Information Theory
Degree in Data Science and Engineering
Lesson 6: Channel coding

Jordi Quer, Josep Vidal

Mathematics Department, Signal Theory and Communications Department
{jordi.quer, josep.vidal}@upc.edu

2019/20 - Q1

Why channel coding?

Let us return to the example of the HDD and compare repetition coding and Shannon channel coding.

For the BSC with $p = 0.1$, the channel capacity is

$$C = 1 - H(0.1) = 0.469 \text{ bits/tr}$$

This implies an average of $1/R = 2.132$ transmissions/bit for zero error transmission with Shannon's channel codes.

Compare it to the 68 transmissions/bit required for the repetition code with a bit error probability of 10^{-15}.

In spite of its great performance, Shannon's coding is impractical because...

- Codewords are infinite length, and
- Even if making them shorter (at the expenses of some residual error), decoding implies exhaustive evaluation of joint typicality with a large number of codewords.

Why channel coding?

We need to design error correction codes that:

- Achieve low probability of error without much rate loss, and
- Have certain internal structure allowing low decoding complexity.

Two typical practical ways to combat errors are:

- ARQ (Automatic Repeat Request) Detect errors and request retransmission, requires a feedback channel and some delay is incurred.
- FEC (Forward Error Correction) Correct errors at the receiver side by introducing redundancy at the transmitter side. It is quicker, does neither require feedback nor buffering at the transmitter, but entails additional complexity at the receiver.

Both are complementary, we will focus on FEC.

Why channel coding?

Channel codes are an intrinsic part of all communication systems...

- Deep space communications: the *Voyager 1* spacecraft transmitted images of Jupiter and Saturn using *convolutional codes* concatenated with Golay codes. Voyager 2 used *Reed-Solomon codes*, and later missions use LDPC codes.
- Cellular communication systems: *3G* and *4G* use convolutional concatenated turbo codes, and *5G* use *LDPC codes* and *Polar codes*.
- Satellite broadcasting: *DVB* use LDPC and *BCH codes*,
- CD, DVD and HDD: used interleaved Reed-Solomon codes. LDPC codes in modern HDD.
- Error-correcting memory controllers. In deep space missions, RAM may be affected by *cosmic rays*. Hamming codes with *triple modular redundancy* are used.
- QRcodes (2D bar codes) Use Reed-Solomon codes and 4 different levels of protection.

Channel codes

A (n, k) channel code $\mathcal{C}$ is a mapping that assigns the k-length sequence of symbols belonging to an alphabet $\mathcal{X}$, to a codeword (a vector of n symbols) belonging to a possibly different alphabet (we adopt the same though).

$$\mathcal{C} \ : \ \mathcal{X}^k \rightarrow \mathcal{X}^n$$
$$\mathbf{c} = \mathcal{C}(x^k)$$

The codeword generated at the transmitter is sometimes received in error $\mathbf{r} = \mathbf{c} + \mathbf{e}$ after having passed through a noisy channel.

Reconstructing $\mathbf{c}$ from the observed $\mathbf{r}$ requires introducing redundancy, by taking $n > k$.

The rate of the code is defined as $R_c = k/n$.

Fields

Encoding and decoding entails arithmetic operations defined in a *field*, $\mathbb{F}$, consisting of a set $\mathcal{X}$ together with two operations:

Addition

Assuming $a, b, c \in \mathcal{X}$...

- The set is closed under addition: $a + b \in \mathbb{F}$
- Associativity: $a + (b + c) = (a + b) + c$
- Commutativity: $a + b = b + a$
- Additive identity element: $a + 0 = a$
- Additive inverse element: $-a \in \mathbb{F}$ such that $a + (-a) = 0$

Fields

Multiplication

Assuming $a, b, c \in \mathcal{X}$...

- The set is closed under multiplication: $ab \in \mathcal{X}$
- Associativity: $a(bc) = (ab)c$
- Commutativity: $ab = ba$
- Distributive over the addition: $a(b + c) = ab + ac$
- Multiplicative identity element: $1a = a$
- Inverse element: $\forall a \neq 0, \exists\, a^{-1}$ such that $a^{-1}a = 1$

A finite field with $q = |\mathcal{X}|$ elements is called a *Galois field*, $GF(q)$, which can only be defined if q is a prime number or an integer power of a prime. Very much used fields in coding are $GF(2^n)$, in particular $n = 8$.

Only $\mathcal{X} = \{0, 1\}$ will be considered in the sequel.

Linear block codes

There are q^n vectors in the vector space $\mathcal{V}_n$ of dimension n in the field $GF(q)$. A subset of this space is called a vector subspace $\mathcal{S}$, if two conditions are met:

- The all-zeros vector is in $\mathcal{S}$, and
- The linear combination of two vectors is in $\mathcal{S}$, i.e. if $\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{S}$ and $\alpha, \beta \in \mathcal{X}$, then $\alpha\mathbf{c}_1 + \beta\mathbf{c}_2 \in \mathcal{S}$

Example. If $n = 4$ and $q = 2$, there are $q^n = 16$ vectors in the vector space $\mathcal{V}_4$ of dimension 4. Two linearly independent codewords, e.g.

$$\mathbf{c}_1 = 0101 \quad \mathbf{c}_2 = 0110$$

form a basis of the 2-dimensional subspace $\mathcal{S}$. □

The scalar product of two vectors is denoted as $\mathbf{c}_i\mathbf{c}_j^T$ and defined as the sum of the products of its elements. Two vectors are orthogonal if its scalar product is zero.

Linear block codes

A collection of vectors orthogonal to $\mathcal{S}$ is a base of the null space (or kernel) $\mathcal{S}_c$ of $\mathcal{S}$. If the dimension of $\mathcal{S}$ is k, the dimension of $\mathcal{S}_c$ is $n-k$.

Example. The codewords

$$\mathbf{c}_3 = 1000 \quad \mathbf{c}_4 = 1001$$

are orthogonal to $\mathbf{c}_1, \mathbf{c}_2$ and generate $\mathcal{S}_c$. □

The weight of a codeword, $w(\mathbf{c}_j)$, is the number of non-zero elements.

The Hamming distance of two codewords $\mathbf{c}_i, \mathbf{c}_j$, $d(\mathbf{c}_i, \mathbf{c}_j)$, is the number of elements in which they differ. Four relevant properties are:

1. Triangular inequality: $d(\mathbf{c}_i, \mathbf{c}_j) \leq d(\mathbf{c}_i, \mathbf{c}_k) + d(\mathbf{c}_k, \mathbf{c}_j)$
2. Non-negativity: $0 \leq d(\mathbf{c}_i, \mathbf{c}_j) \leq n$, with $d(\mathbf{c}_i, \mathbf{c}_j) = 0$ if $\mathbf{c}_i = c_j$
3. Symmetry: $d(\mathbf{c}_i, \mathbf{c}_j) = d(\mathbf{c}_j, \mathbf{c}_i)$
4. $d(\mathbf{c}_i, \mathbf{c}_j) = w(\mathbf{c}_i - \mathbf{c}_j)$, from the definition of the Hamming distance.

Linear block codes

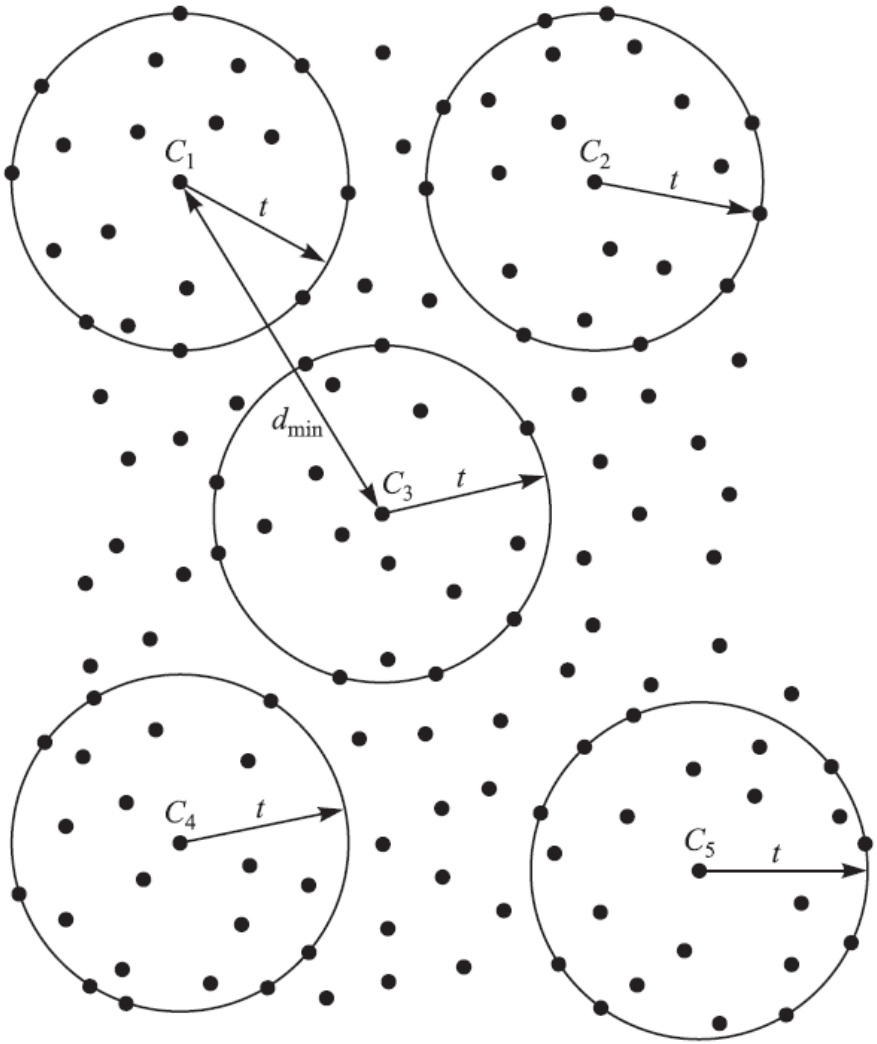
The minimum distance of a code $\mathcal{C}$ is

$$\begin{aligned} d_{min} &= \min\{w(\mathbf{c}_i - \mathbf{c}_j) : \mathbf{c}_i, \mathbf{c}_j \in \mathcal{C}, \mathbf{c}_i \neq \mathbf{c}_j\} \\ &= \min\{w(\mathbf{c}_k) : \mathbf{c}_k \in \mathcal{C}, \mathbf{c}_k \neq \mathbf{0}\} \\ &\triangleq w_{min} \end{aligned}$$

since the linearity of the code implies that a sum of codewords is a codeword.

The key idea behind block codes is: take a number of vectors in $\mathcal{V}_n$ (the codebook) that form a subspace. If due to channel errors any element of the codeword is changed, we can still recover the codeword as long as it is not identified as another codeword.

