3rd BEATS/WIP/CUBAN workshop

Modern Channel Coding Techniques in Source Coding Problems:

an Overview

Giuseppe Caire, Eurecom

Joint work with: S. Verdu, S. Shamai, S. Sesia, K. Narayanan

May 23 – 25, 2005

Outline

- Almost lossless data compression using block codes.
- Schemes and experimental results.
- Applications: Slepian-Wolf, Bidirectional data exchange, Coding with limited feedback.
- Application to “robust” (lossy) source-channel coding.
- Conclusions.

Fixed-Length (Block) Almost Lossless Compression (Shannon, 1948)

1	2	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	n

$\downarrow$

Source Encoder $\longrightarrow$ Error

$\downarrow$

1	2	$\cdots$	$\cdots$	$\cdots$	m

$\downarrow$

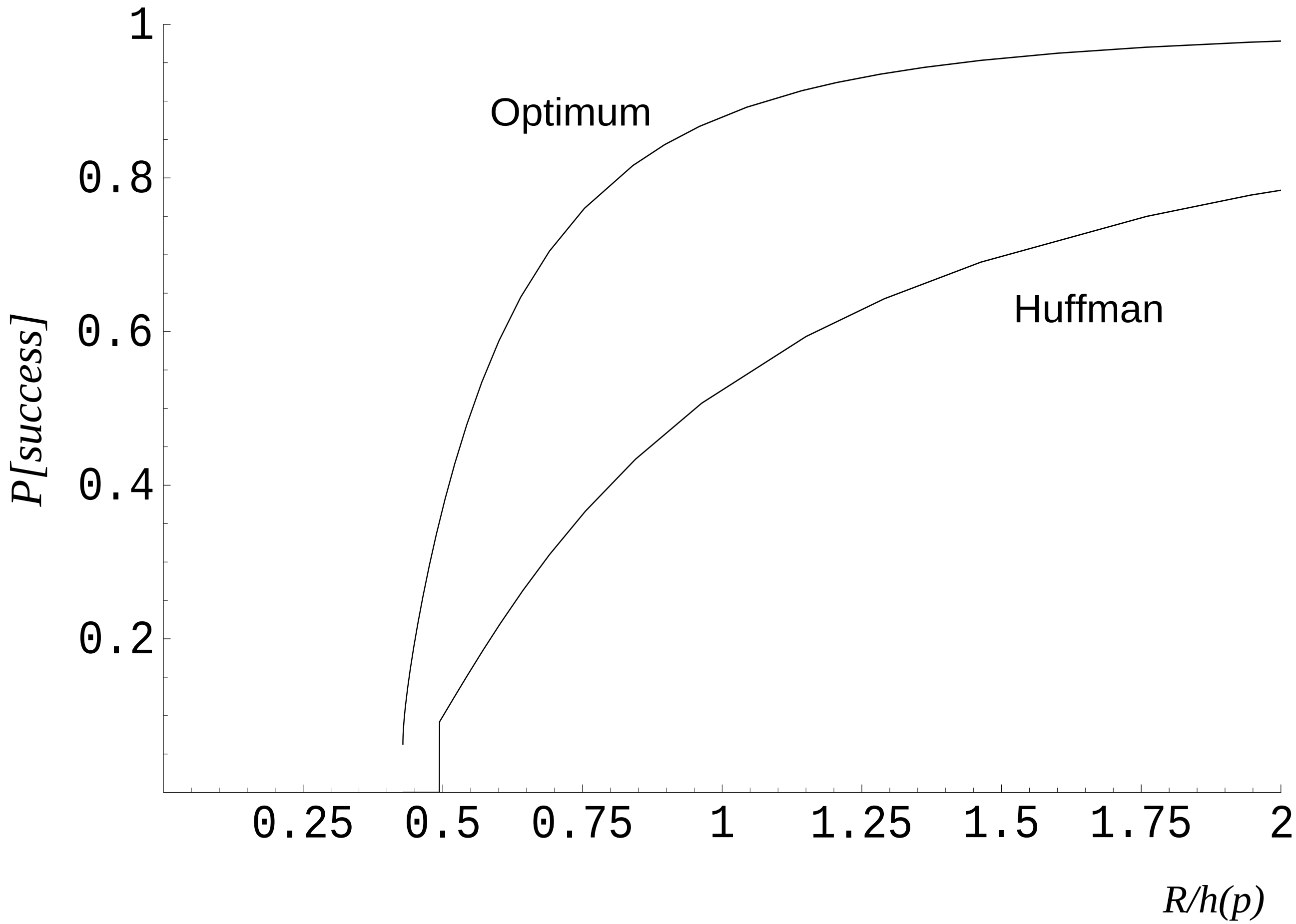
Source Decoder

$\downarrow$

1	2	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	n

Theorem (Shannon-Macmillan) If $\frac{m}{n} > H + \delta$, Probability of block error $\rightarrow 0$.

Example: 7-to-3 code for bias-p coin



- Optimum block source code: encode the 2^3 most probable sourcewords.
- “Block Huffman” code for source with 2^7 values.

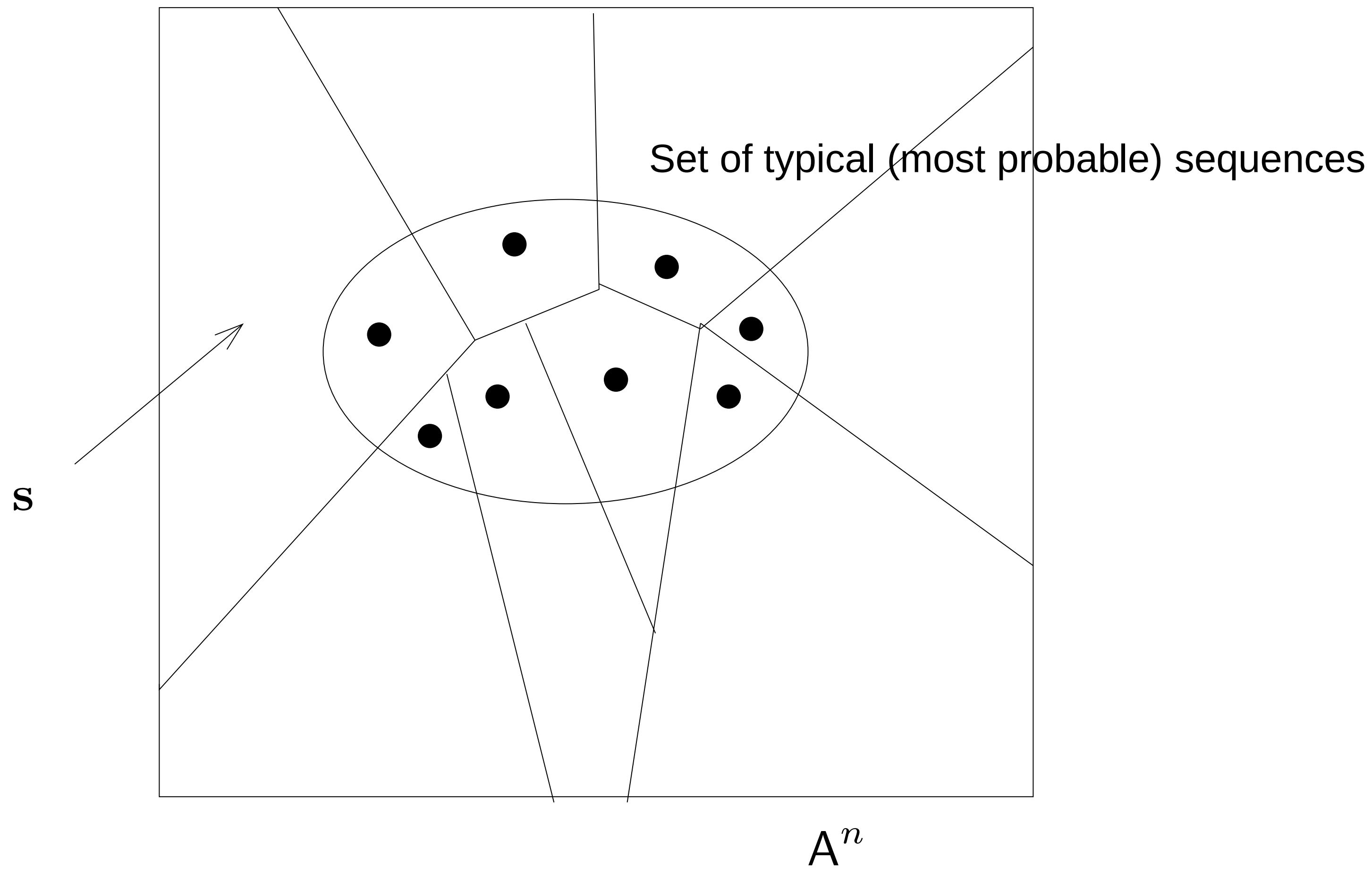
Linear Block Source Codes

Encoder: $\mathbf{H}\mathbf{s} = \mathbf{z}$

$$\begin{bmatrix}
k & p & & q & t & e & z & i & p & j & p & h & u & t \\
s & g & l & k & c & & l & u & & e & m & b & n & e \\
c & & w & f & o & y & x & b & a & x & v & l & s & j \\
g & o & j & f & s & i & e & & j & w & b & a & o & y \\
c & r & t & o & x & d & h & k & e & x & o & h & h & r \\
f & w & q & c & y & i & & g & g & s & u & z & y & t \\
l & t & o & e & q & v & j & m & n & r & k & x & i & k \\
j & & & e & g & l & a & k & u & m & & e & u & x \\
i & f & e & m & x & h & y & o & e & t & b & & f & s \\
s & u & o & k & r & c & & t & j & w & k & q & t & c
\end{bmatrix}
\begin{bmatrix} c \\ l \\ a \\ u \\ d \\ e \\ \ \\ s \\ h \\ a \\ n \\ n \\ o \\ n \end{bmatrix}
=
\begin{bmatrix} p \\ g \\ o \\ l \\ p \\ m \\ u \\ s \\ \ \\ h \end{bmatrix}$$

Decoder: Given the linearly encoded vector $\mathbf{z} = \mathbf{H}\mathbf{s}$, ML decoder chooses
$g_{\mathbf{H}}(\mathbf{z})$: the most likely source vector $\mathbf{u}$ that satisfies $\mathbf{H}\mathbf{u} = \mathbf{z}$.

Coset partition and source typical set



Linear Block Source Codes

- Theorem (Elias, 1955; Csiszár-Körner, 1981)
  - If the source is stationary memoryless with entropy H,
  - $\frac{m}{n} > H + \delta$,
  - the encoding matrix entries are independent, equiprobable on the alphabet,
  - $\Longrightarrow$ average block error probability $\rightarrow 0$.

