1 Description of Established Codes

THE SHANNON LIMIT IS NOT ACHIEVED IN PRACTICE

The non-constructive proof of the noisy channel coding theorem showed that good block codes exist for any noisy channel, and indeed that nearly all block codes are good. But writing down an explicit encoder and decoder that is as good as promised by Shannon is still an unsolved problem.

Nearly all codes are good, but nearly all codes require exponential look-up tables for implementation of the encoder and decoder—exponential in the block length N. And the coding theorem required N to be large.

Most of the explicit families of codes that have been written down have the property that they can achieve a vanishing error probability p_b as $N \to \infty$ only if the rate R also goes to zero.

There is one exception to this statement:

CONCATENATION

An encoder-channel-decoder system $\mathcal{C} \to Q \to \mathcal{D}$ can be viewed as defining a super-channel Q' with a smaller probability of error, and with complex correlations among its errors. We can then create an encoder $\mathcal{C}'$ and decoder $\mathcal{D}'$ for this super-channel Q'. By iterating this process, with each successive code adding a small amount of redundancy to a geometrically increasing block, we can define an explicit sequence of codes with the property that $p_b \to 0$ for some rate $R > 0$ (but not any R up to the capacity C).

In fact there is a proof by Farney that better concatenations exist, which achieve rates up to capacity and have encoding and decoding complexity of order $O(N^4)$. But the proof is non-constructive.

MOST ESTABLISHED CODES ARE LINEAR CODES

Coding theory was born with the work of Hamming, who wrote down a family of error correcting codes each able to correct one error in a block of length N. Since then most established codes have been generalizations of Hamming's codes, starting with 'BCH' BCH codes, to which Reed-Muller codes are closely related. Reed-Solomon codes are a generalization of BCH codes. Goppa codes are a generalization of Reed-Solomon codes.

One might ask, is the reason that the Shannon limit is not achieved because linear codes are inherently not as good as random codes? The answer is no, the noisy channel coding theorem can still be proved for linear codes, and in particular for 'cyclic codes', a class to which BCH and Reed-Solomon codes belong.

Linear codes are easy to implement at the encoding end. Is decoding a linear code also easy? Not necessarily. The general decoding problem (find the maximum likelihood $\mathbf{s}$ in the equation $\mathbf{G}\mathbf{s} + \mathbf{n} = \mathbf{r}$) is in fact NP-complete, *i.e.*, it probably requires exponential computer time. So attention focuses on families of codes (such as those listed above) for which there is a fast decoding algorithm—say, a decoding algorithm that can correct up to T errors in a block of length N. An (N, K, T) code is a linear code which codes K bits into a block of length N and for which there is a decoding algorithm guaranteed to correct any combination of T errors.

REED-MULLER CODES

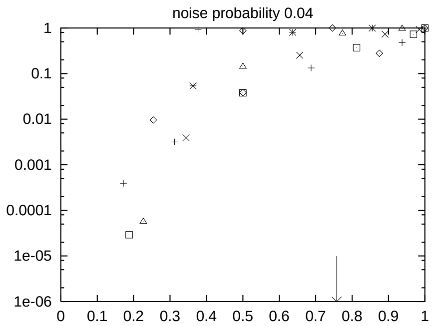
By way of illustration, here is a description of the generator matrix for the Reed-Muller (RM) codes with $N = 16$. Consider the sequence of vectors:

$$\begin{aligned}
\mathbf{v}_0 &= (1111111111111111)\\
\mathbf{v}_1 &= (0000000011111111)\\
\mathbf{v}_2 &= (0000111100001111)\\
\mathbf{v}_3 &= (0011001100110011)\\
\mathbf{v}_4 &= (0101010101010101)\\
\mathbf{v}_{12} &= (0000000000001111)\\
\mathbf{v}_{13} &= (0000000000110011)\\
\mathbf{v}_{14} &= (0000000001010101)\\
\mathbf{v}_{23} &= (0000001100000011)\\
\mathbf{v}_{24} &= (0000010100000101)\\
\mathbf{v}_{34} &= (0001000100010001)
\end{aligned}$$

We can form the following generator matrices. If $\mathbf{G} = [\mathbf{v}_0^{\mathrm{T}}]$, we obtain the $(16, 1, 7)$ RM code, which can be recognized as a repetition code. Taking the first five vectors we can write $\mathbf{G} = [\mathbf{v}_0^{\mathrm{T}}\mathbf{v}_1^{\mathrm{T}}\mathbf{v}_2^{\mathrm{T}}\mathbf{v}_3^{\mathrm{T}}\mathbf{v}_4^{\mathrm{T}}]$ and obtain the (16,5,3) RM code. Or using all 11 vectors, we obtain the (16,11,1) RM code.

