Course on (algebraic aspects of) Convolutional Codes

Diego Napp

Department of Mathematics, Universidad of Aveiro, Portugal

CIDMA

CENTRO DE I&D EM MATEMÁTICA E APLICAÇÕES
CENTER FOR R&D IN MATHEMATICS AND APPLICATIONS

CIMPA RESEARCH SCHOOL

July 5, 2017

My most heartfelt thanks to the organizers

CIMPA RESEARCH SCHOOL

ALGEBRAIC METHODS IN CODING THEORY

Overview

1. Error-correcting codes: From block codes to convolutional codes
   - Basics: Polynomial encoders
2. Distance properties of convolutional codes
   - Maximum Distance Profile (MDP) and Maximum Distance Separable (MDS)
   - Construction of MDP and MDS: Superregular matrices
3. Decoding of Convolutional codes
   - Viterbi algorithm
   - Decoding of convolutional codes over the erasure channel
4. Network coding with convolutional codes
5. Avenues for further research
   - Motivated by applications: Video streaming and storage systems
   - More theoretical: Multidimensional convolutional codes and convolutional codes over $\mathbb{Z}_{p^r}$

Day 3: Decoding of convolutional codes

1. Maximum likelihood decoding: The Viterbi algorithm

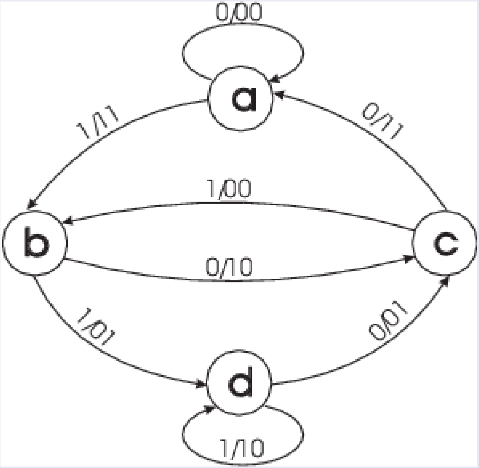


Figure: A convolutional code of rate $1/2$ and 4 states

Example

Notice that for each bit of information, the encoder outputs two bits, so we say that this is a convolutional code of rate $1/2$ and four states, a, b, c and d. We choose a as our initial state.

For instance, if we want to encode the string

$$1\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ \ldots,$$

we would have to take the path $abcabdcbd\ldots$, and the encoded string would be

$$11\ 10\ 11\ 11\ 01\ 01\ 00\ 01\ \ldots$$

In this context, linear block codes may be considered as a particular case of convolutional codes with only one state.

Example

Consider the encoder of the $[3, 2]$ code as a state diagram with 4 arrows going from a to itself. Each arrow corresponds to the 2 input bits, and it gives the output following the input.

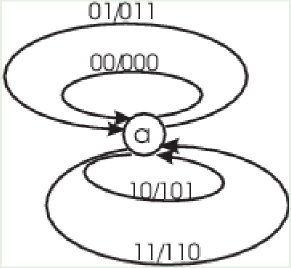
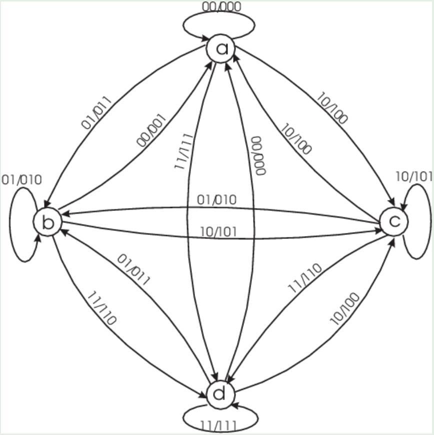