- Theorem for general sources [Verdu-Shamai-Caire, 2003]:
  - If the source has sup-entropy rate $\bar{H}$,
  - $\frac{m}{n} > \bar{H} + \delta$,
  - the encoding matrix entries are independent, equiprobable on the alphabet,
  - $\Longrightarrow$ average block error probability $\rightarrow 0$.

Linear Block Channel Codes

- Additive-noise discrete channel: $\mathbf{y} = \mathbf{x} + \mathbf{u}$
- Linear channel code with parity-check matrix $\mathbf{H}$ $(\mathbf{H}\mathbf{x} = \mathbf{0})$.
- The maximum likelihood decoder selects the codeword $\mathbf{y} - g_{\mathbf{H}}(\mathbf{H}\mathbf{y})$, $\Longleftrightarrow$ it selects the most likely noise realization among those that lead to the same syndrome $\mathbf{H}\mathbf{y}$.
- $\Longrightarrow$ Random Linear codes (independent equiprobable parity-check matrix entries) achieve capacity.

Linear Block Channel Codes
as
Linear Block Source Codes

- Source Encoder = Syndrome Computer:

$$\mathbf{z} = \mathbf{H}\mathbf{s}$$

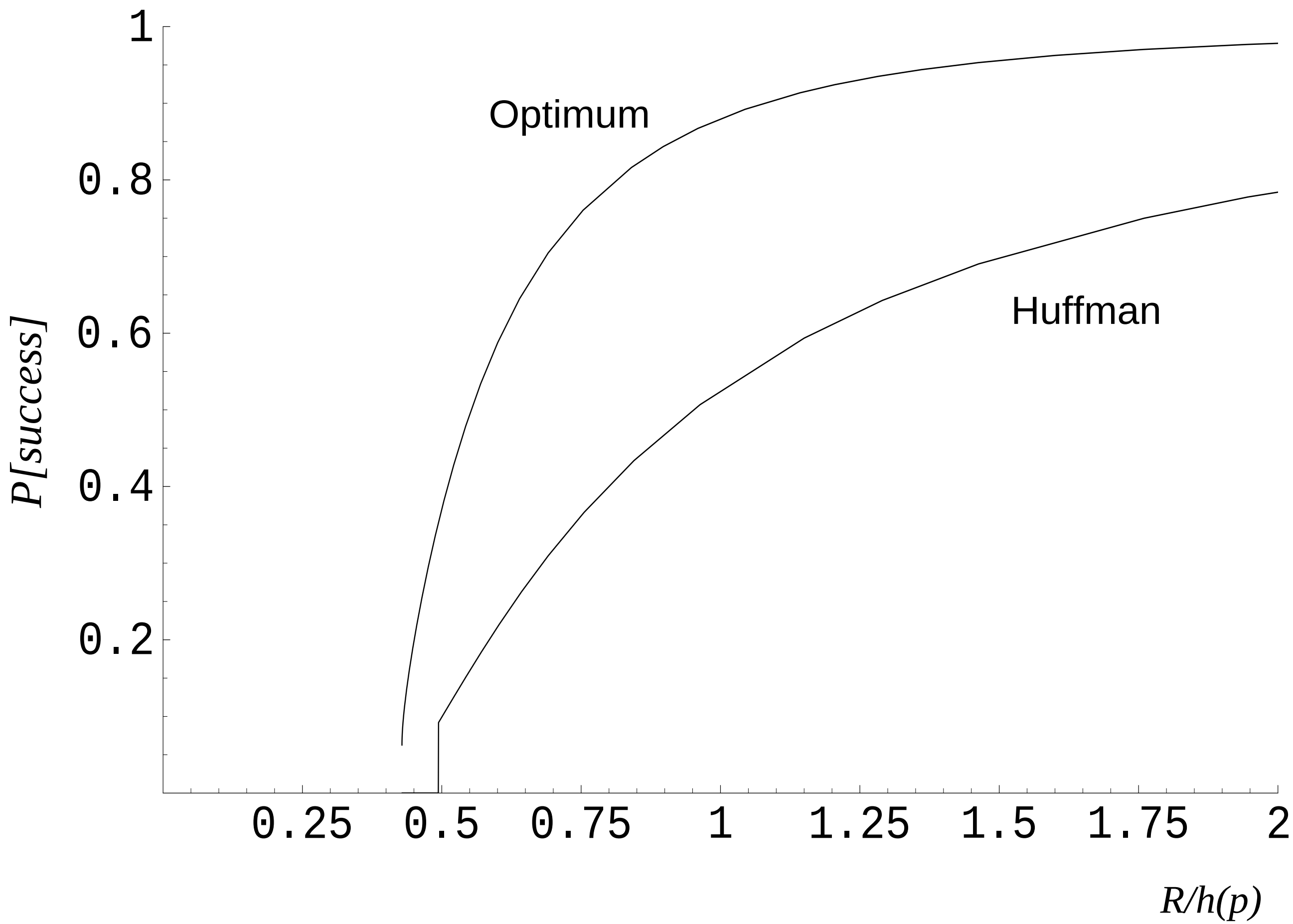
  where $\mathbf{H}$ is the parity-check matrix of a channel code.

- Maximum likelihood Source decoder: $g_{\mathbf{H}}(\mathbf{z})$ (syndrome decoder)

- Block error probability = Block error probability of the channel code when applied to $\mathbf{y} = \mathbf{x} + \mathbf{u}$, and $\mathbf{u}$ has the same distribution as $\mathbf{s}$.

- Not a new idea! (Ancheta, 1976) tried syndrome coding for bias-p coins with $n = 15, 31$ Hamming and BCH codes and could beat run-length coding and Cover's enumerative source encoding only for $p < 0.07$.

Example: 7-to-3 code for bias-p coin



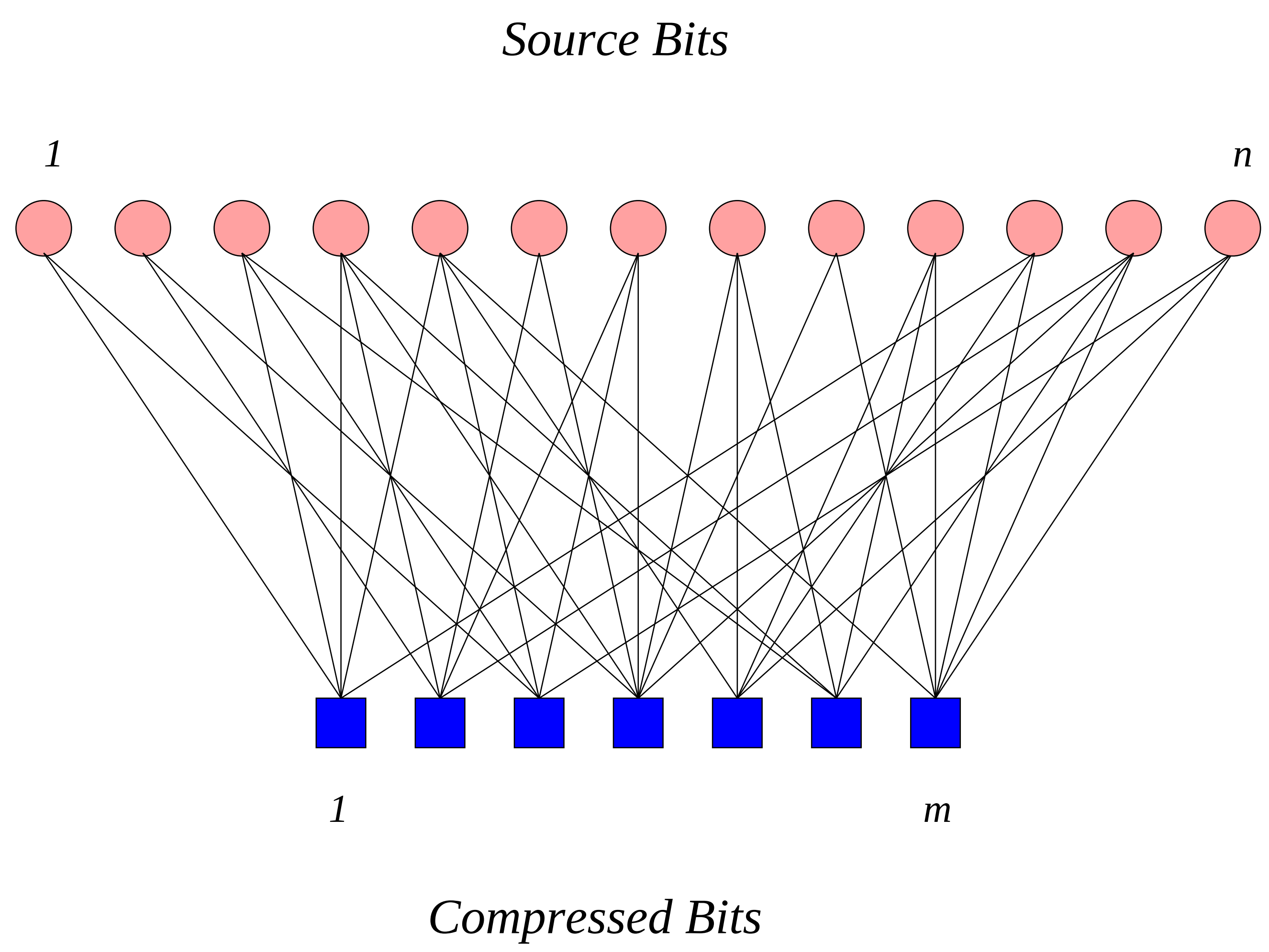
- Optimum code: unique codeword for the 2^3 most probable sourcewords.

  $\longrightarrow$ Hamming (7,4) parity-check matrix $\longleftarrow$.

Linear Block Source Codes vs. Lempel-Ziv

	$Linear Block$	$Lempel - Ziv$
Length	Fixed	Variable
Error	Almost-Lossless	Lossless
Encoding Complexity	$O(n^2)$	$O(n)$
Decoding Complexity	$Exp(n)$	$O(n)$
Source Knowledge	Required at Decoder	Universal

Low-Density Parity-Check (LDPC) Encoding Matrices



Belief-Propagation Decoding (Binary Alphabet)

- A priori source log-ratios

$$\mathcal{L}_k = \log \frac{1 - p_k}{p_k}, \quad k \in \{1, \ldots n\}$$

- Bitnode $k \rightarrow$ Checknodes $\in \mathcal{A}_k$

$$\nu_{k \rightarrow j}^{(t)} = \mathcal{L}_k + \sum_{j' \in \mathcal{A}_k - \{j\}} \mu_{j' \rightarrow k}^{(t-1)}$$

- Checknode $j \rightarrow$ Bitnodes $\in \mathcal{B}_j$

$$\mathbf{z} = \mathbf{H}\mathbf{s}$$

$$\mu_{j \rightarrow k}^{(t)} = (-1)^{z_j} 2 \tanh^{-1} \left(\prod_{k' \in \mathcal{B}_j - \{k\}} \tanh(\nu_{k' \rightarrow j}^{(t)}/2) \right)$$

- Bitnodes tentative decisions

$$\widehat{x}_k^{(t)} = \text{sign}\left\{ \mathcal{L}_k + \sum_{j \in \mathcal{A}_k} \mu_{j \to k}^{(t)} \right\}$$

- Stop if

$$\sum_{k \in \mathcal{B}_j} \widehat{x}_k^{(t)} = z_k, \quad j = 1, \ldots, m$$

Challenges

- Competing with state-of-the-art data compression (arithmetic coding) is very hard, even in terms of block-error probability [fixed-to-fixed] vs. buffer overflow probability [fixed-to-variable].