Linear block codes



The codebook (vectors $\mathbf{c}_i$) contain a fraction of the total set of vectors in $\mathcal{V}_n$ (all black dots), and each codeword can be considered as the center of a sphere of a certain radius t.

Two contradicting goals arise in the design:

1. High efficiency (large R_c, close to channel capacity), by using as many codewords as possible in $\mathcal{V}_n$,
2. High error protection by placing codewords as far as possible from each other, so that corrupted codewords can still be correctly decoded with high probability.

The generator matrix

As an example, let us take a $(n, k) = (127, 92)$ code. From the 2^{127} possible codewords, only $2^{92} \simeq 5 \times 10^{27}$ are selected, but using them still imply a huge lookup table! Fortunatelly we can generate codewords on the fly from the generator matrix.

Define a basis of a k-dimensional space, $\{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_k\}$, and arrange them into a matrix:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \vdots \\ \mathbf{g}_k \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{bmatrix}$$

Given k input symbols $\mathbf{x}_m$, the associated codeword is generated from the vector-matrix product in $\mathbb{F}$:

$$\mathbf{c}_m = \mathbf{x}_m \mathbf{G}$$

That is, codewords are generated as linear combinations of the rows of $\mathbf{G}$.

The generator matrix

By linearly combining rows and permuting columns (Gauss-Jordan elimination), matrix $\mathbf{G}$ can be reduced to its *systematic form*:

$$\mathbf{G} = [\mathbf{P}|\mathbf{I}_k] = \left[\begin{array}{cccc|ccccc} p_{11} & p_{12} & \dots & p_{1n-k} & 1 & 0 & 0 & \dots & 0 \\ p_{21} & p_{22} & \dots & p_{2n-k} & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & & & & & \\ p_{k1} & p_{k2} & \dots & g_{kn-k} & 0 & 0 & 0 & \dots & 1 \end{array}\right]$$

As a result, the codeword is split between parity check symbols and systematic symbols:

$$\mathbf{c}_m = [\mathbf{c}_p|\mathbf{x}_m]$$

The parity check matrix

Associated with the (n, k) code, we can define a $(n, n-k)$ linear dual code, whose generator matrix is $\mathbf{H}^T$. The $n-k$ row vectors of $\mathbf{H}$ form a base of the null space of the rows of $\mathbf{G}$. Any codeword belonging to the code is orthogonal to any codeword generated with the dual code.

Therefore:

$$\mathbf{G}\mathbf{H}^T = \mathbf{0}$$

Solving the linear equation, it follows that

$$\mathbf{H} = [\mathbf{I}_{n-k} | -\mathbf{P}^T]$$

Hence, for any codeword $\mathbf{c}_m\mathbf{H}^T = \mathbf{0}$ we have an algebraic tool that allows detecting errors.

The parity check matrix

Example. A $(5, 2)$ linear block code has a generator matrix:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

The minimum distance is $d_{min} = 3$ and its rate is $R = 2/5$. The parity check matrix is

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

It can be verified that $\mathbf{G}\mathbf{H}^T = \mathbf{0}$. □

Any codeword is orthogonal to $\mathbf{H}^T$, in particular the minimum distance codeword $\mathbf{c}_{d_{min}}\mathbf{H}^T = \mathbf{0}$. Since no other codeword has lower weight, the number of linearly independent rows of $\mathbf{H}$ must be at least $d_{min} - 1$. Since $\mathbf{H}$ generates the null space, $rank(\mathbf{H}) = n - k$. Hence,

$$d_{min} \leq n - k + 1$$

Optimum decoding

Assume that a transmitted codeword $\mathbf{c}$ is received in error $\mathbf{r} = \mathbf{c} + \mathbf{e}$ after having passed through a noisy channel. The decoder operates on $\mathbf{r}$ and returns $\hat{\mathbf{c}} = g(\mathbf{r})$. An error event $\mathcal{E}$ occurs with probability $\Pr(\mathcal{E}|\mathbf{r}) = \Pr(\hat{\mathbf{c}} \neq \mathbf{c}|\mathbf{r})$, and the total error probability is

$$\Pr(\mathcal{E}) = \sum_{\mathbf{r}} \Pr(\mathcal{E}|\mathbf{r}) \Pr(\mathbf{r}).$$

Minimizing the error is equivalent to

$$\underset{\mathbf{c}}{\text{minimize}} \ \Pr(\mathcal{E}|\mathbf{r})$$

or also

$$\underset{\mathbf{c}}{\text{maximize}} \ \Pr(\hat{\mathbf{c}} = \mathbf{c}|\mathbf{r}) \ \Leftrightarrow \ \underset{\mathbf{c}}{\text{maximize}} \ \Pr(\mathbf{r}|\hat{\mathbf{c}} = \mathbf{c}) \Pr(\mathbf{c})$$

that is, $\hat{\mathbf{c}}$ is the most likely codeword given $\mathbf{r}$ (maximum a posteriori decoder). If the channel is memoryless, all errors in the elements of the received codeword are independent so we can write

$$\Pr(\mathbf{r}|\mathbf{c}) = \prod_{i=1}^{n} \Pr(r_i|c_i)$$

Optimum decoding

Let us assume all codewords are equally likely. Since the logarithm is a monotonous increasing function we can use it in the optimization:

$$\begin{aligned}\log \Pr(\mathbf{r}|\mathbf{c}) &= \sum_{i=1}^{n} \log \Pr(r_i|c_i) \\ &= d(\mathbf{r},\mathbf{c}) \log p + (n - d(\mathbf{r},\mathbf{c})) \log(1-p) \\ &= d(\mathbf{r},\mathbf{c}) \log \frac{p}{1-p} + n \log(1-p)\end{aligned}$$

where $d(\mathbf{r},\mathbf{c})$ is the Hamming distance between the received vector and a possible decoded vector. Since $\log \frac{p}{1-p}$ is negative for $p < \frac{1}{2}$ the minimum error criterion finally results in

$$\hat{\mathbf{c}} = \underset{\mathbf{c}}{\operatorname{argmin}}\ d(\mathbf{r},\mathbf{c})$$

Check the figure in slide 11: any black dot corresponding to a possible received vector will be decoded into the closest codeword. Therefore, optimum decoding entails comparison of $\mathbf{r}$ with all possible 2^k codewords! For long codes with large codebooks the complexity may be prohibitive.

Syndrome decoding

Fortunatelly, the algebraic structure of linear block codes allow simple decoding using the parity check matrix $\mathbf{H}$.

We may express $\mathbf{r} = \mathbf{c} + \mathbf{e}$, where $\mathbf{e}$ is an arbitrary error vector in $\mathcal{X}^n$. Applying the parity check matrix:

$$\mathbf{r}\mathbf{H}^T = (\mathbf{c} + \mathbf{e})\mathbf{H}^T = \mathbf{c}\mathbf{H}^T + \mathbf{e}\mathbf{H}^T = \mathbf{e}\mathbf{H}^T = \mathbf{s}$$

$\mathbf{s}$ is an $(n-k)$ dimensional vector called the syndrome. Note that there are 2^n possible errors and only 2^{n-k} syndromes so different error vectors $\mathbf{e}$ may result in the same syndrome.

Error correction is possible if using the standard array $\mathbf{D}$.

Syndrome decoding

Construction of the standard array $\mathbf{D}$: include in the first row all possible codewords $\mathbf{c}_i$ (they form a vector space). In the second row, put all vectors resulting from the sum of a coset leader $\mathbf{e}_2$ (a vector that is not a codeword) and the vectors in the first row. For the third row, take as coset leader $\mathbf{e}_3$ a vector that did not appear in the previous rows and form the third row by summing it to the vectors of the first row. Follow in this way until 2^{n-k} rows are completed:

$$\begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 & \dots & \mathbf{c}_{2^k} \\ \mathbf{e}_2 & \mathbf{c}_2+\mathbf{e}_2 & \mathbf{c}_3+\mathbf{e}_2 & \dots & \mathbf{c}_{2^k}+\mathbf{e}_2 \\ \mathbf{e}_3 & \mathbf{c}_2+\mathbf{e}_3 & \mathbf{c}_3+\mathbf{e}_3 & \dots & \mathbf{c}_{2^k}+\mathbf{e}_3 \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{e}_{2^{n-k}} & \mathbf{c}_2+\mathbf{e}_{2^{n-k}} & \mathbf{c}_3+\mathbf{e}_{2^{n-k}} & \dots & \mathbf{c}_{2^k}+\mathbf{e}_{2^{n-k}} \end{bmatrix}$$

Row j (called the j-th coset) contains all possible received vectors resulting from a particular error pattern $\mathbf{e}_j$, and forms an affine space.

Built in this way, $\mathbf{D}$ contains all possible vectors of $\mathcal{V}_n$.

Syndrome decoding

Using the standard array $\mathbf{D}$...

$$\begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 & \dots & \mathbf{c}_{2^k} \\ \mathbf{e}_2 & \mathbf{c}_2 + \mathbf{e}_2 & \mathbf{c}_3 + \mathbf{e}_2 & \dots & \mathbf{c}_{2^k} + \mathbf{e}_2 \\ \mathbf{e}_3 & \mathbf{c}_2 + \mathbf{e}_3 & \mathbf{c}_3 + \mathbf{e}_3 & \dots & \mathbf{c}_{2^k} + \mathbf{e}_3 \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{e}_{2^{n-k}} & \mathbf{c}_2 + \mathbf{e}_{2^{n-k}} & \mathbf{c}_3 + \mathbf{e}_{2^{n-k}} & \dots & \mathbf{c}_{2^k} + \mathbf{e}_{2^{n-k}} \end{bmatrix}$$

...how to correct errors?

i. Compute the syndrome on the received vector $\mathbf{s} = \mathbf{r}\mathbf{H}^T$

ii. Identify the coset leader with the same syndrome, $\mathbf{e}_m$

iii. Determine the transmitted codeword as $\hat{\mathbf{c}} = \mathbf{r} - \mathbf{e}_m$

Syndrome decoding

Theorem (6.0)

All the 2^k vectors of a coset have the same syndrome. The syndromes for different cosets are different.

Proof. By construction of $\mathbf{D}$ (see slide 19) every vector in a coset has the same syndrome

$$(\mathbf{e}_l + \mathbf{c}_i)\mathbf{H}^T = \mathbf{e}_l\mathbf{H}^T + \mathbf{c}_i\mathbf{H}^T = \mathbf{e}_l\mathbf{H}^T$$

which is the syndrome of the coset leader.

