (erasure) of packets, as well as delays. The Internet standard Transmission Control Procedure (TCP) does this (and more).

Protocols for error correction

A protocol using only forward error correction (FEC) is quite simple, protocol-wise: The decoding algorithm attempts to determine the transmitted codeword. If it fails to find the correct one, obviously an error occurs. This error will propagate up the protocol stack. Therefore the code should be designed relative to the expected noise level in such a way that decoding errors are sufficiently rare.

Hybrid protocols

In practice many protocols are hybrid, in the sense that they use both error correcting codes and error detecting code. A common approach is to encode the information message with an error correcting *inner* code, and then encode the inner codeword with an error detecting *outer* code. This is an example of a concatenated code construction (See Section 3.3.4). If error correction fails, the outer code will detect the error with high probability, and trigger a retransmission.

It is not optimal to retransmit an exact copy of the first codeword, but rather to choose another inner code and thereby use information from the first and the second transmission to determine the information message.

2.1.3 Protocols for latency critical applications

Some protocols, typically at the transport layer, are designed to serve the needs of latency critical applications. Such applications include the distribution of stored multimedia, real-time multimedia. These protocols deal with applications that make up the bulk of Internet traffic (so the protocols should be efficient!), and where the perceived quality of service depends on the timely and regular delivery of multimedia frames to the receiver. The approaches to the stored multimedia and the real-time multimedia cases are different.

Other applications that are sensitive to delay are online games and stock market trading, which typically use short control messages and require yet a different transport layer service. We will return to these issues towards the end of this course, in Section 4.2.

2.1.4 Cryptographic protocols for messaging

Some cryptographic protocols for messaging have interesting properties due to the security service they need to provide. These properties are reflected in the transport services they should be provided with. We will return to also this issue (if time).

2.1.5 Protocols for storage

Viewing storage as communication from a place called “the present” to another place called “the future”, it is straightforward that storage in many respects can be treated as just another example of spatial (place-to-place) communication, perhaps with some peculiar noise characteristics: Noise phenomena can occur during writing to or storage over time in the media, in which cases the protocol cannot rely in error detection. However, in many cases a major noise component occurs during reading the media. For example, physical layer protocols for reading data from magnetic hard disks often employ error detection and ARQ schemes. We will not cover this subject in INF 243.

A major cause of errors in modern cloud storage is the complete failure of disks, or even disk racks. For this reason such storage systems spread the data over multiple disks in different disk racks, in a redundant fashion so that the protocol for reading data can recover any file even in the case of failure of one or more disks. We will discuss such storage systems later in the course.

2.2 Review of probability theory and notation

- The probability of A, written $P(A)$, is a real number between 0 and 1 such that:
  - If $P(A) = 0$, then A is impossible (never happens).
  - If $P(A) = 1$, then A is necessary (always happens).
  - If $0 < P(A) < 1$, then A is possible (may happen).
- But what does it mean?
  - Classical theory: $\frac{\text{Number of favorable cases}}{\text{Number of possible cases}}$.

  - Probabilities as relative frequencies.
  - Probabilities as subjective beliefs.

- A sample space Ω is the set of all possible outcomes from some experiment. Examples include:
  - Throwing a die: $\Omega = \{1, 2, 3, 4, 5, 6\}$.
  - Flipping a coin: $\Omega = \{\text{head}, \text{tail}\}$.
  - Measuring the temperature in Celsius: $\Omega = \{t\colon t \in \mathbb{R}, t > -273.15\}$.
  - Uttering a word: $\Omega = \{u\colon u \text{ is a word}\}$.
- Sample spaces may be discrete or continuous.

- An event E is a subset of the sample space Ω. Examples include:
- Throwing a die: $E = \{4, 5, 6\}$ (it shows at least 4).
- Flipping a coin: $E = \emptyset$ (it shows neither a head nor a tail).
- Measuring the temperature in Celsius: $E = \{t\colon t \in \mathbb{R}, t > 0\}$ (above freezing).
- Uttering a word: $E = \{u\colon u \text{ is a verb}\}$ (uttering a verb).
- For discrete sample spaces, the event space is the power set 2^Ω of the sample space.

If experiments $e_1, \ldots, e_n$ have sample spaces $\Omega_1, \ldots, \Omega_n$, then the composite experiment has sample space $\Omega = \Omega_1 \times \cdots \times \Omega_n$.

- Throwing two dice: $\Omega = \{(m, n)\colon 1 \leq m, n \leq 6\}$.
- Uttering three words: $\Omega = \{(u, v, w)\colon u, v, w \text{ are a words}\}$.

- Let $P(A)$ denote the probability of the event A.
- The three axioms of probability theory are:
  - $P(A) \geq 0$.
  - $P(\Omega) = 1$.
  - If A and B are disjoint, then $P(A \cup B) = P(A) + P(B)$.
- Example: Let N be the sum of two dice. Then,

  - $P(N > 9) = 1/6$.
  - $P(N < 4) = 1/12$.
  - $P(N < 4 \cup N > 9) = 3/12 = 1/4$.

- $P(\Omega \setminus A) = 1 - P(A)$.
- $P(\emptyset) = 0$.
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.
- $P(A \cap B) \leq P(A \cup B) \leq P(A) + P(B)$.
- If $A \subseteq B$, then $P(B \setminus A) = P(B) - P(A)$.
- If $\Omega = A_1 \cup \cdots \cup A_n$, then $P(B) = P(A_1 \cap B) + \cdots + P(A_n \cap B)$.

Proof: Use Venn diagrams.

- The probability of A given B is defined as

$$P(A \mid B) \triangleq \begin{cases} \frac{P(A \cap B)}{P(B)} & \text{if } P(B) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

- Multiplication rule:

$$P(A \cap B) = P(A)P(B \mid A).$$

- Bayes' law:

$$P(A \mid B) = \frac{P(A)P(B \mid A)}{P(B)}.$$

Let N be the sum of two dice. Then,

- $P(N \text{ is even} \cap N > 9) = 1/9$.
- $P(N \text{ is even}) = 1/2$.
- $P(N > 9) = 1/6$.
- $P(N > 9 \mid N \text{ is even}) = 2/9$.
- $P(N \text{ is even} \mid N > 9) = 2/3$.

- If A and B are independent events, then
  - $P(A \cap B) \triangleq P(A)P(B)$.
  - $P(A \mid B) = P(A)$.
  - $P(B \mid A) = P(B)$.
- Example: Let N be the sum of two dice. Then,
  - $P(\text{first die} = 1) = 1/6$.
  - $P(N \text{ is even}) = 1/2$.
  - $P(\text{first die} = 1 \text{ and } N \text{ is even}) = 1/12$.
  - $P(N \text{ is even} \mid \text{first die} = 1) = 1/2$.
  - $P(\text{first die} = 1 \mid N \text{ is even}) = 1/6$.