- Arithmetic coding is easily adapted to source with memory (tree sources, Markov sources).

- Arithmetic coding encodes and decodes the source sequence sequentially and can estimate the source statistics parameters (transition probabilities) sequentially (e.g., using the KT estimator).

- Complexity-wise, arithmetic coding is still much simpler and faster than BP decoding.

Partial answers

- Decoding at the encoder: effectively reducing the block-error probability.
- Back to (limited) variable-length: closed-loop iterative doping (CLID) and incremental redundancy (Fountain codes).
- Dealing with memory: incorporating the memory structure of the source in the BP decoder or using a block-sorting transform (BWT).
- Universality: BWT-based minimum description length model and parameters estimation.
- Results in Caire-Shamai-Verdu, 2003-2004, Caire-Shokrollahi-Shamai-Verdu, 2004.

Decoding at the Encoder

In contrast to channel decoding, the compressor can run an exact copy of the decompression algorithm.

- Library of Parity-Check Matrices:
  choose “best” for the particular source realization.

- Closed-loop iterative doping.

Closed Loop Iterative Doping

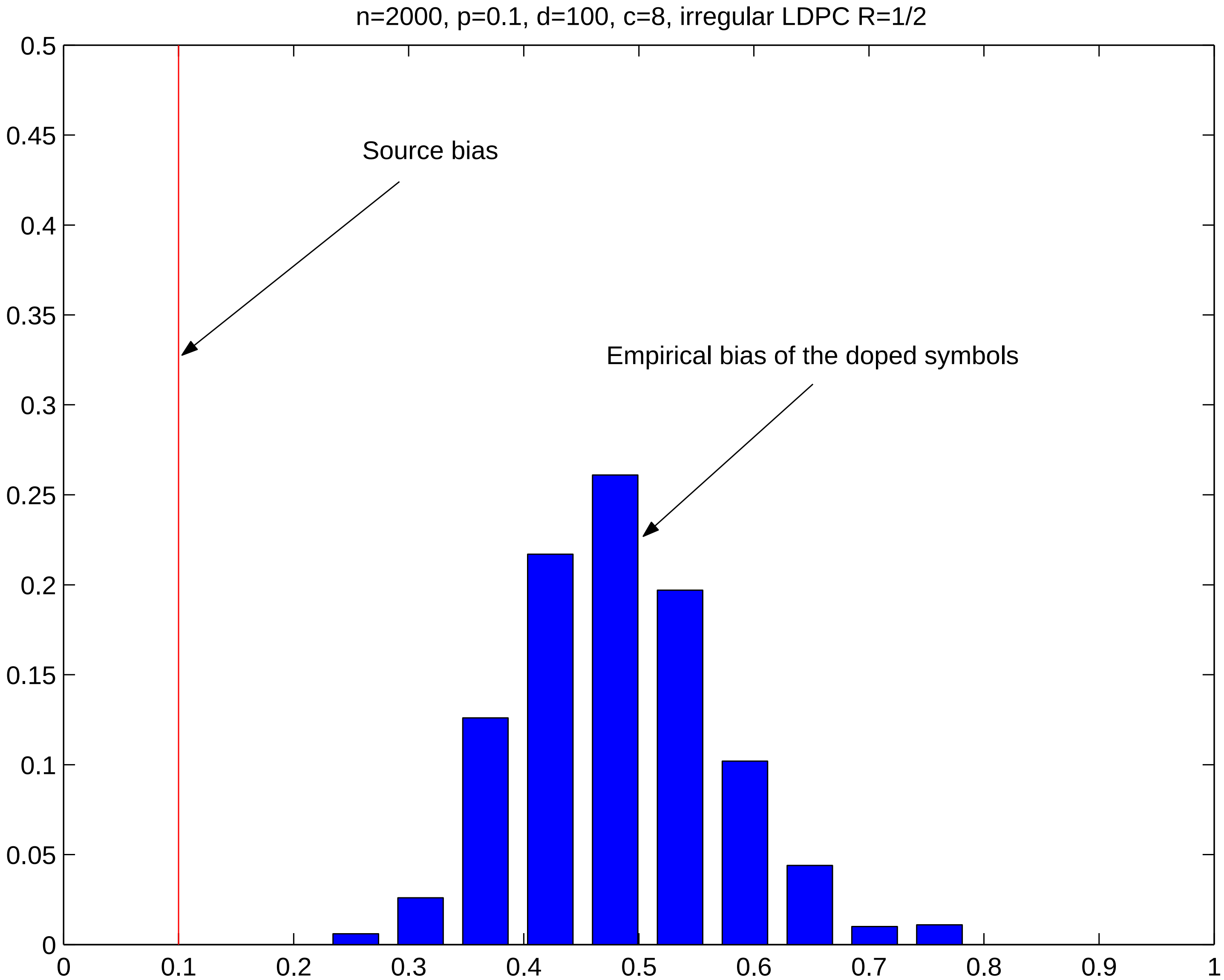
Every D decoder iterations, the source symbol for which the BP algorithm has accumulated the least reliability is communicated to the decoder:

$$\widehat{k} = \arg \min_{k=1,\dots,n} \{|\nu_k^{(t)}|\}, \qquad \nu_k^{(t)} = \mathcal{L}_k + \sum_{j \in \mathcal{A}_k} \mu_{j \to k}^{(t-1)}$$

$$\mathcal{L}_{\widehat{k}} = \begin{cases} +\infty & \text{if } \; x_{\widehat{k}} = 0 \\ -\infty & \text{if } \; x_{\widehat{k}} = 1 \end{cases}$$

- Almost lossless fixed-length version: Fixed number of iterations.
- Lossless variable-length version (zero error probability):
  Dope and iterate until decoded word is correct.
  Choose among the library of $\mathbf{H}$'s that leading to the shortest compressed length.

Empirical Distribution of Doped Bits



Biased Coin Experiment: 500-to-303 code

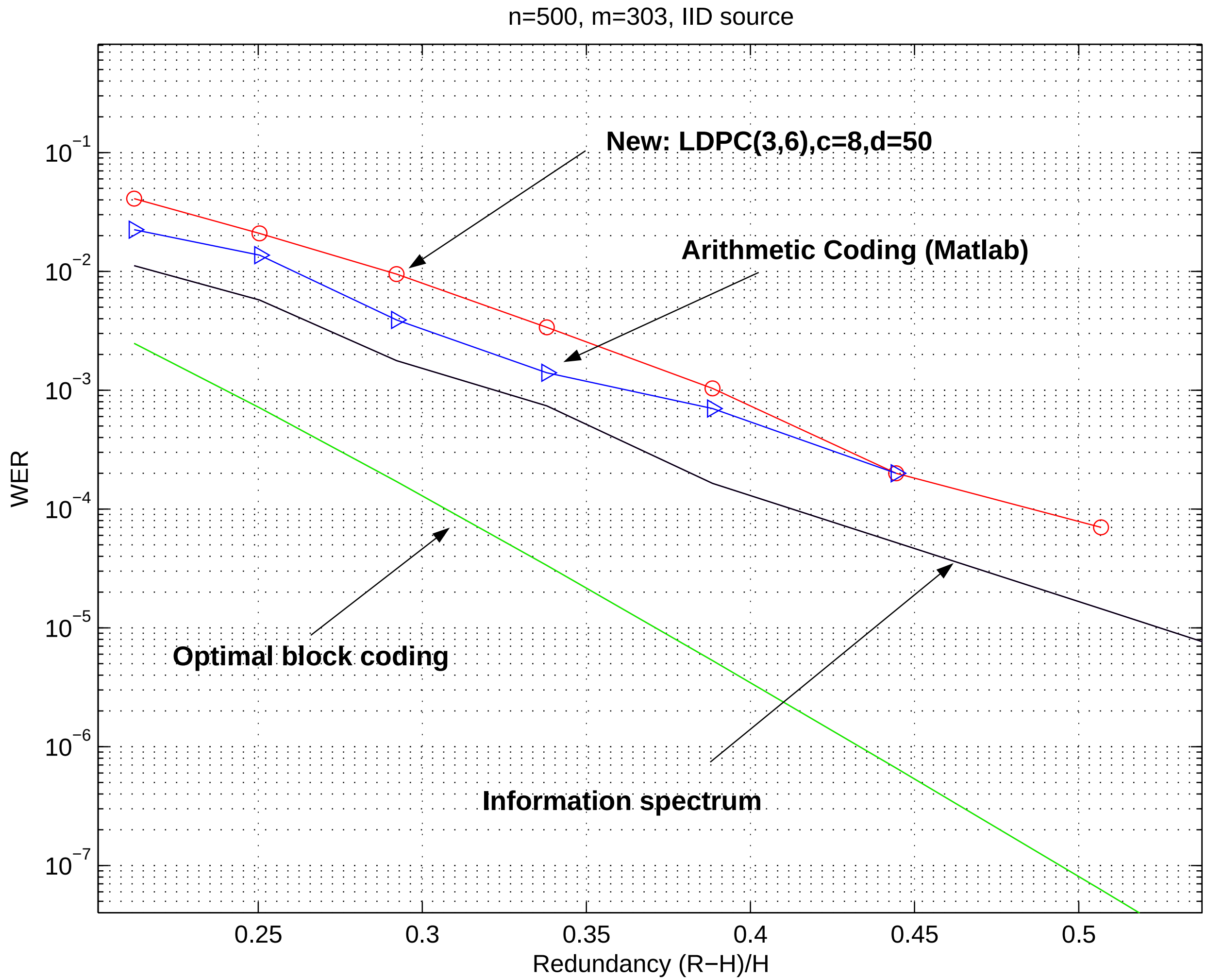


Figure 1: Block error rate of 500-to-303 source codes for a biased coin.

Biased Coin Experiment: 500-to-303 code: Cont.

Optimal block coding: (non-constructible) optimal code that assigns a distinct codeword to the most probable 2^{303} source realizations.

Information Spectrum: “Ideal” variable-length coding scheme where the length of the codeword assigned to $x_1, \ldots x_n$ is equal to $-\log_2 P_{X_1,\ldots X_n}(x_1, \ldots, x_n)$.

