Concatenation of Rank Metric Codes with Convolutional Codes (for Video Streaming)

Diego Napp, Raquel Pinto and Vladimir Sidorenko

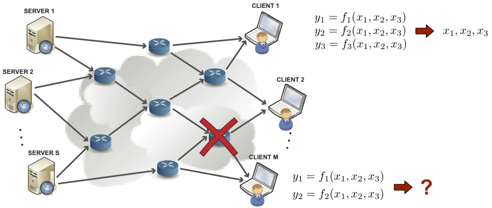
Department of Mathematics, University of Aveiro, Portugal
and
TUM (Technical University of Munich, Germany)
on leave from Institute for Information Transmission Problems, Russian Academy of Science

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

April 4, 2016

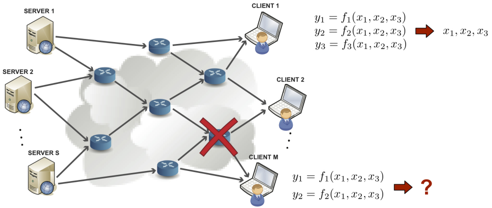
Overview

Linear Network Coding



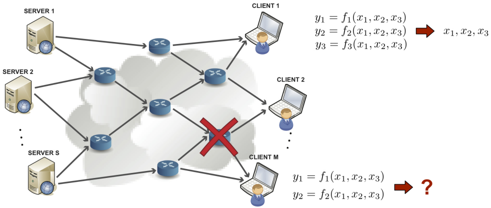
- During one *shot* the transmitter injects a number of packets into the network, each of which may be regarded as a row vector over a finite field $\mathbb{F}_{q^m}$.

Linear Network Coding



- During one *shot* the transmitter injects a number of packets into the network, each of which may be regarded as a row vector over a finite field $\mathbb{F}_{q^m}$.
- These packets propagate through the network. Each node creates a random -linear combination of the packets it has available and transmits this random combination.

Linear Network Coding



- During one *shot* the transmitter injects a number of packets into the network, each of which may be regarded as a row vector over a finite field $\mathbb{F}_{q^m}$.
- These packets propagate through the network. Each node creates a random -linear combination of the packets it has available and transmits this random combination.
- Finally, the receiver collects such randomly generated packets and tries to infer the set of packets injected into the network

Rank metric codes are used in Network Coding

- Rank metric codes are matrix codes $\mathcal{C} \subset \mathbb{F}_q^{m\times n}$, armed with the rank distance

$$d_{\mathrm{rank}}(X,Y) = rank(X - Y), \text{ where } X, Y \in \mathbb{F}_q^{n\times m}.$$

Rank metric codes are used in Network Coding

- Rank metric codes are matrix codes $\mathcal{C} \subset \mathbb{F}_q^{m \times n}$, armed with the rank distance

$$d_{\mathrm{rank}}(X, Y) = rank(X - Y), \text{ where } X, Y \in \mathbb{F}_q^{n \times m}.$$

- For linear (n, k) rank metric codes over $\mathbb{F}_{q^m}$ with $m \geq n$ the following analog of the Singleton bound holds,

$$d_{\mathrm{rank}}(\mathcal{C}) \leq n - k + 1.$$

Rank metric codes are used in Network Coding

- Rank metric codes are matrix codes $\mathcal{C} \subset \mathbb{F}_q^{m \times n}$, armed with the rank distance

$$d_{\mathrm{rank}}(X, Y) = rank(X - Y), \text{ where } X, Y \in \mathbb{F}_q^{n \times m}.$$

- For linear (n, k) rank metric codes over $\mathbb{F}_{q^m}$ with $m \geq n$ the following analog of the Singleton bound holds,

$$d_{\mathrm{rank}}(\mathcal{C}) \leq n - k + 1.$$

- The code that achieves this bound is called Maximum Rank Distance (MRD). Gabidulin codes are a well-known class of MRD codes.

THE IDEA: Multi-shot

- Coding can also be performed over multiple uses of the network, whose internal structure may change at each shot

THE IDEA: Multi-shot

- Coding can also be performed over multiple uses of the network, whose internal structure may change at each shot
- Creating dependencies among the transmitted codewords of different shots can improve the error-correction capabilities (Nobrega, R., Uchoa-Filho (2010), Wachter-Zeh, A., Stinner, M., Sidorenko (2015), Mahmood, R., Badr, A., Khisti(2015)).

