

Notation $\mathcal{A}_X = \{a_1, a_2, \dots, a_r\}$ and $\mathcal{A}_Y = \{b_1, b_2, \dots, b_s\}$ are source and code alphabets.

Assumptions No insertions/deletions, sequential correspondence known, channel is *memoryless*. Channel completely specified by conditional distribution $Q_{j|i} = P(Y = b_j | X = a_i)$

The **binary symmetric channel** has $\mathcal{A}_X = \mathcal{A}_Y = \{0, 1\}$, and flips bits with probability f ; receives correctly with probability $1 - f$.

The **binary erasure channel** has $\mathcal{A}_X = \{0, 1\}$, $\mathcal{A}_Y = \{0, 1, ?\}$, where the channel randomly “loses” bit (replaces with ?) with probability f , correctly transmits with probability $1 - f$.

The **Z-channel** is like the BSC but always transmits 0 correctly, flips 1’s to 0 with probability f .

Transition probabilities:

$$\left[\begin{array}{cc} \text{BSC} \\ 1-f & f \\ f & 1-f \end{array} \right] \left| \left[\begin{array}{cc} \text{BEC} \\ 1-f & 0 \\ f & f \\ 0 & 1-f \end{array} \right] \left| \left[\begin{array}{cc} \text{Z} \\ 1 & f \\ 0 & 1-f \end{array} \right]$$

Note: j index symbols along rows, i along columns in the above, obviously since there are only two input symbols for BEC.

Define $p_i = P(X = a_i)$ as the **channel input distribution**, which (unlike the conditionals above) we control.

The **joint probability** of an input symbol and output symbol $R_{ij} = P(X = a_i)P(Y = b_j | X = a_i) = p_i Q_{j|i}$

The **output probabilities** are then $q_j = P(Y = b_j) = \sum_i p_i R_{ij}$.

The **backwards (posterior) probabilities** of the inputs given the outputs are thus $S_{i|j} = P(X = a_i | Y = b_j) = R_{ij}/q_j$

Input entropy is just $H(X) = \sum_i p_i \log(1/p_i)$

Output entropy is just $H(Y) = \sum_j q_j \log(1/q_j)$

The **joint entropy** of the input/output is $\sum_i \sum_j R_{ij} \log(1/R_{ij})$, $0, \dots, v_{N-1} + v_N = 0$.

Can define the **conditional entropy of the posterior** as just $H(X|Y = b_j) = \sum_i S_{i|j} \log(1/S_{i|j})$

The joint entropy can be expressed as the sum of one of the marginal entropies and a conditional conditioned on that variable, e.g. $H(X, Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$

The **mutual information** is

$$\begin{aligned} I(X; Y) &= H(X) + H(Y) - H(X, Y) \\ &= H(X) + H(Y|X) \\ &= H(Y) + H(X|Y) \end{aligned}$$

A channel’s **capacity** is the maximum mutual information obtainable with any input distribution.

For a BSC the capacity is $1 - H_2(f)$, H_2 is the binary entropy, i.e. $H_2(p) = p \log(1/p) + (1 - p) \log(1/(1 - p))$.

For a Z-channel this is more complicated, because the expression $I(X; Y) = H_2((1 - p_0)(1 - f)) - (1 - p_0)H_2(f)$ depends on p_0 and f [$H_2((1 - p_0)(1 - f))$ is $H(Y)$]

For extensions of a channel, input/output/conditional entropies and mutual information are N times that of the original channel.

Define a code \mathcal{C} for the N th extension as a subset of all possible blocks on N symbols, $\mathcal{C} \subset \mathcal{A}_X^N$, then N th extension is channel with $|\mathcal{C}|$ input symbols and $|\mathcal{A}_Y|^N$ output symbols. When decoding want to decode by choosing w to maximize

$$P(w|b_{j1} \dots b_{jN}) = \frac{P(w)P(b_{j1} \dots b_{jN}|w)}{P(b_{j1} \dots b_{jN})}$$

If $P(w) = 1/|\mathcal{C}| \forall w$ then we’re doing maximum likelihood and maximizing $P(b_{j1} \dots b_{jN}|w)$.