Arithmetic Coding: Nonuniversal, source exactly known.

Code Library: 8 (3,6) regular LDPCs (3 bit overhead).

Closed-Loop Iterative Doping: 50 bits.

Compression/Decompression Scheme: Universal Version

G. Caire, S. Shamai, S. Verdú,
”Universal Data Compression with LDPC Codes”
3rd International Symposium on Turbo codes and related topics
Brest, France, 1 - 5 September 2003

- Pre-processing of the source sequence via the BWT.
- Adaptive Segmentation via MDL principle.
- Communicate to the decoder a quantized version of:
  - segmentation
  - distributions

Random ensemble of Markov sources

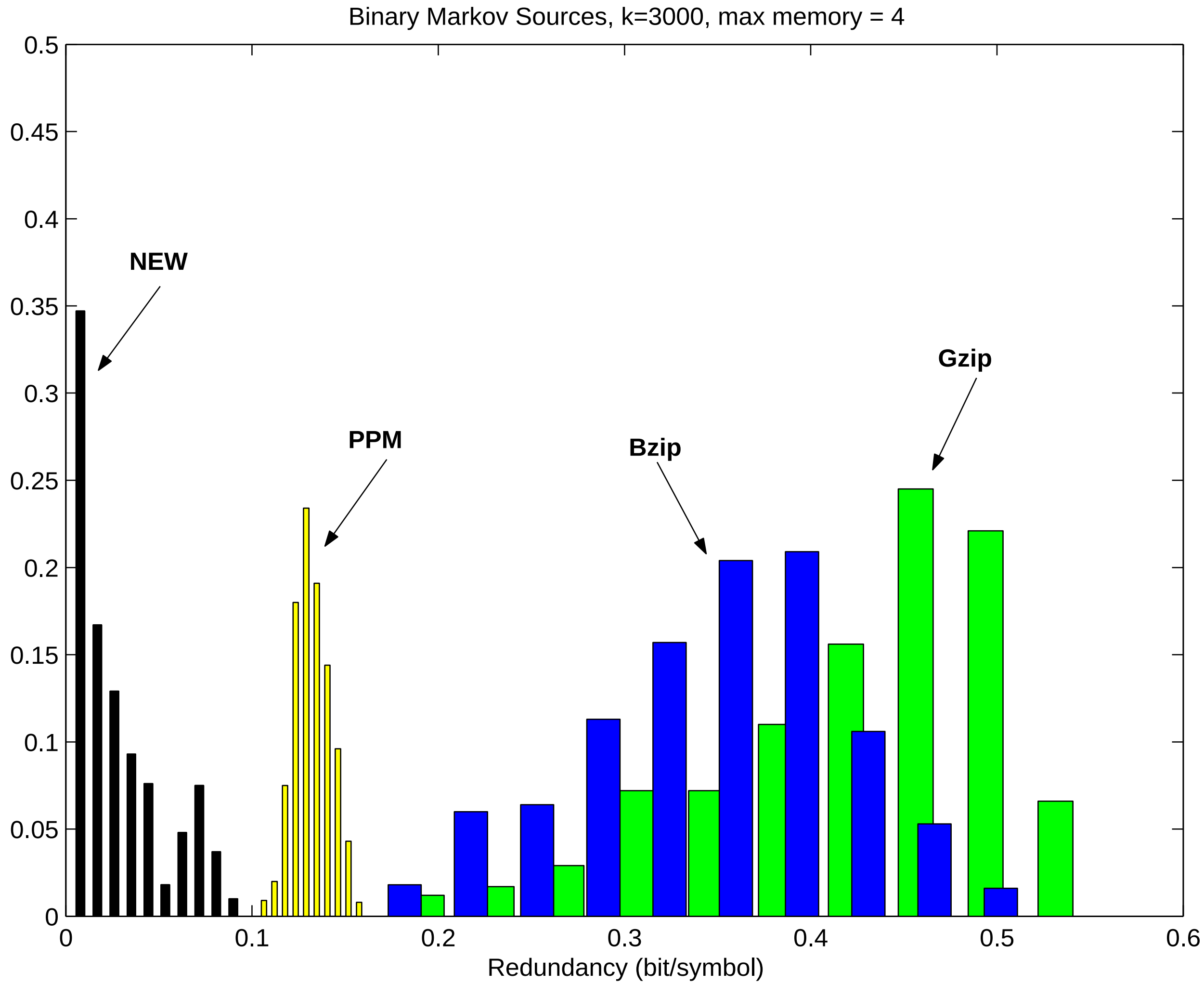


Figure 2: Histogram of redundancies; blocklength = 3000

Random ensemble of Markov sources

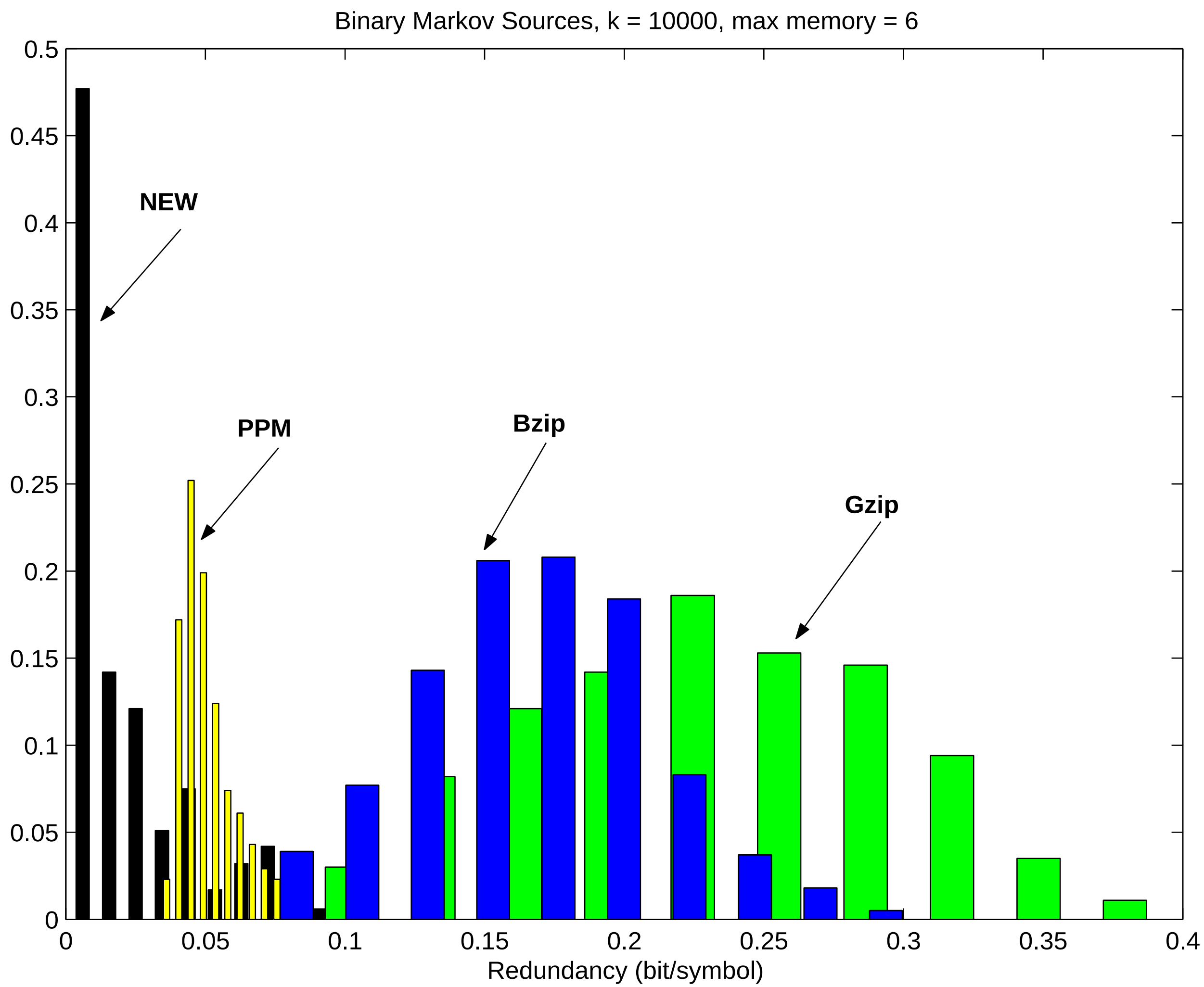


Figure 3: Histogram of redundancies; blocklength = 10000

Library of LDPC matrices vs. Fountain Coding

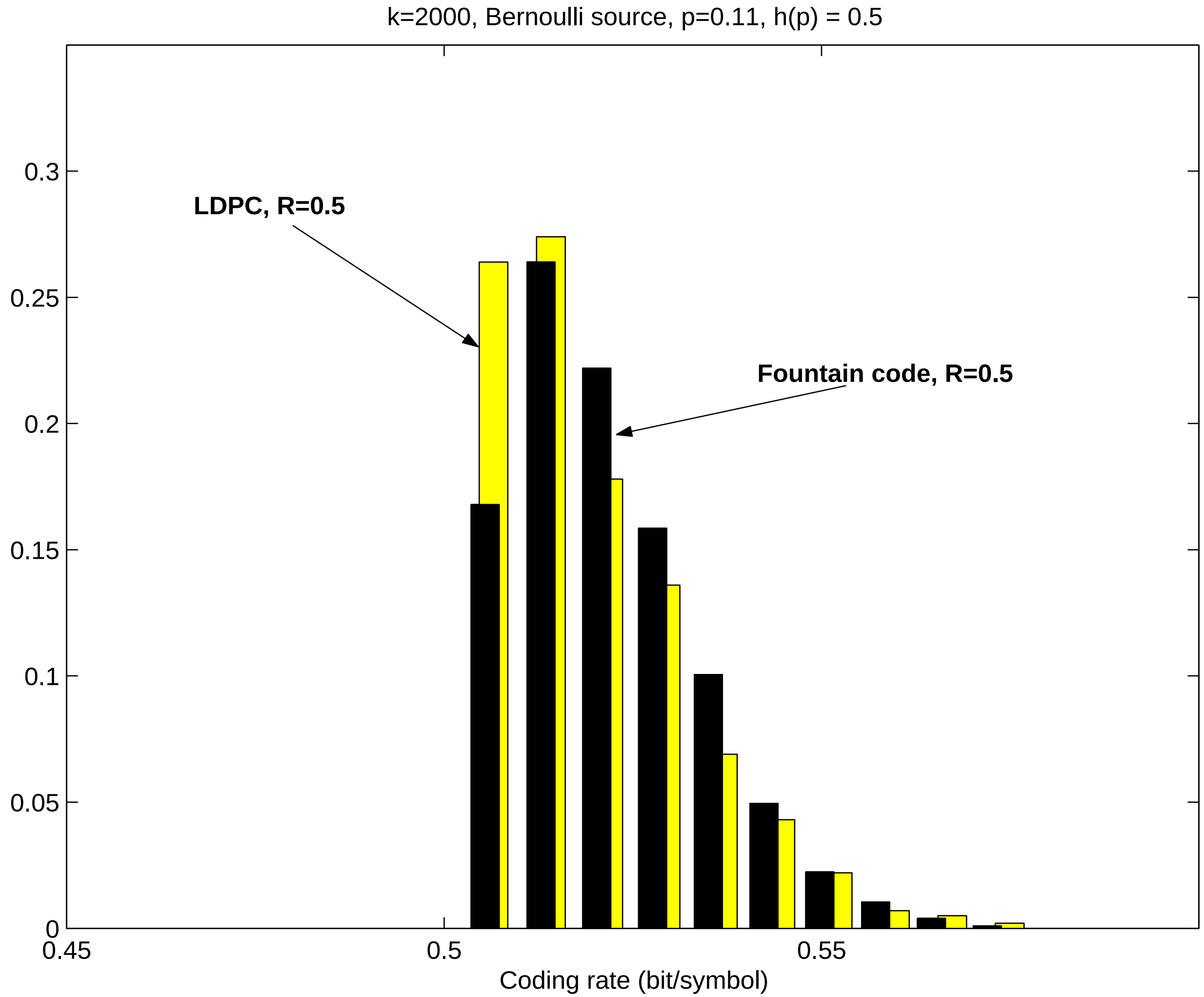


Figure 4: Histogram of redundancies; non-universal $p = 0.11$, blocklength = 2000

