Information-Theoretic Modeling

Lecture 4: Noisy Channel Coding

Jyrki Kivinen

Department of Computer Science, University of Helsinki

Autumn 2012
Lecture 4: Noisy Channel Coding
1 Noisy Channels

- Reliable communication
- Error correcting codes
- Repetition codes
1. Noisy Channels
   - Reliable communication
   - Error correcting codes
   - Repetition codes

2. Channel Coding and Shannon’s 2nd Theorem
   - Channel capacity
   - Codes and rates
   - Channel coding theorem
1. Noisy Channels
   - Reliable communication
   - Error correcting codes
   - Repetition codes

2. Channel Coding and Shannon’s 2nd Theorem
   - Channel capacity
   - Codes and rates
   - Channel coding theorem

3. Hamming Codes
   - Parity Check Codes
   - Hamming (7,4)
1 Noisy Channels
   - Reliable communication
   - Error correcting codes
   - Repetition codes

2 Channel Coding and Shannon’s 2nd Theorem
   - Channel capacity
   - Codes and rates
   - Channel coding theorem

3 Hamming Codes
   - Parity Check Codes
   - Hamming (7,4)
Reliable communication

In practice, most media are not perfect — noisy channels:
In practice, most media are not perfect — **noisy channels**:

- Modem line
Reliable communication

In practice, most media are not perfect — *noisy channels*:

- Modem line
- Satellite link
Reliable communication

In practice, most media are not perfect — noisy channels:

- Modem line
- Satellite link
- Hard disk
In practice, most media are not perfect — *noisy channels*:

- Modem line
- Satellite link
- Hard disk

Can we recover the original message (without errors) from a noisy code string?
We want to minimize two things:
1. Length of the code string.
2. Probability of error.
Error correcting codes

We want to minimize two things:

1. Length of the code string.
2. Probability of error.
Repetition codes

A simple idea: Just repeat the original string many times.
Repetition codes

A simple idea: Just repeat the original string many times.

Repetition codes

A simple idea: Just repeat the original string many times.

TRANSMISSION
Repetition codes

A simple idea: Just repeat the original string many times.

TRANSMISSION
TTTRRAANNNNSSSSMMMMIIIISSSSSSIIII0000NNN
Repetition codes

A simple idea: Just repeat the original string many times.

TRANSMISON
TTTRRAAANNNSSSSMMIIISSSSSIIOONNN
TTTHRAAANNNBSSSSMMIIISSSSWSPIL000ONNG
Repetition codes

A simple idea: Just repeat the original string many times.

TRANSMISSSION
TTTRRRAAAAANNNSSSMMMIIIISSSSSSIIIIOOONNN
TTTHRRRAANNBSSSSMMIIISSSSWSPILOOONNG
TRANSMISS?ION
Repetition codes

A simple idea: Just repeat the original string many times.

```
TRANSMISSIO
TTTRRAAANNNSSSSMMIIIISSSSSSIII000ONNN
TTTHRRRAAANBBSSSSMMMMIIIISSSSWSPIL000ONNG
TRANSMISSION
```

Transmission rate reduced to 1 : 3.
Repetition codes

A simple idea: Just repeat the original string many times.

TRANSMISSIO

TTTRRAAANNNSSSMMMMIIIIISSSSSSIIIOOONNN

TTTHRRRAAANNNBBSSSMMMIIIISSSSWSPILOONNG

TRANSMISSION

Transmission rate reduced to 1 : 3.