For the second part of the theorem, let $\mathbf{e}_j$ and $\mathbf{e}_l$ be the coset leaders of the jth and lth cosets, where $j < l$. Assume that the syndromes of these coset leaders are equal

$$\mathbf{e}_j\mathbf{H}^T = \mathbf{e}_l\mathbf{H}^T$$

$$(\mathbf{e}_j + \mathbf{e}_l)\mathbf{H}^T = \mathbf{0}.$$

This implies that $\mathbf{e}_j + \mathbf{e}_l$ is a codeword in $\mathcal{C}$, say $\mathbf{c}_i$. Thus, $\mathbf{e}_j + \mathbf{e}_l = \mathbf{c}_i$, and $\mathbf{e}_l = \mathbf{e}_j + \mathbf{c}_i$. This implies that $\mathbf{e}_l$ is in the jth coset, which contradicts the construction of $\mathbf{D}$. Therefore, no two cosets have the same syndrome. □

Syndrome decoding

Theorem (6.1)

Every (n, k) linear block code is capable of correcting up to 2^{n-k} error patterns.

Proof. Let $\mathbf{c}_j$ be the transmitted codeword. Only if the error vector induced by the channel $\mathbf{e}$ is one of the 2^{n-k} coset leaders, the received vector is in the j-th column of $\mathbf{D}$. Otherwise, $\mathbf{c}_j + \mathbf{e}$ will be associated to some syndrome, and hence to a different coset leader $\mathbf{e}_l$ plus a non-zero codeword $\mathbf{c}_i$ (remember all possible vectors in $\{0, 1\}^n$ appear in the array). Hence,

$$\mathbf{r} = \mathbf{c}_j + \mathbf{e} = \mathbf{c}_j + \mathbf{e}_l + \mathbf{c}_i = \mathbf{e}_l + (\mathbf{c}_j + \mathbf{c}_i) = \mathbf{e}_l + \mathbf{c}_s$$

and the decoded symbol is $\hat{\mathbf{c}} = \mathbf{c}_s \notin \mathcal{C}$. An error occurs. □

Syndrome decoding

Example (continued). For the $(5, 2)$ linear block code, with $d_{min} = 3$

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

the standard array and the syndromes are (codewords in red):

Standard array				Syndromes
00000	01101	10110	11011	000
00001	01100	10111	11010	011
00010	01111	10100	11001	101
00100	01001	10010	11111	001
01000	00101	11110	10011	010
10000	11101	00110	01011	100
11000	10101	01110	00011	110
10001	11100	00111	01010	111

Here, coset leaders are all vectors of weight 1 plus two vectors of weight 2 selected among those having different syndrome. Note that if $d_{min} = 1$, no vector of weight 1 could have been used as a coset leader. Why? If a coset leader lead us to another codeword, errors cannot be detected.

Syndrome decoding

Syndrome decoding is optimal if coset leaders are properly chosen.

Theorem (6.2)

If the coset leaders are the minimum weight vectors, syndrome decoding minimizes the probability of error.

Proof. Let $\mathbf{r}$ be the received vector. If located in the i-th column of $\mathbf{D}$ and in the l-th coset, then $\mathbf{r}$ is decoded into $\mathbf{c}_i$. Because $\mathbf{r} = \mathbf{e}_l + \mathbf{c}_i$, the distance between $\mathbf{r}$ and $\mathbf{c}_i$ is

$$d(\mathbf{r}, \mathbf{c}_i) = w(\mathbf{r} + \mathbf{c}_i) = w(\mathbf{e}_l + \mathbf{c}_i + \mathbf{c}_i) = w(\mathbf{e}_l)$$

Consider now the distance to another codeword $\mathbf{c}_j$

$$d(\mathbf{r}, \mathbf{c}_j) = w(\mathbf{r} + \mathbf{c}_j) = w(\mathbf{e}_l + \mathbf{c}_i + \mathbf{c}_j) = w(\mathbf{e}_l + \mathbf{c}_s) \geq w(\mathbf{e}_l) = d(\mathbf{r}, \mathbf{c}_i)$$

where the inequality appears if coset leaders are chosen of minimum weight. Finally, check the optimum receiver in slide 17. □

Note that minimum weight errors are the most likely to occur in a BSC.

Syndrome decoding

We are interested in codes with large d_{min}.

Theorem (6.3)

For a (n, k) linear block code $\mathcal{C}$ with minimum distance d_{min}, all the vectors of weight $t = \lfloor (d_{min} - 1)/2 \rfloor$ or less can be used as coset leaders.

Proof. Let $\mathbf{x}, \mathbf{y}$ be two vectors of weight $t = \lfloor (d_{min} - 1)/2 \rfloor$ or less. The weight of the sum is

$$d(\mathbf{x}, \mathbf{y}) = w(\mathbf{x} + \mathbf{y}) \leq w(\mathbf{x}) + w(\mathbf{y}) \leq 2t \leq d_{min}$$

If $\mathbf{x}$ and $\mathbf{y}$ are in the same coset, then $\mathbf{x} = \mathbf{c}_i + \mathbf{e}_k$ and $\mathbf{y} = \mathbf{c}_j + \mathbf{e}_k$ and hence the sum must belong to $\mathcal{C}$ by linearity of the code. But this is impossible because the weight is less than d_{min}. Then, no two vectors of weight less than or equal to t can be in the same coset, and hence all coset leaders can be vectors of weight t or less. □

Syndrome decoding

A block code of minimum distance d_{min} can correct all errors of weight equal or lower than t, and cannot correct all errors of weight $t+1$.

Theorem (6.4)

For an (n,k) linear block code $\mathcal{C}$ with minimum distance d_{min}, if all vectors of weight t or less are used as coset leaders, there is at least one vector of weight $t+1$ that cannot be used as a coset leader.

Proof. Let $\mathbf{v} \in \mathcal{C}$ such that $w(\mathbf{v}) = d_{min}$. Let $\mathbf{x}, \mathbf{y}$ be two vectors satisfying

1. $\mathbf{x} + \mathbf{y} = \mathbf{v}$
2. $\mathbf{x}, \mathbf{y}$ do not have non-zero components in common places.

By 1) and linearity of the code, $\mathbf{x}, \mathbf{y}$ must be in the same coset, and by 2)

$$w(\mathbf{x}) + w(\mathbf{y}) = w(\mathbf{v}) = d_{min}$$

Since $2t+1 \leq d_{min} \leq 2t+2$, by theorem (6.3) if $w(\mathbf{y}) = t+1$ then $w(\mathbf{x})$ must be t or $t+1$. Since both are in the same coset, if $\mathbf{x}$ is chosen as coset leader then $\mathbf{y}$ cannot be chosen. □

Error detection and error correction capabilites

Example (continued). Note that $d_{min} = 3$ and $t = 1$. Let $\mathbf{c} = 01101$ be the transmitted codeword and the error induced by the channel is $\mathbf{e} = 10100$ (not a coset leader). The syndrome is $(\mathbf{c} + \mathbf{e})\mathbf{H}^T = 101$ and the decoded vector is $\mathbf{r} + \mathbf{e}_3 = 11011 \neq \mathbf{c}$. An error is detected but not corrected.
□

Looking into the structure of $\mathbf{D}$ and the theorems above, a block code can

- Not detect $2^k - 1$ errors: those codewords altered into another codeword.
- Detect $2^n - 2^k$ errors: those generating a non-zero syndrome vector; for large n, $2^k - 1 \ll 2^n - 2^k$.
- Correct 2^{n-k} errors: those errors equal to a coset leader; they are a small fraction of those detected for large n.

Block error probability

The probability of error in a codeword is given by the sum of the probability of undetected errors $P_{ud}(\mathcal{E})$ plus de probability of detect but not being able to correct errors $P_{uc}(\mathcal{E})$:

$$\begin{aligned} P(\mathcal{E}) &= P_{ud}(\mathcal{E}) + P_{uc}(\mathcal{E}) \\ &= \Pr(\mathcal{E}|no\,detect)\Pr(no\,detect) + \Pr(\mathcal{E}|detect)\Pr(detect) \end{aligned}$$

Note that this is an upper bound of the bit error probability. Let us evaluate the four terms:

- $\Pr(\mathcal{E}|no\,detect) = 1$, if the error is not detected, it cannot be corrected!
- $\Pr(no\,detect)$ is given in next slide as $P_u(\mathcal{E})$. Depends on the code and on the channel transition probabilities.
- $\Pr(\mathcal{E}|detect)$ is given in the following slides as $P_d(\mathcal{E})$.
- $\Pr(detect) = 1 - \Pr(no\,detect)$

Probability of undetected errors

Define the weight enumerator function (WEF) $A_j(i)$ of the code $\mathcal{C}_j$ as the number of codewords having weight i. The probability of undetected errors for the binary symmetric channel is

$$P_u(\mathcal{E}) = \sum_{i=1}^{n} A_j(i) p^i (1-p)^{n-i}$$

The WEF is known for many codes.

We can compute an upper bound for the undetected errors probability on the whole set Γ of possible (n, k) linear block codes. Since all are constructed using the generator matrix $\mathbf{G}$, there are $2^{k(n-k)}$ possible parity matrices $\mathbf{P}$.

Let us compute a bound on this error for the ensemble of linear block codes so that we can get a bound. If code is selected randomly, $P(\mathcal{C}_j) = 2^{-k(n-k)}$ and then the average probability of error is:

$$\bar{P}_u(\mathcal{E}) = \sum_{j=1}^{|\Gamma|} P(\mathcal{C}_j) P_u(\mathcal{E}|\mathcal{C}_j) = 2^{-k(n-k)} \sum_{j=1}^{n} p^i (1-p)^{n-i} \sum_{j=1}^{|\Gamma|} A_j(i)$$

Probability of undetected errors

A nonzero vector is contained in $2^{(k-1)(n-k)}$ codes or in none of the codes (why?). Because there are $\binom{n}{i}$ vectors of weight i, we have

$$\sum_{j=1}^{|\Gamma|} A_j(i) \leq \binom{n}{i} 2^{(k-1)(n-k)}$$

and hence

$$\bar{P}_u(\mathcal{E}) \leq 2^{-(n-k)} \sum_{i=1}^{n} \binom{n}{i} p^i (1-p)^{n-i} = 2^{-(n-k)} (1-(1-p)^n) \leq 2^{-(n-k)}$$

That is, there exists codes $\mathcal{C}_k$ whose $P_u(\mathcal{E}|\mathcal{C}_k)$ is decreasing faster than exponentially with the number of parity check bits $(n-k)$.

Only a few codes have been discovered to reach this bound.

Probability of detected and non corrected errors

A linear block code $\mathcal{C}$ can correct all errors of weight $t = \lfloor (d_{min} - 1)/2 \rfloor$ and some of weight $t + 1$. The probability of having detected an error but not being able to correct it is:

$$P_d(\mathcal{E}) \leq \sum_{i=t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

Inequality comes from coset leaders of weight larger than t.