For a BSC, maximum likelihood decoding is equivalent to picking the codeword with the smallest hamming distance (number of bit positions differing) to the received message. Rate of a binary code \mathcal{C} is $\log_2 |\mathcal{C}|/N$.

Shannon’s Noisy Coding Theorem states that for any channel with capacity C , any desired error probability $\epsilon > 0$, and any transmission rate $R < C$, there exists a code with some length N having a rate at least R such that the probability of error when decoding this code by maximum likelihood is less than ϵ .

Note that this may require *very long blocks* which are often impractical. One solution: **linear codes**, define the N -vector space over the finite field Z_2 as the input and output alphabet of our channel extension. Codewords are now thought of as binary vectors.

A binary code is *linear* if it is closed under addition (modulo 2). i.e., \mathcal{C} must be a subspace of Z_2^N (implying that the all-zero codewords must be in \mathcal{C}). For non-binary codes we need condition for closure under scalar multiplication too.

A linear binary code with K basis vectors has 2^K codewords (number of configurations of binary coefficients). $N - K$ equations of the form $\mathbf{c} \cdot \mathbf{v} = 0$ define a code with 2^K codewords since they specify a basis for our code’s null space. The N -repetition code is a linear $[N, 1]$ code (represents 1 bit with N bits) specifiable by $N - 1$ check equations $v_1 + v_2 = 0, v_2 + v_3 = 0, \dots, v_{N-1} + v_N = 0$.

Generator matrices have each row containing a basis vector, so a $[N, K]$ code generator has K rows and N columns. Given our message \mathbf{s} we encode it into channel input \mathbf{t} by multiplying $\mathbf{t} = \mathbf{sG}$. Assuming the rows are L.I. we’ll have a different \mathbf{t} for each distinct \mathbf{s} .

Parity check matrices have rows containing the coefficients of $M = N - K$ check equations (so M rows, N columns). If H is a parity check matrix for \mathcal{C} and $\mathbf{v} \in \mathcal{C}$, then $\mathbf{v}H^T = \mathbf{0}$. Almost all codes have more than one such parity check matrix.

The generator matrix for a code is actually a parity check matrix for the complementary code.

Get the same code by doing row operations on parity/generator matrix (in case of generator, same set of codewords, *different* mapping!).

Can get *equivalent* codes by permuting the columns. If the left $K \times K$ block forms the identity, the generator/parity check matrix is in *systematic form*. First K bits are original message and then next $N - K$ bits are “check” bits.

These should end up being 0 after multiplying received with the parity check matrix if it’s a valid codeword.

Hamming distance satisfies the triangle inequality:

$$d(\mathbf{u}, \mathbf{v}) \leq d(\mathbf{u}, \mathbf{w}) + d(\mathbf{w}, \mathbf{v})$$

Parity from generator matrices

Suppose G , a generator for \mathcal{C} , is in systematic form so $G = [I_K | P]$, then a parity check matrix for is given by $H = [-P^T | I_{N-K}]$ (in Z_2 , $-P^T = P^T$).

Minimum distance of a code is the minimum of $d(\mathbf{u}, \mathbf{v})$ over all \mathbf{u}, \mathbf{v} . If the minimum distance is $2t + 1$ then a nearest

neighbour decoder can always successfully correct t errors or less. The minimum distance of a binary linear code is equal to the **minimum weight** of the $2^K - 1$ codewords, where weight is the number of 1's in a codeword.

Also the minimum distance is equal to the number of linearly dependent columns in parity check matrix H .

If H has a column of zeros, $d = 1$. If H has two identical columns, $d \geq 2$.

For binary codes, if all columns are distinct and non-zero, then $d \geq 3$.