If errors independent and symmetric, probability of error reduced to $3(1 - p)p^2 + p^3 \approx 3p^2$, where $p$ is the error rate of the channel.
1. Noisy Channels
   - Reliable communication
   - Error correcting codes
   - Repetition codes

2. Channel Coding and Shannon’s 2nd Theorem
   - Channel capacity
   - Codes and rates
   - Channel coding theorem

3. Hamming Codes
   - Parity Check Codes
   - Hamming (7,4)
We are going to define the *channel capacity* $C$ purely in terms of the probabilistic properties of the channel.
Channel Capacity: basic intuition

- We are going to define the *channel capacity* $C$ purely in terms of the probabilistic properties of the channel.
- We consider encoding messages of $b$ bits into *code words* of $b/R$ bits, for some rate $0 < R < 1$. 
Channel Capacity: basic intuition

- We are going to define the *channel capacity* $C$ purely in terms of the probabilistic properties of the channel.
- We consider encoding messages of $b$ bits into *code words* of $b/R$ bits, for some *rate* $0 < R < 1$.
- *Error* is the event that the original message cannot be correctly decoded from the received code word.
Channel Capacity: basic intuition

- We are going to define the channel capacity $C$ purely in terms of the probabilistic properties of the channel.
- We consider encoding messages of $b$ bits into code words of $b/R$ bits, for some rate $0 < R < 1$.
- Error is the event that the original message cannot be correctly decoded from the received code word.
- We say a rate $R$ is achievable using a channel, if there is an encoding such that the probability of error goes to zero as $b$ increases.
Channel Capacity: basic intuition

- We are going to define the *channel capacity* $C$ purely in terms of the probabilistic properties of the channel.
- We consider encoding messages of $b$ bits into *code words* of $b/R$ bits, for some *rate* $0 < R < 1$.
- *Error* is the event that the original message cannot be correctly decoded from the received code word.
- We say a rate $R$ is *achievable* using a channel, if there is an encoding such that the probability of error goes to zero as $b$ increases.
- The *Source Coding Theorem*, or *Shannon’s Second Theorem*, says rate $R$ is achievable if $R < C$, and not achievable if $R > C$. 
Channel Capacity

- Binary symmetric channel (BSC), error rate $p$:
  \[
  \Pr[y = 1 \mid x = 0] = \Pr[y = 0 \mid x = 1] = p
  \]
  where $x$ is the transmitted and $y$ the received bit.
Binary symmetric channel (BSC), error rate $p$:

$$\Pr[y = 1 \mid x = 0] = \Pr[y = 0 \mid x = 1] = p$$

where $x$ is the transmitted and $y$ the received bit.

We define *channel capacity* as

$$C(p) = 1 - H(p) = 1 - \left[ p \log_2 \frac{1}{p} + (1 - p) \log_2 \frac{1}{1 - p} \right].$$
Channel Capacity

- Binary symmetric channel (BSC), error rate $p$:
  \[
  \Pr[y = 1 \mid x = 0] = \Pr[y = 0 \mid x = 1] = p
  \]
  where $x$ is the transmitted and $y$ the received bit

- We define channel capacity as
  \[
  C(p) = 1 - H(p) = 1 - \left[ p \log_2 \frac{1}{p} + (1 - p) \log_2 \frac{1}{1 - p} \right].
  \]

- For instance, $C(0.1) \approx 0.53$. Ratio about 1 : 2.
Channel Capacity

For channels other than BSC, the channel capacity is more generally defined as

\[ C = \max_{p_X} I(X, Y) = \max_{p_X} (H(Y) - H(Y | X)) \]

- \( X \) is the transmitted and \( Y \) the received symbol
- \( I \) is calculated with respect to \( p_{X,Y}(x, y) = p_X(x)p_{Y|X}(y | x) \)
- \( p_{Y|X} \) is defined by the channel characteristics.
For channels other than BSC, the channel capacity is more generally defined as

\[ C = \max_{p_X} I(X, Y) = \max_{p_X} (H(Y) - H(Y | X)) \]

- \( X \) is the transmitted and \( Y \) the received symbol
- \( I \) is calculated with respect to \( p_{X,Y}(x,y) = p_X(x)p_{Y|X}(y | x) \)
- \( p_{Y|X} \) is defined by the channel characteristics.

