Institute of Operating Systems
and Computer Networks

Technische
Universität
Braunschweig

Secure communication based on noisy input data

Error correcting codes

Stephan Sigg

June 21, 2011

Overview and Structure

05.04.2011 Organisational
15.04.2011 Introduction
19.04.2011 Classification methods (Basic recognition, Bayesian, Non-parametric)
26.04.2011 Classification methods (Linear discriminant, Neural networks)
03.05.2011 Classification methods (Sequential, Stochastic)
10.05.2011 Feature extraction from audio data
17.05.2011 Feature extraction from the RF channel
24.05.2011 Fuzzy Commitment
31.05.2011 Fuzzy Extractors
07.06.2011 Error correcting codes
21.06.2011 Entropy
28.06.2011 Physically unclonable functions

Outline

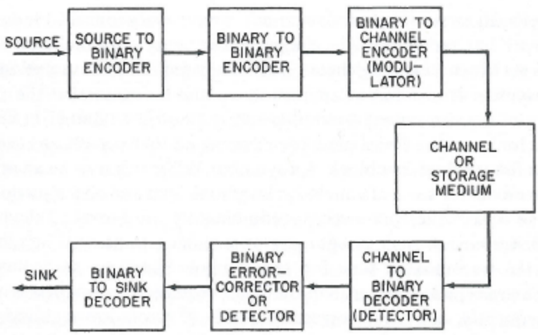
Introduction

We consider a communication system in which the channel between the encoder and the decoder might be impaired by noise

Introduction

By employing error correcting codes, we try to account for possible errors during transmission over the channel

Naturally, error correcting codes can not correct all errors but must be designed to correct the most likely patterns



Introduction

Frequently, the assumption has been taken that each symbol is affected independently by noise

In this case the probability of a given error pattern depends only on the number of errors

In several fields, for instance for communication technology, errors are more likely to occur in blocks of symbols (bursts)

Introduction

Types of Codes

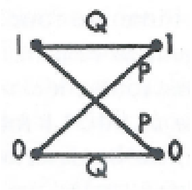
Block codes Breaks the continuous sequence of information digits into k-symbol sections or blocks. It then operates on these blocks independently

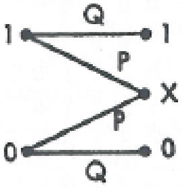
Tree codes operate on the information sequence without breaking it up into independent blocks. An important subclass are convolutional codes since they are simple to implement

Introduction

In order to predict the performance of a code, it is beneficial to have an accurate approximation on the channel behaviour

Most extensively studied channels are the Binary Symmetric Channel and the Binary erasure channel





Outline

Block codes

Let q denote the number of distinct symbols employed on the channel

A block code is a set of M sequences of channel symbols of length n

Decision to which code word a received word belongs may be based on a decoding table

Code Words	1 1 0 0 0	0 0 1 1 0	1 0 0 1 1	0 1 1 0 1
Other Received Words	1 1 0 0 1	0 0 1 1 1	1 0 0 1 0	0 1 1 0 0
	1 1 0 1 0	0 0 1 0 0	1 0 0 0 1	0 1 1 1 1
	1 1 1 0 0	0 0 0 1 0	1 0 1 1 1	0 1 0 0 1
	1 0 0 0 0	0 1 1 1 0	1 1 0 1 1	0 0 1 0 1
	0 1 0 0 0	1 0 1 1 0	0 0 0 1 1	1 1 1 0 1
	1 1 1 1 0	0 0 0 0 0	0 1 0 1 1	1 0 1 0 1
	0 1 0 1 0	1 0 1 0 0	1 1 1 1 1	0 0 0 0 1

Block codes

The probability of correct decoding can be calculated with the help of an assumption on the channel characteristics.