Assume a sphere of radius t around each possible codeword. The number of vectors of $\{0,1\}^n$ in the sphere is

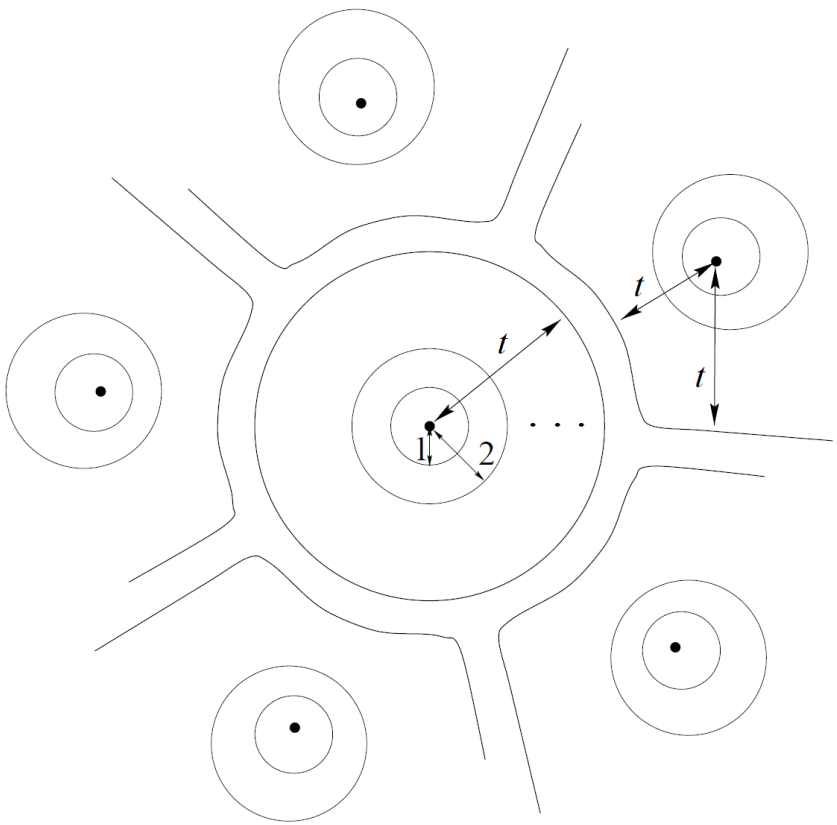
$$1 + \binom{n}{1} + \binom{n}{1} + \cdots + \binom{n}{t} = \sum_{i=0}^{t} \binom{n}{i}$$

Since we have 2^k possible spheres and a total of 2^n vectors, the following inequality must hold

$$2^k \sum_{i=0}^{t} \binom{n}{i} \leq 2^n$$

Probability of detected and non corrected errors

In a perfect code this expression holds with equality, and no error vector of weight $t+1$ can be corrected. The spheres around each codeword fully cover all vectors in the n-dimensional space and do not overlap. This result in minimum probability of error among all codes of the same (n, k) values.



Part of the Hamming space perfectly filled by spheres of radius t centered on the codewords of a perfect code.

Probability of detected and non corrected errors

Exact expressions for the probability of error can also be obtained for quasi-perfect codes, for which all spheres of radius t are disjoint and the spheres of radius $t+1$ overlap (as a consequence of the weight of some coset leaders is $t+1$). No more than $t+1$ errors can be corrected.

A quasi-perfect code can be derived from a perfect code by adding a parity check digit to the end of each codeword, so if a (n, k) code is a perfect binary linear code of minimum distance d, then the code thus constructed is a $(n+1, k)$ nearly perfect binary linear code of minimum distance $d+1$.

Probability of detected and non corrected errors

For quasi perfect codes, the total number of vectors outside the spheres is

$$N_{t+1} = 2^n - 2^k \sum_{i=0}^{t} \binom{n}{i}$$

that we can equally divide among the 2^k codewords so that each sphere is enlarged by adding

$$\beta_{t+1} = 2^{n-k} - \sum_{i=0}^{t} \binom{n}{i}$$

vectors of distance $t+1$ from the transmitted codeword. Consequently, from all $\binom{n}{t+1}$ error vectors of distance $t+1$ we can correct β_{t+1}. Thus the error probability is given by

$$P_d(\mathcal{E}) = \sum_{i=t+2}^{n} \binom{n}{i} p^i (1-p)^{n-i} + \left[\binom{n}{t+1} - \beta_{t+1} \right] p^{t+1} (1-p)^{n-t-1}$$

Perfect codes

The only perfect codes known so far are:

- Hamming codes
- Golay $(23, 12)$ codes
- Trivial $(n, 2)$ codes with n odd and $d_{min} = 2$

Why are they so rare? It is hard to pack spheres in high dimensional spaces.

Bounds on d_{min}

Given that t determines the error rate performance of the code, we need to compute a value for d_{min} given a code. Unfortunately there is no simple way of computing it, and many bounds have been derived. The bound shown below provide necessary conditions for R_c given a desired distance d_{min}.

- Singleton bound
  Use the bound $d_{min} \leq n - k + 1$. Since $d_{min} - 1 \simeq 2t$, a high power correction is achieved if $n - k$ is large, at least, $n - k \geq 2t$. Codes for which this bound is met with equality are called Maximum Distance Separable codes (only repetition codes and Reed-Solomon codes).

$$\delta_{min} = \frac{d_{min}}{n} \leq 1 - R_c + \frac{1}{n}$$

for large n

$$\delta_{min} \leq 1 - R_c$$

Bounds on d_{min}

- Hamming bound
  Depart from the inequality

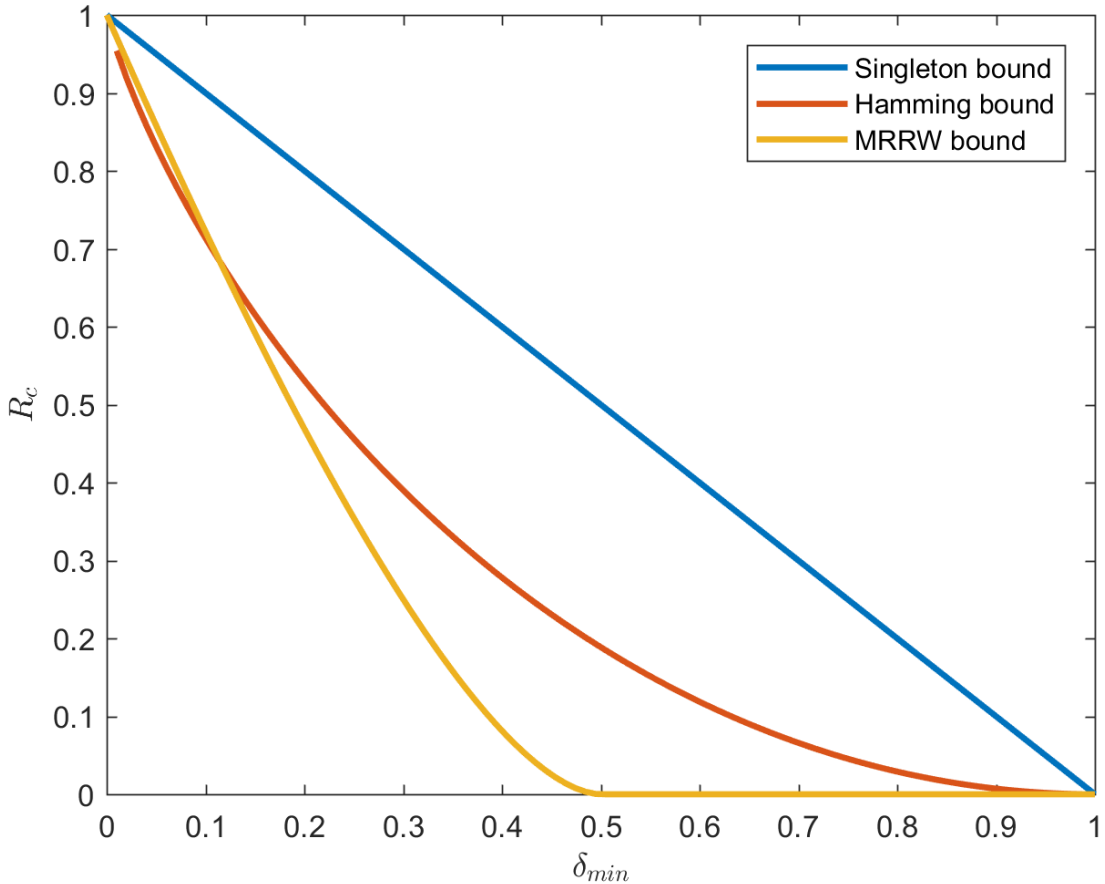
$$2^{(n-k)} \geq \sum_{i=0}^{t} \binom{n}{i}$$

and take logarithm on both sides

$$1 - R_c \geq \frac{1}{n} \log \sum_{i=0}^{t} \binom{n}{i} \simeq \frac{1}{n} \log 2^{nH\left(\frac{t}{n}\right)} = H\left(\frac{t}{n}\right) \simeq H\left(\frac{\delta_{min}}{2}\right)$$

where approximations are for large n, using Newton's binomial and Stirling's approximation. The bound is tight for perfect codes.

Bounds on d_{min}



Asymptotic bounds for R_c versus δ_{min}: high error correction capabilities imply reduced code rate.

Outstanding block codes

Some well known and widely used block codes are:

- Hamming codes
- Cyclic codes
- Golay $(23, 12)$ codes
- Reed-Solomon codes
- BCH codes
- Low Density Parity Check (LDPC) codes

Hamming codes

Richard Hamming

In 1950, only two years after Shannon's paper, the mathematician *Richard Hamming*, office mate of Shannon at Bell Telephone Labs, asserted that by proper encoding of information, errors induced by a noisy channel can be reduced to any desired level without sacrificing the rate of information. Coding theory started with him.

Hamming codes have been used over the years for their high rate and low complexity of decoding.