- Let $\Omega = \{\omega_1, \ldots, \omega_n\}$ denote the output space of a random experiment.
- A stochastic variable X with values from the alphabet $\{x_1, \ldots, x_L\}$ is a mapping from Ω to $\{x_1, \ldots, x_L\}$ as follows:

$$X : \Omega \to \{x_1, \ldots, x_L\}.$$

- The basic events are associated with a probability measure P, and we write $\{X = x_i\}$ for the event $\{\omega : X(\omega) = x_i\}$ and $P(X = x_i)$ for its probability.
- A probability mass function (pmf) f_X for X is a mapping from $\{x_1, \ldots, x_L\}$ to the real numbers such that

$$f_X(x_i) = P(X = x_i), i = 1, 2, \ldots, L$$

  and

$$f_X(x_i) \geq 0 \, \forall i \text{ and } \sum_{i=1}^{L} f_X(x_i) = 1.$$

- Statistical independence:

$$f_{YZ}(y_i, z_j) = f_Y(y_i) f_Z(z_j)$$

- Conditional probability mass function (pmf) is defined as

$$f_{X|Y}(x_i | y_j) \triangleq \frac{f_{XY}(x_i, y_j)}{f_Y(y_j)}$$

- The expectation of a discrete random variable X with values from the alphabet $\{x_1, \dots, x_L\}$ is defined as

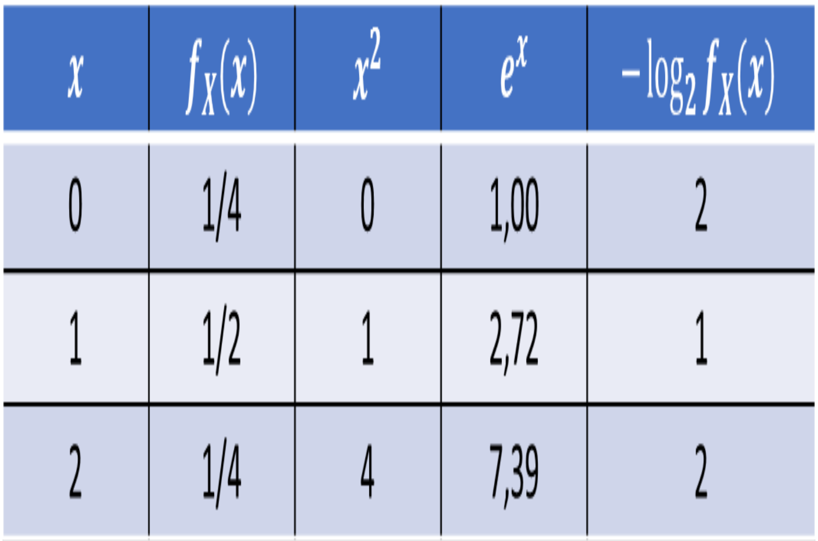
$$\mathbb{E}[X] = \overline{X} \triangleq \sum_{i=1}^{L} x_i f_X(x_i).$$

- The expectation of an arbitrary function $F(X)$ of X is

$$\mathbb{E}[F(X)] = \overline{F(X)} \triangleq \sum_{i=1}^{L} F(x_i) f_X(x_i).$$

- The variance of X is defined as

$$\mathrm{Var}(X) \triangleq \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \sum_{i=1}^{L} (x_i - \mathbb{E}[X])^2 f_X(x_i) = \mathbb{E}\left[X^2\right] - (\mathbb{E}[X])^2.$$



x	$f_X(x)$	x^2	e^x	$-\log_2 f_X(x)$
0	1/4	0	1,00	2
1	1/2	1	2,72	1
2	1/4	4	7,39	2

$$\overline{X} = 1, \quad \overline{X^2} = 3/2, \quad \overline{\mathrm{e}^X} = 3.46, \quad \overline{f_X} = 3/8, \quad \overline{-\log f_X} = 3/2$$

- Let X be a Bernoulli variable, i.e., a random variable which takes the value 1 with probability p and the value 0 with probability $q = 1 - p$.

- Then,

$$\mathbb{E}[X] = 1 \cdot p + 0 \cdot q = p.$$

- The variance becomes

$$\mathrm{Var}(X) = \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2 = 1^2 \cdot p + 0^2 \cdot q - p^2 = p(1-p) = pq.$$

- Let $X_1, \ldots, X_n$ be discrete random variables, and $Y = \sum_{i=1}^n c_i X_i$ where c_i are arbitrary real numbers. Note that the variables may be dependent. Then,

$$\mathbb{E}[Y] = \sum_{i=1}^n c_i \mathbb{E}[X_i].$$

- Proof: Assume $n = 2$ (the general case follows by induction). By the basic definition of expected value, we get

$$\begin{aligned}
\mathbb{E}[c_1X_1 + c_2X_2] &= \sum_{\forall x_1} \sum_{\forall x_2} (c_1x_1 + c_2x_2) P(X_1 = x_1, X_2 = x_2) \\
&= \sum_{\forall x_1} c_1x_1 \sum_{\forall x_2} P(X_1 = x_1, X_2 = x_2) + \sum_{\forall x_2} c_2x_2 \sum_{\forall x_1} P(X_1 = x_1, X_2 = x_2) \\
&= c_1 \sum_{\forall x_1} x_1 P(X_1 = x_1) + c_2 \sum_{\forall x_2} x_2 P(X_2 = x_2) \\
&= c_1\mathbb{E}[X_1] + c_2\mathbb{E}[X_2].
\end{aligned}$$

- Let $X_1, \ldots, X_n$ be independent discrete random variables. Then,

$$\mathbb{E}[X_1 \cdots X_n] = \mathbb{E}[X_1] \cdots \mathbb{E}[X_n].$$

- Proof: Assume $n = 2$ (the general case follows by induction). By the basic definition of expected value, we get

$$\begin{aligned}
\mathbb{E}[X_1X_2] &= \sum_{\forall x_1} \sum_{\forall x_2} x_1x_2 P(X_1 = x_1, X_2 = x_2) \\
&= \sum_{\forall x_1} \sum_{\forall x_2} x_1x_2 P(X_1 = x_1) P(X_2 = x_2) \\
&= \sum_{\forall x_1} x_1 P(X_1 = x_1) \sum_{\forall x_2} x_2 P(X_2 = x_2) \\
&= \mathbb{E}[X_1] \cdot \mathbb{E}[X_2].
\end{aligned}$$