In a Binary Symmetric channel, the probability of correctly decoding the word 11000 is calculated as

$$1P^0Q^5 + 5P^1Q^4 + 2P^2Q^3$$

Code Words	1 1 0 0 0	0 0 1 1 0	1 0 0 1 1	0 1 1 0 1
Other Received Words	1 1 0 0 1	0 0 1 1 1	1 0 0 1 0	0 1 1 0 0
	1 1 0 1 0	0 0 1 0 0	1 0 0 0 1	0 1 1 1 1
	1 1 1 0 0	0 0 0 1 0	1 0 1 1 1	0 1 0 0 1
	1 0 0 0 0	0 1 1 1 0	1 1 0 1 1	0 0 1 0 1
	0 1 0 0 0	1 0 1 1 0	0 0 0 1 1	1 1 1 0 1
	1 1 1 1 0	0 0 0 0 0	0 1 0 1 1	1 0 1 0 1
	0 1 0 1 0	1 0 1 0 0	1 1 1 1 1	0 0 0 0 1

Block codes

Linear block codes

For linear block codes we require a set of k basis vectors $\vec{g}$ (generator vectors) of length n

Basis vectors are linear independent vectors that span the basis of a vector space

These vectors are considered as rows of a matrix G

The row-space of G defines the linear code V and code vectors $\vec{v}$ are linear combinations of rows in G

It is important that the vectors g are linear independent since otherwise different linear combinations of vectors would lead to identical code vectors

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

Block codes

Linear block codes

Data vectors $\overrightarrow{d}$ define which generator vectors g are combined to a code vector $\overrightarrow{v}$

We define a matrix H of rank $n-k$ whose row space is a basis of vectors orthogonal to each vector in G (null space)

Since each code vector $\overrightarrow{v}$ is the result of a linear combination of generator vectors $\overrightarrow{g}$, we have

$$\overrightarrow{v} H^T = \overrightarrow{0}$$

In the case of errors in the code vector, the result is hence

$$(\overrightarrow{v} + \overrightarrow{e}) H^T \neq \overrightarrow{0}$$

iff $(\overrightarrow{v} + \overrightarrow{e}) \notin \overrightarrow{\alpha} G$

Block codes

Linear block codes

In the case of errors in the code vector, the result is hence

$$(\overrightarrow{v} + \overrightarrow{e})H^T \neq \overrightarrow{0}$$

The error vector $\overrightarrow{e}$ then defines the linear combination of rows of H^T that lead to the syndrome:

$$\begin{aligned} & (\overrightarrow{v} + \overrightarrow{e})H^T \\ = & \overrightarrow{v}H^T + \overrightarrow{e}H^T \\ = & \overrightarrow{0} + \overrightarrow{e}H^T \\ = & \overrightarrow{s} \end{aligned}$$

H is spanned by basis vectors$\rightarrow \overrightarrow{s}$ defines uniquely the error vectors that occurred.

Block codes

Linear block codes

Example

Let

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

H states that the sum of the first two digits and the sum of digits one, three and four of every code word must be zero.

Block codes

Linear block codes

The minimum distance for a block code equals the minimum weight of its nonzero vectors. For a block code with $q = 2$ and $n = 5$, the set of vectors

$$
\begin{array}{c}
(\; 0 \; 0 \; 0 \; 0 \; 0 \;) \\
(\; 1 \; 0 \; 0 \; 1 \; 1 \;) \\
(\; 0 \; 1 \; 0 \; 1 \; 0 \;) \\
(\; 1 \; 1 \; 0 \; 0 \; 1 \;) \\
(\; 0 \; 0 \; 1 \; 0 \; 1 \;) \\
(\; 1 \; 0 \; 1 \; 1 \; 0 \;) \\
(\; 0 \; 1 \; 1 \; 1 \; 1 \;) \\
(\; 1 \; 1 \; 1 \; 0 \; 0 \;)
\end{array}
$$

have a minimum weight and therefore a minimum distance of 2

Outline

Convolutional codes

Tree codes do not break the information sequence into blocks and handle them independently.

A tree code associates a code sequence with an information sequence

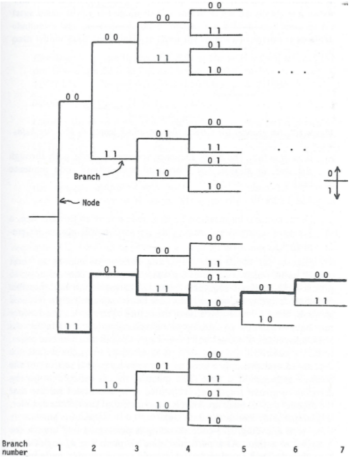
The code sequence is defined by a tree labelled with binary sub-sequences at its edges

A 0-information bit translates to an upward step in the tree, a 1-information bit to a downward step.