Hamming codes

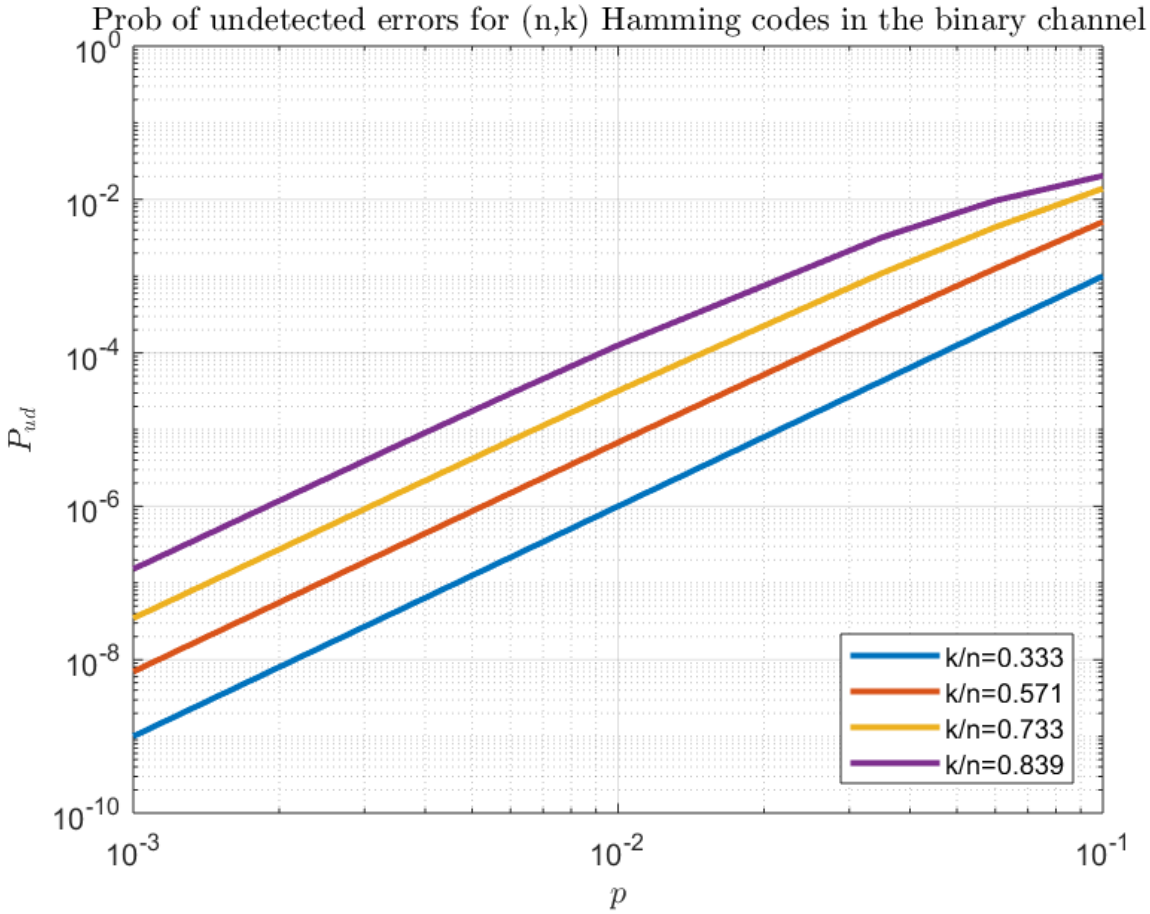
These are codes of $d_{min} = 3$. For any $m = n - k \geq 3$ there is a code such that

- Codeword length: $n = 2^m - 1$
- Information symbols: $k = 2^m - m - 1$
- Error correcting capability: $t = 1$
- Rate: $R_c = 1 - \frac{m}{2^m - 1}$
- Parity check matrix $\mathbf{H}$: its $2^m - 1$ columns contain all integers of m binary digits except zero.
- Weight enumerator function, is the set of coefficients of the polynomial

$$A(z) = \frac{1}{n+1}\left[(1+z)^n + n(1-z)(1-z^2)^{(n-1)/2}\right]$$

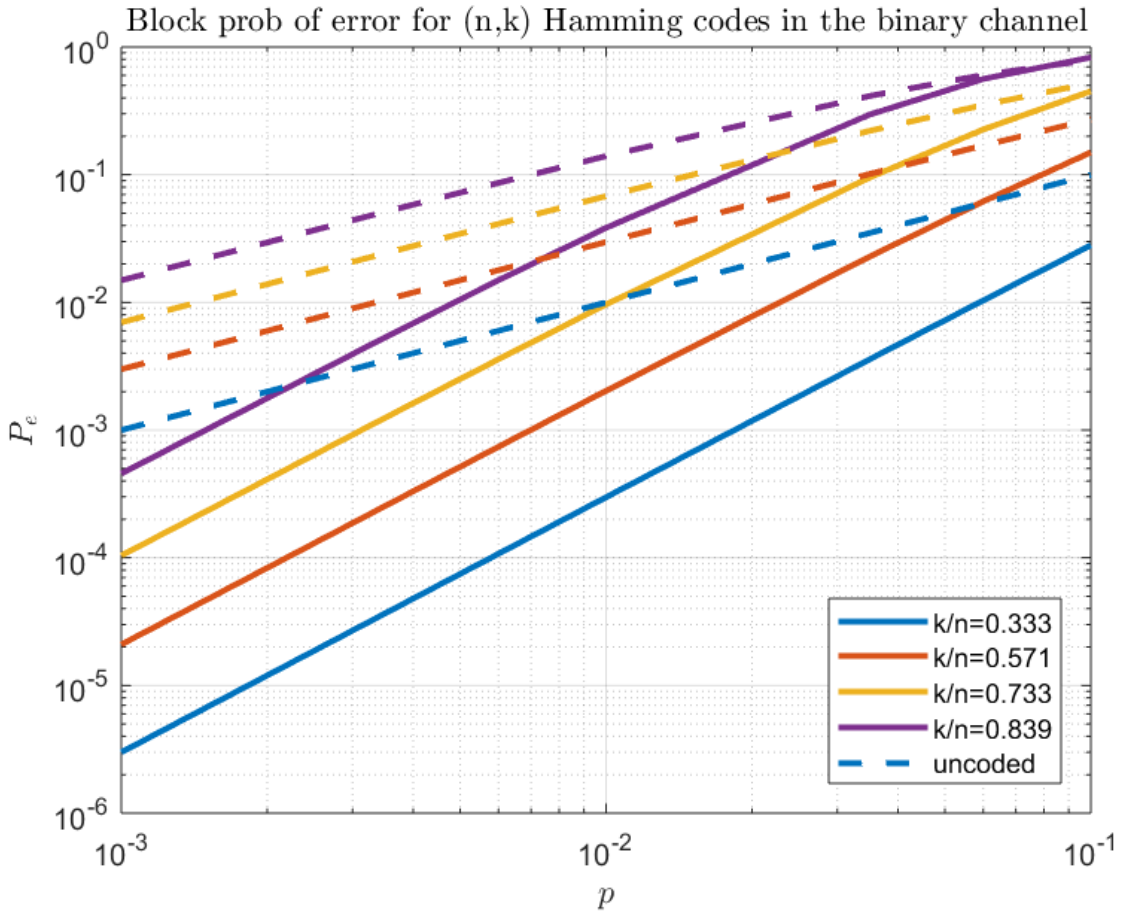
Probabilities of error and of undetected errors can be readily obtained.

Hamming codes



Probability of undetected errors for several (n, k) Hamming codes versus transition probability of a BSC

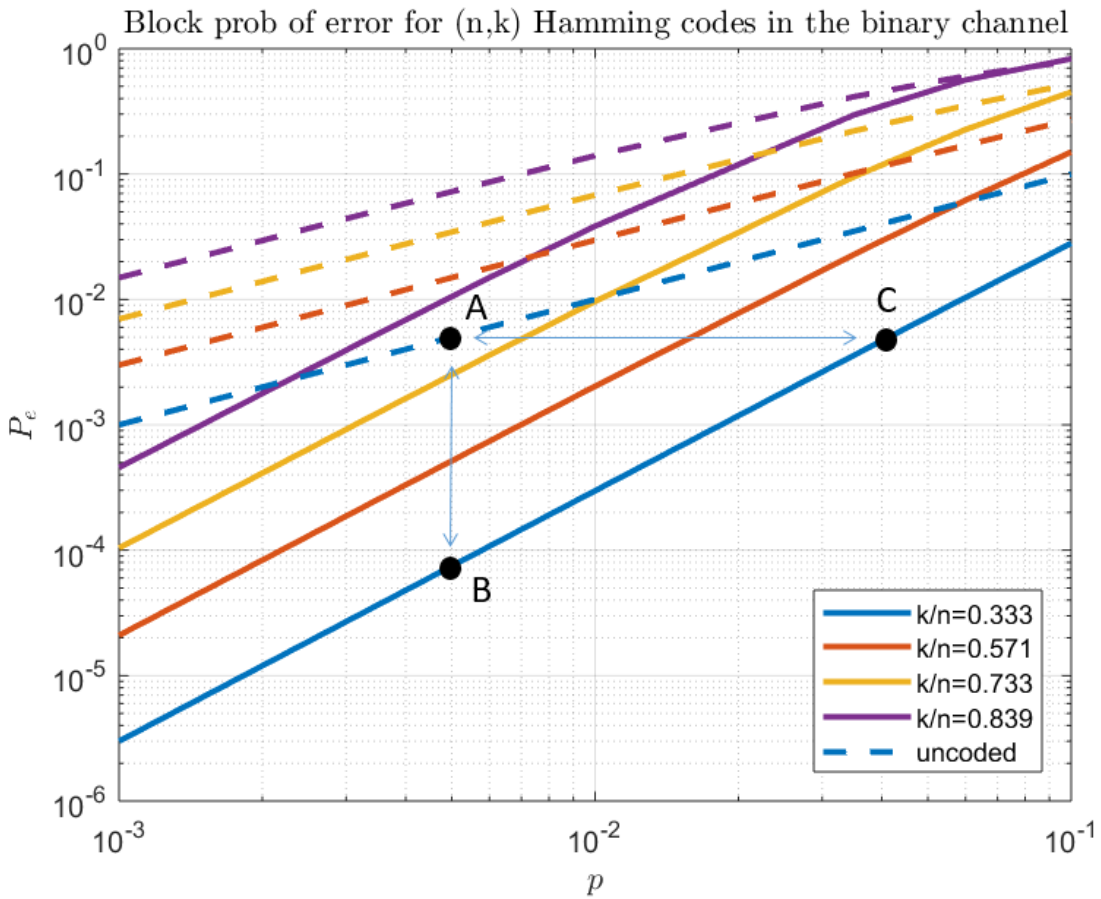
Hamming codes



Block probability of detected and not corrected errors, for several (n, k) Hamming codes versus transition probability of a BSC. Uncoded block probability in dashed line.

Benefits of channel coding

- $A \to B$ Given a BSC with a certain p, we can reduce error by increasing redundancy (reducing bit transmission rate)
- $A \to C$ Given a probability of error for the BSC, we can tolerate worse channel p values, by increasing redundancy.