Figure: Encoder of a $[3, 2, 2]$ linear block code

Example

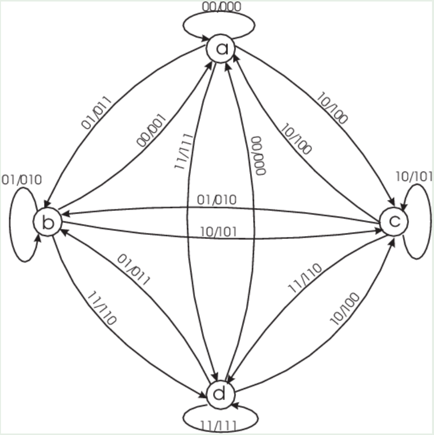


A convolutional code of rate 2/3 and 4 states. The string

$$01\ 11\ 10\ 00\ 10\ 01\ \ldots,$$

would take the path

Example



A convolutional code of rate $2/3$ and 4 states. The string

$$01\ 11\ 10\ 00\ 10\ 01\ \ldots,$$

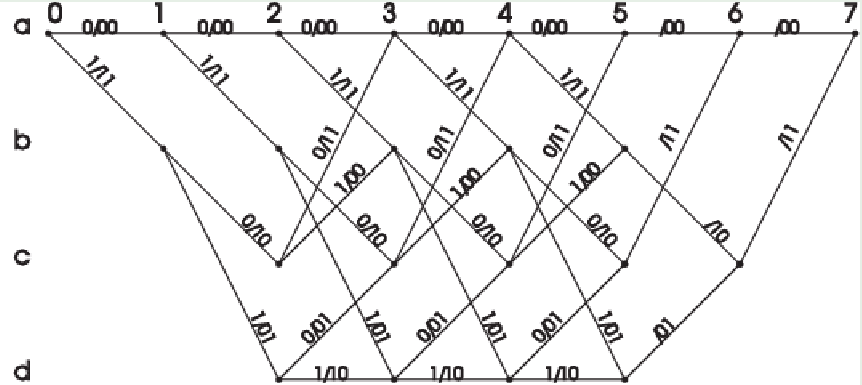
would take the path *abdcacb* . . ., and the encoded string would be

$$011\ 110\ 100\ 001\ 100\ 010\ \ldots.$$

Trellis structure of convolutional codes

Example

It is convenient to keep track of the time during the encoding: we can label the states as a_i, b_i, c_i and d_i, where, say, a_i means that we visit the state a_i at time (or iteration) i. Edges connect a state i with a state $i+1$ only, so the edges play now the role of the arrows in the state diagram.



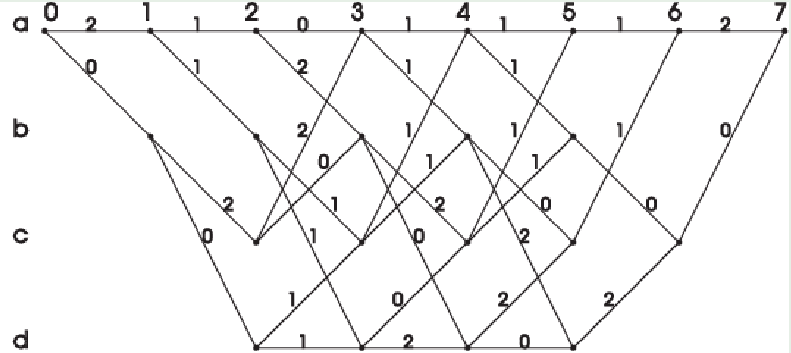
Trellis structure of convolutional codes

Example

Assume that we receive the sequence

$$11\ 01\ 00\ 01\ 10\ 10\ 11.$$

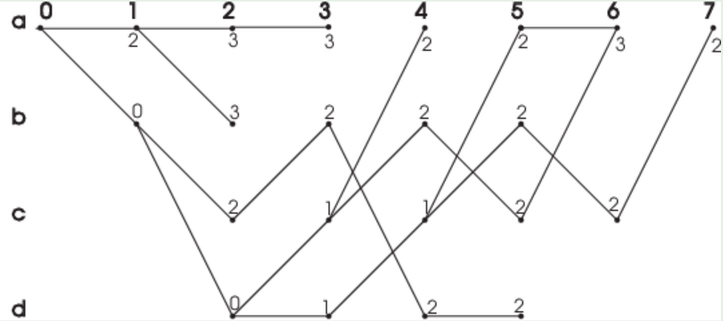
This means, at time 1 we receive 11, at time 2 we receive 01, etc. We label the edges in the trellis with the distances between the received sequence and the output bits.



Trellis structure of convolutional codes

Example

Erasing the paths with higher weights.



Exercise

Using the code given by the encoder of Figure 1 and $k = 8$, decode the received string

$$10\ 11\ 01\ 10\ 00\ 01\ 10\ 11\ 01$$

using the Viterbi Decoding Algorithm.

2. Decoding of convolutional codes over the erasure channel

It has been recently shown (Rosenthal 2012) that the decoding of convolutional codes over the erasure channel can be very efficient. This opens up many many interesting possible areas where convolutional codes can be applied.

Definition

The Erasure channel is a communication channel, where symbols either arrive correctly or the receiver knows that they are in error (or did not arrive at all).

Example

The Internet can be viewed as an erasure channel: packets are usually protected by cyclic redundancy-check codes, so the receiver can know when a packet is in error, or the buffer is full and packets are dropped out. Another example is Storage systems.

We take large alphabets, e.g., $\mathbb{F} = \mathbb{F}_{2^{1000}}$ We will show that the decoding of these convolutional codes over the erasure channel is possible in polynomial time by simple linear algebra.

Theorem

Let $\mathcal{C}$ be convolutional code of rate k/n and degree δ with $d_{j_0}^c$ j_0-th column distance. If in any sliding window of length $(j_0 + 1)n$ at most $d_{j_0}^c - 1$ erasures occur then we can recover completely the transmitted sequence.

The best scenario happens when the convolutional code is MDP.