Convolutional codes

Example: Encoding of 101100

A well-designed decoder chooses the branch that leads to the path with smaller hamming distance of a sub-sequence



Convolutional codes

Due to the structure of tree-codes, the error probability is not as easy to calculate as for block codes.

Errors in previous code words might impact code words transmitted later.

Convolutional codes

Let F_i denote a matrix whose k_0 rows are linearly independent vectors over $GF(q)$

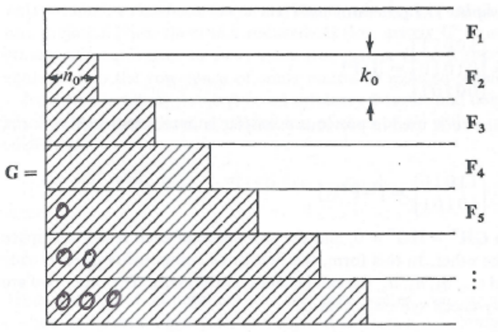
Further assume that the first $(i-1)n_0$ columns of F_i are zero while some of the $(i-1)n_0+1$ through in_0 columns are nonzero

A linear tree code is defined as

$$G = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ \vdots \end{bmatrix}$$

The minimum weight of a code word whose first n_0 digits are not all zero equals the minimum distance d of the code

Convolutional codes



Convolutional codes

A code sequence is obtained as

$$c = iG$$

A class of linear tree codes, called convolutional codes is achieved by defining the matrices F_i to be shifted versions of F_1:

$$G = \begin{bmatrix} G_0 & G_1 & G_2 & \dots & G_{m-2} & G_{m-1} \\ & G_0 & G_1 & \dots & G_{m-3} & G_{m-2} \\ & & G_0 & \dots & G_{m-4} & G_{m-3} \\ & & & \ddots & & \\ & & & & G_0 & G_1 \\ & & & & & G_0 \end{bmatrix}$$

The matrices G_i have k_0 rows and n_0 columns.

Convolutional codes

For arbitrary $k_0 \times (n_0 - k_0)$ matrices P_i and identity matrices I this code generator matrix is combinatorially equivalent to one in echelon canonical form:

$$G = \begin{bmatrix} IP_0 & \mathbf{0}P_1 & \mathbf{0}P_2 & \dots & \mathbf{0}P_{m-2} & \mathbf{0}P_{m-1} \\ & IP_0 & \mathbf{0}P_1 & \dots & \mathbf{0}P_{m-3} & \mathbf{0}P_{m-2} \\ & & IP_0 & \dots & \mathbf{0}P_{m-4} & \mathbf{0}P_{m-3} \\ & & & \ddots & & \\ & & & & IP_0 & \mathbf{0}P_1 \\ & & & & & IP_0 \end{bmatrix}$$

Convolutional codes

The corresponding nullspace is spanned by

$$H = \begin{bmatrix} P_0^T I & & & \\ P_1^T \mathbf{0} & P_0^T I & & \\ \vdots & \vdots & \ddots & \\ P_{m-1}^T \mathbf{0} & P_{m-2}^T \mathbf{0} & \dots & P_0^T I \end{bmatrix}$$

Outline

Burst-Correcting Convolutional codes

The basic idea behind all burst-correcting convolutional codes is that the digits involved in the decoding of a particular digit are spread in time so that only one, or at most a few can be affected by a single burst of errors.

The simplest way to achieve this is spreading by interleaving

The data stream is then broken into i independent streams

Either symbols or short blocks of symbols are interleaved by i other symbol or block streams

The parameter i is called the interleaving degree

Burst-Correcting Convolutional codes

The parity check matrix of an interleaved code is derived from the parity check matrix of the non-interleaved code:

$$H_1 = \begin{bmatrix} 01 & & & \\ 10 & 01 & & \\ 10 & 10 & 01 & \\ 00 & 10 & 10 & 01 \end{bmatrix} \quad H_2 = \begin{bmatrix} 01 & & & & & & \\ 00 & 01 & & & & & \\ 10 & 00 & 01 & & & & \\ 00 & 10 & 00 & 01 & & & \\ 10 & 00 & 10 & 00 & 01 & & \\ 00 & 10 & 00 & 10 & 00 & 01 & \\ 00 & 00 & 10 & 00 & 10 & 00 & 01 \end{bmatrix}$$

Interleaving degree 2

Burst-Correcting Convolutional codes

The coding rate is unaffected by the interleaving

Therefore, arbitrary long, nearly optimal burst-correcting convolutional codes can be formed by interleaving convolutional codes.