THE IDEA: Multi-shot

- Coding can also be performed over multiple uses of the network, whose internal structure may change at each shot
- Creating dependencies among the transmitted codewords of different shots can improve the error-correction capabilities (Nobrega, R., Uchoa-Filho (2010), Wachter-Zeh, A., Stinner, M., Sidorenko (2015), Mahmood, R., Badr, A., Khisti(2015)).
- One standard way to impose correlation of codewords over time is by means of convolution codes.

THE IDEA: Multi-shot

- Coding can also be performed over multiple uses of the network, whose internal structure may change at each shot
- Creating dependencies among the transmitted codewords of different shots can improve the error-correction capabilities (Nobrega, R., Uchoa-Filho (2010), Wachter-Zeh, A., Stinner, M., Sidorenko (2015), Mahmood, R., Badr, A., Khisti(2015)).
- One standard way to impose correlation of codewords over time is by means of convolution codes.
- We propose to use a concatenated code derived by combining a rank metric code (as inner code) and a convolutional code (as outer code)

THE IDEA: Multi-shot

- Coding can also be performed over multiple uses of the network, whose internal structure may change at each shot
- Creating dependencies among the transmitted codewords of different shots can improve the error-correction capabilities (Nobrega, R., Uchoa-Filho (2010), Wachter-Zeh, A., Stinner, M., Sidorenko (2015), Mahmood, R., Badr, A., Khisti(2015)).
- One standard way to impose correlation of codewords over time is by means of convolution codes.
- We propose to use a concatenated code derived by combining a rank metric code (as inner code) and a convolutional code (as outer code)
- We show how this scheme add complex dependencies to data streams in a quite simple way

Block codes vs convolutional codes

$$\ldots u_2,\ u_1,\ u_0 \xrightarrow{G} \ldots v_2 = u_2 G,\ v_1 = u_1 G,\ v_0 = u_0 G$$

represented in a polynomial fashion

$$\cdots + u_2 D^2 + u_1 D + u_0 \xrightarrow{G} \cdots + \underbrace{u_2 G}_{v_2} D^2 + \underbrace{u_1 G}_{v_1} D + \underbrace{u_0 G}_{v_0}$$

substitute G by $G(D) = G_0 + G_1 D + \cdots + G_s D^s$?

$$\ldots u_2 D^2 + u_1 D + u_0 \xrightarrow{G(D)} \ldots \underbrace{(u_2 G_0 + u_1 G_1 + u_0 G_2)}_{v_2} D^2 + \underbrace{(u_1 G_0 + u_0 G_1)}_{v_1} D + \underbrace{u_0 G_0}_{v_0}$$

Block codes vs convolutional codes

$$\dots u_2,\ u_1,\ u_0 \xrightarrow{\quad G \quad} \dots v_2 = u_2 G,\ v_1 = u_1 G,\ v_0 = u_0 G$$

represented in a polynomial fashion

$$\cdots + u_2 D^2 + u_1 D + u_0 \xrightarrow{\quad G \quad} \cdots + \underbrace{u_2 G}_{v_2} D^2 + \underbrace{u_1 G}_{v_1} D + \underbrace{u_0 G}_{v_0}$$

substitute G by $G(D) = G_0 + G_1 D + \cdots + G_s D^s$?

$$\dots u_2 D^2 + u_1 D + u_0 \xrightarrow{G(D)} \dots \underbrace{(u_2 G_0 + u_1 G_1 + u_0 G_2)}_{v_2} D^2 + \underbrace{(u_1 G_0 + u_0 G_1)}_{v_1} D + \underbrace{u_0 G_0}_{v_0}$$

Block codes: $\mathcal{C} = \{uG\} = \mathrm{Im}_{\mathbb{F}} G \sim \{u(D)G\} = \mathrm{Im}_{\mathbb{F}(D)} G$
Convolutional codes: $\mathcal{C} = \{u(D)G(D)\} = \mathrm{Im}_{\mathbb{F}((D))} G(D)$

A convolutional code $\mathcal{C}$ is a $\mathbb{F}((D))$-subspace of $\mathbb{F}^n((D))$.