Intuition:

- \( Y \) should carry a lot of information
- Knowing \( X \) should remove most of the uncertainty about \( Y \)
- We can get a favorable \( p_X \) by choosing a suitable coding.
Channel Capacity

For channels other than BSC, the channel capacity is more generally defined as

$$C = \max_{p_X} I(X, Y) = \max_{p_X} (H(Y) - H(Y \mid X))$$

- $X$ is the transmitted and $Y$ the received symbol
- $I$ is calculated with respect to $p_{X,Y}(x, y) = p_X(x)p_{Y\mid X}(y \mid x)$
- $p_{Y\mid X}$ is defined by the channel characteristics.

Intuition:
- for a large capacity, we want $Y$ to carry a lot of information
Channel Capacity

For channels other than BSC, the channel capacity is more generally defined as

$$C = \max_{p_X} I(X, Y) = \max_{p_X} (H(Y) - H(Y | X))$$

- $X$ is the transmitted and $Y$ the received symbol
- $I$ is calculated with respect to $p_{X,Y}(x, y) = p_X(x)p_{Y|X}(y | x)$
- $p_{Y|X}$ is defined by the channel characteristics.

Intuition:
- for a large capacity, we want $Y$ to carry a lot of information
- however, knowing $X$ should remove most of the uncertainty about $Y$
Channel Capacity

For channels other than BSC, the channel capacity is more generally defined as

\[ C = \max_{p_X} I(X, Y) = \max_{p_X} (H(Y) - H(Y \mid X)) \]

- \( X \) is the transmitted and \( Y \) the received symbol
- \( I \) is calculated with respect to \( p_{X,Y}(x,y) = p_X(x)p_{Y \mid X}(y \mid x) \)
- \( p_{Y \mid X} \) is defined by the channel characteristics.

Intuition:
- for a large capacity, we want \( Y \) to carry a lot of information
- however, knowing \( X \) should remove most of the uncertainty about \( Y \)
- we can get a favourable \( p_X \) by choosing a suitable coding.
Example 1: BSC
Choosing uniform $p_X$ gives the maximum $I(X; Y) = 1 - H(p)$
Example 1: BSC
Choosing uniform $p_X$ gives the maximum $I(X; Y) = 1 - H(p)$
(exercise)
Channel Capacity

Example 1: BSC
Choosing uniform $p_X$ gives the maximum $I(X; Y) = 1 - H(p)$
(exercise)

Example 2: Noisy typewriter
Channel Capacity

Example 1: BSC
Choosing uniform $p_X$ gives the maximum $I(X; Y) = 1 - H(p)$
(exercise)

Example 2: Noisy typewriter
- The maximum is obtained for uniform $p_X$ (symmetricity)
Channel Capacity

**Example 1: BSC**
Choosing uniform $p_X$ gives the maximum $I(X; Y) = 1 - H(p)$ (exercise)

**Example 2: Noisy typewriter**
- The maximum is obtained for uniform $p_X$ (symmetricity)
- with uniform $p_X$, also $p_Y$ is uniform over 26 symbols
  \[ H(Y) = \log_2 26 \]
Channel Capacity

Example 1: BSC
Choosing uniform $p_X$ gives the maximum $I(X; Y) = 1 - H(p)$ (exercise)

Example 2: Noisy typewriter
- The maximum is obtained for uniform $p_X$ (symmetricity)
- with uniform $p_X$, also $p_Y$ is uniform over 26 symbols
  $\Rightarrow H(Y) = \log_2 26$
- if $X$ is known, there are two equally probable values $Y$
  $\Rightarrow H(Y | X) = \log_2 2 = 1$
Channel Capacity

Example 1: BSC
Choosing uniform $p_X$ gives the maximum $I(X; Y) = 1 - H(p)$ (exercise)

Example 2: Noisy typewriter
- The maximum is obtained for uniform $p_X$ (symmetricity)
- with uniform $p_X$, also $p_Y$ is uniform over 26 symbols
  $\Rightarrow H(Y) = \log_2 26$
- if $X$ is known, there are two equally probable values $Y$
  $\Rightarrow H(Y | X) = \log_2 2 = 1$
- so $I(X; Y) = \log_2 26 - 1 = \log_2 13$ (capacity 13 bits per transmission)
For simplicity, we consider BSC unless we say otherwise.