CLID versus open-loop incremental redundancy

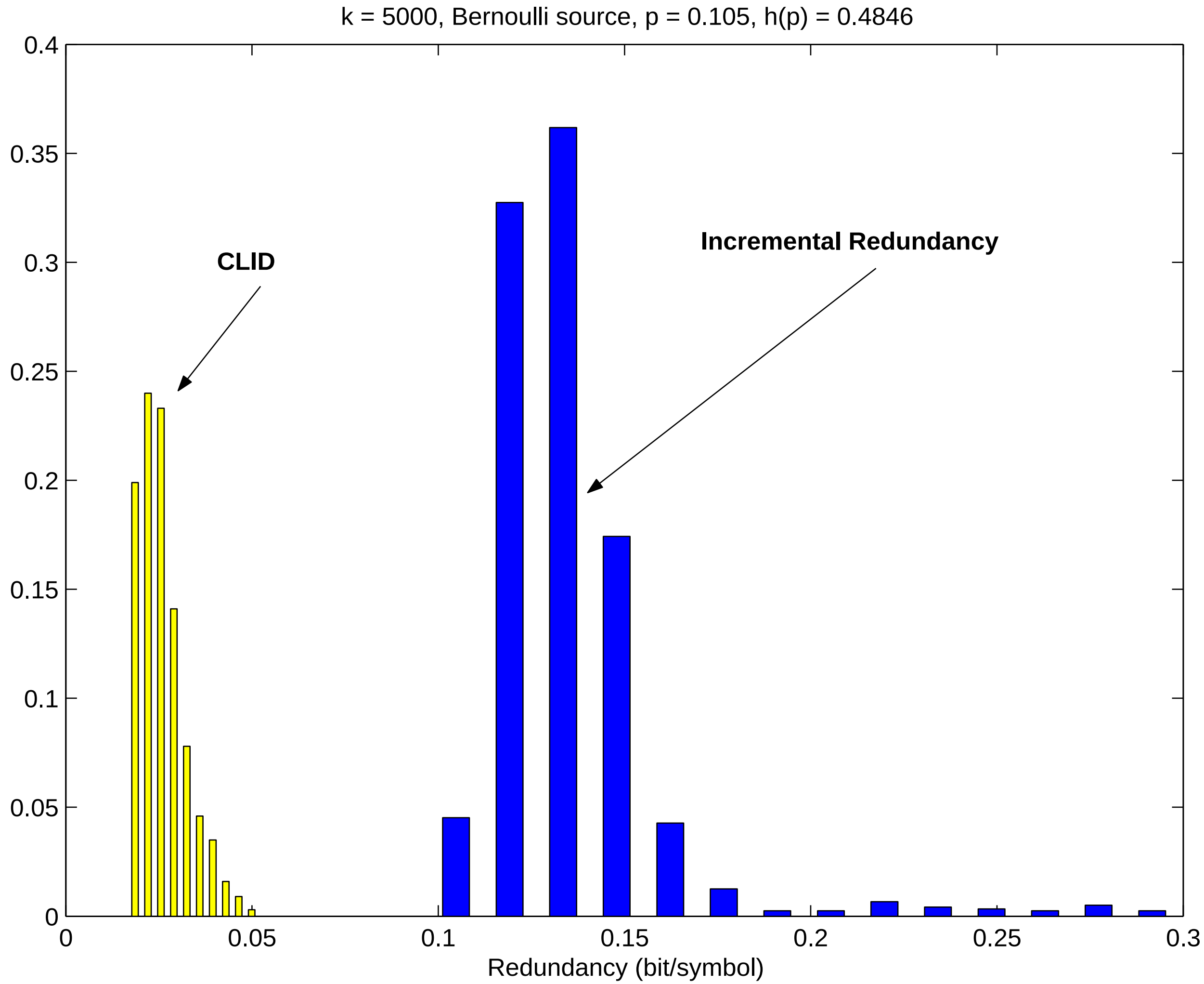


Figure 5: Histogram of redundancies; non-universal $p = 0.105$, blocklength = 5000

Applications: Slepian-Wolf data compression

- Slepian-Wolf separate compression of correlated sources is one of the corner-stones of multiterminal information theory.

- Although known since a long time, it has recently received renewed interest because of new emerging applications (e.g., sensor networks).

- Slepian-Wolf region:

$$R_1 \geq H(X|Y), \quad R_2 \geq H(Y|X), \quad R_1 + R_2 \geq H(X,Y)$$

- Focus on the vertex $R_2 = H(Y)$, $R_1 = H(X|Y)$. Problem: ENC1 must encode X at the conditional entropy without knowing the realization of Y.

Slepian-Wolf random binning

- Random binning: form a random code of size $2^{n(H(X)+2\epsilon)}$, $\sim P_X$, and randomly partition it into $2^{n(H(X|Y)+\epsilon)}$ bins of size $2^{n(I(X;Y)+\epsilon)}$.

- Encoding: assuming $\mathbf{x}$ typical, output the index of the bin that contains $\mathbf{x}$.

- Decoding: the decoder looks into the bin for the $\widehat{\mathbf{x}}$ jointly typical with $\mathbf{y}$.

- For any $\epsilon > 0$ and sufficiently large n the probability of encoding and decoding errors vanishes.

Slepian-Wolf compression is a channel coding problem

- Assume $\mathcal{X} = \mathbb{F}_q$: S-W operating at $(R_1 = H(X|Y), R_2 = H(Y))$ can be implemented optimally by linear block codes (Wyner scheme).

- Let $\mathbf{H}_1 \in \mathbb{F}_q^{m \times n}$ be a parity-check matrix. A computes the syndrome $\mathbf{s}_1 = \mathbf{H}_1 \mathbf{x}$ and sends it to B.

- B has $\mathbf{y}$ and $\mathbf{s}_1$ and reconstructs $\mathbf{x}$ by ML decoding:

$$\widehat{\mathbf{x}} = \arg \max_{\mathbf{x}:\mathbf{H}_1\mathbf{x}=\mathbf{s}_1} P_{X|Y}(\mathbf{x}|\mathbf{y})$$

- If $m/n > H(X|Y)$, for any $\epsilon > 0$ there exist matrices $\mathbf{H}_1$ such that $P_e < \epsilon$ for sufficiently large n.

- S-W is as hard as achieving channel capacity (with small block error rate!!!).

Applications: bidirectional data-exchange

- Caire-Shamai-Verdu, 2004.
- Let A and B have data vectors $\mathbf{x} \in \mathcal{X}^n$ and $\mathbf{y} \in \mathcal{Y}^n$, respectively.
- $\mathbf{x}, \mathbf{y}$ are generated according to some known joint pmf $P_{X,Y}$.
- A and B are connected by a bidirectional lossless channel.
- Goal: make $\mathbf{x}$ known to B and $\mathbf{y}$ known to B with minimal rate.

Bidirectional data exchange rate region

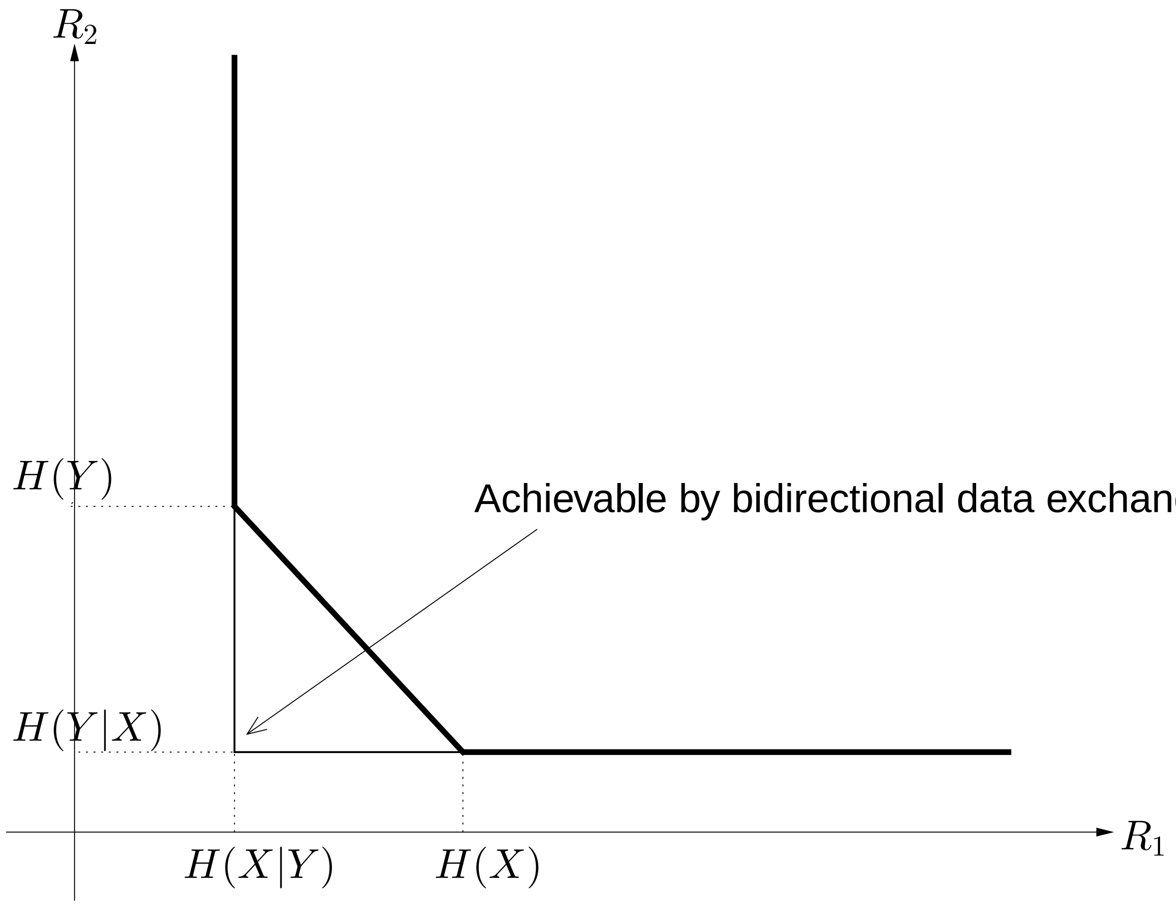
- Proposition:

$$R_1 \geq H(X|Y), \quad R_2 \geq H(Y|X)$$

- Achievability: use Slepian-Wolf in both directions.