A convolutional code $\mathcal{C}$ is a $\mathbb{F}((D))$-subspace of $\mathbb{F}^n((D))$.

A matrix $G(D)$ whose rows form a basis for $\mathcal{C}$ is called an encoder.
If $\mathcal{C}$ has rank k then we say that $\mathcal{C}$ has rate k/n.

$$\mathcal{C} \;=\; \mathrm{Im}_{\mathbb{F}((D))} G(D) = \left\{ u(D)G(D) : \; u(D) \in \mathbb{F}^k((D)) \right\}$$

A convolutional code $\mathcal{C}$ is a $\mathbb{F}((D))$-subspace of $\mathbb{F}^n((D))$.

A matrix $G(D)$ whose rows form a basis for $\mathcal{C}$ is called an encoder. If $\mathcal{C}$ has rank k then we say that $\mathcal{C}$ has rate k/n.

$$\mathcal{C} \quad = \quad \mathrm{Im}_{\mathbb{F}((D))} G(D) = \left\{ u(D) G(D) : \; u(D) \in \mathbb{F}^k((D)) \right\}$$

Remark

One can also consider the ring of polynomials $\mathbb{F}[D]$ instead of Laurent series $\mathbb{F}((D))$ and define $\mathcal{C}$ as a $\mathbb{F}[D]$-module of $\mathbb{F}^n[D]$.

Historical Remarks

- Convolutional codes were introduced by Elias (1955)
- Became widespread in practice with the Viterbi decoding. Widely implemented codes in (wireless) communications. The field is typically $\mathbb{F}_2$ and the rate and degree are small so that the Viterbi decoding algorithm is efficient.
- Renewed interest for convolutional codes over large alphabets trying to fully exploit the potential of convolutional codes.
- Decoding over the erasure channel is easy (Rosenthal et al. 2012).

MDS convolutional codes over $\mathbb{F}$

The Hamming weight of a polynomial vector

$$v(D) = \sum_{i \in \mathbb{N}} v_i D^i = v_0 + v_1 D + v_2 D^2 + \cdots + v_\nu D^\nu \in \mathbb{F}[D]^n,$$

defined as $\mathrm{wt}(v(D)) = \sum_{i=0}^{\nu} \mathrm{wt}(v_i)$.

MDS convolutional codes over $\mathbb{F}$

The Hamming weight of a polynomial vector

$$v(D) = \sum_{i \in \mathbb{N}} v_i D^i = v_0 + v_1 D + v_2 D^2 + \cdots + v_\nu D^\nu \in \mathbb{F}[D]^n,$$

defined as $\mathrm{wt}(v(D)) = \sum_{i=0}^{\nu} \mathrm{wt}(v_i)$.

The free distance of a convolutional code $\mathcal{C}$ is given by,

$$d_{\mathrm{free}}(\mathcal{C}) \quad = \quad \min \{\mathrm{wt}(v(D)) \ | \ v(D) \in \mathcal{C} \quad \text{and} \quad v(D) \neq 0\}$$

MDS convolutional codes over $\mathbb{F}$

The Hamming weight of a polynomial vector

$$v(D) = \sum_{i \in \mathbb{N}} v_i D^i = v_0 + v_1 D + v_2 D^2 + \cdots + v_\nu D^\nu \in \mathbb{F}[D]^n,$$

defined as $\mathrm{wt}(v(D)) = \sum_{i=0}^{\nu} \mathrm{wt}(v_i)$.

The free distance of a convolutional code $\mathcal{C}$ is given by,

$$d_{\mathrm{free}}(\mathcal{C}) \quad = \quad \min \left\{ \mathrm{wt}(v(D)) \;\middle|\; v(D) \in \mathcal{C} \quad \text{and} \quad v(D) \neq 0 \right\}$$

- For block codes ($\delta = 0$) we know that maximum value is given by the Singleton bound: $n - k + 1$

MDS convolutional codes over $\mathbb{F}$

The Hamming weight of a polynomial vector

$$v(D) = \sum_{i \in \mathbb{N}} v_i D^i = v_0 + v_1 D + v_2 D^2 + \cdots + v_\nu D^\nu \in \mathbb{F}[D]^n,$$

defined as $\mathrm{wt}(v(D)) = \sum_{i=0}^{\nu} \mathrm{wt}(v_i)$.