Corollary

Let $\mathcal{C}$ be an MDP convolutional code of rate k/n and degree δ. If in any sliding window of length $(L + 1)n$ at most $(L + 1)(n - k)$ erasures occur in the transmitted sequence then we can completely recover the transmitted sequence in polynomial time in δ.

- Assume we have correctly received or recovered a sequence until instant t.
- We solve a full rank system, where $\star$ represent blocks where some erasures have occurred.

Idea of the process

$$\left[\begin{array}{ccccccccc} H_\nu & H_{\nu-1} & \dots & H_{\nu-L} & \dots & H_0 & & & \\ & H_\nu & \dots & H_{\nu-L+1} & \dots & H_1 & H_0 & & \\ & & \ddots & & & & & \ddots & \\ & & & H_\nu & \dots & H_L & H_{L-1} & \dots & H_0 \end{array}\right] \begin{bmatrix} \mathbf{v}_{t-\nu} \\ \vdots \\ \mathbf{v}_{t-1} \\ \star \\ \star \\ \vdots \\ \star \end{bmatrix} = 0$$

- The algorithm requires only linear algebra; it has polynomial time complexity
- Adaptative process: we do not need to take the largest L window we are allowed, we can choose smaller window sizes depending on the distribution of the erasures

Example

Assume we have an $[202, 101]$ MDS blok code and a MDP convolutional code of comparable window length. Both are capable to recover 100 of erasures in windows of 200 symbols, so 50%.
Suppose we receive correctly until instant t, followed by

$$\cdots vv|\overbrace{\star\star\cdots\star\star}^{60}\overbrace{vvv\cdots vv}^{80}\overbrace{\star\star\cdot\star\star}^{60}vv|vv\cdots$$

In a 200-symbol window the block code cannot decode: $120 > 100$ erasures $\rightarrow$ skip (and discard) the block.

A $(2, 1, 50)$ MDP convolutional code can decode in a 120-symbol window to recover 60 erasures, then slide to the next window to recover the rest!
We have flexibility in choosing the size and position of the sliding window.

Example

What could we do below, where there are spaces with clean enough memory, but still, the erasures are too concentrated in the beginning of the block to uniquely solve the systems???

$$\cdots vv|\overbrace{\star\star\cdots\star\star}^{22}\overbrace{vv\star\star vv\star\star\cdots vv\star\star}^{180(90)}|\overbrace{vvv\cdots vv}^{202}\overbrace{\star\star\cdot\star\star}^{60}vv|vv\cdots$$

MDP cannot correct it ...How can we do better?

- If we could move in the other direction along the sequence, the erasures are less concentrated and it would be easier to recover.
- We would like MDP behavior in the backwards process as well: to control the column distances of the reverse code.

Proposition:

Let $\mathcal{C}$ be an (n, k, δ)-code with minimal generator matrix $G(z)$. Let $\overline{G}(z)$ be the matrix obtained by replacing each entry $g_{ij}(z)$ of $G(z)$ by $\overline{g_{ij}}(z) := z^{\delta_j} g_{ij}(z^{-1})$, where δ_j is the j-th column degree of $G(z)$. Then, $\overline{G}(z)$ is a minimal generator matrix of an (n, k, δ)-code $\overline{\mathcal{C}}$, and

$$\mathbf{v}_0 + \mathbf{v}_1 z + \cdots + \mathbf{v}_{s-1} z^{s-1} + \mathbf{v}_s z^s \in \mathcal{C}$$

if and only if

$$\mathbf{v}_s + \mathbf{v}_{s-1} z^{+} \cdots + \mathbf{v}_1 z^{s-1} + \mathbf{v}_0 z^s \in \overline{\mathcal{C}}.$$

$\overline{\mathcal{C}}$ is the **reverse code** of $\mathcal{C}$.

Definition

Let $\mathcal{C}$ be an MDP (n, k, δ) convolutional code. We say that $\mathcal{C}$ is a reverse-MDP convolutional code if the reverse code $\overline{\mathcal{C}}$ is as well an MDP code.

Theorem

Let k, n and δ be positive integers. An (n, k, δ) reverse-MDP convolutional code exists over a sufficiently large field.

Example

Now reverse-MDP can recover the sequence.

$$\cdots vv \Big| \overbrace{\star\star\cdots\star\star}^{22}\overbrace{vv\star\star vv\star\star\cdots vv\star\star}^{180(90)} \Big| \overbrace{vvv\cdots vv}^{202}\overbrace{\star\star\cdot\star\star}^{60} vv \Big| vv\cdots$$

Example

Now reverse-MDP can recover the sequence.

$$\cdots vv|\overbrace{\star\star\cdots\star\star}^{22}\overbrace{vv\star\star vv\star\star\cdots vv\star\star}^{180(90)}|\overbrace{vvv\cdots vv}^{202}\overbrace{\star\star\cdot\star\star}^{60}vv|vv\cdots$$

Main problem

We know very little about reverse-MDP.

Thanks for your attention