- Messages we want to send are blocks of $b$ bits.
Codes and rates

For simplicity, we consider BSC unless we say otherwise.

- *Messages* we want to send are blocks of $b$ bits.
  Thus, there are $M = 2^b$ possible messages.
Codes and rates

For simplicity, we consider BSC unless we say otherwise.

- *Messages* we want to send are blocks of \( b \) bits. Thus, there are \( M = 2^b \) possible messages.
- We encode a message into *code words* of \( n \) bits.
Codes and rates

For simplicity, we consider BSC unless we say otherwise.

- *Messages* we want to send are blocks of $b$ bits. Thus, there are $M = 2^b$ possible messages.
- We encode a message into *code words* of $n$ bits. So generally we need $n \geq \log_2 M = b$. 
Codes and rates

For simplicity, we consider BSC unless we say otherwise.

- **Messages** we want to send are blocks of $b$ bits. Thus, there are $M = 2^b$ possible messages.
- We encode a message into *code words* of $n$ bits. So generally we need $n \geq \log_2 M = b$.
- Notation:

\[
W \in \{1, \ldots, M\} : \text{(index of) a message}
\]
\[
X_n = f(W) \in \{0, 1\}^n : \text{code word for message } W
\]
\[
Y_n \in \{0, 1\}^n : \text{received code word (noisy version of } X_n)\]
\[
\hat{W} = g(Y_n) \in \{1, \ldots, M\} : \text{our guess about what the correct message was.}
\]
Codes and rates

For simplicity, we consider BSC unless we say otherwise.

- **Messages** we want to send are blocks of \( b \) bits. Thus, there are \( M = 2^b \) possible messages.
- We encode a message into code words of \( n \) bits. So generally we need \( n \geq \log_2 M = b \).

**Notation:**
- \( W \in \{1, \ldots, M\} \): (index of) a message
Codes and rates

For simplicity, we consider BSC unless we say otherwise.

- *Messages* we want to send are blocks of $b$ bits. Thus, there are $M = 2^b$ possible messages.

- We encode a message into *code words* of $n$ bits. So generally we need $n \geq \log_2 M = b$.

- **Notation:**
  - $W \in \{1, \ldots, M\}$: (index of) a message
  - $X^n = f(W) \in \{0,1\}^n$: code word for message $W$
For simplicity, we consider BSC unless we say otherwise.

- **Messages** we want to send are blocks of \( b \) bits.
  Thus, there are \( M = 2^b \) possible messages.

- We encode a message into **code words** of \( n \) bits.
  So generally we need \( n \geq \log_2 M = b \).

- **Notation:**
  - \( W \in \{1, \ldots, M\} \): (index of) a message
  - \( X^n = f(W) \in \{0,1\}^n \): code word for message \( W \)
  - \( Y^n \in \{0,1\}^n \): received code word (noisy version of \( X^n(W) \)
Codes and rates

For simplicity, we consider BSC unless we say otherwise.

- **Messages** we want to send are blocks of $b$ bits. Thus, there are $M = 2^b$ possible messages.

- We encode a message into *code words* of $n$ bits. So generally we need $n \geq \log_2 M = b$.

- **Notation:**
  - $W \in \{1, \ldots, M\}$: (index of) a message
  - $X^n = f(W) \in \{0, 1\}^n$: code word for message $W$
  - $Y^n \in \{0, 1\}^n$: received code word (noisy version of $X^n(W)$)
  - $\hat{W} = g(Y^n) \in \{1, \ldots, M\}$: our guess about what the correct message was.
For simplicity, we consider BSC unless we say otherwise.