The free distance of a convolutional code $\mathcal{C}$ is given by,

$$d_{\mathrm{free}}(\mathcal{C}) = \min \{ \mathrm{wt}(v(D)) \mid v(D) \in \mathcal{C} \quad \text{and} \quad v(D) \neq 0 \}$$

- For block codes ($\delta = 0$) we know that maximum value is given by the Singleton bound: $n - k + 1$
- This bound can be achieve if $|\mathbb{F}| > n$

Theorem

Rosenthal and Smarandache (1999) showed that the free distance of convolutional code of rate k/n and degree δ must be upper bounded by

$$d_{\mathrm{free}}(\mathcal{C}) \leq (n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1. \tag{1}$$

A code achieving (1) is called Maximum Distance Separable (MDS).

Definition

Another important distance measure for a convolutional code is the jth column distance $d_j^c(\mathcal{C})$, (introduced by Costello), given by

$$d_H^j(\mathcal{C}) \quad = \quad \min\left\{\mathrm{wt}(v_{[0,j]}(D)) \;\mid\; v(D) \in \mathcal{C} \;\text{ and }\; v_0 \neq 0\right\}$$

where $v_{[0,j]}(D) = v_0 + v_1 D + \ldots + v_j D^j$ represents the j-th truncation of the codeword $v(D) \in \mathcal{C}$.

Definition

Another important distance measure for a convolutional code is the jth column distance $d_j^c(\mathcal{C})$, (introduced by Costello), given by

$$d_H^j(\mathcal{C}) \quad = \quad \min \left\{ \mathrm{wt}(v_{[0,j]}(D)) \ \mid \ v(D) \in \mathcal{C} \ \text{ and } \ v_0 \neq 0 \right\}$$

where $v_{[0,j]}(D) = v_0 + v_1 D + \ldots + v_j D^j$ represents the j-th truncation of the codeword $v(D) \in \mathcal{C}$.

The column distances satisfy

$$d_H^0 \leq d_H^1 \leq \cdots \leq \lim_{j \to \infty} d_H^j(\mathcal{C}) = d_{\mathrm{free}}(\mathcal{C}) \leq (n-k) \left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1 \right) + \delta + 1.$$

The j-th column distance is upper bounded as following

$$d_H^j(\mathcal{C}) \leq (n-k)(j+1)+1, \tag{2}$$

The j-th column distance is upper bounded as following

$$d_H^j(\mathcal{C}) \leq (n-k)(j+1)+1, \tag{2}$$

If $\mathcal{C}$ has the best possible column distance profile then it is called Maximum Distance Profile (MDP).

The j-th column distance is upper bounded as following

$$d_H^j(\mathcal{C}) \leq (n-k)(j+1)+1, \tag{2}$$

If $\mathcal{C}$ has the best possible column distance profile then it is called Maximum Distance Profile (MDP).

The construction of MDP boils down to the construction of Superregular matrices (difficult over *small* fields).

Performance over the erasure channel

Theorem (Rosenthal et al. 2012)

Let $\mathcal{C}$ be convolutional code and $d_H^{j_0}$ its $j = j_0$ -th column distance.

If in any sliding window of length $(j_0 + 1)n$ at most $d_H^{j_0} - 1$ erasures occur then we can recover completely the transmitted sequence.

Performance over the erasure channel

Theorem (Rosenthal et al. 2012)

Let $\mathcal{C}$ be convolutional code and $d_H^{j_0}$ its $j = j_0$ -th column distance.

If in any sliding window of length $(j_0 + 1)n$ at most $d_H^{j_0} - 1$ erasures occur then we can recover completely the transmitted sequence.

Remark

The best scenario happens when the convolutional code is MDP.

Example

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

A $[202, 101]$ MDS block code can correct 101 erasures in a window of 202 symbols (recovering rate $\frac{101}{202}$): $\Rightarrow$ cannot correct this window.

A $(2, 1, 50)$ MDP convolutional code has also 50% error capability. $(L+1)n = 101 \times 2 = 202$. Take a window of 120 symbols, correct and continue until you correct the whole window.