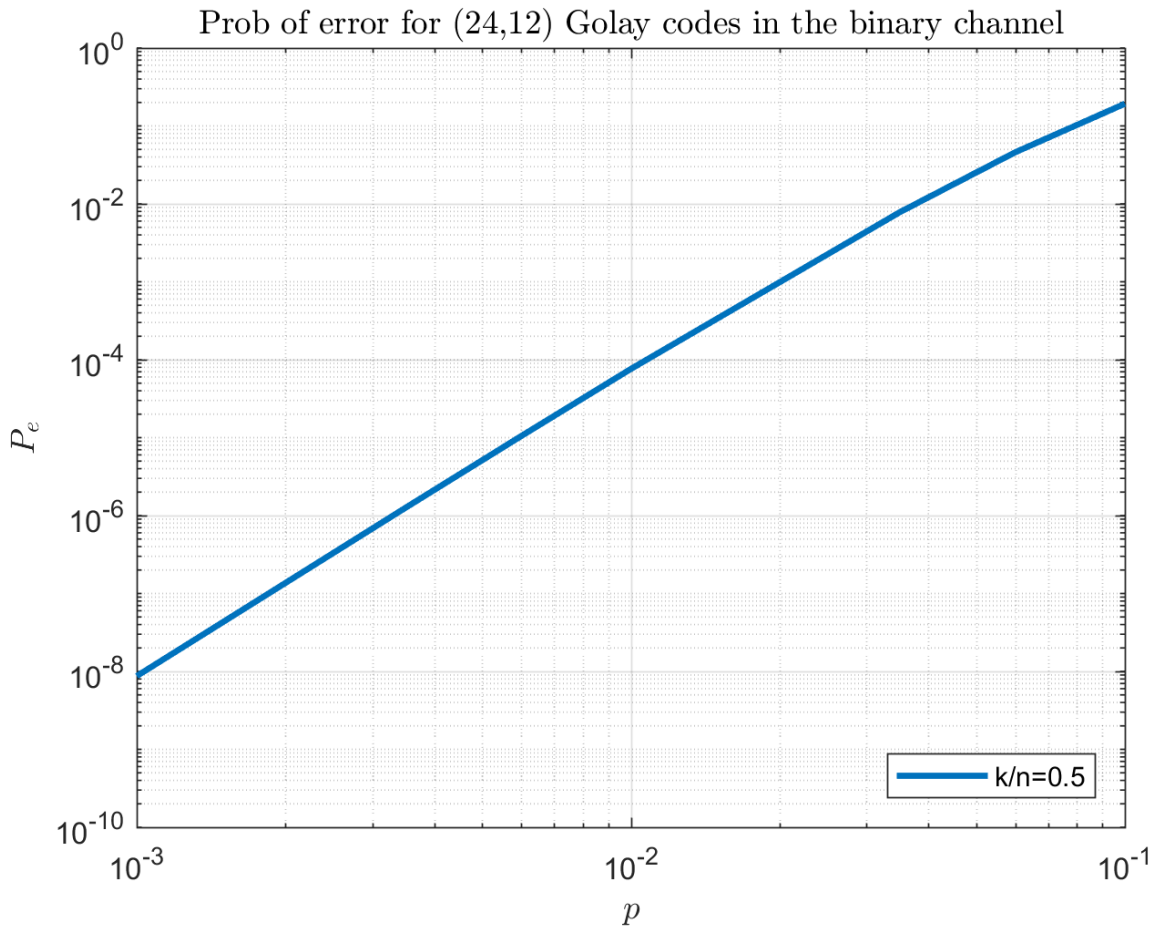
Golay $(23, 12)$ code

Derived by the French mathematician *Marcel Golay* in 1949, it is a cyclic code: if $\mathbf{c}$ is a codeword, any cyclic shift of $\mathbf{c}$ is also a codeword. $d_{min} = 7$.

- Codeword length: 23
- Information symbols: 12
- Error correcting capability: $t = 3$
- Rate: $R_c = 12/23$
- Weight enumerator function, known.

Usually a parity check bit is added, and then $n = 24$ and $d_{min} = 8$, in this case it can detect up to 4 errors and correct 3 or less.

Golay $(23, 12)$ code



Probability of detected and not corrected errors for the Golay code versus transition probability of a BSC

Reed-Solomon codes

Derived by the US mathematicians and engineers Irving Reed and Gustave Solomon in 1960, probably the most widely used codes. They are cyclic codes, and have the largest possible d_{min} among all linear block codes of a given (n, k) and are defined in $GF(2^k)$.

- Alphabet size: $q = 2^k$ symbols
- Codeword length: $N = q - 1 = 2^k - 1$
- Information symbols: $K = 1, 2, \ldots, N - 1$
- Minimum distance: $D_{min} = N - K + 1$
- Error correcting capability: $t = \frac{D_{min}-1}{2}$
- Rate: $R_c = K/N$
- Weight enumerator function, known.

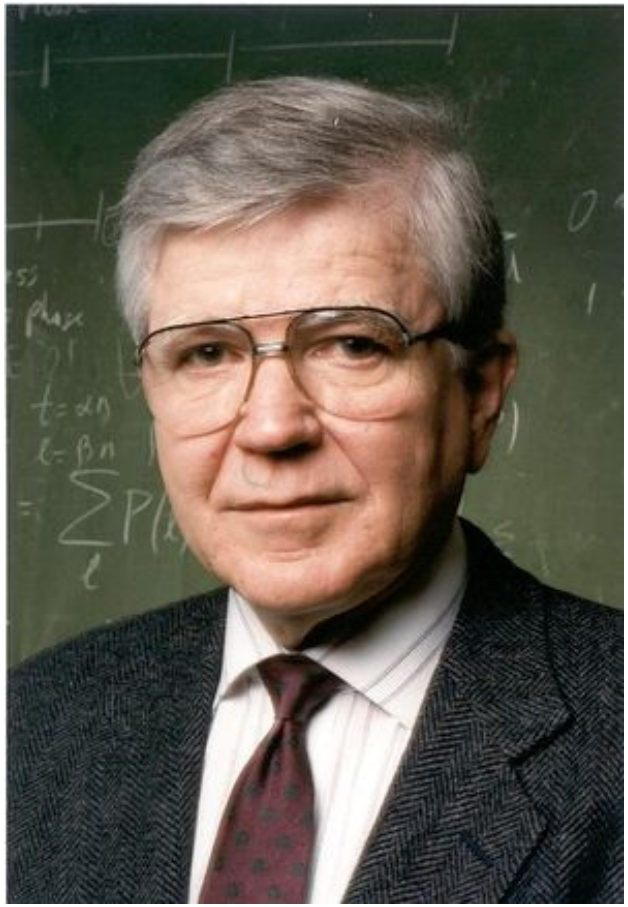
Reed-Solomon codes

Example. To explore the advantages of nonbinary codes consider these two codes:

- Binary (7,3) code. 2^7 possible codewords, only 2^3 are used, i.e. $\frac{1}{16}$ of the vectors.
- Ternary (7,3) code (defined in $GF(q), q = 3$). There are $2^{7q} = 2^{21} = 2,097,152$ input binary words, of which $2^{3q} = 2^9 = 512$ are used, i.e. only $\frac{1}{4096}$ of the vectors.

Since a small fraction of vectors are used as codewords, large values of D_{min} can be obtained.

LDPC codes



Robert Gallager

Discovered by *Robert Gallager* in this PhD thesis of 1960. They have good minimum distance[1] and are able to approach Shannon capacity bound.

The parity check matrix has a low density of ones. In *regular LDPC codes*, $\mathbf{H}$ contains w_c ones in each column and $w_r = w_c \frac{n}{n-k}$ ones in each row, where $w_c \ll min(n, n-k)$, $w_r \ll min(n, n-k)$. The code rate is

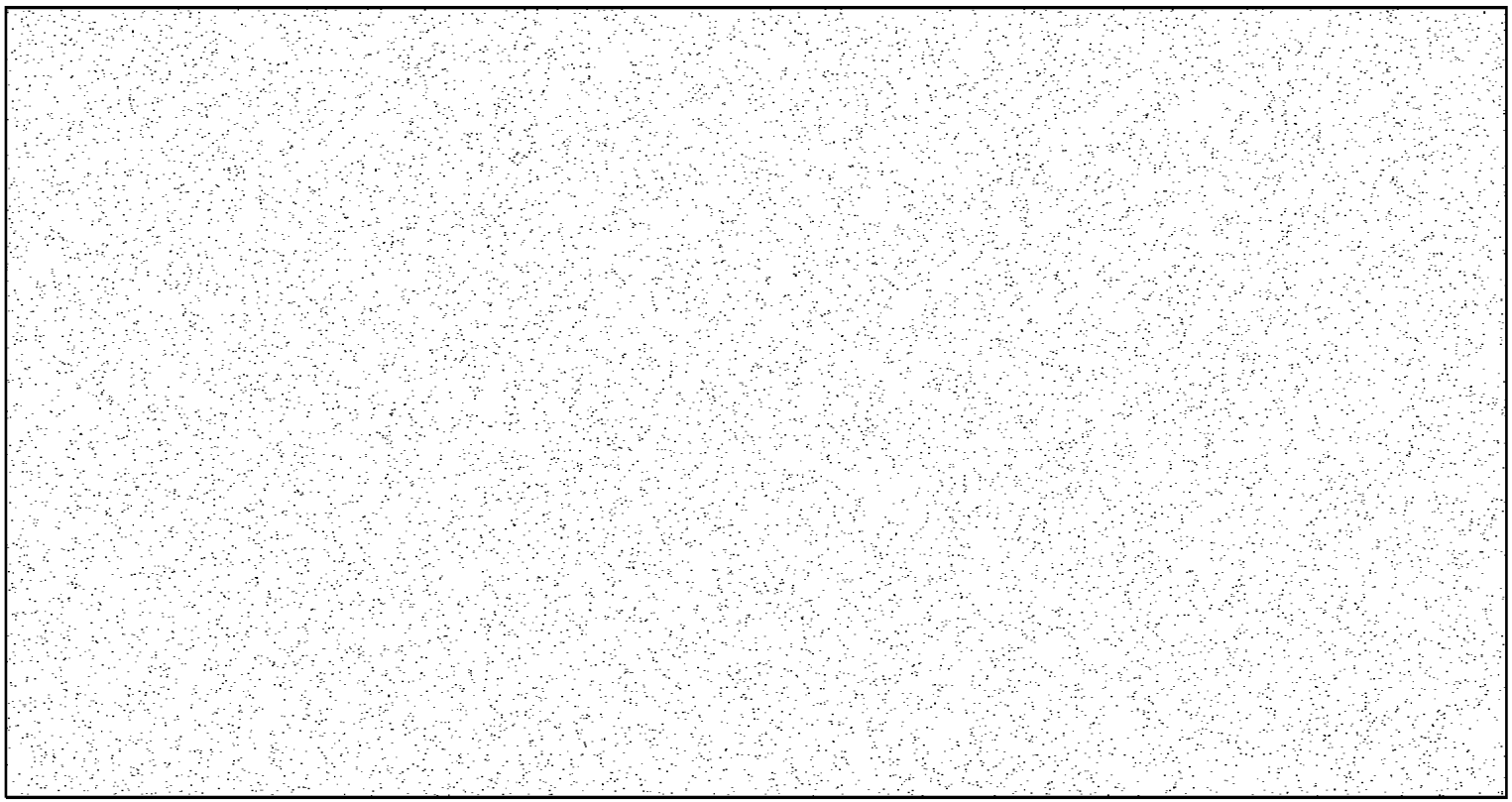
$$R = k/n = 1 - w_c/w_r.$$

LDPC codes perform near the Shannon limit for large block lengths. For example, simulations show a bit error rate of 10^{-6} for a rate of 90% of channel capacity, with block length of 10^7.