- Furthermore, if $Y = \sum_{i=1}^n c_i X_i$ where c_i, $i = 1, \ldots, n$, are arbitrary real numbers, then

$$\mathrm{Var}(Y) = \sum_{i=1}^n c_i^2 \mathrm{Var}(X_i).$$

- Proof: Assume $n = 2$ (the general case follows by induction). By the basic definition of variance, we get

$$\begin{aligned}\mathrm{Var}(Y) &= \mathbb{E}\left[Y^2\right] - (\mathbb{E}[Y])^2 = \mathbb{E}\left[c_1^2X_1^2 + 2c_1c_2X_1X_2 + c_2^2X_2^2\right] - (c_1\mathbb{E}[X_1] + c_2\mathbb{E}[X_2])^2 \\ &= c_1^2\mathbb{E}\left[X_1^2\right] + 2c_1c_2\mathbb{E}[X_1]\mathbb{E}[X_2] + c_2^2\mathbb{E}\left[X_2^2\right] - \\ &\quad \left(c_1^2(\mathbb{E}\left[X_1\right])^2 + 2c_1c_2\mathbb{E}[X_1]\mathbb{E}[X_2] + c_2^2(\mathbb{E}[X_2])^2\right) \\ &= c_1^2\left(\mathbb{E}\left[X_1^2\right] - \mathbb{E}([X_1])^2\right) + c_2^2\left(\mathbb{E}\left[X_2^2\right] - (\mathbb{E}[X_2])^2\right) \\ &= c_1^2\mathrm{Var}(X_1) + c_2^2\mathrm{Var}(X_2).\end{aligned}$$

- The covariance of two random variables X_1 and X_2 is defined as

$$\mathrm{Cov}(X_1, X_2) \triangleq \mathbb{E}[(X_1 - \mathbb{E}[X_1])(X_2 - \mathbb{E}[X_2])] = \mathbb{E}[X_1X_2] - \mathbb{E}[X_1]\mathbb{E}[X_2].$$

- Note that when X_1 and X_2 are independent, then $\mathbb{E}[X_1X_2] = \mathbb{E}[X_1]\mathbb{E}[X_2]$ and $\mathrm{Cov}(X_1, X_2) = 0$.

- When all values for X have the same probability, X has a uniform distribution, and $f_X(x_i) = 1/L$ for all $i = 1, \ldots, L$.
- Expectation: $\mathbb{E}[X] = \sum_{i=1}^L x_i \cdot 1/L = 1/L \sum_{i=1}^L x_i$.
- Variance: $\mathrm{Var}(X) = \sum_{i=1}^L x_i^2 \cdot 1/L - 1/L^2(\sum_{i=1}^L x_i)^2 = \frac{\sum_{i=1}^L x_i^2 - 1/L(\sum_{i=1}^L x_i)^2}{L}$.

- If $X_1, \ldots, X_n$ are independent, identically distributed (i.i.d.) random variables, all Bernoulli distributed with success probability p, then $Y = \sum_{i=1}^n X_i$ follows a binomial distribution.

- The binomial distribution with parameters n and p is the discrete probability distribution of the number of successes in a sequence of n independent experiments, each asking a yes-no question, and each with its own Boolean-valued outcome: a random variable containing a single bit of information: success/yes/true/one (with probability p) or failure/no/false/zero (with probability $q = 1 - p$).

- The probability of getting exactly k successes in n trials is given by the pmf

$$f_Y(k) = \binom{n}{k} p^k(1-p)^{n-k}$$

for $k = 0, \ldots, n$, where $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ is the binomial coefficient.

- Let $Y = \sum_{i=1}^n X_i$ follow a binomial distribution with parameters n and p, where $X_1, \ldots, X_n$ are i.i.d. Bernoulli variables with success probability p.

- Then,

$$\mathbb{E}[Y] = \sum_{i=1}^{n} \mathbb{E}[X_i] = np.$$

- The variance becomes

$$\mathrm{Var}(Y) = \sum_{i=1}^{n} \mathrm{Var}(X_i) = npq.$$

- For continuous random variables, summations are replaced by integrals. Expectation and variance are defined in an analogous manner and we talk about the probability density function (pdf) instead of the probability mass function.

- The expectation of a continuous random variable X with pdf $f_X(x)$ is defined as

$$\mathbb{E}[X] = \overline{X} \triangleq \int_{x=-\infty}^{\infty} x f_X(x)\,\mathrm{d}x.$$

- The variance of X is defined as

$$\mathrm{Var}(X) \triangleq \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \int_{x=-\infty}^{\infty} (x - \mathbb{E}[X])^2 f_X(x)\,\mathrm{d}x = \mathbb{E}\left[X^2\right] - (\mathbb{E}[X])^2.$$

- When all values for X between a and $b > a$ are equally likely and values outside have zero probability, then X is said to be uniformly distributed on the interval $[a, b]$ and

$$f_X(x) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq x \leq b, \\ 0 & \text{otherwise.} \end{cases}$$

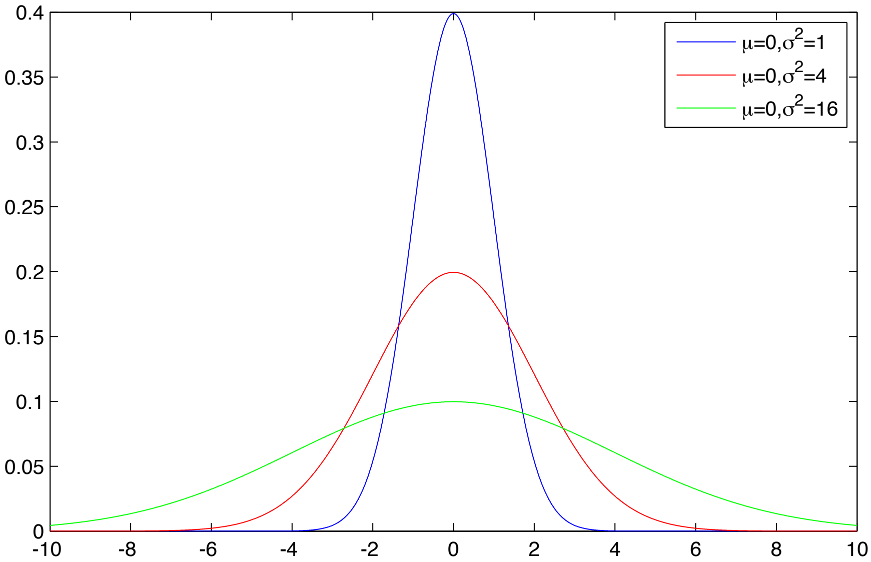
- Expectation:

$$\mathbb{E}[X] = \int_{x=a}^{b} \frac{x}{b-a}\,\mathrm{d}x = \frac{b^2 - a^2}{2(b-a)} = \frac{a+b}{2}.$$

- Variance:

$$\begin{aligned} \mathrm{Var}(X) &= \int_{x=a}^{b} \frac{x^2}{b-a}\,\mathrm{d}x - \left(\frac{a+b}{2}\right)^2 = \frac{b^3 - a^3}{3(b-a)} - \left(\frac{a+b}{2}\right)^2 \\ &= \frac{b^2 + ab + a^2}{3} - \left(\frac{a+b}{2}\right)^2 = \frac{(b-a)^2}{12}. \end{aligned}$$

- The normal (or Gaussian) distribution is a very common continuous probability distribution.



