Part III – Advanced Coding Techniques

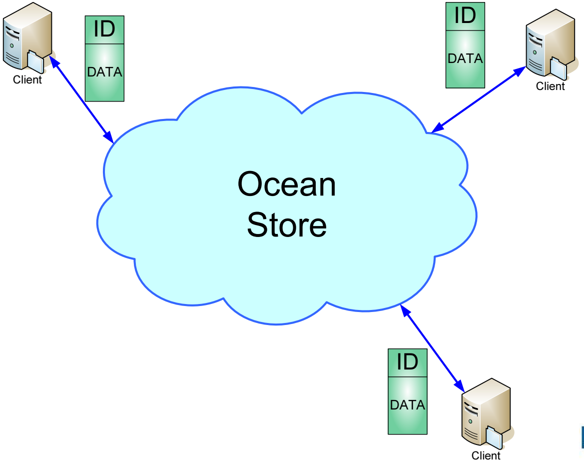
José Vieira

SPL — Signal Processing Laboratory
Departamento de Electrónica, Telecomunicações e Informática / IEETA
Universidade de Aveiro, Portugal

2010

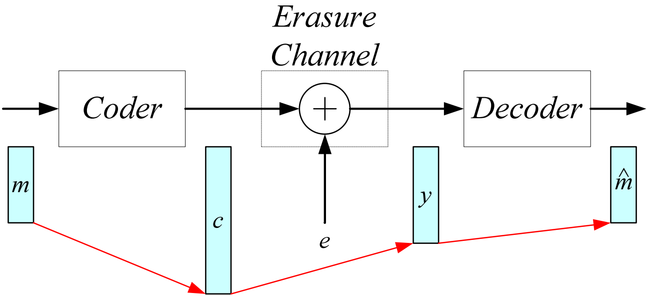
Ocean Store

The Infinite disk



Error Correction Code

Unpredictable Channel Conditions



Digital Fountains

Questions

- Dissemination of data: How to encode data files to distribute them by a huge number of disks around the world?
- Resilience: How can we encode data files to make any encoded data useful?
- Ratelessness: How to generate an infinite number of codewords?
- Answer: random coding!

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Digital Fountains

Questions

- Dissemination of data: How to encode data files to distribute them by a huge number of disks around the world?
- Resilience: How can we encode data files to make any encoded data useful?
- Ratelessness: How to generate an infinite number of codewords?
- Answer: random coding!

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Questions

- Dissemination of data: How to encode data files to distribute them by a huge number of disks around the world?
- Resilience: How can we encode data files to make any encoded data useful?
- Ratelessness: How to generate an infinite number of codewords?
- Answer: random coding!

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Digital Fountains

Questions

- Dissemination of data: How to encode data files to distribute them by a huge number of disks around the world?
- Resilience: How can we encode data files to make any encoded data useful?
- Ratelessness: How to generate an infinite number of codewords?
- Answer: random coding!

Essential Textbooks and Papers

- David J. C. Mackay, "Information Theory, Inference and Learning Algorithms", Cambridge, 2004
- Mackay, D. J. C., "Fountain Codes", IEE Proceedings - Communications, Vol.152, N.6, pp.1062-1068, December, 2005
- Luby, Michael, "LT Codes", Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'02), pp.271-280, IEEE, November, 2002
- Maymounkov, Petar, "Online Codes", New York University, New York, November, 2002
- Fragouli, Christina, Boudec, Jean-Yves Le, and Widmer, Jörg, "Network Coding: Na Instant Primer", ACM SIGCOMM Computer Communication Review, Vol.36, N.1, pp.63-68, January, 2006

MAP tele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Suplementar Papers

- Shokrollahi, Amin, "Raptor Codes", IEEE Transactions on Information Theory, Vol.52, N.6, pp.2551-2567, June, 2006
- Shamai, Shlomo, Telatar, I. Emre, and Verdú, Sergio, "Fountain Capacity", IEEE Transactions on Information Theory, Vol.53, N.11, pp.4372-4376, November, 2007
- Dimakis, Alexandros G., Prabhakaran, Vinod, and Ramchandran, Kannan, "Decentralized Erasure Codes for Distributed Networked Storage", IEEE Transactions on Information Theory, Vol.52, N.6, pp.2809-2816, June, 2006

Outline

1. **Linear Codes in any Field**
   - Correcting Erasures
   - Correcting Errors
2. **Coding Matrices**
   - Structured Matrices
   - Random Matrices
   - Sparse Random Matrices
3. **Coding with Sparse Random Matrices**
   - Fountain Codes
   - Applications

One linear combination

$$\underset{1\times 1}{\begin{bmatrix} c_1 \end{bmatrix}} = \underset{1\times K}{\begin{bmatrix} - & g_1 & - \end{bmatrix}} \underset{K\times 1}{\begin{bmatrix} | \\ m \\ | \end{bmatrix}}$$

Two linear combinations

$$\underbrace{\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}}_{2\times 1} = \underbrace{\begin{bmatrix} - & g_1 & - \\ - & g_2 & - \end{bmatrix}}_{2\times K} \underbrace{\begin{bmatrix} | \\ m \\ | \end{bmatrix}}_{K\times 1}$$

Adding redundancy to a signal — $N > K$

$$\underset{N\times 1}{\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}} = \underset{N\times K}{\begin{bmatrix} - & g_1 & - \\ - & g_2 & - \\ & \vdots & \\ - & g_N & - \end{bmatrix}} \underset{K\times 1}{\begin{bmatrix} | \\ m \\ | \end{bmatrix}}$$

$$\boxed{c = Gm}$$

Adding redundancy to a signal — $N > K$

$$\underset{N\times 1}{\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}} = \underset{N\times K}{\begin{bmatrix} - & g_1 & - \\ - & g_2 & - \\ & \vdots & \\ - & g_N & - \end{bmatrix}} \underset{K\times 1}{\begin{bmatrix} | \\ m \\ | \end{bmatrix}}$$

$$\boxed{c = Gm}$$

Outline

Erasures

Erasures

Lost samples at **known** positions

How to recover the message m from incomplete c?

$$\underbrace{\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix}}_{N\times 1} = \underbrace{\begin{bmatrix} - & g_1 & - \\ - & g_2 & - \\ - & g_3 & - \\ - & g_4 & - \\ - & g_5 & - \end{bmatrix}}_{N\times K} \underbrace{\begin{bmatrix} | \\ m \\ | \end{bmatrix}}_{K\times 1}$$

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Erasures

If only L samples of c are received we have:

$$\underset{N\times 1}{\begin{bmatrix} c_1 \\ \cancel{c_2} \\ c_3 \\ \cancel{c_4} \\ c_5 \end{bmatrix}} = \underset{N\times K}{\begin{bmatrix} - & g_1 & - \\ - & \cancel{g_2} & - \\ - & g_3 & - \\ - & \cancel{g_4} & - \\ - & g_5 & - \end{bmatrix}} \underset{K\times 1}{\begin{bmatrix} | \\ m \\ | \end{bmatrix}}$$

Define $J = \{1, 3, 5\}$ as the set of the received samples

MAP tele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Erasures

Solve the following system of equations to obtain the original signal m

$$\underset{L\times 1}{\begin{bmatrix} c_1 \\ c_3 \\ c_5 \end{bmatrix}} = \underset{L\times K}{\begin{bmatrix} - & g_1 & - \\ - & g_3 & - \\ - & g_5 & - \end{bmatrix}} \underset{K\times 1}{\begin{bmatrix} | \\ m \\ | \end{bmatrix}}$$

$$c(J) = G(J)m$$

Erasures

Depending on L (number of received samples), there are three possible situations

- **L** $<$ **K** – Underdetermined system of equations. In general not enough information to recover m uniquely. Additional restrictions can be imposed in order to get an unique solution.
- **L** $=$ **K** – Determined system of equations, one solution (max.).