We have flexibility in choosing the size and position of the sliding window.

The proposed concatenation scheme

- $\mathcal{C}_O$ an (K, k, δ) convolutional code over the field $\mathbb{F}_{q^{mn}}$

The proposed concatenation scheme

- $\mathcal{C}_O$ an (K, k, δ) convolutional code over the field $\mathbb{F}_{q^{mn}}$
- (Hamming) distance $d_{\mathrm{free}}(\mathcal{C}_O)$, column distance $d^j_H(\mathcal{C}_O)$

The proposed concatenation scheme

- $\mathcal{C}_O$ an (K, k, δ) convolutional code over the field $\mathbb{F}_{q^{mn}}$
- (Hamming) distance $d_{\mathrm{free}}(\mathcal{C}_O)$, column distance $d_H^j(\mathcal{C}_O)$
- minimal basic encoder $G_O(D)$

The proposed concatenation scheme

- $\mathcal{C}_O$ an (K, k, δ) convolutional code over the field $\mathbb{F}_{q^{mn}}$
- (Hamming) distance $d_{\mathrm{free}}(\mathcal{C}_O)$, column distance $d^j_H(\mathcal{C}_O)$
- minimal basic encoder $G_O(D)$
- $\mathcal{C}_I$ a rank metric code with (rank) distance $d_{\mathrm{rank}}(\mathcal{C}_I)$ and encoder G_I

The proposed concatenation scheme

- $\mathcal{C}_O$ an (K, k, δ) convolutional code over the field $\mathbb{F}_{q^{mn}}$
- (Hamming) distance $d_{\mathrm{free}}(\mathcal{C}_O)$, column distance $d_H^j(\mathcal{C}_O)$
- minimal basic encoder $G_O(D)$
- $\mathcal{C}_I$ a rank metric code with (rank) distance $d_{\mathrm{rank}}(\mathcal{C}_I)$ and encoder G_I

$u(D) = u_0 + u_1 D + u_2 D^2 + \cdots \in \mathbb{F}_{q^{mn}}[D]^k$ the information vector.

$$\cdots + u_2 D^2 + u_1 D + u_0 \xrightarrow{G_O(D)} \cdots + v_2 D^2 + v_1 D + v_0$$

The proposed concatenation scheme

- $\mathcal{C}_O$ an (K, k, δ) convolutional code over the field $\mathbb{F}_{q^{mn}}$
- (Hamming) distance $d_{\text{free}}(\mathcal{C}_O)$, column distance $d_H^j(\mathcal{C}_O)$
- minimal basic encoder $G_O(D)$
- $\mathcal{C}_I$ a rank metric code with (rank) distance $d_{\text{rank}}(\mathcal{C}_I)$ and encoder G_I

$u(D) = u_0 + u_1 D + u_2 D^2 + \cdots \in \mathbb{F}_{q^{mn}}[D]^k$ the information vector.

$$\cdots + u_2 D^2 + u_1 D + u_0 \xrightarrow{G_O(D)} \cdots + v_2 D^2 + v_1 D + v_0$$

We divide

$$v_i = (v_i^0, v_i^1, \ldots, v_i^{K-1})$$

We identify $v_i^j \in \mathbb{F}_{q^{mn}}$ with a vector $V_i^j \in \mathbb{F}_{q^m}^n$ (for a given basis of $\mathbb{F}_{q^m}^n$) and write

$$V_i = (V_i^0, V_i^1, \ldots, V_i^{K-1})$$

and therefore

$$V(D) = V_0 + V_1 D + V_2 D^2 + \cdots \in \mathbb{F}_{q^m}^n[D]^K.$$

We identify $v_i^j \in \mathbb{F}_{q^{mn}}$ with a vector $V_i^j \in \mathbb{F}_{q^m}^n$ (for a given basis of $\mathbb{F}_{q^m}^n$) and write

$$V_i = (V_i^0, V_i^1, \ldots, V_i^{K-1})$$

and therefore

$$V(D) = V_0 + V_1 D + V_2 D^2 + \cdots \in \mathbb{F}_{q^m}^n[D]^K.$$

Finally, the **codewords** $X(D)$ of $\mathcal{C}$ are obtained through the matrix $G_I \in \mathbb{F}_{q^m}^{n \times N}$ in the following way:

$$X_i^j = V_i^j G_I,$$

$$X_i = (X_i^0, X_i^1, \ldots, X_i^{K-1})$$

and

$$X(D) = X_0 + X_1 D + X_2 D^2 + \ldots \in \mathcal{C} \subset \mathbb{F}_q^{m \times n}[D]^N.$$