- The pdf of a Gaussian random variable X is
$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$$
where μ is the expectation, σ^2 is the variance, and $\mathrm{e} = 2.71\ldots$ is the base of the natural logarithm.

- When a random variable X is distributed according to a Gaussian distribution with mean μ and variance σ^2, it is written as
$$X \sim \mathcal{N}(\mu, \sigma^2).$$

- Let $X_1, \ldots, X_n$ be independent random variables where $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$. Then, $Y = \sum_{i=1}^n X_i$ is Gaussian distributed with mean $\mu = \sum_{i=1}^n \mu_i$ and variance $\sigma^2 = \sum_{i=1}^n \sigma_i^2$.

- Sketch of proof:
  - The characteristic function of a random variable X is defined as $\phi_X(t) \triangleq \mathbb{E}[\mathrm{e}^{\sqrt{-1}tX}]$.
  - It can be shown that the characteristic function of a Gaussian random variable X with mean μ and variance σ^2 is
$$\phi_X(t) = \mathrm{e}^{\sqrt{-1}t\mu-\sigma^2t^2/2}.$$
  - Also, the characteristic function of a sum of two independent variables X_1 and X_2 is equal to the product of the individual functions since
$$\phi_{X_1+X_2}(t) = \mathbb{E}[\mathrm{e}^{\sqrt{-1}t(X_1+X_2)}] = \mathbb{E}[\mathrm{e}^{\sqrt{-1}tX_1}]\mathbb{E}[\mathrm{e}^{\sqrt{-1}tX_2}] = \phi_{X_1}(t)\phi_{X_2}(t).$$

  - Combining the two results above gives the desired result since no two distinct distributions can both have the same characteristic function.

- The law of large numbers is a theorem that describes the result of performing the same experiment a large number of times. According to the law, the average of the results obtained from a large number of trials should be close to the expected value, and will tend to become closer as more trials are performed.
- Let $X_1 \ldots, X_n$ be i.i.d. random variables with finite mean μ.
- The sample average is defined as

$$S_n = \frac{X_1 + \cdots + X_n}{n}.$$

- The weak law of large numbers states that for any positive number ϵ,

$$\lim_{n \to \infty} P\left(|S_n - \mu| > \epsilon\right) = 0.$$

- The strong law of large numbers states that

$$P\left(\lim_{n \to \infty} S_n = \mu\right) = 1.$$

- Suppose $X_1, \ldots, X_n$ are i.i.d. random variables with mean μ and variance $\sigma^2 < \infty$. Then as n approaches infinity, the random variables $\sqrt{n}(S_n - \mu)$ converge in distribution to $\mathcal{N}(0, \sigma^2)$.
- The usefulness of the theorem is that the distribution $\sqrt{n}(S_n - \mu)$ approaches normality regardless of the shape of the distribution of the individual X_i.
- There are other versions of the theorem where the random variables X_i are not required to be identically distributed, but still independent. Conditions on higher order moments and their growth rates are however imposed.

The *variance* of X, in turn, represents the average distance from the mean, defined as

$$\mathrm{Var}(X) \triangleq \mathbb{E}\left[(X - \mathbb{E}[X])^2\right] = \sum_{i=1}^{m} (x_i - \mathbb{E}[X])^2 f_X(x_i) = \mathbb{E}\left[X^2\right] - (\mathbb{E}[X])^2.$$

2.3 Communication Channels

It is a common approach to apply coding to a precisely defined channel model. That way, it is possible to provide mathematically based statements of the coded system's behaviour, including, especially, the probabilities of decoder failures. Of course, real world channels may be subject to complex noisy processes that are hard to model. For this reason, the "precisely defined channel model" is usually an idealized and theoretical channel model.

In this section we will study communication channels from the theoretical (Section 2.3.1) and practical (Section 2.3.2) points of view.

2.3.1 Theoretical channel models

Discrete Memoryless Channels (DMCs)

A discrete memoryless channel can be considered as a random function that transforms an input symbol from an input alphabet $\mathcal{A}$ of u symbols into an output symbol from an output alphabet $\mathcal{B}$ with v symbols, so that the conditional probability mass function

$$f_{X|Y}(y|x)$$

is known for all possible channel input symbols x and all possible channel output symbols y. See Fig 2.1.

The binary symmetric channel (BSC) and other symmetric channels BSC: $\mathcal{A} = \mathcal{B} = \{0,1\}$ and $P(Y = 0|X = 0) = P(Y = 1|X = 1) = 1 - p$ and for some value $p \in [0,1]$.

qSC: $\mathcal{A} = \mathcal{B} = \{0,\ldots,q-1\}$ and $P(Y = X) = 1 - p, P(Y \neq X) = p/(q-1)$ for some value $p \in [0,1]$.

Asymmetric channels An asymmetric channel is one where the error probability depends on the input symbol. This can be a model for an optical channel (photons may in rare cases disappear inside an optical fiber, but will not spontaneously appear) or, with 0 and 1 switched, a model where a signal may appear but will never disappear (think writing on white paper).

Erasure channels BEC: $\mathcal{A} = \{0,1\}, \mathcal{B} = \{0,1,?\}$ and $P(Y = 0|X = 0) = P(Y = 1|X = 1) = 1 - p$, $P(Y =?|X = 0) = P(Y =?|X = 1) = p$ for some value $p \in [0,1]$.