$$\hat{m} = G(J)^{-1}c(J)$$

- **L** $>$ **K** – Overdetermined system of equations. In general there is not an unique solution. In the field $\mathbb{R}$ we can choose the least squares solution, the one that best approximates all the equations in L_2 sense.

$$\hat{m} = (G(J)^T G(J))^{-1} G(J)^T c(J) = G(J)^\dagger c(J)$$

Erasures

Depending on L (number of received samples), there are three possible situations

- **L** $<$ **K** – Underdetermined system of equations. In general not enough information to recover m uniquely. Additional restrictions can be imposed in order to get an unique solution.
- **L** $=$ **K** – Determined system of equations, one solution (max.).

$$\hat{m} = G(J)^{-1}c(J)$$

- **L** $>$ **K** – Overdetermined system of equations. In general there is not an unique solution. In the field $\mathbb{R}$ we can choose the least squares solution, the one that best approximates all the equations in L_2 sense.

$$\hat{m} = (G(J)^T G(J))^{-1} G(J)^T c(J) = G(J)^{\dagger} c(J)$$

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Erasures

Depending on L (number of received samples), there are three possible situations

- **L** $<$ **K** – Underdetermined system of equations. In general not enough information to recover m uniquely. Additional restrictions can be imposed in order to get an unique solution.
- **L** $=$ **K** – Determined system of equations, one solution (max.).

$$\hat{m} = G(J)^{-1} c(J)$$

- **L** $>$ **K** – Overdetermined system of equations. In general there is not an unique solution. In the field $\mathbb{R}$ we can choose the least squares solution, the one that best approximates all the equations in L_2 sense.

$$\hat{m} = (G(J)^T G(J))^{-1} G(J)^T c(J) = G(J)^{\dagger} c(J)$$

Outline

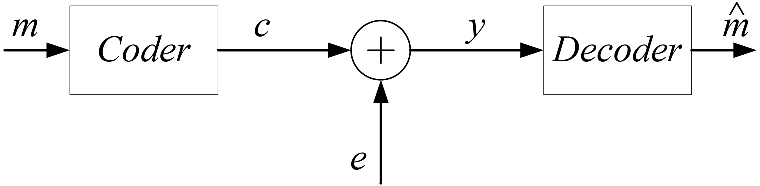
1. **Linear Codes in any Field**
   - Correcting Erasures
   - Correcting Errors
2. **Coding Matrices**
   - Structured Matrices
   - Random Matrices
   - Sparse Random Matrices
3. **Coding with Sparse Random Matrices**
   - Fountain Codes
   - Applications

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Correcting Errors

Errors

Errors: Lost samples at **unknown** positions



How can we find the errors positions?

Correcting Errors

- Consider an $N \times N$ orthogonal matrix partitioned in the following way:

$$F = \left[\begin{array}{c|c} & \\ G & H \\ & \\ N \times K & N \times (N-K) \end{array} \right]$$

- $\begin{bmatrix} G^T \\ H^T \end{bmatrix} [GH] = \begin{bmatrix} G^T G & G^T H \\ H^T G & H^T H \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$
- So we have $H^T G = 0$

Correcting Errors

- Consider an $N \times N$ orthogonal matrix partitioned in the following way:

$$F = \left[\begin{array}{c|c} G & H \\ {\scriptstyle N\times K} & {\scriptstyle N\times(N-K)} \end{array} \right]$$

- $\begin{bmatrix} G^T \\ H^T \end{bmatrix} [GH] = \begin{bmatrix} G^T G & G^T H \\ H^T G & H^T H \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$
- So we have $H^T G = 0$

Correcting Errors

- Consider an $N \times N$ orthogonal matrix partitioned in the following way:

$$F = \left[\begin{array}{c|c} G & H \\ \scriptstyle N\times K & \scriptstyle N\times(N-K) \end{array} \right]$$

- $\begin{bmatrix} G^T \\ H^T \end{bmatrix} [GH] = \begin{bmatrix} G^T G & G^T H \\ H^T G & H^T H \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$
- So we have $H^T G = 0$

Correcting Errors

- Received y and given F, if there are no errors, then

$$y = c = Gm$$

- We can use the matrix H to test the received signal y

$$s = H^T y = H^T c = \underbrace{H^T G}_{=0} m = 0$$

Correcting Errors

- Received y and given F, if there are no errors, then

$$y = c = Gm$$

- We can use the matrix H to test the received signal y

$$s = H^T y = H^T c = \underbrace{H^T G}_{=0} m = 0$$

Correcting Errors

- The received vector y is a corrupted version of c at unknown positions, example

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ e_3 \\ 0 \\ 0 \end{bmatrix}$$

$$y = c + e$$

- The matrix H is used to verify that the received signal y is a codeword

$$s = H^T y = H^T c + H^T e = H^T e \neq 0$$

- The *syndrome* s is a linear combination of the columns of H^T where the e_i are the coefficients.

Correcting Errors

- The received vector y is a corrupted version of c at unknown positions, example

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ e_3 \\ 0 \\ 0 \end{bmatrix}$$

$$y = c + e$$

- The matrix H is used to verify that the received signal y is a codeword

$$s = H^T y = H^T c + H^T e = H^T e \neq 0$$

- The *syndrome* s is a linear combination of the columns of H^T where the e_i are the coefficients.

MAP tele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Correcting Errors

- The received vector y is a corrupted version of c at unknown positions, example

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ e_3 \\ 0 \\ 0 \end{bmatrix}$$

$$y = c + e$$

- The matrix H is used to verify that the received signal y is a codeword

$$s = H^T y = H^T c + H^T e = H^T e \neq 0$$

- The *syndrome* s is a linear combination of the columns of H^T where the e_i are the coefficients.

Correcting Errors

- The received vector y is a corrupted version of c at unknown positions, example

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ e_3 \\ 0 \\ 0 \end{bmatrix}$$

$$y = c + e$$

- The matrix H is used to verify that the received signal y is a codeword

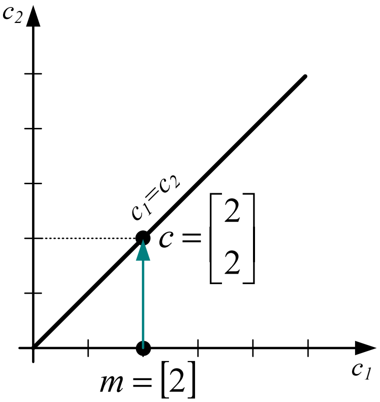
$$s = H^T y = H^T c + H^T e = H^T e \neq 0$$

- The *syndrome* s is a linear combination of the columns of H^T where the e_i are the coefficients.