- **Messages** we want to send are blocks of $b$ bits. Thus, there are $M = 2^b$ possible messages.

- We encode a message into *code words* of $n$ bits. So generally we need $n \geq \log_2 M = b$.

- **Notation:**
  - $W \in \{1, \ldots, M\}$: (index of) a message
  - $X^n = f(W) \in \{0, 1\}^n$: code word for message $W$
  - $Y^n \in \{0, 1\}^n$: received code word (noisy version of $X^n(W)$)
  - $\hat{W} = g(Y^n) \in \{1, \ldots, M\}$: our guess about what the correct message was.

- The *rate* of the code is $R = (\log_2 M)/n$. 
Codes and rates

Let $\lambda_w$, for $w \in \{1, \ldots, M\}$, denote the probability that message $w$ was sent but not correctly received.
Codes and rates

Let $\lambda_w$, for $w \in \{1, \ldots, M\}$, denote the probability that message $w$ was sent but not correctly received.

We can write this as

$$\lambda_w = \sum_{y \notin g^{-1}(w)} p(y \mid X = f(w))$$

where $g^{-1}(w) = \{y \mid g(y) = w\}$.
Codes and rates

Let $\lambda_w$, for $w \in \{1, \ldots, M\}$, denote the probability that message $w$ was sent but not correctly received.

We can write this as

$$
\lambda_w = \sum_{y \notin g^{-1}(w)} p(y \mid X = f(w))
$$

where $g^{-1}(w) = \{ y \mid g(y) = w \}$.

Average error: $\bar{\lambda} = \frac{1}{M} \sum_w \lambda_w$
Codes and rates

Let $\lambda_w$, for $w \in \{1, \ldots, M\}$, denote the probability that message $w$ was sent but not correctly received.

We can write this as

$$\lambda_w = \sum_{y \notin g^{-1}(w)} p(y \mid X = f(w))$$

where $g^{-1}(w) = \{ y \mid g(y) = w \}$.

Average error: $\bar{\lambda} = \frac{1}{M} \sum_w \lambda_w$

Maximum error: $\lambda_{\text{max}} = \max_w \lambda_w$
A rate $R$ is *achievable* if there is a sequence of codes, for increasingly large code word lengths $n$, such that as $n$ goes to infinity, the maximum error $\lambda_{\text{max}}$ goes to zero.
A rate $R$ is *achievable* if there is a sequence of codes, for increasingly large code word lengths $n$, such that as $n$ goes to infinity, the maximum error $\lambda_{\text{max}}$ goes to zero.

**Channel Coding Theorem**

If $R < C$, where $C$ is the channel capacity, then rate $R$ is achievable.

If $R > C$, then rate $R$ is not achievable.
Channel coding theorem

A rate $R$ is *achievable* if there is a sequence of codes, for increasingly large code word lengths $n$, such that as $n$ goes to infinity, the maximum error $\lambda_{\text{max}}$ goes to zero.

**Channel Coding Theorem**

If $R < C$, where $C$ is the channel capacity, then rate $R$ is achievable.

If $R > C$, then rate $R$ is not achievable.

In other words, for any given $\epsilon > 0$ and $R < C$, for large enough $b$ we can encode messages of $b$ bits into code words of $n = b/R$ bits so that the probability of error is at most $\epsilon$. 
Channel coding theorem

A rate $R$ is achievable if there is a sequence of codes, for increasingly large code word lengths $n$, such that as $n$ goes to infinity, the maximum error $\lambda_{\text{max}}$ goes to zero.

**Channel Coding Theorem**

If $R < C$, where $C$ is the channel capacity, then rate $R$ is achievable.

If $R > C$, then rate $R$ is not achievable.

In other words, for any given $\epsilon > 0$ and $R < C$, for large enough $b$ we can encode messages of $b$ bits into code words of $n = \frac{b}{R}$ bits so that the probability of error is at most $\epsilon$.