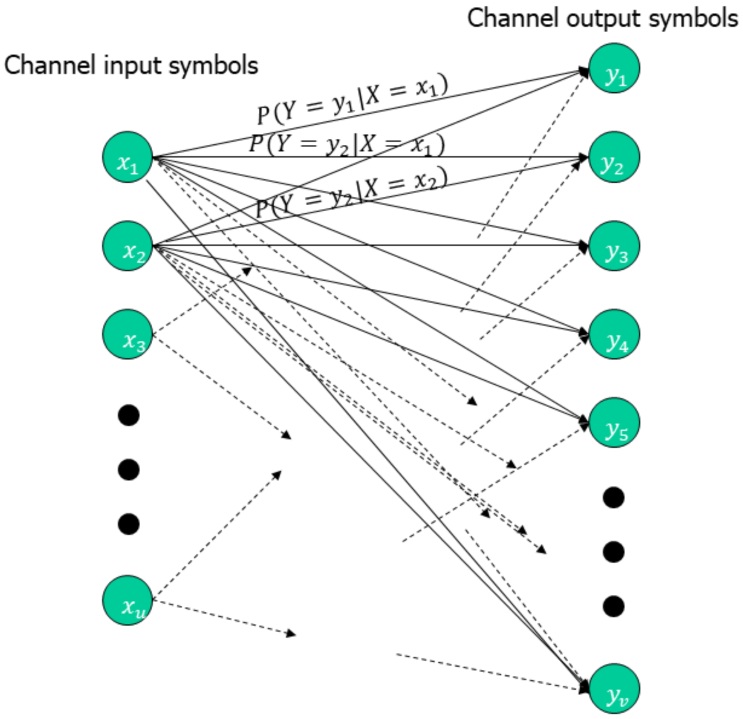


Figure 2.1: Discrete memoryless channel.

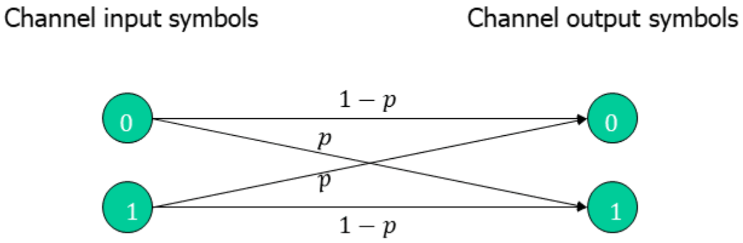


Figure 2.2: Binary symmetric channel.

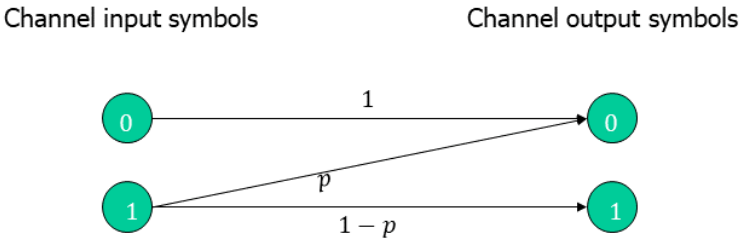


Figure 2.3: Z-channel.

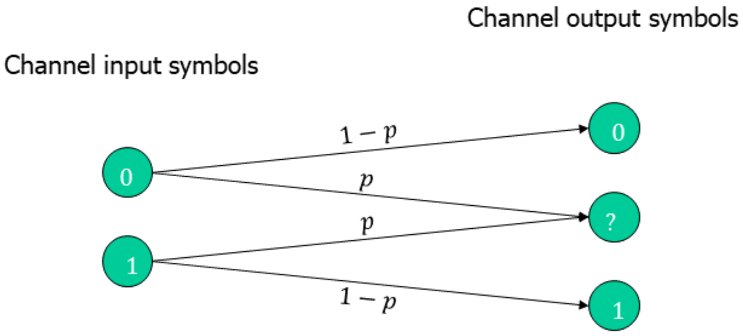


Figure 2.4: Binary erasure channel.

qEC: $\mathcal{A} = \{0, \ldots, q-1\}, \mathcal{B} = \{0, \ldots, q-1, ?\}$ and $P(Y = X) = 1 - p, P(Y = ?) = p$ for some value $p \in [0, 1]$.

Other theoretical channels that act on single symbols

The Additive White Noise Gaussian (AWGN) channel The Additive White Noise Gaussian (AWGN) channel is time-discrete (symbols are transmitted and received at discrete time instants). Usually input symbols belong to a finite set of real values (e.g., -1 and +1), while the output symbol is equal to the input symbol plus a noise sample. The noise sample follows a zero-mean, gaussian distributed sample, so the output can in principle be any real number. The intensity of the noise is measured as the ratio D between the variance of (average) signal power at the receiver and the gaussian noise, and is usually measured in the engineering way in terms of decibels (dB), or $10 \log_{10} D$. A high dB number means a good channel with little noise. For computational purposes, the AWGN channel is sometimes discretized to a discrete-output channel.

Other theoretical channels that act on multiple symbols

Burst channels Gilbert channel: Discrete channel with memory. Error probability depends on previous errors according to a Markov process. In the Gilbert channel, there are two channel states, *Good* and *Bad*. In the *Good* state, the channel acts as a BSC with a small crossover probability. In the *Bad* state, the channel acts as a BSC with a large crossover probability. The channel switches between these two states according to a Markov process. This model can be generalized to one with more than two states, and thus model a discrete fading channel.

Insertion/deletion channels An insertion (deletion) channel acts by inserting (resp. deleting) symbols in a transmitted sequences. This can happen for example in writing text, or in processing of DNA sequences.

Transposition channels A transposition occurs when two neighbor symbols are switched. This is also common in handwritten or typed text.

Bitshift channels A bitshift is a kind of transposition on a binary channel where location of 1s carry the information. For example, on a binary channel it can make sense to parse an encoded sequence into substrings of zeros terminated by a 1 (0001,01,1,000001) where the length of the sequences correspond to the encoded information. If a 1 is shifted, this will change the information content. Such errors can for example occur in certain magnetic recording channels.

2.3.2 Physical communication channels

Any way of transmitting or storing information is a “channel”. In particular, many channels used to transmit electromagnetic signals fit well into the channel model of an AWGN, where the noise may be caused by electric currents in the receiving antennas, background noise, or combinations of many other interfering sources.

It is a normal approach to abstract the physical channel into a theoretical channel by simplifying the description of the channel. However, one should be aware that the effect of physical errors on an encoded sequences can be complicated to predict, and depends also on the modulation (i.e. the representation as logical symbols by physical signals.)

Interference channels

Sometimes, other communication sessions using the same frequencies may dominate the noise. This may cause a noise component which is not uniformly distributed over the entire frequency spectra, *i.e.* is not white, and hence “colored”. This is an important problem, and there are coding theoretic approaches to solve it, but this is beyond the scope of this course.

Fading channels

In addition to the random noise, wireless channels may be disturbed by *fading*, which means that the signal-to-noise ratio varies in time in a random and unpredictable way, especially if locations of the receiver or the sender or the geometric context of the channel change during transmission. The effect is that the channel behaves as an AWGN with varying noise level over the duration of the transmission.