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Correcting Errors

Example: repetition code

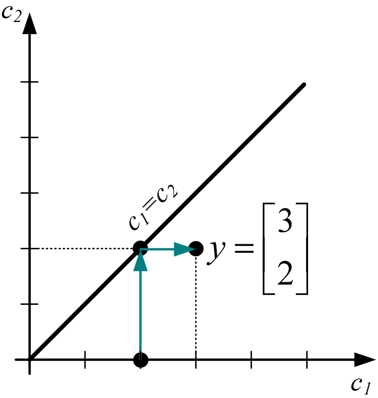


- Repetition code:

$$F = \begin{array}{c} \begin{array}{cc} G & H \end{array} \\ \left[\begin{array}{c|c} 1 & 1 \\ 1 & -1 \end{array} \right] \end{array}$$

- Suppose we code

$$c = Gm = \begin{bmatrix} 1 \\ 1 \end{bmatrix} [2] = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$



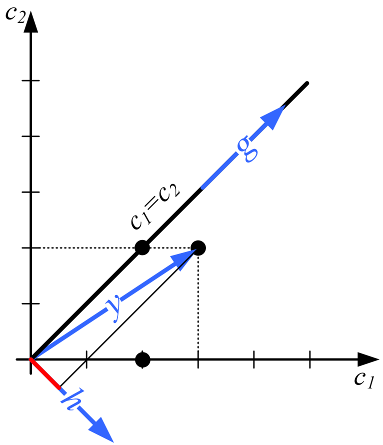
- Repetition code:

$$F = \begin{array}{c} \begin{array}{cc} G & \quad H \end{array} \\ \left[\begin{array}{c|c} 1 & 1 \\ 1 & -1 \end{array}\right] \end{array}$$

- Suppose we code

$$c = Gm = \begin{bmatrix}1\\1\end{bmatrix}[2] = \begin{bmatrix}2\\2\end{bmatrix}$$

- If an error occurs $y = c + e = \begin{bmatrix}2\\2\end{bmatrix} + \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}3\\2\end{bmatrix}$



- Repetition code:

$$F = \begin{array}{cc} G & H \\ \end{array} \left[\begin{array}{c|c} 1 & 1 \\ 1 & -1 \end{array}\right]$$

- Suppose we code

$$c = Gm = \begin{bmatrix} 1 \\ 1 \end{bmatrix} [2] = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

- If an error occurs $y = c + e =$

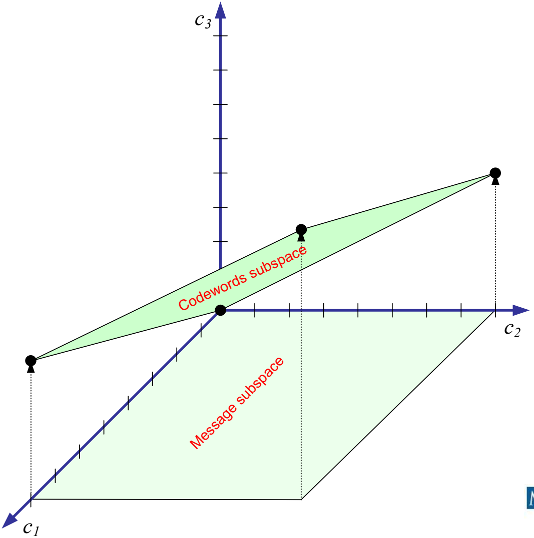
$$\begin{bmatrix} 2 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

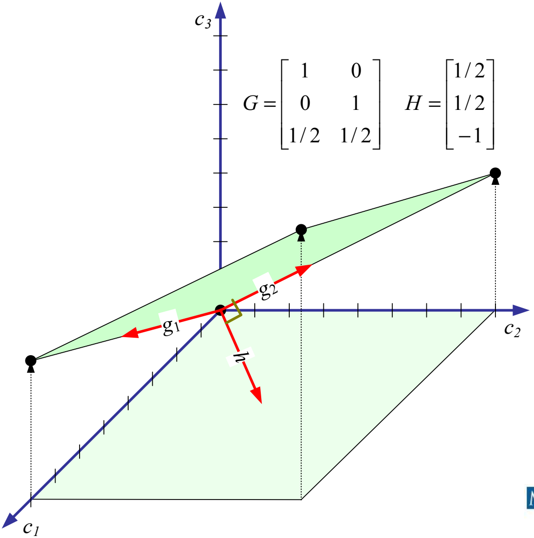
- We can verify that the received vector y has an error

$$H^T y = [1 - 1] \begin{bmatrix} 3 \\ 2 \end{bmatrix} = 1 \neq 0$$

Linear Codes

Coding with Real Numbers

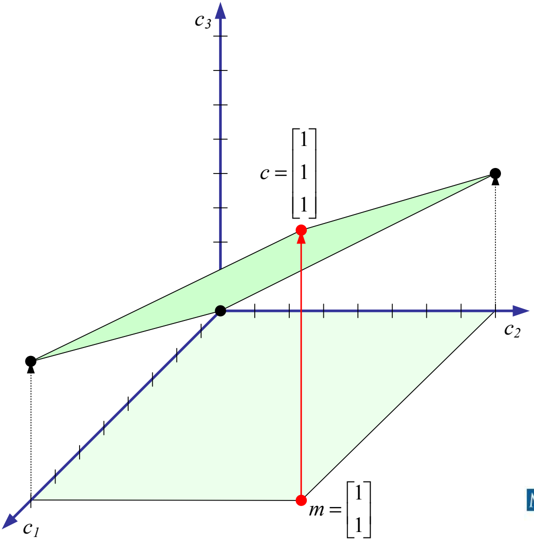




c_3
$G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1/2 & 1/2 \end{bmatrix}$
$H = \begin{bmatrix} 1/2 \\ 1/2 \\ -1 \end{bmatrix}$
g_2
g_1
h
c_2
c_1

Linear Codes

Coding with Real Numbers



- Linear code: $F = \begin{array}{c} \begin{array}{ccc} \quad G & & \quad H \end{array} \\ \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1/2 & 1/2 & -2 \end{array}\right] \end{array}$
- To code the message $m = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ we get $c = Gm = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
- If an error occurs $y = c + e = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1/2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1/2 \end{bmatrix}$
- To verify that the received vector y has an error

$$s = H^T y = \begin{bmatrix} 1 & 1 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1/2 \end{bmatrix} = 1 \neq 0$$

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Linear Codes

Coding with Real Numbers

- Linear code: $F = \begin{array}{c} \begin{array}{cc|c} \quad\quad G & & \quad H \end{array} \\ \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1/2 & 1/2 & -2 \end{array}\right] \end{array}$

- To code the message $m = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ we get $c = Gm = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

- If an error occurs $y = c + e = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1/2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1/2 \end{bmatrix}$

- To verify that the received vector y has an error

$$s = H^T y = \begin{bmatrix} 1 & 1 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1/2 \end{bmatrix} = 1 \neq 0$$

Linear Codes

Coding with Real Numbers

- Linear code: $F = \begin{array}{c} \begin{array}{cc|c} \quad\; G & & \;\; H \end{array} \\ \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1/2 & 1/2 & -2 \end{array}\right] \end{array}$
- To code the message $m = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ we get $c = Gm = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
- If an error occurs $y = c + e = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1/2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1/2 \end{bmatrix}$
- To verify that the received vector y has an error

$$s = H^T y = \begin{bmatrix} 1 & 1 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1/2 \end{bmatrix} = 1 \neq 0$$

- Linear code: $F = \begin{array}{c} \begin{array}{ccc} \quad G \quad & & H \end{array} \\ \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1/2 & 1/2 & -2 \end{array}\right] \end{array}$

- To code the message $m = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ we get $c = Gm = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

- If an error occurs $y = c + e = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1/2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1/2 \end{bmatrix}$

- To verify that the received vector y has an error

$$s = H^T y = \begin{bmatrix} 1 & 1 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1/2 \end{bmatrix} = 1 \neq 0$$

Correcting Errors

The Syndrome as a linear combination of columns of H^T

$$\begin{bmatrix} s_1 \\ \vdots \\ s_{N-K} \end{bmatrix} = \overset{H^T}{\begin{bmatrix} | & | & & | \\ h_1 & h_2 & \cdots & h_N \\ | & | & & | \end{bmatrix}} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ \vdots \\ e_N \end{bmatrix}$$

$$\begin{bmatrix} s_1 \\ \vdots \\ s_{N-K} \end{bmatrix} = e_1 \begin{bmatrix} | \\ h_1 \\ | \end{bmatrix} + e_2 \begin{bmatrix} | \\ h_2 \\ | \end{bmatrix} + \cdots + e_N \begin{bmatrix} | \\ h_N \\ | \end{bmatrix}$$