Interleaving an (n, k) block code that corrects bursts of length b to a degree i produces an (ni, ki) block code with burst-correcting ability bi

Interleaving an (mn_0, mk_0) convolutional code that corrects bursts of length b to a degree i produces an $(mn_0(i-1)+n_0, mk_0(i-1)+k_0)$ convolutional code with burst-correcting ability bi

Burst-Correcting Convolutional codes

Berlekamp-Preparata-Massey Codes

Classical $(2n_0^2, 2n_0^2 - 2n_0)$ BPM codes have a parity-check matrix of the form

$$H = [B_0 B_1 B_2 \dots B_{2n_0-1}]$$

With B_i down-shifted from B_{i-1} as

$$B_i = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix} B_{i-1}$$

Burst-Correcting Convolutional codes

Berlekamp-Preparata-Massey Codes

An n-tuple that has all its 1's in the 0-th and the i-th block can be represented as

$$E = E_0\mathbf{000}\dots E_i\mathbf{00}\dots\mathbf{0}$$

If B_0 is chosen so that $EH^T \neq \mathbf{0}$ for all choices of E_0, E_i and i the code can correct all length n_0 bursts of errors

In order for this to occur, it must be that
$E_0E_i[B_0B_i]^T \neq \mathbf{0}; i \in [1, 2n_0 - 1]$

(It must not be possible that n_0 errors can create an all-0 code block which could result in $E_0E_i[B_0B_i]^T = \mathbf{0}$

Burst-Correcting Convolutional codes

Berlekamp-Preparata-Massey Codes

The challenging part of defining BPM-codes is to find a matrix B_0 that confines the assumption given above.

Outline

Reed-Muller codes

The Reed-Muller codes are a class of binary group codes covering a wide range of rate and minimum distance

for any m and $r < m$ there is a reed-Muller code for which

$$\begin{aligned}
n &= 2^m \\
k &= 1 + \binom{m}{1} + \cdots + \binom{m}{r} \\
n - k &= 1 + \binom{m}{1} + \cdots + \binom{m}{m-r-1} \\
d &= 2^{m-r} \text{(minimum weight)}
\end{aligned}$$

Reed-Muller codes

Let $\overrightarrow{v_0}$ be a vector whose 2^m components are all 1-s and let $\overrightarrow{v_1}, \overrightarrow{v_2}, \ldots, \overrightarrow{v_m}$ be the rows of a matrix that has all possible m-tuples as columns.

Example for $m = 4$:

$$
\begin{aligned}
\overrightarrow{v_0} &= 1111111111111111 \\
\overrightarrow{v_4} &= 0000000011111111 \\
\overrightarrow{v_3} &= 0000111100001111 \\
\overrightarrow{v_2} &= 0011001100110011 \\
\overrightarrow{v_1} &= 0101010101010101 \\
\overrightarrow{v_4}\,\overrightarrow{v_3} &= 0000000000001111 \\
\overrightarrow{v_4}\,\overrightarrow{v_2} &= 0000000000110011 \\
\overrightarrow{v_4}\,\overrightarrow{v_1} &= 0000000001010101 \\
\overrightarrow{v_3}\,\overrightarrow{v_2} &= 0000001100000011 \\
\overrightarrow{v_3}\,\overrightarrow{v_1} &= 0000010100000101 \\
\overrightarrow{v_2}\,\overrightarrow{v_1} &= 0001000100010001 \\
\overrightarrow{v_4}\,\overrightarrow{v_3}\,\overrightarrow{v_2} &= 0000000000000011 \\
\overrightarrow{v_4}\,\overrightarrow{v_3}\,\overrightarrow{v_1} &= 0000000000000101 \\
\overrightarrow{v_4}\,\overrightarrow{v_2}\,\overrightarrow{v_1} &= 0000000000010001 \\
\overrightarrow{v_3}\,\overrightarrow{v_3}\,\overrightarrow{v_1} &= 0000000100000001 \\
\overrightarrow{v_4}\,\overrightarrow{v_3}\,\overrightarrow{v_2}\,\overrightarrow{v_1} &= 0000000000000001
\end{aligned}
$$