The decoding algorithm for RM codes makes use of the periodic properties of these vectors and is similar to a fast Fourier transform.

To illustrate how far Reed-Muller codes are from achieving the Shannon limit, the following figure shows the block error probability versus the rate of RM codes with N up to 1024, when applied to a binary symmetric channel with noise probability 0.04. Each point style corresponds to a different value of N. The arrow shows the capacity of the channel. Any point to the left of this arrow is achievable, according to Shannon, for sufficiently large N. But evidently as p_B goes to zero, the rate of these codes also goes to zero.



MOST LINEAR CODES ARE EXPRESSED IN THE LANGUAGE OF GALOIS THEORY

Let me explain why Galois fields are an appropriate language. First, a definition and some examples.

A Field F is a set $F = \{0, F'\}$such that

1. F forms a group[1] under an addition operation '+', with 0 being the identity;
2. F' forms a group[2] under a multiplication operation '$\cdot$'; multiplication of any element by 0 yields 0;

[1]In fact an Abelian group, *i.e.*, one that satisfies commutativity—$a + b = b + a$.

[2]Also an Abelian group.

3. these operations satisfy the distributive rule $(a+b)\cdot c = a\cdot c + b\cdot c$.

For example, the real numbers form a field, with '+' and '·' denoting ordinary addition and multiplication.

A Galois Field $GF(q)$ is a field with a finite number of elements q.

A unique Galois field exists for any $q = p^m$, where p is a prime number and m is a positive integer; and not for any other q.

$GF(2)$. The addition and multiplication tables for $GF(2)$ are as follows:

+	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1

These are the rules of addition and multiplication modulo 2.

$GF(4)$. The tables for $GF(4)$ are:

+	0	1	A	B
0	0	1	A	B
1	1	0	B	A
A	A	B	0	1
B	B	A	1	0

·	0	1	A	B
0	0	0	0	0
1	0	1	A	B
A	0	A	B	1
B	0	B	1	A

These are *not* the rules of addition and multiplication modulo 4. So how can $GF(4)$ be described? It turns out that the elements can be related to *polynomials*. Consider polynomial functions of x of degree 1 and with coefficients that are integers modulo 2.

Element	Polynomial	Bit pattern
0	0	00
1	1	01
A	x	10
B	$x+1$	11

These polynomials obey the addition and multiplication rules of $GF(4)$ *if* addition and multiplication are modulo the polynomial x^2+x+1. For example, $B\cdot B = x^2+2x+1 = x = A$.

Each element may also be represented as a bit pattern as shown in the above table, with addition bitwise modulo 2, and multiplication defined with an appropriate carry operation.

Why are Galois fields relevant to linear codes? Imagine generalizing a binary generator matrix $\mathbf{G}$ and binary vector $\mathbf{s}$ to a matrix and vector with elements from a larger set, and generalizing the addition and multiplication operations that define the product $\mathbf{Gs}$. In order to produce an appropriate input for a symmetric channel, it would be convenient if, for random $\mathbf{s}$, the product $\mathbf{Gs}$ produced all elements in the enlarged set with equal probability. This is easiest to ensure if these elements form a group under both addition and multiplication, because then these coding operations do not break the symmetry among the elements. When two random elements of a multiplicative group are multiplied together, all elements are produced with equal probability. This is not true of other sets such as the integers, for which the multiplication operation is more likely to give rise to some elements (the composite numbers) than others. Galois fields, by their definition, avoid such symmetry breaking effects.

OTHER CHANNEL MODELS

Most of the codes mentioned above are designed in terms of the binary symmetric channel, but coding theorists usually keep more complex channels in mind also.

Burst error channels are important models in practice. Reed-Solomon codes use Galois fields with large numbers of elements (*e.g.*, 2^{16}), and thereby automatically achieve a degree of burst error tolerance in that even if 17 successive bits are corrupted, only 2 successive symbols in the GF representation are corrupted. Concatenation can give further fortuitous protection against burst errors. The Reed-Solomon codes used on digital compact discs are able to correct any burst of errors of length 4000.