This is also known as Shannon’s Second Theorem (the first one being the Source Coding Theorem).
Channel Coding Theorem—So what?

Assume you want to transmit data with probability of error $10^{-15}$ over a BSC, $p = 0.1$. Using a repetition code, we need to make the message 63 times as long as the source string. Shannon's result says twice as long is enough. If you want probability of error $10^{-100}$, Shannon's result still says that twice is enough! However the messages you encode need to be sufficiently long!
Channel coding theorem

Channel Coding Theorem—So what?

Assume you want to transmit data with probability of error $10^{-15}$ over a BSC, $p = 0.1$. Using a repetition code, we need to make the message 63 times as long as the source string.
Channel Coding Theorem—So what?

Assume you want to transmit data with probability of error $10^{-15}$ over a BSC, $p = 0.1$. Using a repetition code, we need to make the message 63 times as long as the source string. (Exercise: Check the math. Hint: binomial distribution.)
Channel Coding Theorem—So what?

Assume you want to transmit data with probability of error $10^{-15}$ over a BSC, $p = 0.1$. Using a repetition code, we need to make the message 63 times as long as the source string.

Shannon’s result says twice as long is enough.
Channel Coding Theorem—So what?

Assume you want to transmit data with probability of error $10^{-15}$ over a BSC, $p = 0.1$. Using a repetition code, we need to make the message 63 times as long as the source string.

Shannon’s result says twice as long is enough.

If you want probability of error $10^{-100}$, Shannon’s result still says that twice is enough!
Channel coding theorem

Channel Coding Theorem—So what?

Assume you want to transmit data with probability of error $10^{-15}$ over a BSC, $p = 0.1$. Using a repetition code, we need to make the message 63 times as long as the source string.

Shannon’s result says twice as long is enough.

If you want probability of error $10^{-100}$, Shannon’s result still says that twice is enough!

However the messages you encode need to be sufficiently long!
The proof of Channel Coding Theorem (which we won’t cover) is based on choosing $M$ code words, each $n$ bits long, completely at random.
Channel coding theorem

- The proof of Channel Coding Theorem (which we won’t cover) is based on choosing $M$ code words, each $n$ bits long, completely at random.
- To decode $y$, just pick $w$ for which $f(w)$ is closest to $y$. 

If $\log_2 M < nR$, then the expected error rate, over random choice of code books, is very small. If random code books are good on average, then surely the best single code book is at least as good. However, in practice we need specific codes that have high rates and are easy to compute. Finding such is difficult and out of scope for this course. We will next give a simple example to illustrate the basic idea.
The proof of Channel Coding Theorem (which we won’t cover) is based on choosing $M$ code words, each $n$ bits long, completely at random.

To decode $y$, just pick $w$ for which $f(w)$ is closest to $y$.

If $\log_2 M < nR$, then the expected error rate, over random choice of code books, is very small.
The proof of Channel Coding Theorem (which we won’t cover) is based on choosing $M$ code words, each $n$ bits long, completely at random.

To decode $y$, just pick $w$ for which $f(w)$ is closest to $y$.

If $\log_2 M < nR$, then the expected error rate, over random choice of code books, is very small. This is the tricky part.
The proof of Channel Coding Theorem (which we won’t cover) is based on choosing $M$ code words, each $n$ bits long, completely at random.

To decode $y$, just pick $w$ for which $f(w)$ is closest to $y$.

If $\log_2 M < nR$, then the expected error rate, over random choice of code books, is very small.

If random code books are good on average, then surely the best single code book is at least as good.
The proof of Channel Coding Theorem (which we won’t cover) is based on choosing $M$ code words, each $n$ bits long, completely at random.

To decode $y$, just pick $w$ for which $f(w)$ is closest to $y$.

If $\log_2 M < nR$, then the expected error rate, over random choice of code books, is very small.

If random code books are good on average, then surely the best single code book is at least as good.