[1] X.Y. Hu, M.P.C. Fossorier, E. Eleftheriou, C. Di, T. Richardson and R. Urbanke, "On the Computation of the Minimum Distance of Low-Density Parity-Check Codes" IEEE Intl. Conference on Communications (ICC), Paris, France, 20-24 June 2004.

LDPC codes

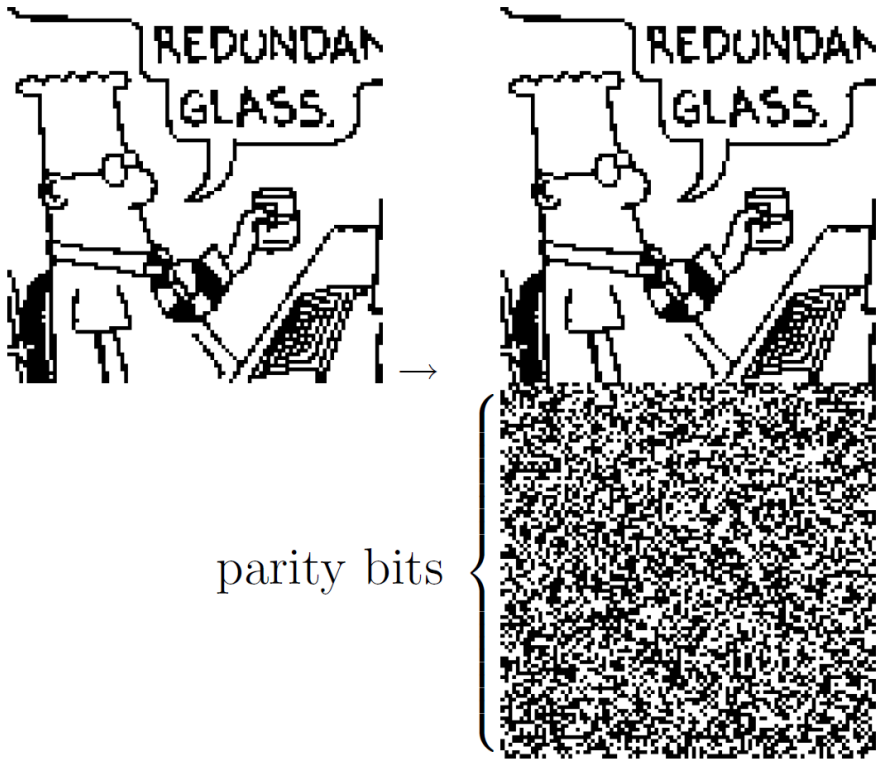
$\mathbf{H} =$



A low-density parity-check matrix with $n = 20000$ columns of weight $w_c = 3$ and $n - k = 10000$ rows of weight $w_r = 6$.

Examples taken from D. Mackay, *Information Theory, Inference, and Learning Algorithms*, Cambridge Press, 2003

LDPC codes



Systematic and parity bits for the encoded source.
The parity bits depend on nearly half the input bits.

LDPC codes

- Encoding. It is known that only irregular LDPC codes are able to achieve close to capacity rates at low probability of error[1]. These codes have a variable number of ones in each row and column of $\mathbf{H}$.
- Decoding. Optimum decoding is an NP-hard problem. Many suboptimum iterative algorithms exist that take advantage of the sparse structure of matrix $\mathbf{H}$.

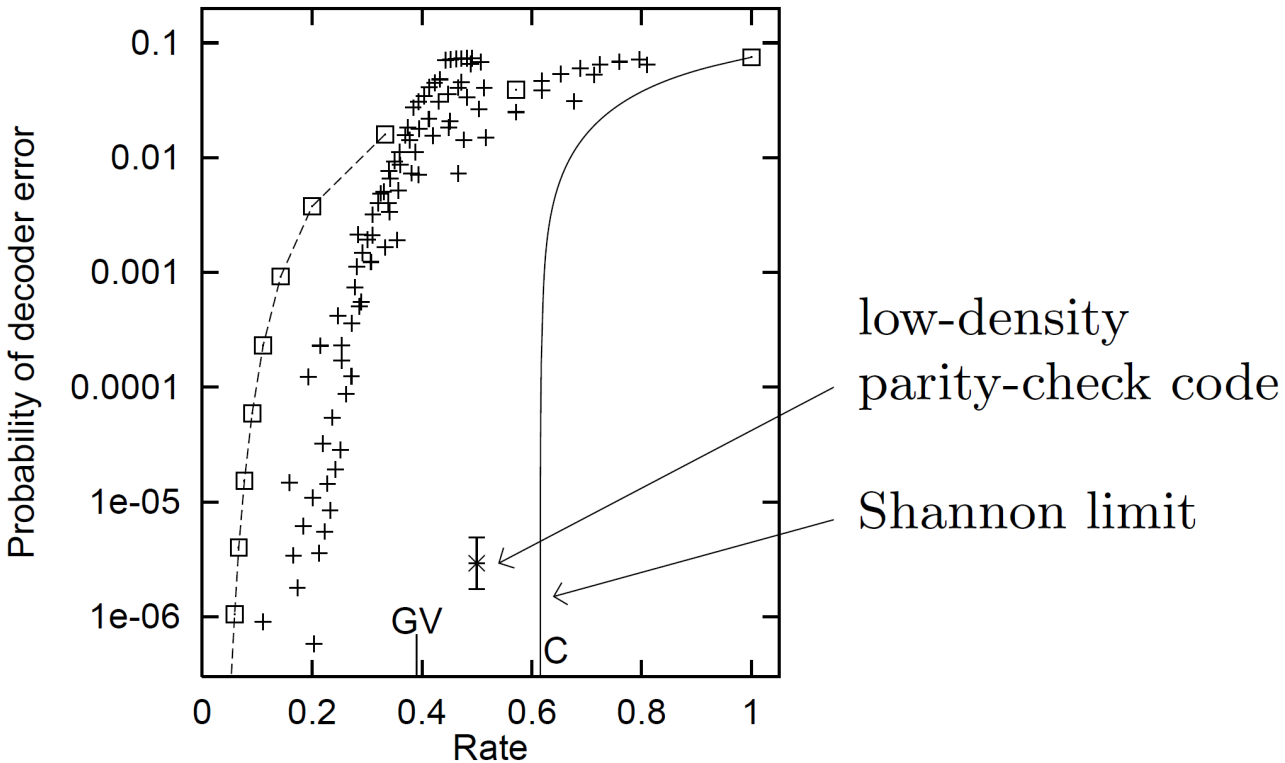
  The bit-flipping family of algorithms operate basically as follows:

  1: compute the syndrome $\mathbf{s} = \mathbf{r}\mathbf{H}^T$
  2: `iter=0`
  3: `if` ($\mathbf{s} == \mathbf{0}$ `or iter > max_num_iter`) `then stop`
  4: `otherwise`, take the nonzero components of $\mathbf{s}$ and flip those components of $\mathbf{r}$ that appear in the largest number of unsatisfied parity check equations
  5: `iter=iter+1`
  6: `go to 1`

[1] S.Y. Chung, G.D. Forney Jr., T. Richardson, R. Urbanke, "On the Design of Low-Density Parity-Check Codes within 0.0045 dB of the Shannon Limit," IEEE Commun. Lett., vol. 5, pp. 58–60, 2001

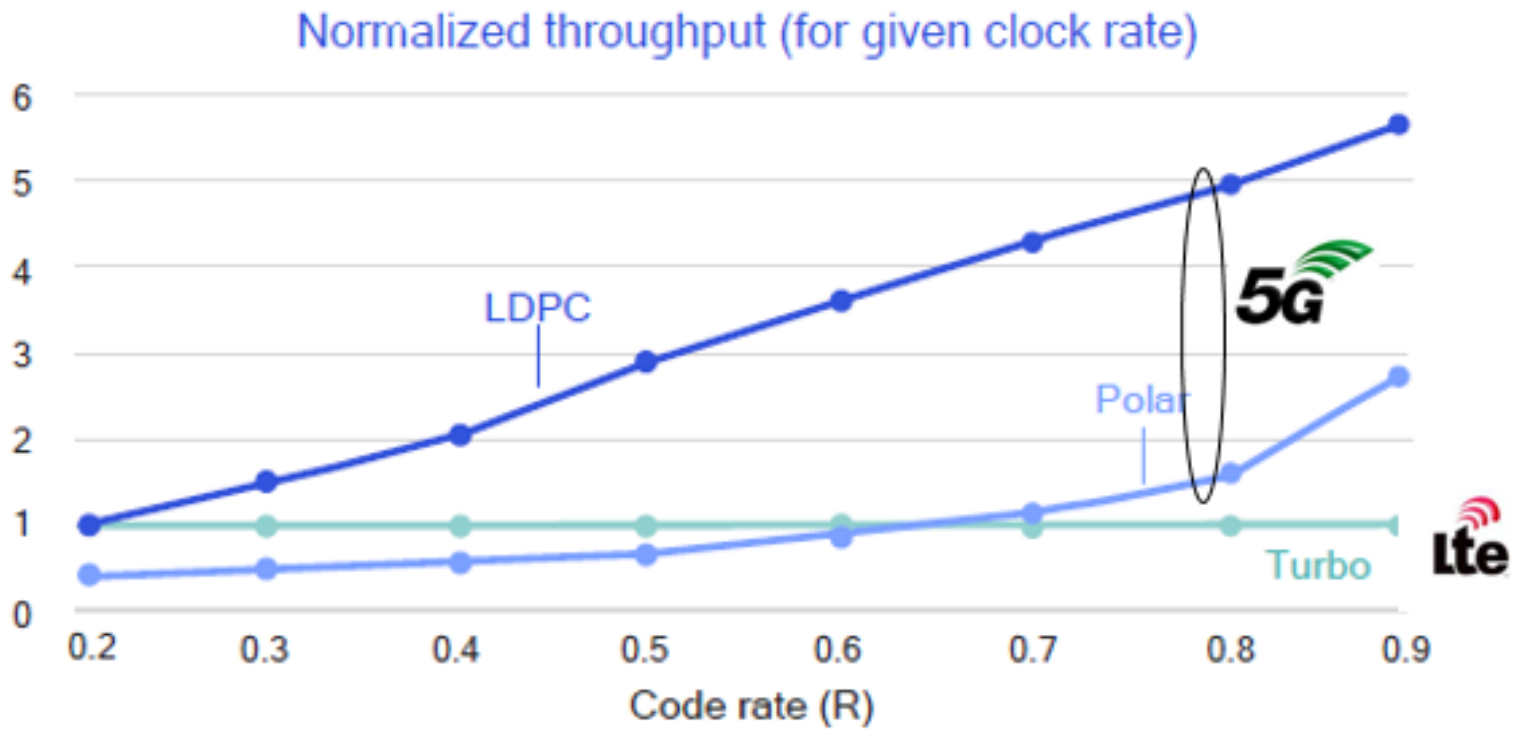
Iterative decoding of an LDPC code for a transmission received over a BSC channel transition probability $q = 0.075$. The decoder halts after the $13th$ iteration when the best guess violates no parity checks. The final decoding is error free.