The BSC is an inadequate channel model for a second reason: many channels have *real outputs*. For example, a binary input x may give rise to a probability distribution over a real output y. Codes whose decoders can handle real outputs (log likelihood ratios) are therefore important. 'Convolutional codes' are such codes.

2 Decoding by variational free energy minimization

See this web page:
`ftp://131.111.48.8/pub/mackay/abstracts/fe.html`

3 The Gaussian Channel

The Gaussian Channel has a real input x and a real output y. The conditional distribution of y given x is a Gaussian distribution:

$$P(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-(y-x)^2/2\sigma^2\right].$$

[I will not introduce extra notation to distinguish probability densities from probabilities.]

Note that this channel has a continuous input and output but is discrete in time.

Notation. If a Gaussian variable y has mean x and variance σ^2 then we may write:

$$y \sim \text{Normal}(x,\sigma^2), \text{ or } y \sim \mathcal{N}(x,\sigma^2).$$

Why is this a useful channel model? And what rate of error-free information communication can be achieved over this channel?

MOTIVATION FOR GAUSSIAN CHANNEL

Consider a real (electrical, say) channel with inputs and outputs that are continuous in time. We put in $x(t)$, which is some sort of band-limited signal, and out comes $y(t) =$

$x(t)+n(t)$. Our transmission has a power cost. The average power of a transmission of length T may be constrained thus:

$$\int_0^T dt\,[x(t)]^2/T \leq P.$$

The received signal is assumed to differ from $x(t)$ by additive noise $n(t)$ (for example Johnson noise) which we will model as white Gaussian noise. The magnitude of this noise is quantified by the *noise spectral density* N_0, which depends on the effective temperature of the system.

How could such a channel be used to communicate information? Consider transmitting a set of N real numbers $\{x_n\}_{n=1}^N$ in a signal of duration T made up of a weighted combination of orthonormal basis functions $\phi_n(t)$,

$$x(t) = \sum_{n=1}^{N} x_n \phi_n(t),$$

where $\int_0^T dt\,\phi_n(t)\phi_m(t) = \delta_{nm}$. The receiver can then compute:

$$\begin{aligned} y_n &\equiv \int_0^T dt\,\phi_n(t)y(t) = x_n + \int_0^T dt\,\phi_n(t)n(t) \\ &= x_n + n_n. \end{aligned}$$

If there were no noise, then y_n would equal x_n. The white Gaussian noise $n(t)$ adds scalar noise n_n to the estimate y_n. This noise is Gaussian: $n_n \sim \mathrm{Normal}(0, N_0/2)$, where N_0 is the spectral density introduced above. Thus a continuous channel used in this way is equivalent to the Gaussian channel defined above. The power constraint $\int_0^T dt\,[x(t)]^2 \leq PT$ defines a constraint on the numbers x_n,

$$\sum_n x_n^2 \leq PT \Rightarrow \overline{x_n^2} \leq \frac{PT}{N}$$

Before returning to the Gaussian channel, let us define the **bandwidth** (measured in Hertz) of the continuous channel to be:

$$W = \frac{N}{2T},$$

where N is the maximum number of orthonormal functions that can be produced in an interval of length T. This definition can be justified by imagining creating a band-limited signal of duration T from orthonormal cosine and sine curves of maximum frequency W. The number of orthonormal functions is $N = 2WT$. This definition relates to the Nyquist sampling theorem: if the highest frequency present in a signal is W, then the signal can be fully determined from its values at a series of discrete sample points separated by the Nyquist interval $\Delta t = 1/2W$ seconds.

So the use of a real continuous channel with bandwidth W, noise spectral density N_0 and power P is equivalent to $N/T = 2W$ uses per second of a Gaussian channel with $\sigma^2 = N_0/2$ and subject to the constraint $\overline{x_n^2} \leq P/2W$.

4 Capacity of Gaussian channel

Until now we have only measured the joint, marginal, and conditional entropy of discrete variables. There are two issues we must address—the infinite length of the real line, and the infinite precision of real numbers.