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Correcting Errors

The Syndrome as a linear combination of columns of H^T

$$\begin{bmatrix} s_1 \\ \vdots \\ s_{N-K} \end{bmatrix} = \begin{array}{c} H^T \\ \begin{bmatrix} | & | & & | \\ h_1 & h_2 & \cdots & h_N \\ | & | & & | \end{bmatrix} \end{array} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ \vdots \\ e_N \end{bmatrix}$$

$$\begin{bmatrix} s_1 \\ \vdots \\ s_{N-K} \end{bmatrix} = e_1 \begin{bmatrix} | \\ h_1 \\ | \end{bmatrix} + e_2 \begin{bmatrix} | \\ h_2 \\ | \end{bmatrix} + \cdots + e_N \begin{bmatrix} | \\ h_N \\ | \end{bmatrix}$$

Correcting Errors

Brute force approach

Problem

Find the linear combination of vectors h_i that best approximates s

- The error vector e is a sparse vector, so we want the sparsest solution
- Brute force approach: test all error patterns
- Equivalent to solve the following optimization problem:

Problem

$$min \|e\|_0 \quad s.t. \quad s = H^T e$$

Problem

Find the linear combination of vectors h_i that best approximates s

- The error vector e is a sparse vector, so we want the sparsest solution
- Brute force approach: test all error patterns
- Equivalent to solve the following optimization problem:

Problem

$$min \|e\|_0 \quad s.t. \quad s = H^T e$$

Problem

Find the linear combination of vectors h_i that best approximates s

- The error vector e is a sparse vector, so we want the sparsest solution
- Brute force approach: test all error patterns
- Equivalent to solve the following optimization problem:

Problem

$$min \|e\|_0 \quad s.t. \quad s = H^T e$$

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Correcting Errors

Brute force approach

- Search for the "best" linear combination

$$\begin{bmatrix} s_1 \\ \vdots \\ s_{N-K} \end{bmatrix} = e_1 \begin{bmatrix} | \\ h_1 \\ | \end{bmatrix} + e_2 \begin{bmatrix} | \\ h_2 \\ | \end{bmatrix} + \cdots + e_N \begin{bmatrix} | \\ h_N \\ | \end{bmatrix}$$

- Find the minimum of $\|s - \hat{s}\|_2$ for all error patterns and each number of errors

n		Nº Combinations
1	$\hat{s} = e_i h_i$	N
2	$\hat{s} = e_i h_i + e_j h_j$	$\binom{N}{2}$
$\vdots$	$\vdots$	$\vdots$
L	$\hat{s} = \sum_{i=J} e_i h_i$	$\sum_{n=1}^{L} \binom{N}{n}$

- This is a *NP* hard combinatorial problem.

MAP tele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Correcting Errors

Brute force approach

- Search for the "best" linear combination

$$\begin{bmatrix} s_1 \\ \vdots \\ s_{N-K} \end{bmatrix} = e_1 \begin{bmatrix} | \\ h_1 \\ | \end{bmatrix} + e_2 \begin{bmatrix} | \\ h_2 \\ | \end{bmatrix} + \cdots + e_N \begin{bmatrix} | \\ h_N \\ | \end{bmatrix}$$

- Find the minimum of $\|s - \hat{s}\|_2$ for all error patterns and each number of errors

n		N^o Combinations
1	$\hat{s} = e_i h_i$	N
2	$\hat{s} = e_i h_i + e_j h_j$	$\binom{N}{2}$
$\vdots$	$\vdots$	$\vdots$
L	$\hat{s} = \sum_{i=J} e_i h_i$	$\sum_{n=1}^{L} \binom{N}{n}$

- This is a *NP* hard combinatorial problem.

Correcting Errors

Brute force approach

- Search for the "best" linear combination

$$\begin{bmatrix} s_1 \\ \vdots \\ s_{N-K} \end{bmatrix} = e_1 \begin{bmatrix} | \\ h_1 \\ | \end{bmatrix} + e_2 \begin{bmatrix} | \\ h_2 \\ | \end{bmatrix} + \cdots + e_N \begin{bmatrix} | \\ h_N \\ | \end{bmatrix}$$

- Find the minimum of $\|s - \hat{s}\|_2$ for all error patterns and each number of errors

n		N^o Combinations
1	$\hat{s} = e_i h_i$	N
2	$\hat{s} = e_i h_i + e_j h_j$	$\binom{N}{2}$
$\vdots$	$\vdots$	$\vdots$
L	$\hat{s} = \sum_{i=J} e_i h_i$	$\sum_{n=1}^{L} \binom{N}{n}$

- This is a *NP* hard combinatorial problem.

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

- Solution 1: Use coding matrices G and parity check matrices H with a convenient **structure**:
  - Hamming
  - DFT - (BCH cyclic codes)
  - DCT
  - etc.
- Solution 2: Use **random matrices** and L_1 minimization to obtain a sparse solution for the error vector
- Solution 3: Use **sparse random matrices** and fast algorithms that take advantage of sparsity (LDPC and LT codes)

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

- Solution 1: Use coding matrices G and parity check matrices H with a convenient **structure**:
  - Hamming
  - DFT - (BCH cyclic codes)
  - DCT
  - etc.
- Solution 2: Use **random matrices** and L_1 minimization to obtain a sparse solution for the error vector
- Solution 3: Use **sparse random matrices** and fast algorithms that take advantage of sparsity (LDPC and LT codes)

- Solution 1: Use coding matrices G and parity check matrices H with a convenient **structure**:
  - Hamming
  - DFT - (BCH cyclic codes)
  - DCT
  - etc.
- Solution 2: Use **random matrices** and L_1 minimization to obtain a sparse solution for the error vector
- Solution 3: Use **sparse random matrices** and fast algorithms that take advantage of sparsity (LDPC and LT codes)

Outline

Solution 1

Coding with structured matrices

- Choose β_i as the roots of unity in any field (finite or not). Note that the roots of unity are the solutions of $a^n = 1$
- Construct the Vandermonde matrix

$$\begin{bmatrix} \beta_0^0 & \beta_1^0 & \cdots & \beta_{N-1}^0 \\ \beta_0^1 & \beta_1^1 & \cdots & \beta_{N-1}^1 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_0^{N-1} & \beta_1^{N-1} & \cdots & \beta_{N-1}^{N-1} \end{bmatrix} \quad \text{with} \quad \beta_i \neq \beta_j$$

- These codes can correct at least $\frac{N-K}{2}$ errors

MAP tele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Solution 1

Coding with the DFT

- In a Galois field, only certain values of N have roots of unity
- In the Complex field $\mathbb{C}$ the roots of unity of order N are $\beta_i = e^{j\frac{2\pi}{N}i}$ - DFT matrix
- These codes are known as the BCH codes
- A codeword c is generated by evaluating the IDFT of a zero padded message vector m

$$\begin{bmatrix} | \\ | \\ c \\ | \\ | \end{bmatrix} = \begin{bmatrix} & & \\ & IDFT & \\ & & \end{bmatrix} \begin{bmatrix} | \\ m \\ | \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} & & \\ & G & \\ & & \end{bmatrix} \begin{bmatrix} | \\ m \\ | \end{bmatrix}$$

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Solution 1

Coding with the DFT

- The syndrome s is part of the DFT of e

$$s = H^T e$$

- The complete equation will be

$$\begin{bmatrix} s' \\ s \end{bmatrix} = \begin{bmatrix} G^T \\ H^T \end{bmatrix} e$$

- If we have a way of obtaining s' then we could calculate the error e by inverse transform.
- As e is sparse with only L values different from zero, s is a linear combination of only L components:

$$s_n = \sum_{k=1}^{L} a_k s_{n-k}$$

- If we know $2L$ values of the syndrome we can correct L errors

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Solution 1

Coding with the DFT

- The syndrome s is part of the DFT of e

$$s = H^T e$$

- The complete equation will be

$$\begin{bmatrix} s' \\ s \end{bmatrix} = \begin{bmatrix} G^T \\ H^T \end{bmatrix} e$$