- Converse: let a genie give $\mathbf{y}$ to the encoder in A, the minimum rate to communicate $\mathbf{x}$ to B is still $H(X|Y)$. The same argument holds in the reverse direction.

The Slepian-Wolf and data exchange regions



Scenario: multicast over heterogeneous networks

- A is a server that sends a source file $\mathbf{x}$ to a client B via an unreliable high-throughput forward link (e.g., DVB-S downlink, DVB-T downlink, DAB).

- After demodulation/decoding, B has $\mathbf{y} = \mathbf{x} + \mathbf{e}$, a corrupted version of $\mathbf{x}$.

- A and B are also connected by a bidirectional reliable data link (e.g., GPRS, UMTS, Wired Internet).

- The bidirectional reliable link has low throughput (or high-cost).

- We wish to use efficiently the reliable link to transfer $\mathbf{x}$ to B in a fully reliable way by exploiting the fact that B has already received $\mathbf{y}$.

Example: BEC “correlation” (one-directional)

- Suppose that $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$ are linked by a BEC with erasure probability p. and focus on A $\rightarrow$ B.
- Using a Fountain Code, A produces $m > np$ parity bits $\mathbf{s}_1$ and sends them to B via the reliable link.
- With high probability, B is able to decode $\mathbf{x}$ since it has received $\approx (1-p)n$ symbols via $\mathbf{y}$ and additional $> np$ symbols via $\mathbf{s}_1$.
- Communication takes place at rate slightly above $H(X|Y) = p$ in direction A $\rightarrow$ B and zero in the reverse direction.
- Problem: very heavy demand on the performance of the Fountain Code.

Example: BEC “correlation” (bi-directional)

- Idea! The bidirectional link affords significant simplification.
- B encodes the erasure pattern using standard data compression and sends it to A.
- A just sends the erased symbols to B.
- No encoding/decoding is needed, and communication take place at rate $\approx H(Y|X) = h(p)$ in direction B $\rightarrow$ A and $\approx H(X|Y) = p$ in direction A $\rightarrow$ B.
- In the BEC “correlation” case, we achieve the S-W rates in both directions for data exchange and avoid complicated coding and decoding.

Low-complexity bi-directional CLID algorithm.

- Step 1. A sends $\mathbf{s}_1 = \mathbf{H}\mathbf{x}$ to B, and B sends $\mathbf{s}_2 = \mathbf{H}\mathbf{y}$ to A.

- Step 2. Both terminals have $\mathbf{s} = \mathbf{s}_1 - \mathbf{s}_2 = \mathbf{H}(\mathbf{x} - \mathbf{y})$: they run locally exact copies of the BP decoder and use CLID over the bidirectional link to recover $\mathbf{e}$ exactly.

- At each CLID round, d symbols of $\mathbf{x}$ are exchanged in direction A $\rightarrow$ B and d symbols of $\mathbf{y}$ in the same positions are exchanged in direction B $\rightarrow$ A, such that both terminals can compute the value of $\mathbf{e}$ in the doped positions.

Rate of the scheme.

- This scheme achieves $H(X_i - Y_i) + \Delta$ in both directions, where Δ is due to CLID and to the suboptimality (gap from capacity) of the LDPC code.

- In general, $\max\{H(X|Y), H(Y|X)\} \leq H(X_i - Y_i)$ unless X and Y are non-redundant and $\mathbf{e} = \mathbf{x} - \mathbf{y}$ is i.i.d..

- Remark: this is the case where $\mathbf{x}$ is a compressed source and $\mathbf{y}$ is the result of channel decoding with sufficient interleaving (e.g., multicast of compressed video).

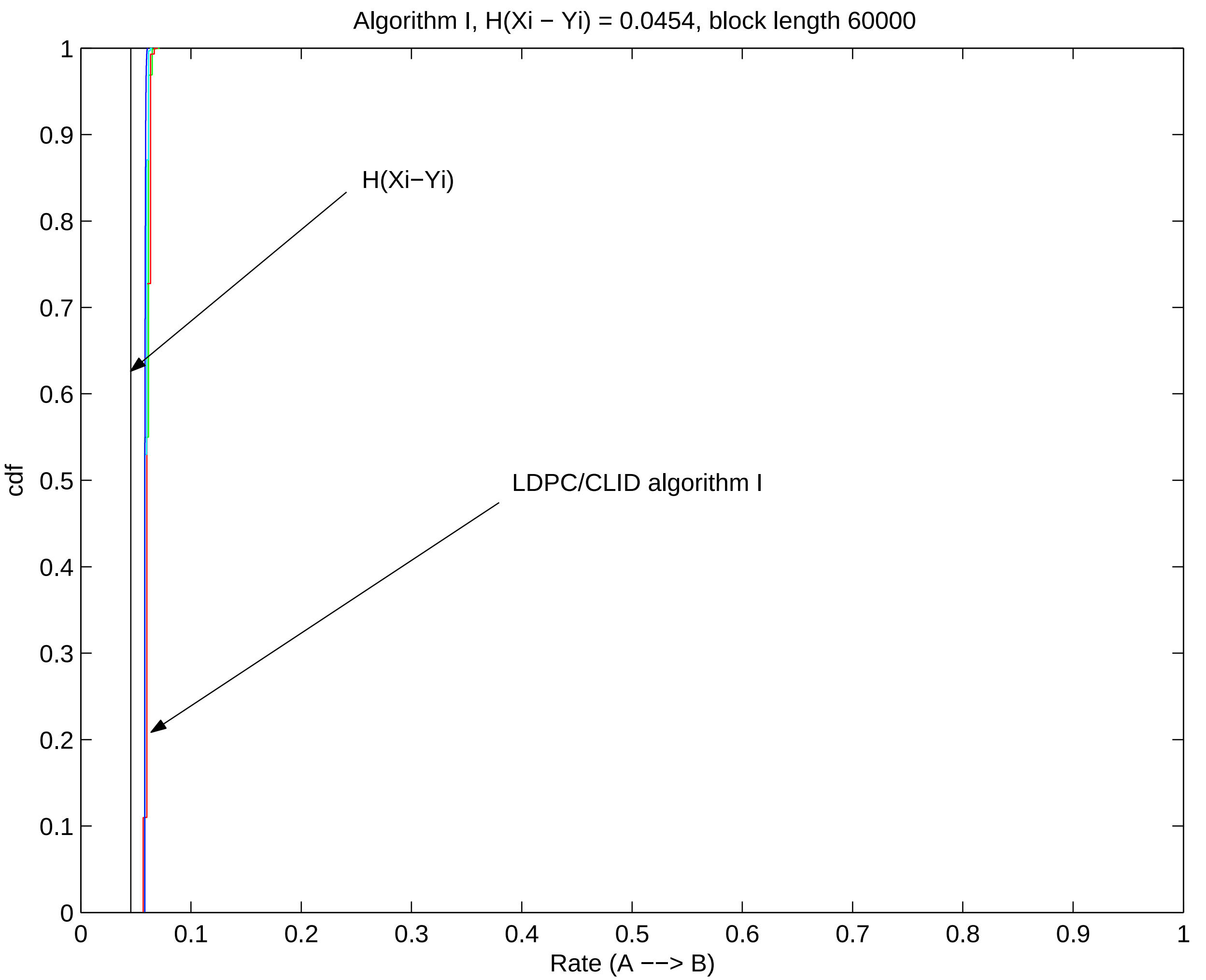
Numerical experiment

- Let $\mathbf{x}$ be a binary i.i.d. uniform vector and $\mathbf{y} = \mathbf{x} + \mathbf{e}$, with $\mathbf{e}$ i.i.d. Bernoulli-p_e.
- We choose $p_e = 0.005$, yielding $H(X_i - Y_i) = H(e_i) = 0.0454$ bit/symbol.
- We used an irregular LDPC ensemble with block length $n = 60000$ and degree sequences optimized using approximated density evolution and linear programming:

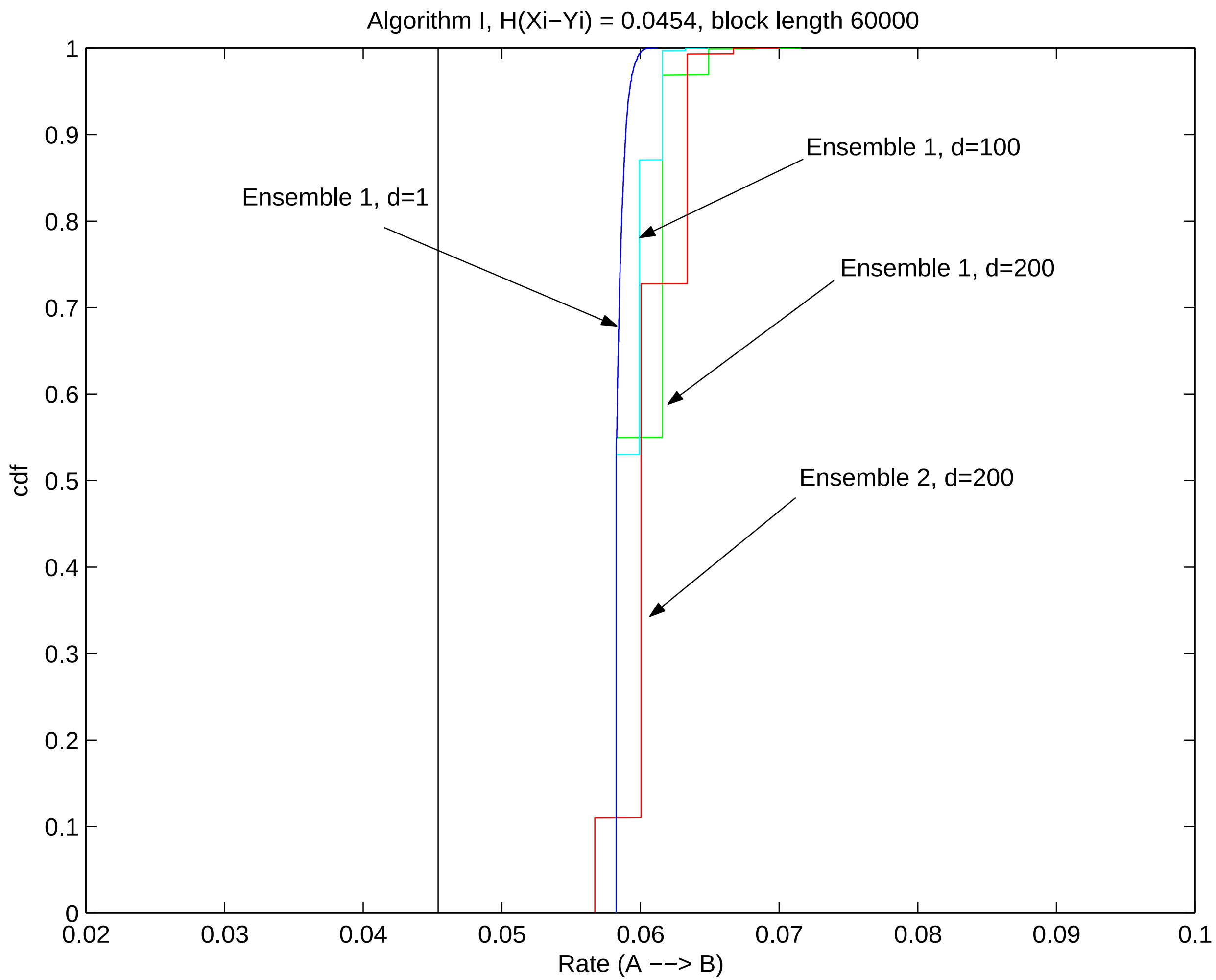
$$\lambda(x) = 0.3479x^2 + 0.3178x^9 + 0.0386x^{30} + 0.2957x^{31}, \quad \rho(x) = 0.5x^{107} + 0.5x^{108}$$

- The compression coding rate is 0.0583, corresponding to 28% redundancy w.r.t. the S-W limit.
- At each CLID round, a block of d CLID bits are sent. We considered the cases $d = 1, 10, 50, 100, 200$.

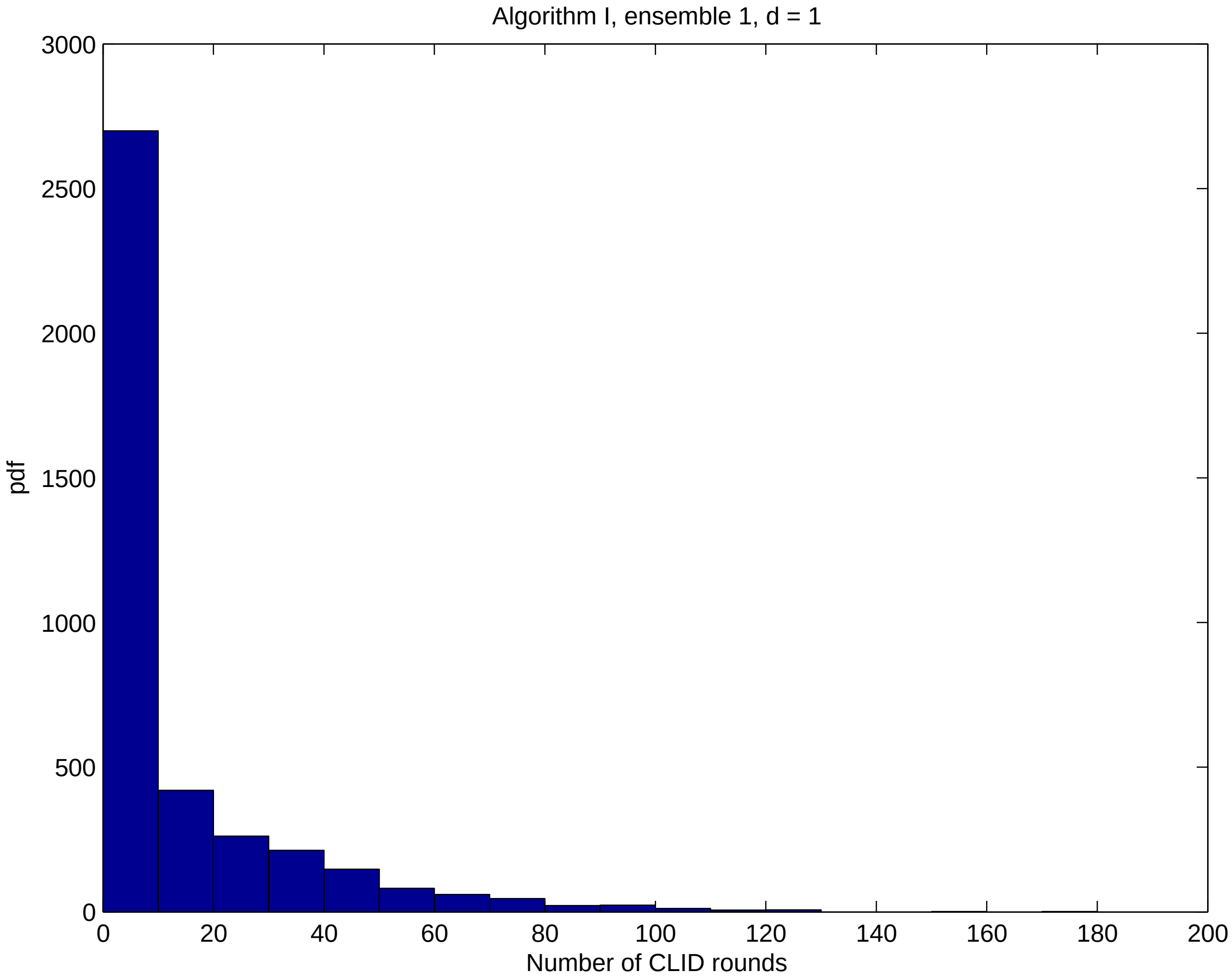
Rate cumulative distribution



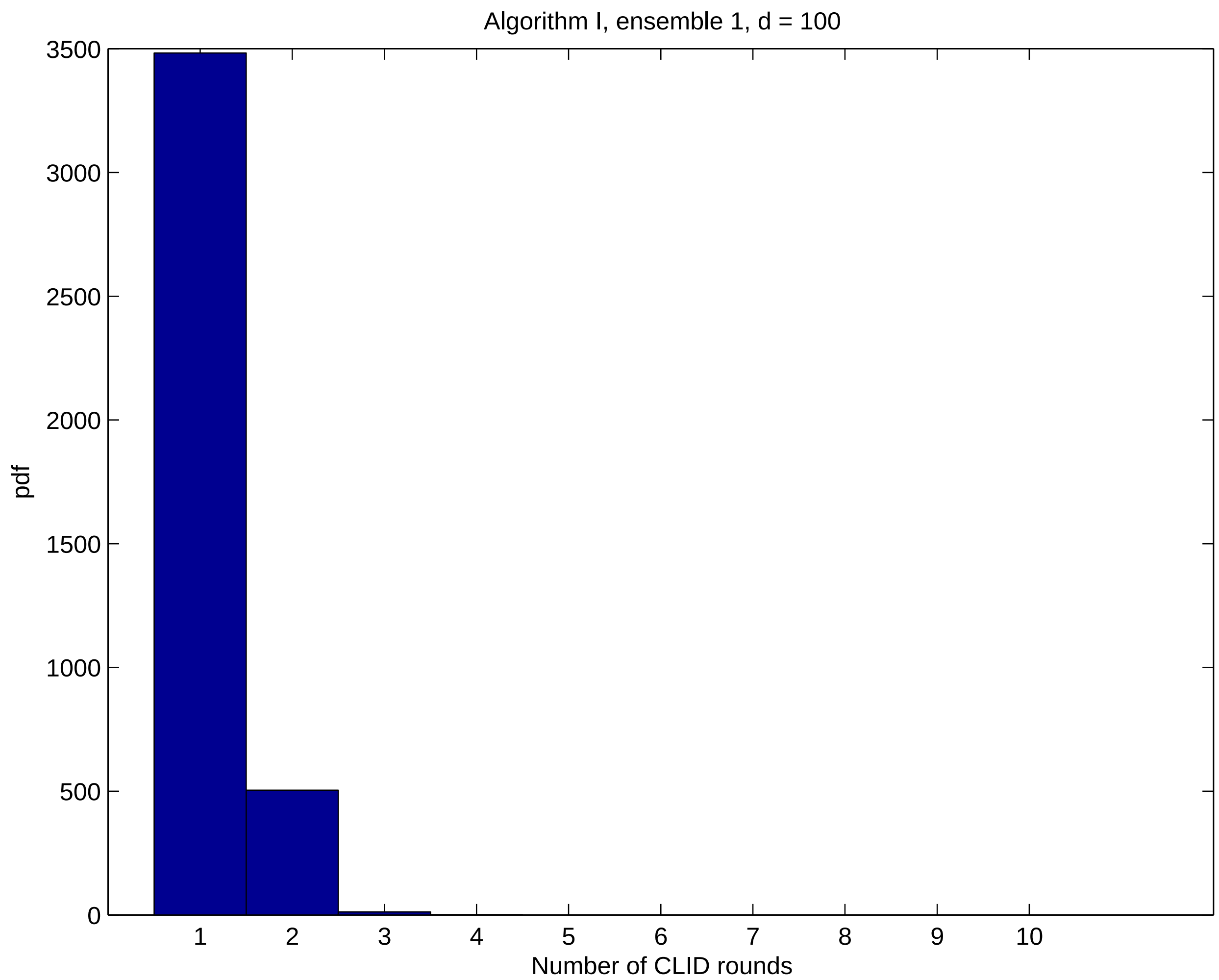
Rate cumulative distribution (detail)



Histogram of the number of protocol CLID rounds, $d = 1$



Histogram of the number of protocol CLID rounds, $d = 100$



Applications: coding with limited feedback

- It is well-known that feedback does not increase capacity of memoryless channels.
- Nevertheless, feedback may achieve a target P_e with much smaller block length and/or much less complexity.
- Shannon's feedback: the encoder at time k knows the channel output $y_1, \ldots, y_{k-1}$.
- Burnashev's exponent:

$$E_f(R) = \overline{D}(1 - R/C)$$

where C is the channel capacity and

$$\overline{D} = \max_{a,b \in \mathcal{X}} D(P_{Y|X=a} \| P_{Y|X=b})$$

A two-step scheme achieving Burnashev's exponent

- Step 1: the information message is transmitted using a good near-capacity achieving code for the underlying DMC (without feedback).