INFINITE INPUTS

Clearly if we are allowed to put *any* real number x into the Gaussian channel, we could just write an entire message as a single very large decimal number followed by a long enough string of zeros to make the probability of error as small as desired. The amount of error-free information conveyed in just a single transmission would be infinite. It is therefore conventional to introduce a *cost function* $v(x)$ for every input x, and constrain codes to have '...a maximum average cost $\bar{v}$...' A generalized channel coding theorem, including a cost function for the inputs, can be proved for the discrete channels discussed previously—see, *e.g.*, McEliece. The result is a channel capacity $C(\bar{v})$ that is a function of the permitted cost. For the Gaussian channel we will assume a cost

$$v(x) = x^2$$

such that the 'average power' $\overline{x^2}$ of the input is constrained. We have already motivated this cost function above in the case of real electrical channels in which the physical power consumption is indeed quadratic in x. The constraint $\overline{x^2} = \bar{v}$ makes it impossible to communicate infinite information over the Gaussian channel.

INFINITE PRECISION

It is tempting to define joint, marginal, and conditional entropies for real variables simply by replacing summations by integrals, but this is not a well defined operation. As we discretize an interval into smaller and smaller divisions, the entropy of the discrete distribution diverges (as the logarithm of the granularity). It is a no-no to take the logarithm of a dimensional quantity such as a probability density $P(x)$ (whose dimensions are $[x]^{-1}$).

There is one information measure, however, which has a well-behaved limit, namely the mutual information. For the discrete case,

$$H(X;Y) = \sum_{x,y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}.$$

Now because the argument of the log is a ratio of two probabilities over the same space, it is (for non-pathological densities) OK to have $P(x,y)$, $P(x)$ and $P(y)$ be probability densities and replace the sum by an integral:

$$\begin{aligned} H(X;Y) &= \int dx\,dy\,P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \\ &= \int dx\,dy\,P(x)P(y|x) \log \frac{P(y|x)}{P(y)}. \end{aligned}$$

We can now ask these questions for the Gaussian channel: (a) what probability distribution $P(x)$ maximizes the mutual information (subject to the constraint $\overline{x^2} = \bar{v}$)? and (b) does the maximal mutual information, as with the discrete channel, measure the maximum error free communication rate of this real channel?

MAXIMIZATION

Introduce a Lagrange multiplier λ for the power constraint and another, μ, for the constraint of normalization of $P(x)$.

$$\begin{aligned} F &= H(X;Y) - \lambda \int dx P(x) x^2 - \mu \int dx P(x) \\ &= \int dx\, P(x) \left[\int dy\, P(y|x) \log \frac{P(y|x)}{P(y)} - \lambda x^2 - \mu \right] \end{aligned}$$

We make the functional derivative with respect to $P(x^*)$.

$$\begin{aligned} \frac{\delta F}{\delta P(x^*)} &= \int dy\, P(y|x^*) \log \frac{P(y|x^*)}{P(y)} - \lambda {x^*}^2 - \mu \\ &\quad - \int dx\, P(x) \int dy\, P(y|x) \frac{1}{P(y)} \frac{\delta P(y)}{\delta P(x^*)}. \end{aligned}$$

The final factor $\delta P(y)/\delta P(x^*)$ is found using $P(y) = \int dx\, P(x) P(y|x)$ to be $P(y|x^*)$, and the whole of the last term collapses in a puff of smoke to 1, which can be absorbed into the μ term.

We now substitute $P(y|x) = \exp(-(y-x)^2/2\sigma^2)/\sqrt{2\pi\sigma^2}$ and set the derivative to zero:

$$\int dy P(y|x) \log \frac{P(y|x)}{P(y)} - \lambda x^2 - \mu' = 0$$

$$\Rightarrow \int dy \frac{\exp(-(y-x)^2/2\sigma^2)}{\sqrt{2\pi\sigma^2}} \log [P(y)\sigma] = -\lambda x^2 - \mu' - \frac{1}{2}$$

This condition must be satisfied by $\log [P(y)\sigma]$ for all x.

Writing a Taylor expansion of $\log [P(y)\sigma] = a + by + cy^2 \ldots$, it is evident that it can only be a quadratic function $\log [P(y)\sigma] = a + cy^2$. (Any higher order terms y^p, $p > 2$, would produce terms in x^p that are not present on the right hand side.) Therefore $P(y)$ is Gaussian. Working backwards we observe that we can obtain this optimal output distribution by using a Gaussian input distribution $P(x)$.