However, in practice we need specific codes that have high rates and are easy to compute. Finding such is difficult and out of scope for this course.
The proof of Channel Coding Theorem (which we won’t cover) is based on choosing $M$ code words, each $n$ bits long, completely at random.

To decode $y$, just pick $w$ for which $f(w)$ is closest to $y$.

If $\log_2 M < nR$, then the expected error rate, over random choice of code books, is very small.

If random code books are good on average, then surely the best single code book is at least as good.

However, in practice we need specific codes that have high rates and are easy to compute. Finding such is difficult and out of scope for this course. We will next give a simple example to illustrate the basic idea.
1. Noisy Channels
   - Reliable communication
   - Error correcting codes
   - Repetition codes

2. Channel Coding and Shannon’s 2nd Theorem
   - Channel capacity
   - Codes and rates
   - Channel coding theorem

3. Hamming Codes
   - Parity Check Codes
   - Hamming (7,4)
Hamming Codes

Parity Check Codes

One way to detect and correct errors is to add *parity checks* to the codewords:
One way to detect and correct errors is to add *parity checks* to the codewords:

- If we add a parity check bit at the end of each codeword we can detect one (but not more) error per codeword.
Parity Check Codes

One way to detect and correct errors is to add *parity checks* to the codewords:

- If we add a parity check bit at the end of each codeword we can detect one (but not more) error per codeword.
- By clever use of more than one parity bits, we can actually identify where the error occurred and thus also *correct errors*. 
Parity Check Codes

One way to detect and correct errors is to add \textit{parity checks} to the codewords:

- If we add a parity check bit at the end of each codeword we can detect one (but not more) error per codeword.
- By clever use of more than one parity bits, we can actually identify where the error occurred and thus also \textit{correct errors}.
- Designing ways to add as few parity bits as possible to correct and detect errors is a \textit{really} hard problem.
Hamming (7,4)

4 data bits ($d_1, d_2, d_3, d_4$), 3 parity bits ($p_1, p_2, p_3$)
Hamming (7,4)

source string 1011, parity bits 010
Hamming (7,4)

error in data bit $d_2$ (0 $\rightarrow$ 1) is identified and corrected
Hamming (7,4)

two errors can be detected but not corrected
Advanced Error Correcting Codes

The Hamming (7,4) code is an example of a code that can detect and correct errors at rate greater than $1:2$. More complex Hamming codes, like Hamming (8,4), Hamming (11,7), etc. can correct and/or detect more errors. The present state-of-the-art is based on so-called low-density parity-check (LDPC) codes, which likewise include a number of parity check bits. Massive research effort: At ISIT-09 conference, 12 sessions (4 talks in each) about LDPC codes.
Advanced Error Correcting Codes

The Hamming (7,4) code is an example of a code that can detect and correct errors at rate greater than 1 : 2.

More complex Hamming codes, like Hamming (8,4), Hamming (11,7), etc. can correct and/or detect more errors.
The Hamming (7,4) code is an example of a code that can detect and correct errors at rate greater than 1 : 2.

More complex Hamming codes, like Hamming (8,4), Hamming (11,7), etc. can correct and/or detect more errors.

The present state-of-the-art is based on so called low-density parity-check (LDPC) codes, which likewise include a number of parity check bits.
Advanced Error Correcting Codes

The Hamming (7,4) code is an example of a code that can detect and correct errors at rate greater than $1:2$.

More complex Hamming codes, like Hamming (8,4), Hamming (11,7), etc. can correct and/or detect more errors.

The present state-of-the-art is based on so called *low-density parity-check* (LDPC) codes, which likewise include a number of parity check bits.

Massive research effort: At ISIT-09 conference, 12 sessions (4 talks in each) about LDPC codes.
Next topics

Back to noiseless source coding
Next topics

Back to noiseless source coding

- prefix codes and Kraft Inequality
Next topics

Back to noiseless source coding

- prefix codes and Kraft Inequality
- coding algorithms: Shannon coding, Huffman coding, arithmetic coding