- If we have a way of obtaining s' then we could calculate the error e by inverse transform.
- As e is sparse with only L values different from zero, s is a linear combination of only L components:

$$s_n = \sum_{k=1}^{L} a_k s_{n-k}$$

- If we know $2L$ values of the syndrome we can correct L errors

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Solution 1

Coding with the DFT

- The syndrome s is part of the DFT of e

$$s = H^T e$$

- The complete equation will be

$$\begin{bmatrix} s' \\ s \end{bmatrix} = \begin{bmatrix} G^T \\ H^T \end{bmatrix} e$$

- If we have a way of obtaining s' then we could calculate the error e by inverse transform.
- As e is sparse with only L values different from zero, s is a linear combination of only L components:

$$s_n = \sum_{k=1}^{L} a_k s_{n-k}$$

- If we know $2L$ values of the syndrome we can correct L errors

Solution 1

Coding with the DFT

- The syndrome s is part of the DFT of e

$$s = H^T e$$

- The complete equation will be

$$\begin{bmatrix} s' \\ s \end{bmatrix} = \begin{bmatrix} G^T \\ H^T \end{bmatrix} e$$

- If we have a way of obtaining s' then we could calculate the error e by inverse transform.
- As e is sparse with only L values different from zero, s is a linear combination of only L components:

$$s_n = \sum_{k=1}^{L} a_k s_{n-k}$$

- If we know $2L$ values of the syndrome we can correct L errors

- The syndrome s is part of the DFT of e

$$s = H^T e$$

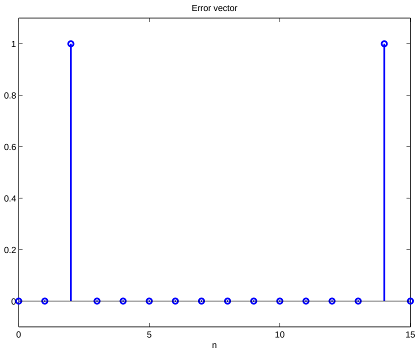
- The complete equation will be

$$\begin{bmatrix} s' \\ s \end{bmatrix} = \begin{bmatrix} G^T \\ H^T \end{bmatrix} e$$

- If we have a way of obtaining s' then we could calculate the error e by inverse transform.
- As e is sparse with only L values different from zero, s is a linear combination of only L components:

$$s_n = \sum_{k=1}^{L} a_k s_{n-k}$$

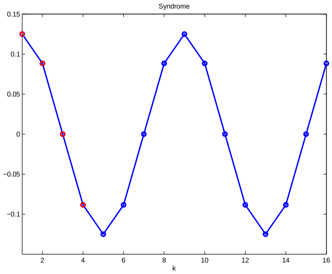
- If we know $2L$ values of the syndrome we can correct L errors



- Consider the following example:
- $N = 16$
- Two errors at $J = \{2, 14\}$

Solution 1

Decoding example



- In this example, due to the error vector symmetry, the syndrome is a real vector
- We have the following difference equation

$$s_n = a_1 s_{n-1} + a_2 s_{n-2}$$

- We have two unknowns and with the four known elements of the syndrome we can form

$$\begin{cases} s_3 = a_1 s_2 + a_2 s_1 \\ s_4 = a_1 s_3 + a_2 s_2 \end{cases}$$

- Due to the structure of the coding matrix G and the parity check matrix H, the syndrome reconstruction is very sensitive to burst of errors
- This is a direct consequence of the structure of the matrix H. Contiguous row vectors are almost colinear, leading to bad conditioned system of equations
- To improve the reconstruction stability, we have to modify the structure of H
- On real number codes we have stability problems. On finite fields those codes behave poorly for burst errors
- **Conclusion**: The coding matrix structure is fundamental for both fields: Real and Finite

Solution 1

Stability Problems

- Due to the structure of the coding matrix G and the parity check matrix H, the syndrome reconstruction is very sensitive to burst of errors
- This is a direct consequence of the structure of the matrix H. Contiguous row vectors are almost colinear, leading to bad conditioned system of equations
- To improve the reconstruction stability, we have to modify the structure of H
- On real number codes we have stability problems. On finite fields those codes behave poorly for burst errors
- **Conclusion**: The coding matrix structure is fundamental for both fields: Real and Finite

Solution 1

Stability Problems

- Due to the structure of the coding matrix G and the parity check matrix H, the syndrome reconstruction is very sensitive to burst of errors
- This is a direct consequence of the structure of the matrix H. Contiguous row vectors are almost colinear, leading to bad conditioned system of equations
- To improve the reconstruction stability, we have to modify the structure of H
- On real number codes we have stability problems. On finite fields those codes behave poorly for burst errors
- **Conclusion**: The coding matrix structure is fundamental for both fields: Real and Finite

Solution 1

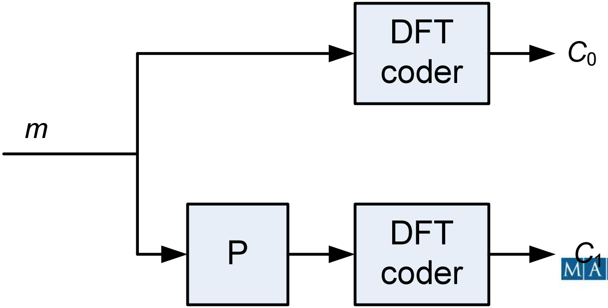
Stability Problems

- Due to the structure of the coding matrix G and the parity check matrix H, the syndrome reconstruction is very sensitive to burst of errors
- This is a direct consequence of the structure of the matrix H. Contiguous row vectors are almost colinear, leading to bad conditioned system of equations
- To improve the reconstruction stability, we have to modify the structure of H
- On real number codes we have stability problems. On finite fields those codes behave poorly for burst errors
- **Conclusion**: The coding matrix structure is fundamental for both fields: Real and Finite

- Due to the structure of the coding matrix G and the parity check matrix H, the syndrome reconstruction is very sensitive to burst of errors
- This is a direct consequence of the structure of the matrix H. Contiguous row vectors are almost colinear, leading to bad conditioned system of equations
- To improve the reconstruction stability, we have to modify the structure of H
- On real number codes we have stability problems. On finite fields those codes behave poorly for burst errors
- **Conclusion**: The coding matrix structure is fundamental for both fields: Real and Finite

Solution $1 + 1/2$
Turbo Codes

- The first attempt to randomise the coding matrix structure was achieved with turbo codes
- The message is codded with two different coding matrices usually the DFT and a column permuted version
- Those codes perform better than the BCH codes for both fields. They come close to the Shannon limit



Outline

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Solution 2

Coding with random matrices

- Consider a $N \times K$ random real number coding matrix G

$$c = Gm$$

- To correct errors we need to evaluate the syndrome with a parity check matrix H which must obey the condition

$$H^T G = 0$$

- The columns of H should be orthogonal to the columns of G. This can be obtained by applying the Gram-Schmidt orthogonalization algorithm to a $N \times N$ random matrix

Solution 2

Coding with random matrices

Problem

How to solve the underdetermined system of equations

$$s = H^T e$$

- As H has no structure a general method must be found
- Additional restrictions must be applied to the vector e in order to define an unique solution, e.g.:
  - Minimum energy - $\min \|e\|_2$ (L_2 *norm*)
  - Sparsest - $\min \|e\|_0$ (L_0 *pseudonorm*)

Problem

How to solve the underdetermined system of equations

$$s = H^T e$$

- As H has no structure a general method must be found
- Additional restrictions must be applied to the vector e in order to define an unique solution, e.g.:
  - Minimum energy - $\min \|e\|_2$ (L_2 *norm*)
  - Sparsest - $\min \|e\|_0$ (L_0 *pseudonorm*)