Internet transmission

IP packets sent through the Internet are almost always “correct” if they arrive to the intended receiver. However, packets may be lost due to noisy (wireless links, or due to router overload. In both cases, the channel may be well approximated by a q-ary erasure channel.

Storage channels

Storage channels include the different situations corresponding to e.g. magnetic storage, flash memories, handwriting, or distributed storage. It is worth noting that in some cases, the noise occurs in the reading process rather than in the writing process. Hence, an ARQ protocol for reading an makes sense.

We will give special attention in this course to distributed storage, where a file is stored on multiple recording devices in order to increase robustness against device failures in a an efficient way. The coding problem in this case is closely related to that of an erasure channel.

Other channels

Other real-world channels also show up, sometimes in unexpected(?) situations:

Cryptography and Cybersecurity Coding theory is a useful tool in cryptography and other security applications. Examples include secret sharing, code based post-quantum cryptography, and coding for wire tap channels.

Compressed sensing Certain remote sensing techniques (e. g. in magnetic resonance sensing or seismography) involve sending out electromagnetic or acoustic signals into an environment, and measuring the signal echoed from physical objects. By combining the received echoed signals, one can create an image of the physical target objects. However, this method fragile to noise and errors. By selecting the transmitted signals carefully and formulating the combining of the received echo signals as a decoding problem, significant efficiency gains can be achieved.

DNA Quaternary codes , Error detection, DNA transmission of genetic information through sexual reproduction, insertion/deletion errors (Levenshtein distance)

2.4 Basic information theory

This section contains a brief introduction to information theory. A standard reference is [3].

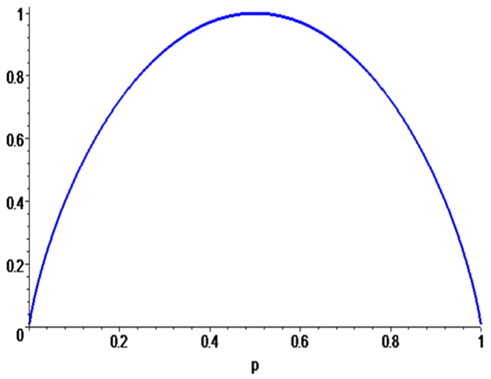


Figure 2.5: Binary entropy function $h(p)$.

Information theory grew out of the landmark paper [4] by Claude Shannon. The word "information" is used in everyday language in slightly different meanings. Shannon formalized it to signify "the amount of reduction of uncertainty about a variable obtained by observing the value of another variable."

Let X be a discrete stochastic variable with pmf (probability mass function) $f_X(x_j), j = 1, \ldots, m$ as defined in Section 2.2. "Uncertainty" here refers to the situation when the value of X is not yet known. Shannon used the word *entropy*, borrowed from thermodynamics, to formalize this concept of uncertainty. The entropy of X is[1] the statistical expectation of the negative of the logarithm of the probability of each outcome of X,

$$H(X) \triangleq \mathbb{E}[-\log X] = -\sum_{i=1}^{m} f_X(x_i) \log f_X(x_i) \tag{2.1}$$

By convention, the logarithm is assumed to be calculated in base 2 (that is, if $y = 2^z$, then $\log y = \log_2 y = z$), in which case the unit of measure of the entropy is a binary digit - a *bit*.

Example 5. *When $m = 2$ and the probabilities of the two outcomes are p and $1 - p$, respectively, the entropy function is conveniently expressed as a function of p. It is common to call this the* binary entropy function $h(p)$, *where*

$$h(p) = -p \log_2(p) - (1 - p) \log_2(1 - p).$$

The binary entropy function has a maximum value of 1 (bit) and is shown in Figure 2.5.

Analogously, one can also define the conditional entropy $H(X|Y)$ of X with respect

[1] Strange as it seems, this definition provides a useful measure of entropy which behaves consistent with intuitive "axioms" that one may impose.

to some other variable Y,

$$H(X|Y) = -\sum_{i=1}^{m}\sum_{j=1}^{n} f_{XY}(x_i, y_j)\log f_{X|Y}(x_i|y_j),$$

where m and n, as before, are the number of possible values for X and Y, respectively. The *mutual information* between the two discrete random variables X and Y is defined as

$$I(X;Y) \triangleq \mathbb{E}[I(X = x; Y = y)] = \sum_{i=1}^{m}\sum_{j=1}^{n} f_{XY}(x_i, y_j)\log\frac{f_{X|Y}(x_i|y_j)}{f_X(x_i)} \tag{2.2}$$

It is easy to show that $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$. The symmetry of this relation between X and Y accounts for the word "mutual". In words, the mutual information quantifies how much uncertainty about X (or Y, respectively) by observing the value of Y (or X, respectively.)

The significance of the mutual information $I(X;Y)$ with respect to inference is that it quantifies *how much* one can learn about X by observing Y. This quantification makes it possible to compare and select the most efficient among different methods of obtaining information.

2.5 Communication channels revisited

In this section we return to communication channels.

Claude Shannon revolutionized our way of thinking about communication and storage when he introduced information theory in [4]. Information theory studies the behaviour and properties of both information sources and communication channels. In particular, Shannon introduced *channel coding* as the process of encoding data for robust transmission over noisy channels. Let us make it a bit more precise what we mean by a source and a channel.

Shannon's main result on channel coding can be expressed *informally* as follows (see INF 144/ INF 242 for details). Consider an information source $\mathcal{S}$ that produces uniformly distributed information symbols from an alphabet $\mathcal{A}$, and a discrete memoryless channel with input alphabet $\mathcal{A}$ and output alphabet $\mathcal{B}$ so that the conditional probability

$$f_{Y|X}(Y = y|X = x)$$

is known for all possible channel input symbols x and all possible channel output symbols y. From this conditional probability distribution, it is possible to calculate a number C called the *channel capacity.* Obviously, when the communication channel is random as described here, there is a probability that the receiver of the transmitted data may make an error in guessing what information was sent. To this end, an *error correcting code* is used.

The channel capacity C is defined as

$$C = \max_{f_X} I(X;Y),$$

where the maximization runs over all pmfs for the input symbols X.

An error correcting code for the source $\mathcal{S}$ has an *encoder*, that maps k information symbols from the source $\mathcal{S}$ into n encoded symbols from the same alphabet $\mathcal{A}$, at a code rate of $R = k/n$. These n encoded symbols are passed through the noisy channel, resulting in n channel output symbols. Based on these n output symbols, a *decoder* will attempt to determine the transmitted codeword. The significance of the channel capacity C is given by the *Channel Coding Theorem*:

Theorem 6 (The Channel Coding Theorem)**.** *There exists an encoder of any code rate $R < C$, so that an optimum decoder has error probability less than some positive δ for δ arbitrarily close to zero, provided k (and n) are large enough.*

The converse result is that for $R > C$, the error probability cannot approach zero regardless of the code length. Indeed, the error probability will rapidly get too large for practical use as the rate grows beyond C.