GAUSSIAN INPUT DISTRIBUTION

If $P(x) = \text{Normal}(0, v)$ and $P(y|x) = \text{Normal}(x, \sigma^2)$ then $P(y) = \text{Normal}(0, v + \sigma^2)$ and

$$\begin{aligned} P(x|y) &\propto P(y|x) P(x) \\ &\propto \exp(-(y-x)^2/2\sigma^2) \exp(-x^2/2v) \\ &= \text{Normal}\left(\frac{v}{v+\sigma^2}\, y\, ,\, (1/v + 1/\sigma^2)^{-1} \right). \end{aligned}$$

The mutual information of this optimized distribution is

$$C = \frac{1}{2} \log \left(1 + \frac{v}{\sigma^2} \right)$$

This is an important result. We see that the capacity of the Gaussian channel is a function of the *signal to noise ratio* v/σ^2.

NOISY CHANNEL CODING THEOREM (AGAIN)

We have evaluated a maximal mutual information. Does it correspond to a maximum possible rate of error-free information transmission? One way of proving this relationship is simply to define a sequence of discrete channels, all based on the Gaussian channel, with increasing numbers of inputs and outputs, and prove that the maximum mutual information of these channels tends to C.

A more intuitive argument for the coding theorem may be preferred.

GEOMETRICAL VIEW OF NCCT: SPHERE PACKING

Consider a sequence $\mathbf{x} = (x_1, \ldots x_N)$ of inputs, and the corresponding output $\mathbf{y}$, as defining two points in an N dimensional space. For large N, the noise power is very likely to be close (fractionally) to $N\sigma^2$. The output $\mathbf{y}$ is therefore very likely to be close to the surface of a sphere of radius $\sqrt{N\sigma^2}$ centred on $\mathbf{x}$. Similarly, if the original signal $\mathbf{x}$ is generated at random subject to an average power constraint $\overline{x^2} = v$, then $\mathbf{x}$ is likely to lie close to a sphere, centred on the origin, of radius $\sqrt{Nv}$; and because the total average power of $\mathbf{y}$ is $v + \sigma^2$, the received signal $\mathbf{y}$ is likely to lie on the surface of a sphere of radius $\sqrt{N(v+\sigma^2)}$, centred on the origin.

The volume of an N-dimensional sphere of radius r is

$$V = \frac{\pi^{N/2}}{\Gamma(N/2+1)} r^N.$$

Now consider making a communication system based on non-confusable inputs $\mathbf{x}$, that is, inputs whose spheres do not overlap. The maximum number M of non-confusable inputs is given by dividing the volume of the sphere of probable $\mathbf{y}$s by the volume of the sphere for $\mathbf{y}$ given $\mathbf{x}$:

$$M \leq \left(\frac{\sqrt{N(v+\sigma^2)}}{\sqrt{N\sigma^2}} \right)^N$$

Thus the capacity is bounded by:

$$C = \frac{1}{N} \log M \leq \frac{1}{2} \log \left(1 + \frac{v}{\sigma^2} \right).$$

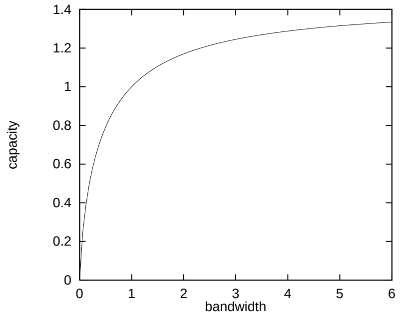
A more detailed argument using the law of large numbers can establish equality.

BACK TO THE CONTINUOUS CHANNEL

Recall that the use of a real continuous channel with bandwidth W, noise spectral density N_0 and power P is equivalent to $N/T = 2W$ uses per second of a Gaussian channel with $\sigma^2 = N_0/2$ and subject to the constraint $\overline{x_n^2} \leq P/2W$. Substituting the result for the capacity of the Gaussian channel, we find the capacity of the continuous channel to be:

$$C = W \log \left(1 + \frac{P}{N_0 W} \right) \text{ bits per second.}$$

This formula gives insight into the tradeoffs of practical communication. Imagine that we have a fixed power constraint. What is the best bandwidth to make use of that power? Introducing $W_0 = P/N_0$, *i.e.*, the bandwidth for which the signal to noise ratio is 1, The following figure shows $C/W_0 = W/W_0 \log(1 + W_0/W)$ as a function of W/W_0. The capacity increases to an asymptote of $W_0 \log e$. It is dramatically better (in terms of capacity) to transmit at a low signal to noise ratio over a large bandwidth, than with high signal to noise in a narrow bandwidth.


1.4
1.2
1
0.8
0.6
0.4
0.2
0
capacity
0
1
2
3
4
5
6
bandwidth