Problem

How to solve the underdetermined system of equations

$$s = H^T e$$

- As H has no structure a general method must be found
- Additional restrictions must be applied to the vector e in order to define an unique solution, e.g.:
  - Minimum energy - $\min \|e\|_2$ (L_2 *norm*)
  - Sparsest - $\min \|e\|_0$ (L_0 *pseudonorm*)

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Solution 2

Coding with random matrices

Problem

How to solve the underdetermined system of equations

$$s = H^T e$$

- As H has no structure a general method must be found
- Additional restrictions must be applied to the vector e in order to define an unique solution, e.g.:
  - Minimum energy - $\min \|e\|_2$ (L_2 *norm*)
  - Sparsest - $\min \|e\|_0$ (L_0 *pseudonorm*)

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Solution 2

Coding with random matrices

Problem

How to solve the underdetermined system of equations

$$s = H^T e$$

- As H has no structure a general method must be found
- Additional restrictions must be applied to the vector e in order to define an unique solution, e.g.:
  - Minimum energy - $\min \|e\|_2$ (L_2 *norm*)
  - Sparsest - $\min \|e\|_0$ (L_0 *pseudonorm*)

Solution 2

L_0 to L_1 equivalence

- Donoho and Elad in 2001 founded empirically and theoretically that instead of solving the hard L_0 problem to find the sparsest solution

Problem

$$\min \|e\|_0 \quad s.t. \quad s = H^T e$$

- They could solve the easiest L_1 problem and under certain conditions, still obtain the same sparsest solution

Problem

$$\min \|e\|_1 \quad s.t. \quad s = H^T e$$

- This problem can be solved by Linear Programing using the Simplex algorithm or Interior Point methods

Solution 2

L_0 to L_1 equivalence

- Donoho and Elad in 2001 founded empirically and theoretically that instead of solving the hard L_0 problem to find the sparsest solution

Problem

$$\min \|e\|_0 \quad s.t. \quad s = H^T e$$

- They could solve the easiest L_1 problem and under certain conditions, still obtain the same sparsest solution

Problem

$$\min \|e\|_1 \quad s.t. \quad s = H^T e$$

- This problem can be solved by Linear Programing using the Simplex algorithm or Interior Point methods

MAP tele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

- Donoho and Elad in 2001 founded empirically and theoretically that instead of solving the hard L_0 problem to find the sparsest solution

Problem

$$\min \|e\|_0 \quad s.t. \quad s = H^T e$$

- They could solve the easiest L_1 problem and under certain conditions, still obtain the same sparsest solution

Problem

$$\min \|e\|_1 \quad s.t. \quad s = H^T e$$

- This problem can be solved by Linear Programing using the Simplex algorithm or Interior Point methods

Solution 2

Coding with random matrices

- We can write the equation to solve in the following form

$$\begin{bmatrix} s_1 \\ \vdots \\ s_{N-K} \end{bmatrix} = e_1 \begin{bmatrix} | \\ h_1 \\ | \end{bmatrix} + e_2 \begin{bmatrix} | \\ h_2 \\ | \end{bmatrix} + \cdots + e_N \begin{bmatrix} | \\ h_N \\ | \end{bmatrix}$$

- The syndrome s is a linear combination of L vectors h_i
- We want to find the linear combination of vectors h_i that better "explains" the syndrome using the smallest number of vectors h_i

Solution 2

Coding with random matrices

$$L < \frac{1 + 1/M(H^T)}{2} = ebp$$

- *ebp* is the Equivalent Break Point and is an estimate of the maximum number of correctable errors
- $M(A)$ is the mutual incoherence of matrix A

Definition

$$M(H^T) = \max_{i \neq j} \left| h_i^T h_j \right|, \quad \textit{such that} \quad \|h_k\|_2 = 1$$

- How to choose H?
- All the sets of $N - K$ columns of H^T should be linearly independent
- Ideally, $h_i^T h_j \approx 0 \qquad i \neq j$

$$L < \frac{1 + 1/M(H^T)}{2} = ebp$$

- *ebp* is the Equivalent Break Point and is an estimate of the maximum number of correctable errors
- $M(A)$ is the mutual incoherence of matrix A

Definition

$$M(H^T) = \max_{i \neq j} \left| h_i^T h_j \right|, \quad \textit{such that} \quad \|h_k\|_2 = 1$$

- How to choose H?
- All the sets of $N - K$ columns of H^T should be linearly independent
- Ideally, $h_i^T h_j \approx 0 \qquad i \neq j$

Solution 2

Coding with random matrices

$$L < \frac{1 + 1/M(H^T)}{2} = ebp$$

- *ebp* is the Equivalent Break Point and is an estimate of the maximum number of correctable errors
- $M(A)$ is the mutual incoherence of matrix A

Definition

$$M(H^T) = \max_{i \neq j} \left| h_i^T h_j \right|, \quad \text{such that} \quad \|h_k\|_2 = 1$$

- How to choose H?
- All the sets of $N - K$ columns of H^T should be linearly independent
- Ideally, $h_i^T h_j \approx 0 \qquad i \neq j$

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Solution 2

Coding with random matrices

$$L < \frac{1 + 1/M(H^T)}{2} = ebp$$

- *ebp* is the Equivalent Break Point and is an estimate of the maximum number of correctable errors
- $M(A)$ is the mutual incoherence of matrix A

Definition

$$M(H^T) = \max_{i \neq j} \left| h_i^T h_j \right|, \quad \textit{such that} \quad \|h_k\|_2 = 1$$

- How to choose H?
- All the sets of $N - K$ columns of H^T should be linearly independent
- Ideally, $h_i^T h_j \approx 0 \qquad i \neq j$

$$L < \frac{1 + 1/M(H^T)}{2} = ebp$$

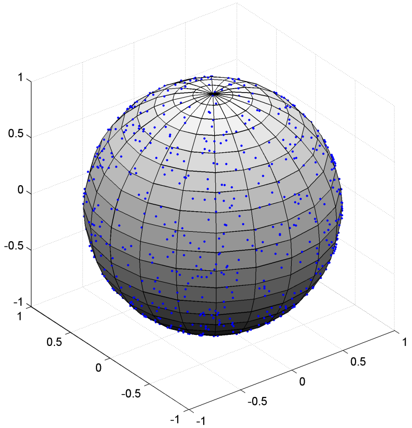
- *ebp* is the Equivalent Break Point and is an estimate of the maximum number of correctable errors
- $M(A)$ is the mutual incoherence of matrix A

Definition

$$M(H^T) = \max_{i \neq j} \left| h_i^T h_j \right|, \quad \textit{such that} \quad \|h_k\|_2 = 1$$

- How to choose H?
- All the sets of $N - K$ columns of H^T should be linearly independent
- Ideally, $h_i^T h_j \approx 0 \qquad i \neq j$

Random matrices are the solution



Outline

Solution 3

Coding with random sparse matrices

- With random sparse matrices is possible to find efficient algorithms to code and decode.
- This algorithms make use of the sparsity and avoid the slower L1 optimisation
- We will introduce two different types of codes that uses sparse matrices
  - LDPC codes – Low-Density Parity-Check codes [Gallager 1968]
  - LT codes – The first rateless erasure codes [Luby 2002]
  - Online codes – Almost the first rateless erasure code [Maymounkov 2002]

Solution 3

Coding with random sparse matrices

- With random sparse matrices is possible to find efficient algorithms to code and decode.
- This algorithms make use of the sparsity and avoid the slower L1 optimisation
- We will introduce two different types of codes that uses sparse matrices
  - LDPC codes – Low-Density Parity-Check codes [Gallager 1968]
  - LT codes – The first rateless erasure codes [Luby 2002]
  - Online codes – Almost the first rateless erasure code [Maymounkov 2002]