LDPC codes



Error probability of the low-density parity-check code for the BSC with $q = 0.075$, compared with algebraic codes. Squares: repetition codes and Hamming (7,4) code. Crosses: BCH codes.

LDPC codes



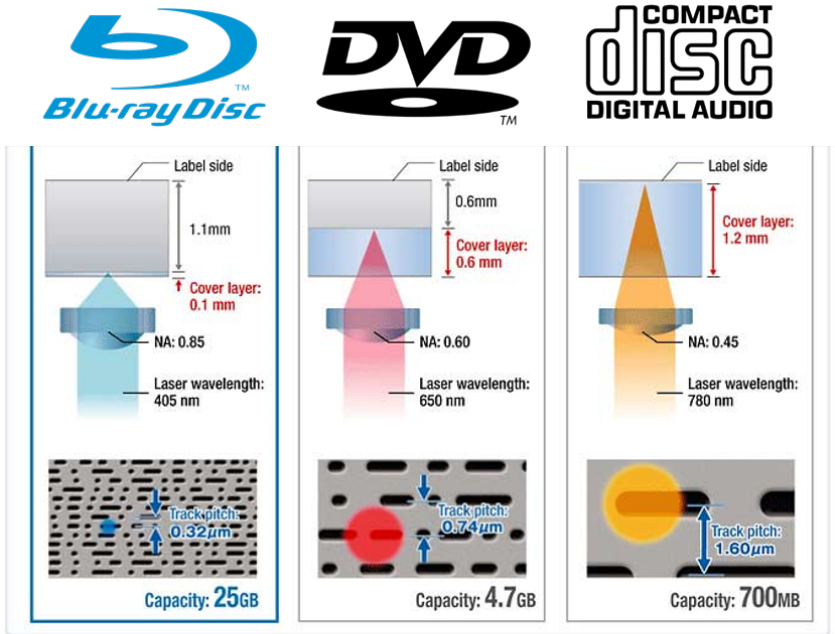
Codes in wireless communication systems. In 5G, LDPC have been selected for user data and polar codes for control data. In 4G, turbo codes were selected. The advantage of LDPC is that the clock rate of the hardware does not have to be increased with the rate of the code increases.

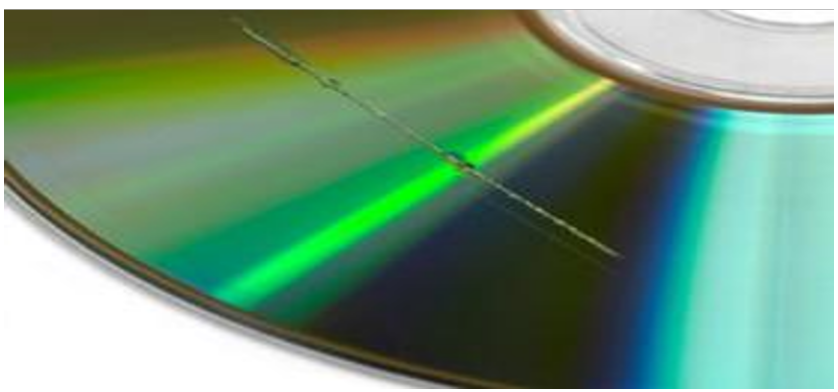
Interleaving

Most of codes have been designed for statistically independent errors (like in BSC). In many channels however, errors appear in a bursty way, like in cellular communication system, or when reading an optical support with damaged surface.

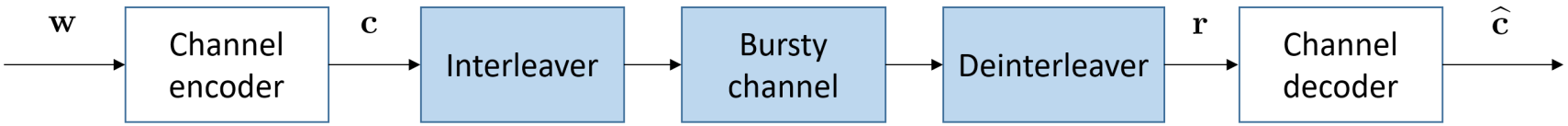
A burst of errors of length b is a sequence of b-bit errors where the first and last are 1. An (n, k) linear block code can correct bursts of length $b \leq \lfloor \frac{1}{2}(n-k) \rfloor$.

A way to deal with bursty errors is to interleave the data, so that the channel is turned into a channel with independent errors, and use a conventional error correcting code:

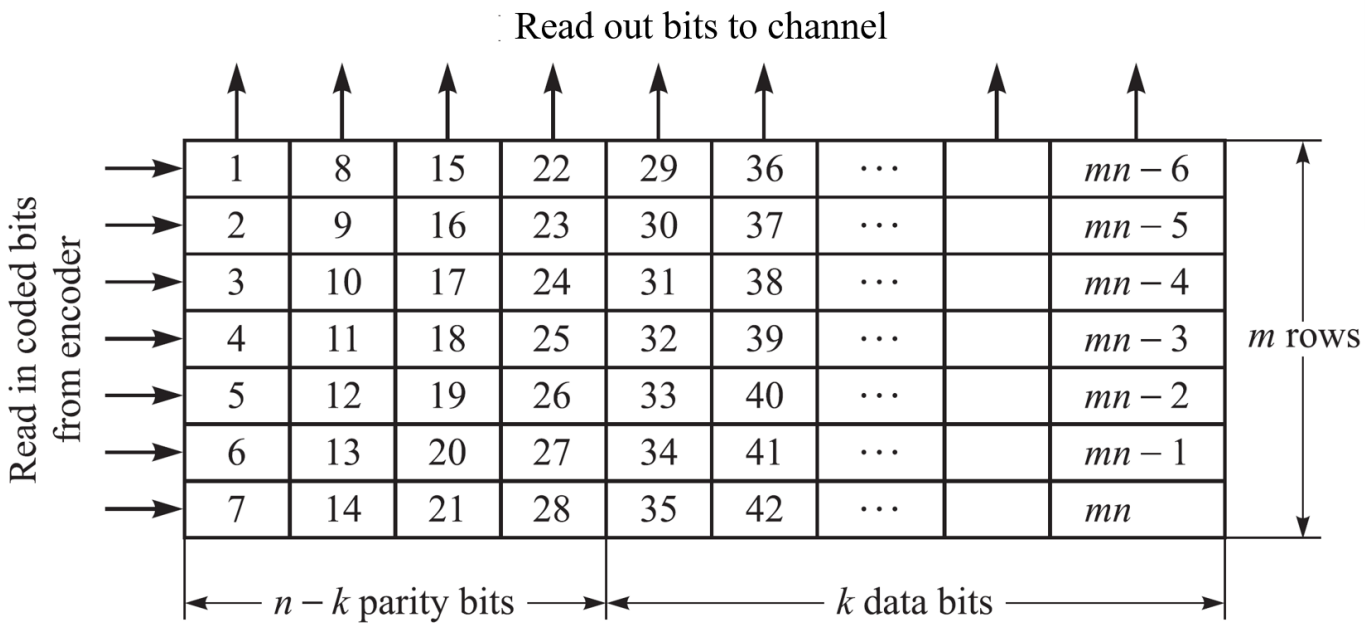




Interleaving



The interleaver reorders the encoded data and transmits it over the channel. At the receiver, the deinterleaver reorders the data and passes it to the decoder. As a result, errors are spread out in time and appear independent. A block interleaver of degree m, is formed by m rows and n columns...



Interleaving

At the deinterleaver, symbols are stored in the matrix and read row-wise. As a result, a burst of $l = mb$ errors is broken into m bursts of length b.

Thus, an (n, k) code that can handle bursts of errors of length b can be combined with an interleaver to create an interleaved (mn, mk) block code that can handle bursts of length mb.

Example. In Compact Disk, a Reed-Solomon $R_c = 24/28$ is concatenated with an interleaver and another Reed-Solomon code of rate $R_c = 28/32$ (Cross Interleave Reed Solomon Code - CIRC). The overall code rate of $3/4$, and $d_{min} = 5$. Codes are designed in the Galois field $GF(2^8)$.

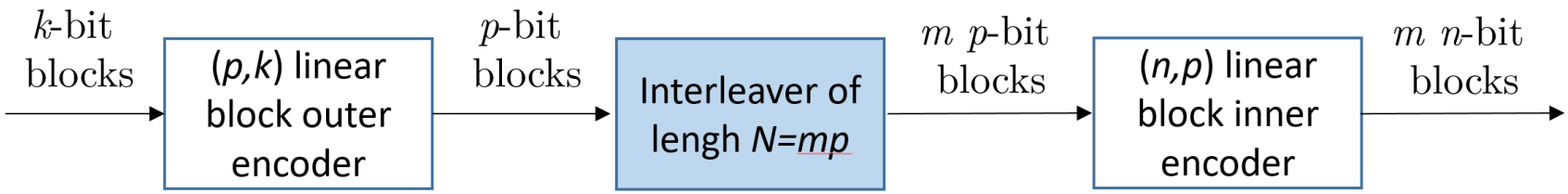
The maximum fully correctable burst length of CIRC is up to 4000 bits (2.5mm of disk surface). After CIRC, the residual error rate is less than 10^{-9}, which is enough for CD.

Interleaving

For CD-ROM, this rate is too high so the standard of CD-ROM had added the EDC (Error Detection Code) and ECC (Error Correction Code). This method can make the error rate is less than 10^{-13} which satisfy the requirement of CD-ROM.

Concatenated codes

Code concatenation, together with interleaving, is used to increase the codeword length and improve error performance. The minumum distance is the product of minimum distances.



Although not optimal, decoding is usually done sequentially for each code.

Wrap up

- Coding is an intrinsic part of any information communication or storage system.
- The number of codes proposed since 1950 is huge, the use of one or another depends on the performance sought (in terms of rate and probability of error), the conditions of the channel (error induced, bursty or not bursty) and the complexity allowed for decoding.
- Design of codes and efficient decoding algorithms an active area of research.

Recommended bibliography:
S. Lin, D. J, Costello, *Error Control Coding*, Second edition, Prentice Hall, 2004