Reed-Muller codes

Example for $m = 4, r = 2$:

$$
\begin{aligned}
\vec{v_0} &= 1111111111111111\\
\vec{v_4} &= 0000000011111111\\
\vec{v_3} &= 0000111100001111\\
\vec{v_2} &= 0011001100110011\\
\vec{v_1} &= 0101010101010101\\
\vec{v_4}\vec{v_3} &= 0000000000001111\\
\vec{v_4}\vec{v_2} &= 0000000000110011\\
\vec{v_4}\vec{v_1} &= 0000000001010101\\
\vec{v_3}\vec{v_2} &= 0000001100000011\\
\vec{v_3}\vec{v_1} &= 0000010100000101\\
\vec{v_2}\vec{v_1} &= 0001000100010001\\
\vec{v_4}\vec{v_3}\vec{v_2} &= 0000000000000011\\
\vec{v_4}\vec{v_3}\vec{v_1} &= 0000000000000101\\
\vec{v_4}\vec{v_2}\vec{v_1} &= 0000000000010001\\
\vec{v_3}\vec{v_3}\vec{v_1} &= 0000000100000001\\
\vec{v_4}\vec{v_3}\vec{v_2}\vec{v_1} &= 0000000000000001
\end{aligned}
$$

The r-th order Reed-Muller code is formed by using as a basis the vectors $\vec{v_0}, \vec{v_1}, \ldots, \vec{v_m}$ and all vector products of r or fewer of these vectors

Reed-Muller codes

Decoding of Reed-Muller codes

Assume a second order $(16, 11)$ code for $m = 4$.

The r-th order Reed-Muller code is formed by using as a basis the vectors $\vec{v_0}, \vec{v_1}, \ldots, \vec{v_m}$ and all vector products of r or fewer of these vectors

The 11 information symbols are denoted by

$$a_0, a_4, a_3, a_2, a_1, a_{43}, a_{42}, a_{41}, a_{32}, a_{31}, a_{21}$$

The codevector is then

$$\begin{aligned} a_0\vec{v_0} &+ a_4\vec{v_4} + a_3\vec{v_3} + a_2\vec{v_2} + a_1\vec{v_1} + a_{43}\vec{v_4}\vec{v_3} \\ &+ a_{42}\vec{v_4}\vec{v_2} + a_{41}\vec{v_4}\vec{v_1} + a_{32}\vec{v_3}\vec{v_2} + a_{31}\vec{v_3}\vec{v_1} + a_{21}\vec{v_2}\vec{v_1} \\ &= (b_1, b_2, \ldots, b_n) \end{aligned}$$

Example for $m = 4, r = 2$:

$$\begin{aligned} \vec{v_0} &= 1111111111111111 \\ \vec{v_4} &= 0000000011111111 \\ \vec{v_3} &= 0000111100001111 \\ \vec{v_2} &= 0011001100110011 \\ \vec{v_1} &= 0101010101010101 \\ \vec{v_4}\vec{v_3} &= 0000000000001111 \\ \vec{v_4}\vec{v_2} &= 0000000000110011 \\ \vec{v_4}\vec{v_1} &= 0000000001010101 \\ \vec{v_3}\vec{v_2} &= 0000001100000011 \\ \vec{v_3}\vec{v_1} &= 0000010100000101 \\ \vec{v_2}\vec{v_1} &= 0001000100010001 \end{aligned}$$

Reed-Muller codes

Decoding of Reed-Muller codes

Determine the a's from a noisy vector.

Note that, in the absence of errors:

$$b_1 + b_2 + b_3 + b_4 = a_{21}$$
$$b_5 + b_6 + b_7 + b_8 = a_{21}$$
$$b_9 + b_{10} + b_{11} + b_{12} = a_{21}$$
$$b_{13} + b_{14} + b_{15} + b_{16} = a_{21}$$

Example for $m = 4, r = 2$:

$$\begin{aligned}
\vec{v_0} &= 1111111111111111 \\
\vec{v_4} &= 0000000011111111 \\
\vec{v_3} &= 0000111100001111 \\
\vec{v_2} &= 0011001100110011 \\
\vec{v_1} &= 0101010101010101 \\
\vec{v_4}\vec{v_3} &= 0000000000001111 \\
\vec{v_4}\vec{v_2} &= 0000000000110011 \\
\vec{v_4}\vec{v_1} &= 0000000001010101 \\
\vec{v_3}\vec{v_2} &= 0000001100000011 \\
\vec{v_3}\vec{v_1} &= 0000010100000101 \\
\vec{v_2}\vec{v_1} &= 0001000100010001
\end{aligned}$$

We have in general 2^{m-r} independent determinations of a_{21}

This means that $\frac{2^{m-r}}{2} - 1 = 2^{m-r-1} - 1$ errors can be corrected.