2.5.1 Discrete Memoryless Channels Revisited

The binary symmetric channel and other symmetric channels

The capacity of the BSC:

$$C_{BSC} = 1 - h_2(p)$$

In any case, if $p < 1/2$ (which we can usually assume. Why?), the most likely error vector is the one with lowest Hamming weight (the fewest number of errors). Hence the best decoder (the maximum likelihood decoder) is the one that decodes to the codeword which is closest (in Hamming distance) to the received word.

Erasure channels

The capacity of the BEC:

$$C_{BEC} = 1 - p$$

Decoding: Solving a set of linear equations. Works if number of erased symbols is less than the number of (independent) parity check equations.

2.6 Finite Fields and associated algorithms

Shannon's channel coding theorem refers to an "optimum decoder". While Shannon was a rather practical engineer in many respects, it is worth noting that in the context of this proof the "optimum decoder" is just a function that returns the *most probably transmitted codeword given the received vector of symbols.* Trivially, this function can be calculated by brute force, but this becomes infeasible as soon as the size of the parameters grows beyond toy examples.

Indeed, the decoding problem has been shown to be NP-hard (Berlekamp, McEliece, van Tilborg 1979) *in general.* In order to build a hope of designing an efficient decoder, and also to be able to reason about code properties, we need to impose a structure on the codes. This involves putting conditions on the alphabets used, in such a way that we can use tools from linear algebra. Roughly speaking, we need to make sure that we can perform basic arithmetic operations that behave in a well defined way on alphabet elements (symbols).

2.6.1 Finite Rings and Fields

Groups, rings, and fields are sets with associated arithmetic operations. (For those of us with a background in informatics, it may be useful to think of these constructions as classes of objects, with the arithmetic operations as class methods.)

Rings

A set R, with associated arithmetic operations addition $(+)$, multiplication $(\cdot)$, subtraction $(-)$ and special elements 0,1 is a ring with identity if the following axioms are satisfied:

1. $\forall a, b \in R, a + b = b + a$ (addition is commutative)
2. $\forall a, b, c \in R, (a + b) + c = a + (b + c)$ (addition is associative)
3. $\forall a \in R, a + (-a) = 0$ (so 0 *zero element* and $-a$ is the negative of a)

4. $\forall a \in R, a + 0 = a$ (so 0 is a neutral element in addition, or a *zero element*)
5. $\forall a, b, c \in R, (a \cdot b) \cdot c = a \cdot (b \cdot c)$ (multiplication is associative)
6. $\forall a \in R, a \cdot 1 = 1 \cdot a = a$ (so 1 is a neutral element in multiplication, or a *identity element*)
7. $\forall a, b, c \in R, a \cdot (b + c) = (a \cdot b) + (a \cdot c)$, and $(a + b) \cdot c = (a \cdot c) + (b \cdot c)$ (distributive laws)

Examples of rings include the sets of integers, rational numbers, real numbers, complex numbers, and $n \times n$ real matrices. The natural numbers, on the other hand, do not constitute a ring.

A *commutative* ring is one where also the following axiom is satisfied:

8. $\forall a, b \in R, a \cdot b = b \cdot a$ (multiplication is commutative)

The ring is a *finite ring* if it has a finite number of elements.

Fields

A field is a commutative ring R which also satisfies

9. $\forall a \in R \setminus \{0\}$, there exists an element b such that $a \cdot b = 1$ (b is the multiplicative inverse of a)
10. the elements 0 and 1 are distinct.

It follows that a field has no *zero divisors*, i.e. there does not exist a pair of nonzero field elements a, b so that $a \cdot b = 0$. The field is a *finite field* if it has a finite number of elements.

2.6.2 Prime fields

Example 7. *The simplest finite field is the binary field, made up of the elements* $\{0, 1\}$ *and the arithmetic operations* $0+0 = 1+1 = 0$, $0+1 = 1+0 = 1$, $0 \cdot 0 = 0 \cdot 1 = 1 \cdot 0 = 0$, *and* $1 \cdot 1 = 1$. *This field is also sometimes called* F_2, *or alternatively* $GF(2)$ *("Galois Field".)*

Example 8. *The ternary field consists of the set of integers mod 3,* $Z_3 = \{0, 1, 2\}$*, and addition and multiplication defined as the corresponding integer operations, but where the result is finally reduced mod 3 so that all resulting values again lie in* Z_3*. Thus,* $0+0 = 1+2 = 2+1 = 0$*,* $0+1 = 1+0 = 2+2 = 1$*,* $0+2 = 2+0 = 1+1 = 2$*,* $0 \cdot a = a \cdot 0 = 0$*,* $1 \cdot 1 = 2 \cdot 2 = 1$*,* $1 \cdot 2 = 2 \cdot 1 = 2$*.*

Example 9. *Similarly, from any prime p, the set* $Z_p = \{0, 1, \ldots, p-1\}$ *with addition and multiplication defined as the corresponding integer operations, but where the result is finally reduced mod p so that all resulting values again lie in* Z_p*.* Z_p *is a finite field, also called GF(p), and since it is based on a prime number, it is also called a* prime *field.*

Example 10. *But for computer applications we often prefer to work with, say, m-dimensional binary vectors, which can be naturally associated with m-bit integers. Thus instead of primes, we would like to work with sets* $Z_{2^m} = \{0, 1, \ldots, 2^m - 1\}$ *with addition and multiplication defined as the corresponding integer operations, but where the result is finally reduced mod* 2^m*. This is* not *a field. For example, consider the set* $Z_4 = \{0, 1, 2, 3\}$*. Here the element* 2 *does not have a multiplicative inverse. Instead,* $2 \cdot 2 (mod 4) = 0$*, so* 2 *is a* zero divisor*.*

2.6.3 Extension fields

Example 10 shows that it requires some more thought to define a finite field with a non-prime number of elements. However, since we reallyreally want fields with 2^m elements ($GF(2^m)$), we will need to dredefine the arithmetic operations.

The trick is the same as the one used to define complex numbers from real numbers. In the process of defining complex numbers, we make use of imaginary numbers, which are the solutions of equations that do not have solutions in the base field. Thus a complex number $x + yi$ can be thought of as a two dimensional vector (x, y), where x and y are real numbers and the imaginary number $i = \sqrt{-1}$.