Solution 3

Coding with random sparse matrices

- With random sparse matrices is possible to find efficient algorithms to code and decode.
- This algorithms make use of the sparsity and avoid the slower L1 optimisation
- We will introduce two different types of codes that uses sparse matrices
  - LDPC codes – Low-Density Parity-Check codes [Gallager 1968]
  - LT codes – The first rateless erasure codes [Luby 2002]
  - Online codes – Almost the first rateless erasure code [Maymounkov 2002]

Solution 3

LDPC codes

- The parity check matrix is highly sparse: the number of nonzero elements grows linearly with N
- Due to sparsity low complexity algorithms exists
- The parity check matrix can be generated randomly but must obey certain rules
- Usually N is very large (1000 to 10000 or more)
- Usually the coding matrix is not sparse, which implies a coding complexity quadratic with N
- As N becomes large the LDPC codes approach the Shannon limit [MacKay1999]
- An "optimal" LDPC code can get within ≈ 0.005 dB of channel capacity

Solution 3

LDPC codes

- The parity check matrix is highly sparse: the number of nonzero elements grows linearly with N
- Due to sparsity low complexity algorithms exists
- The parity check matrix can be generated randomly but must obey certain rules
- Usually N is very large (1000 to 10000 or more)
- Usually the coding matrix is not sparse, which implies a coding complexity quadratic with N
- As N becomes large the LDPC codes approach the Shannon limit [MacKay1999]
- An "optimal" LDPC code can get within ≈ 0.005 dB of channel capacity

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

- The parity check matrix is highly sparse: the number of nonzero elements grows linearly with N
- Due to sparsity low complexity algorithms exists
- The parity check matrix can be generated randomly but must obey certain rules
- Usually N is very large (1000 to 10000 or more)
- Usually the coding matrix is not sparse, which implies a coding complexity quadratic with N
- As N becomes large the LDPC codes approach the Shannon limit [MacKay1999]
- An “optimal” LDPC code can get within ≈ 0.005 dB of channel capacity

Solution 3

LDPC codes

- The parity check matrix is highly sparse: the number of nonzero elements grows linearly with N
- Due to sparsity low complexity algorithms exists
- The parity check matrix can be generated randomly but must obey certain rules
- Usually N is very large (1000 to 10000 or more)
- Usually the coding matrix is not sparse, which implies a coding complexity quadratic with N
- As N becomes large the LDPC codes approach the Shannon limit [MacKay1999]
- An "optimal" LDPC code can get within ≈ 0.005 dB of channel capacity

MAP tele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Solution 3

LDPC codes

- The parity check matrix is highly sparse: the number of nonzero elements grows linearly with N
- Due to sparsity low complexity algorithms exists
- The parity check matrix can be generated randomly but must obey certain rules
- Usually N is very large (1000 to 10000 or more)
- Usually the coding matrix is not sparse, which implies a coding complexity quadratic with N
- As N becomes large the LDPC codes approach the Shannon limit [MacKay1999]
- An "optimal" LDPC code can get within ≈ 0.005 dB of channel capacity

Solution 3

LDPC codes

- The parity check matrix is highly sparse: the number of nonzero elements grows linearly with N
- Due to sparsity low complexity algorithms exists
- The parity check matrix can be generated randomly but must obey certain rules
- Usually N is very large (1000 to 10000 or more)
- Usually the coding matrix is not sparse, which implies a coding complexity quadratic with N
- As N becomes large the LDPC codes approach the Shannon limit [MacKay1999]
- An "optimal" LDPC code can get within ≈ 0.005 dB of channel capacity

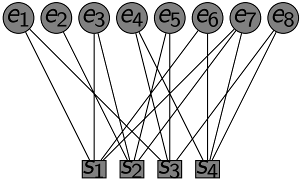
Solution 3

LDPC codes example

The LDPC parity check matrix can be represented by a Tanner graph

$$\begin{array}{c} H^T \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

$$s = H^T e$$



Outline

What is a Digital Fountain ?

- Is a new paradigm for data transmission that changes the standard approach where a user must receive an ordered stream of data symbols to one where the user must receive enough symbols to reconstruct the original information.
- With a Digital Fountain is possible to generate an infinite data stream from a K symbol file. Once the receiver gets any K symbols from the stream it can reconstruct the original message.

Digital Fountain ?

The name Digital Fountain comes from the analogy with a water fountain filling a glass of water. The glass must be filled up, not with some specific drops of water.

One linear combination

$$\underset{1\times 1}{\begin{bmatrix} c_1 \end{bmatrix}} = \begin{bmatrix} - & \underset{1\times K}{g_1} & - \end{bmatrix} \underset{K\times 1}{\begin{bmatrix} | \\ m \\ | \end{bmatrix}}$$

Two linear combinations

$$\underset{2\times 1}{\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}} = \underset{2\times K}{\begin{bmatrix} - & g_1 & - \\ - & g_2 & - \end{bmatrix}} \underset{K\times 1}{\begin{bmatrix} | \\ m \\ | \end{bmatrix}}$$

Infinite number of linear combinations

$$\underset{N\times 1}{\begin{bmatrix} c_1 \\ c_2 \\ \vdots \end{bmatrix}} = \underset{N\times K}{\begin{bmatrix} - & g_1 & - \\ - & g_2 & - \\ & \vdots & \end{bmatrix}} \underset{K\times 1}{\begin{bmatrix} | \\ m \\ | \end{bmatrix}}$$

- On Fountain Codes, each line g_i of G is generated online
- Each g_i has only a finite number of 1's (degree)
- The number of 1's is a random variable with distribution ρ
- The symbols to combine (XOR) are chosen randomly
- We can generate linear combinations as needed
- The receiver can be filled with codewords until it is sufficient to decode the original message
- The K original symbols can be recovered from $K(1+\epsilon)$ coded symbols with probability $1-\delta$

MAPtele DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

- On Fountain Codes, each line g_i of G is generated online
- Each g_i has only a finite number of 1's (degree)
- The number of 1's is a random variable with distribution ρ
- The symbols to combine (XOR) are chosen randomly
- We can generate linear combinations as needed
- The receiver can be filled with codewords until it is sufficient to decode the original message
- The K original symbols can be recovered from $K(1+\epsilon)$ coded symbols with probability $1-\delta$

- On Fountain Codes, each line g_i of G is generated online
- Each g_i has only a finite number of 1's (degree)
- The number of 1's is a random variable with distribution ρ
- The symbols to combine (XOR) are chosen randomly
- We can generate linear combinations as needed
- The receiver can be filled with codewords until it is sufficient to decode the original message
- The K original symbols can be recovered from $K(1+\epsilon)$ coded symbols with probability $1-\delta$

- On Fountain Codes, each line g_i of G is generated online
- Each g_i has only a finite number of 1's (degree)
- The number of 1's is a random variable with distribution ρ
- The symbols to combine (XOR) are chosen randomly
- We can generate linear combinations as needed
- The receiver can be filled with codewords until it is sufficient to decode the original message
- The K original symbols can be recovered from $K(1+\epsilon)$ coded symbols with probability $1-\delta$

- On Fountain Codes, each line g_i of G is generated online
- Each g_i has only a finite number of 1's (degree)
- The number of 1's is a random variable with distribution ρ
- The symbols to combine (XOR) are chosen randomly
- We can generate linear combinations as needed
- The receiver can be filled with codewords until it is sufficient to decode the original message
- The K original symbols can be recovered from $K(1+\epsilon)$ coded symbols with probability $1-\delta$