Reed-Muller codes

Decoding of Reed-Muller codes

Similar determinations can be made for

$a_{31}, a_{32}, a_{41}, a_{42}, a_{43}$

after these values are determined,

$$
\begin{aligned}
& a_{43}\,\vec{v_4}\,\vec{v_3} + a_{42}\,\vec{v_4}\,\vec{v_2} + a_{41}\,\vec{v_4}\,\vec{v_1} \\
+ \quad & a_{32}\,\vec{v_3}\,\vec{v_2} + a_{31}\,\vec{v_3}\,\vec{v_1} + a_{21}\,\vec{v_2}\,\vec{v_1}
\end{aligned}
$$

can be subtracted from the received vector to achieve

$$
\begin{aligned}
r' &= a_0\,\vec{v_0} + a_4\,\vec{v_4} + a_3\,\vec{v_3} + a_2\,\vec{v_2} + a_1\,\vec{v_1} \\
&= (b'_1, b'_2, \ldots, b'_n)
\end{aligned}
$$

Example for $m = 4, r = 2$:

$$
\begin{aligned}
\vec{v_0} &= 1111111111111111 \\
\vec{v_4} &= 0000000011111111 \\
\vec{v_3} &= 0000111100001111 \\
\vec{v_2} &= 0011001100110011 \\
\vec{v_1} &= 0101010101010101 \\
\vec{v_4}\,\vec{v_3} &= 0000000000001111 \\
\vec{v_4}\,\vec{v_2} &= 0000000000110011 \\
\vec{v_4}\,\vec{v_1} &= 0000000001010101 \\
\vec{v_3}\,\vec{v_2} &= 0000001100000011 \\
\vec{v_3}\,\vec{v_1} &= 0000010100000101 \\
\vec{v_2}\,\vec{v_1} &= 0001000100010001
\end{aligned}
$$

Reed-Muller codes

Decoding of Reed-Muller codes

The next set of coefficients can be determined in a similar way

There are eight equations that a_1 should satisfy:

$$\begin{aligned} a_1 &= b'_1 + b'_2 = b'_3 + b'_4 = b'_5 + b'_6 = b'_7 + b'_8 \\ &= b'_9 + b'_{10} = b'_{11} + b'_{12} = b'_{13} + b'_{14} = b'_{15} + b'_{16} \end{aligned}$$

Similar equations hold for a_2, a_3, a_4

Finally, in the absence of errors,

$$r' - a_4 \overrightarrow{v_4} - a_3 \overrightarrow{v_3} - a_2 \overrightarrow{v_2} - a_1 \overrightarrow{v_1} = a_0 \overrightarrow{v_0}$$

Outline

BCH codes

BCH-codes

BHC-codes as a class are the best known nonrandom codes for channels in which errors affect successive symbols independently.

Let α be an element of $GF(q^m)$

For any specified m_0 and d_0 the code generated by $g(X)$ is a BCH code iff $g(X)$ is the polynomial of lowest degree over $GF(q)$ for which $\alpha^{m_0}, \alpha^{m_0+1}, \ldots, \alpha^{m_0+d_0-2}$ are roots

The length n of the code is the order e of α.

BCH codes

BCH-codes

The minimum distance of the codes is at least d_0

Reed-Solomon codes are a subclass of BCH-codes with $m = m_0 = 1$

Outline

Questions?

Stephan Sigg
sigg@ibr.cs.tu-bs.de

Literature

C.M. Bishop: Pattern recognition and machine learning, Springer, 2007.

P. Tulys, B. Skoric, T. Kevenaar: Security with Noisy Data – On private biometrics, secure key storage and anti-counterfeiting, Springer, 2007.

W.W.Peterson, E.J. Weldon, Error-Correcting Codes, MIT press, 1972.

R.O. Duda, P.E. Hart, D.G. Stork: Pattern Classification, Wiley, 2001.