Distance notions

Definition

The *sum rank distance* of $\mathcal{C}$ is defined as

$$d_{SR}(\mathcal{C}) = \min_{0 \neq X(D) \in \mathcal{C}} \text{rank}\,(X(D)) := \min_{0 \neq X(D) \in \mathcal{C}} \sum_{i \geq 0} \text{rank}\,(X_i)$$

where

$$\text{rank}\,(X_i) := \sum_{j=0}^{K-1} \text{rank}\,(X_i^j).$$

And the *column sum rank distance* of $\mathcal{C}$ is defined as

$$d_{SR}^j(\mathcal{C}) = \min_{X(D) \in \mathcal{C} \text{ and } X_0^0 \neq 0} \sum_{i=0}^{j} \text{rank}\,(X_i),$$

We assume throughout the paper that $m > N$.

Theorem

The Sum Rank distance of the concatenated code $\mathcal{C}$ is

$$d_{SR}(\mathcal{C}) = d_{\mathrm{free}}(\mathcal{C}_O) \times d_{\mathrm{rank}}(\mathcal{C}_I).$$

Theorem

The Column Sum Rank distance of $\mathcal{C}$ is

$$d^j_{SR}(\mathcal{C}) = d^j_H(\mathcal{C}_O) \times d_{\mathrm{rank}}(\mathcal{C}_I).$$

where $d^j_H(\mathcal{C}_O)$ is the column distance of $\mathcal{C}$.

Performance

- The rank metric code (inner code) takes care of the errors and erasures (deletion and injection of packets) during the transmission.
- The convolutional code (outer code) deals only of the *erasures*.

Performance

- The rank metric code (inner code) takes care of the errors and erasures (deletion and injection of packets) during the transmission.
- The convolutional code (outer code) deals only of the *erasures*.

Theorem

If in any sliding window of $\mathcal{C}$ of length (j_0+1) at most $d_{SR}^{j_0}-1$ packet losses occur, then we can completely recover the information sequence.

Previous Theorem has some drawbacks:

- Only necessary conditions. There are erasure patterns that do not satisfy the condition of the Theorem but still can be recovered.
- One has to wait for the whole sequence to arrive. One would rather decode *sequentially*.

Previous Theorem has some drawbacks:

- Only necessary conditions. There are erasure patterns that do not satisfy the condition of the Theorem but still can be recovered.
- One has to wait for the whole sequence to arrive. One would rather decode *sequentially*.

Theorem

Let $d_H^0(\mathcal{C}_o), d_H^1(\mathcal{C}_o), \ldots, d_H^L(\mathcal{C}_o)$ be the distance profile of $\mathcal{C}_o$. Let L_i be the number of packet losses at time instant i. Assume that we have been able to correctly decode up to an instant $t-1$. Then, we can completely decode up to an instant T, $t \leq T$ iff

$$\sum_{i=0}^{s} L_{T-i+t} \leq d_H^s(\mathcal{C}_o) d_{Rank}(\mathcal{C}_I) - 1 \text{ for } s = 0, 1, \ldots, T.$$

Remains to be investigated

- The last Theorem suggests an algorithm: Exploit the structure of the equations to solve.

Remains to be investigated

- The last Theorem suggests an algorithm: Exploit the structure of the equations to solve.
- Simulations: Rate of success in recovering for a given probability of losing a packet.

Remains to be investigated

- The last Theorem suggests an algorithm: Exploit the structure of the equations to solve.
- Simulations: Rate of success in recovering for a given probability of losing a packet.
- Little is known about how to construct good convolutional codes:
  - MDP or equivalently: Construction of superregular matrices over small fields.
  - Constructions tailor-made to deal with *burst of erasures* (in both Hamming and Rank metric)

Thanks for the (financial) support!