Fountain Codes

Concept

- On Fountain Codes, each line g_i of G is generated online
- Each g_i has only a finite number of 1's (degree)
- The number of 1's is a random variable with distribution ρ
- The symbols to combine (XOR) are chosen randomly
- We can generate linear combinations as needed
- The receiver can be filled with codewords until it is sufficient to decode the original message
- The K original symbols can be recovered from $K(1+\epsilon)$ coded symbols with probability $1-\delta$

DOCTORAL PROGRAMME IN TELECOMMUNICATIONS

Fountain Codes

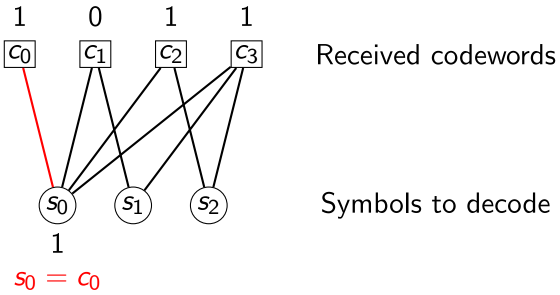
Concept

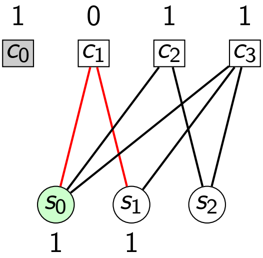
- On Fountain Codes, each line g_i of G is generated online
- Each g_i has only a finite number of 1's (degree)
- The number of 1's is a random variable with distribution ρ
- The symbols to combine (XOR) are chosen randomly
- We can generate linear combinations as needed
- The receiver can be filled with codewords until it is sufficient to decode the original message
- The K original symbols can be recovered from $K(1+\epsilon)$ coded symbols with probability $1-\delta$

1. Find a codeword c with all message symbols decoded except one
2. Recover that message symbol $m_x = c \oplus m_1 \oplus m_2 \oplus \cdots \oplus m_{i-1}$ where $m_1, m_2, \ldots, m_{i-1}$ are the recovered symbols associated with c
3. Apply the previous steps until no more message symbols left

Online Code

Decoding Example 1

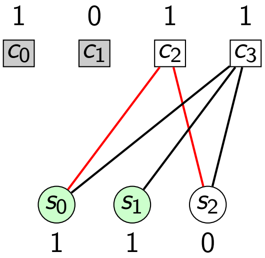




$$s_1 = c_1 \oplus s_0$$

Online Code

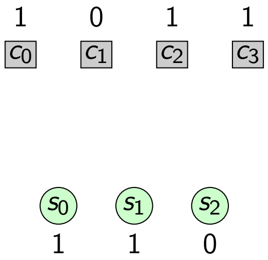
Decoding Example 3



$s_2 = c_2 \oplus s_0$

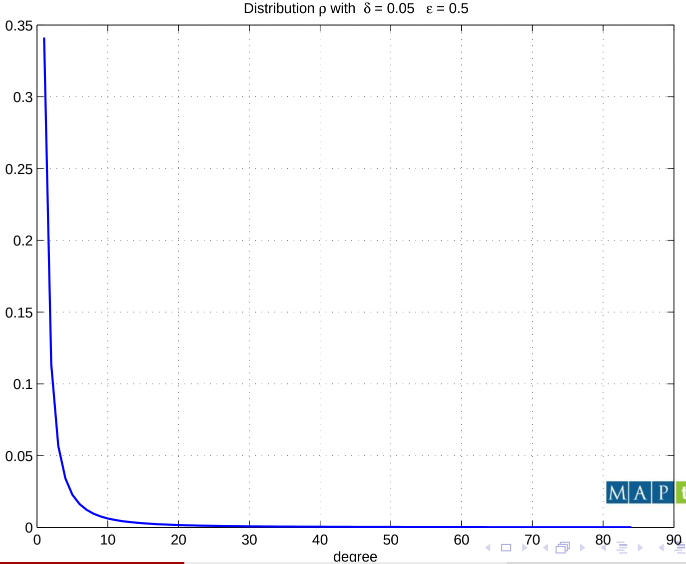
Online Code

Decoding Example 4



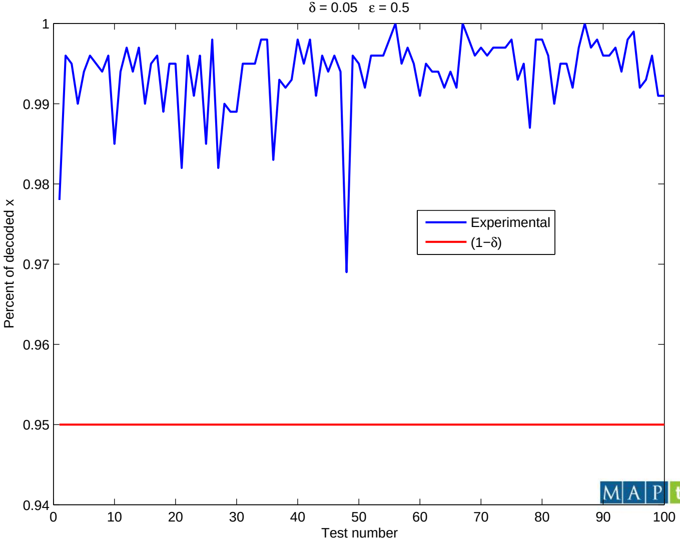
Done

Online Code
Distribution Example



Online Code

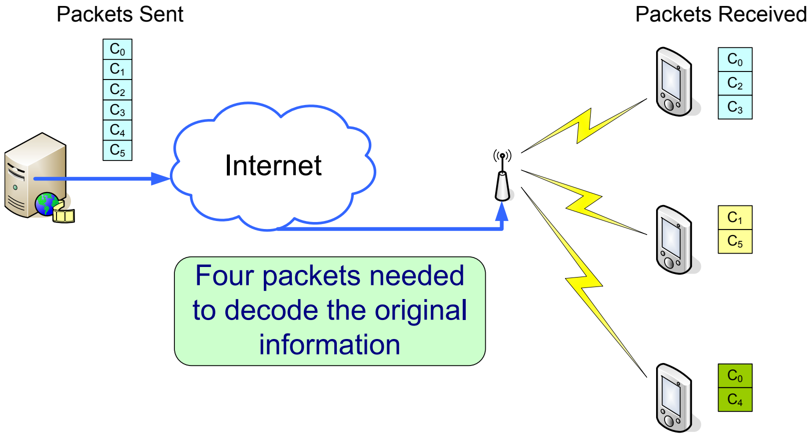
Decoding simulation



Outline

Fountain Codes

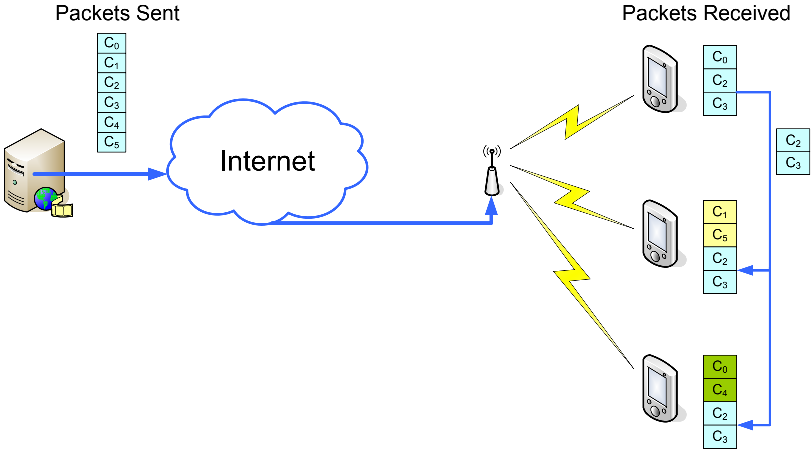
Applications



Four packets needed to decode the original information

Fountain Codes

Applications



Fountain Codes

Applications