In order to make a similar extension of the binary numbers, we use a similar trick. A polynomial $f(x) = f_0 + f_1 x + \cdots + f_m x^m$ of degree m is *binary* if all the coefficients f_j are binary. It is *irreducible* if it is not divisible by any other binary polynomial of degree larger than 0 but less than m. This means that there is no solution in F_2 to the equation $f(x) = 0$ (or in other words, no roots in F_2 of $f(x)$. It can be shown that

Theorem 11. *Any irreducible polynomial over* $GF(2)$ *of degree* m *is a divisor of* $x^{2^m-1}+1$*.*

If an irreducible polynomial $f(x)$ of degree m does not divide any $x^n + 1$ for any $n < 2^m - 1$, then $f(x)$ is called a *primitive* polynomial.

Example 12. *Let* $m = 2$*, so that we will make a finite field with* 2^2 *elements. We will select the primitive polynomial* $f(x) = x^2 + x + 1$ *to define the field* $GF(2^2)$*. (Check: This is* *a polynomial without roots in GF(2),* $(x^2+x+1)(x+1) = x^3+1 = x^{2^2-1}+1$*, and f(x) is then obviously primitive.) Now* invent *a new element* α *so that* $f(\alpha) = 0$*. Then* $\alpha \notin GF(2)$*. Similar to the construction of complex numbers, we can construct the field* $GF(2^2)$ *as follows. The new extension field will inherit the elements* 0 *and* 1*, and then we can add the new element* α*. Then we can also add the final element* $\alpha \cdot \alpha = \alpha^2 = \alpha + 1$*. Thus we use the new element* α *as a generator of the field. If we go on, we get* $\alpha^3 = \alpha^2 + \alpha = 1$*. For convenience, we can think of a dual representation of elements of* $GF(2^2)$*: On one hand, each element is either* 0 *or a power of* α*. Multiplication of nonzero elements then becomes just addition of the exponents. On the other hand, each element can be associated with a binary 2-dimensional vector.* 1 *and* α *corresponds to basis vectors, and the Galois fields* $GF(2^2)$ *can be viewed as a vector space over* $GF(2)$*. Addition in* $GF(2^2)$ *is then carried out as addition of binary vectors. Multiplication can be seen as a right shift of these vectors, where the overflowing bit will be reintroduced in the vector according to the equation given by* $f(\alpha) = 0$*.*

0	1	α	α^2
00	10	01	11

Example 13. *Let* $m = 4$*, so we will make a finite field with* 2^4 *elements. We will select the primitive polynomial* $f(x) = x^4 + x + 1$ *to define the field* $GF(2^4)$*. Now* invent *a new element* α *so that* $f(\alpha) = 0$*. Then* $\alpha \notin GF(2)$*. Similar to the construction of complex numbers, we can construct the field* $GF(2^2)$ *as follows. The new extension field will inherit the elements* 0 *and* 1*, and then we can add the new element* α*. Then by successively multiplying with the generator element* α*, we can include the elements* $\alpha^2, \alpha^3, \cdots, \alpha^{14}$*. The two representations will be as shown in this table:*

0	1	α	α^2	α^3	α^4	α^5	α^6	α^7	α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
0000	1000	0100	0010	0001	1100	0110	0011	1101	1010	0101	1110	0111	1111	1011	1001

If we try to generate α^{15}*, we use* $f(\alpha) = 0$ *so* $\alpha^{15} = \alpha^{12} + \alpha^{11} = (0111) + (1111) = (1000) = 1$*. Notice that the base field* $GF(2)$ *is always a subset (and a subfield) of the extension field, this applies in general for extension fields.*

The *characteristic* of a finite field is the smallest integer ℓ so that if you add the same element ℓ times you get zero. The characteristic is always a prime. For an extension field $GF(2^m)$, the characteristic is 2. In fact, extension fields can be extended in the same fashion as above, so we can generate nested extension fields. The extension field inherits the characteristic of its base field.

2.7 Linear Algebra over Finite Fields

Once we have a finite field with the proper definitions of addition and multiplication in place, linear algebra works as “we are used to.”

2.7.1 Vectors, Matrices, Vector spaces

A vector of length (or dimension) n over $GF(q)$ is an ordered collection of n field elements, in the form

$$\underline{v} = (v_1, \cdots, v_n)$$

A vector can be multiplied by a field element, then the multiplication applies to each field element:

$$c\underline{v} = (cv_1, \cdots, cv_n)$$

A linear combination of vectors $\underline{v_1}, \cdots, \underline{v_\ell}$ is a sum

$$\sum_{i=1}^{\ell} c_i \underline{v_i} \tag{2.3}$$

for some set of coefficients $c_1, \cdots, c_\ell$. If there is a set of coefficients so that the linear combination is zero, then the vectors $\underline{v_1}, \cdots, \underline{v_\ell}$ are linearly dependent. A set of ℓ linearly independent vectors $\underline{v_1}, \cdots, \underline{v_\ell}$ over $GF(q)$ defines an ℓ-dimensional vector space, which is the set of all linear combinations on the form of (2.3) when $(c_1, \cdots, c_\ell)$ runs through all the q^ℓ possible values in $GF(q)^\ell$.

A $k \times n$ matrix is a k-dimensional vector of n-dimensional vectors, usually represented as k rows of n-dimensional columns, where the rows are row vectors and the columns are column vectors. The rank of the matrix is the maximum number r such that every subset of at most r rows (or columns) is linearly independent.

The row (respectively, column) space of a matrix is a vector

A vector-vector product $\underline{uv}$ gives the scalar value $\sum_i u_i v_i$. A $k \times n$ matrix M can be multiplied by an n-dimensional column vector $\underline{v}^\top$, written as $\underline{Mv}^\top$. The result is a column vector where the elements are the scalar product of each row by $\underline{v}^\top$. Similarly matrices can be multiplied in the usual way.

Most of the codes we will consider will be vector spaces over some finite field $GF(q)$. In particular we will consider codes over $GF(2)$, and to some extent codes over $GF(2^m)$.

2.7.2 Gaussian elimination

A matrix $\underline{M}$ is naturally associated with the set of equations given by $\underline{Mx}^\top = \underline{c}^\top$, where $\underline{x}$ is a vector of unknown variables to be solved for and $\underline{c}$ is a set of coefficients (if $\underline{c}$ is all zeros the equation set is homogeneous). The standard procedure for solving this set of equations is to apply Gaussian elimination, which by replacing rows of $\underline{M}$ by linear combinations of the rows iteratively transforms the matrix $\underline{M}$ (and possibly $\underline{c}$) into a new matrix $\underline{M'}$ (and possibly $\underline{c'}$) so that $\underline{M'}$ is in row reduced echelon form.