- Step 2: now the encoder knows exactly the state of the decoder and sends an accept/deny message by using repetition coding, i.e., by repeating L times the letter a (accept) or b (deny) where a and b are chosen to achieve $\overline{D}$.

- If "deny" is decoded, the whole block is re-transmitted. If "accept" is decoded, transmission of the current block stops.

- An error occurs when "accept" is decoded while "deny" was transmitted.

Practical (limited) feedback schemes

- ARQ
- Hybrid-ARQ
- Essentially based on ACK/NACK: performance dominated by the ability of the decoder to detect its error status.
- A problem of practical interest: devise schemes that attain both high reliability and low complexity for given fixed feedback rate R_f (ratio of feedback symbols over forward channel symbols).

Proposed scheme

- Caire, Shamai, Verdu 2005.
- Serial concatenation of two LDPC codes: outer coding $\mathbf{b} \mapsto \mathbf{x}_1$, inner coding $\mathbf{x}_1 \mapsto \mathbf{x}_2$.
- Step 1: upon reception of the channel output $\mathbf{y}$, the decoder runs BP over the joint Tanner graph and obtains $\widehat{\mathbf{x}}_1$, such that, assuming interleaving, $\mathbf{e}_1 = \widehat{\mathbf{x}} - \mathbf{x}_1 \sim$ Bernoulli-p.
- Main feedback: the decoder computes the syndrome $\mathbf{z}_1 = \mathbf{H}_1 \hat{\mathbf{x}}_1 = \mathbf{H}_1 \mathbf{e}_1$. If $\mathbf{z}_1 = 0$, the receiver declares the subvector of $\hat{\mathbf{x}}_1$ as the output. If $\mathbf{z}_1 \neq 0$, $\mathbf{z}_1$ is sent over the feedback link.
- The encoder knows $\mathbf{x}_1$ and $\mathbf{z}_1$ and the decoder knows $\hat{\mathbf{x}}_1$ and $\mathbf{z}_1$, but neither knows $\mathbf{e}_1$.

- Step 2: both encoder and decoder run BP simultaneously on the Tanner graph of the outer code with syndrome equal to $\mathbf{z}_1$, and variable nodes associated to the elements of $\mathbf{e}_1$.

- Problem: similar to the bidirectional data exchange, but CLID cannot be applied since the forward link is noisy.

- Answer: we use a noisy version of the CLID algorithm ... the “dirty CLID”.

- Let ℓ the position of the least reliable symbol at a given doping iteration, then the encoder sends $x_{1,\ell}$ through the forward channel which outputs $y_{1,\ell}$. The decoder computes the MAP estimates $\widetilde{x}_{1,\ell}$ and $\widetilde{e}_{1,\ell} = \widetilde{x}_{1,\ell} - \widehat{x}_{1,\ell}$ and feeds it back to the encoder via the noiseless feedback link.

- Both encoder and decoder have the same noisy observation of $e_{1,\ell}$ and can update their BP algorithm by incorporating the additional message

$$(-1)^{\widehat{e}_{1,\ell}} \log \mathcal{L}$$

for the ℓ-th variable node.

- The BP algorithms at the encoder and decoder remain synchronized. The decoding process stops at an iteration when its guess $\widehat{\mathbf{e}}_1$ satisfies the parity-check equations, i.e.,

$$\mathbf{z}_1 = \mathbf{H}_1 \widehat{\mathbf{e}}_1$$

Design options

- The outer code must be optimized for a BSC with probability of error p. Hence, $r_1 < 1 - h(p)$.

- Letting $\mathbb{E}[d]$ the expected number of doped bits, the resulting feedback rate is given by

$$R_f = \frac{nr_2(1 - r_1) + \mathbb{E}[d]}{n + \mathbb{E}[d]} \approx r_2(1 - r_1) + \eta$$

  where η is small (doping redundancy).

- Two extreme cases.

- Choosing a trivial inner code and an outer code optimized for the forward channel, we get $r_2 = 1$ and $r_1 = C - \epsilon$, with some gap-to-capacity $\epsilon > 0$ which is a code design parameter. This choice yields $R_f \approx 1 - C + \epsilon + \eta$.

- Choosing an inner code optimized for the forward channel yields $r_2 = C - \epsilon$ and very small p, such that $r_1 = 1 - h(p) - \epsilon \approx 1$. This yields

$$R_f = (C - \epsilon)(h(p) + \epsilon) + \eta$$

which can be made as small as desired.

Robust joint source channel coding

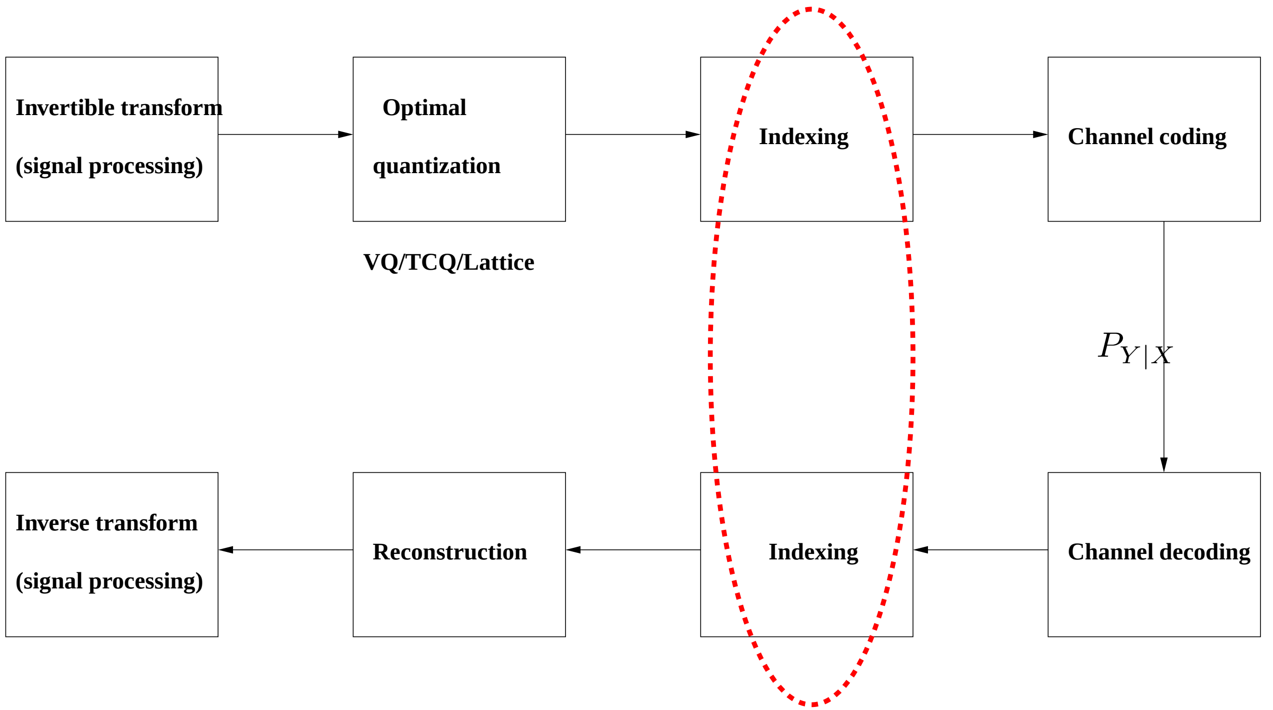
- Data compression via linear block codes, from the pure data compression point of view, is not competitive with respect to state-of-the-art data compression.

- ON THE BAD SIDE: residual error probability, complexity of BP decoding.

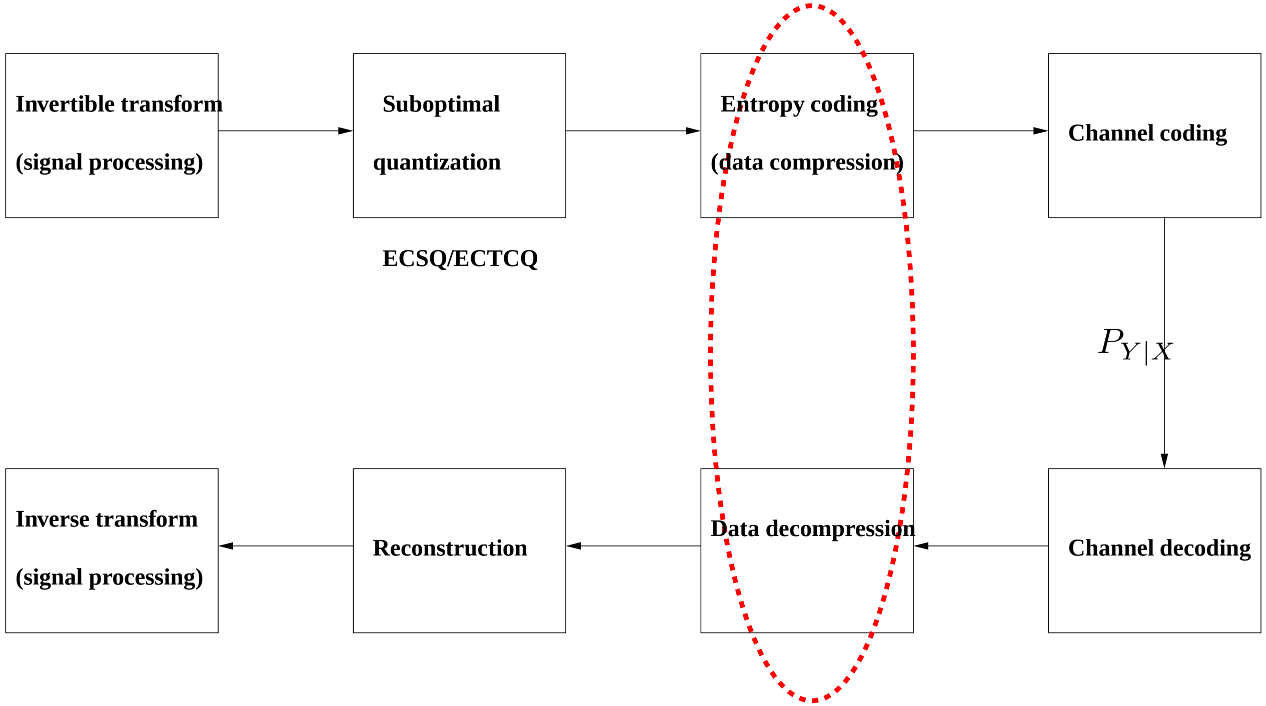
- ON THE GOOD SIDE: linear mapping has generally a well conditioned inverse (non-catastrophic!): a small Hamming distance between encoded sequences reflects into a small Hamming distance of the source reconstructed sequence.

- Standard state-of-the-art data compression has generally catastrophic inverse: one bit error yields a random reconstructed sequence, almost independent of the independent sequence.

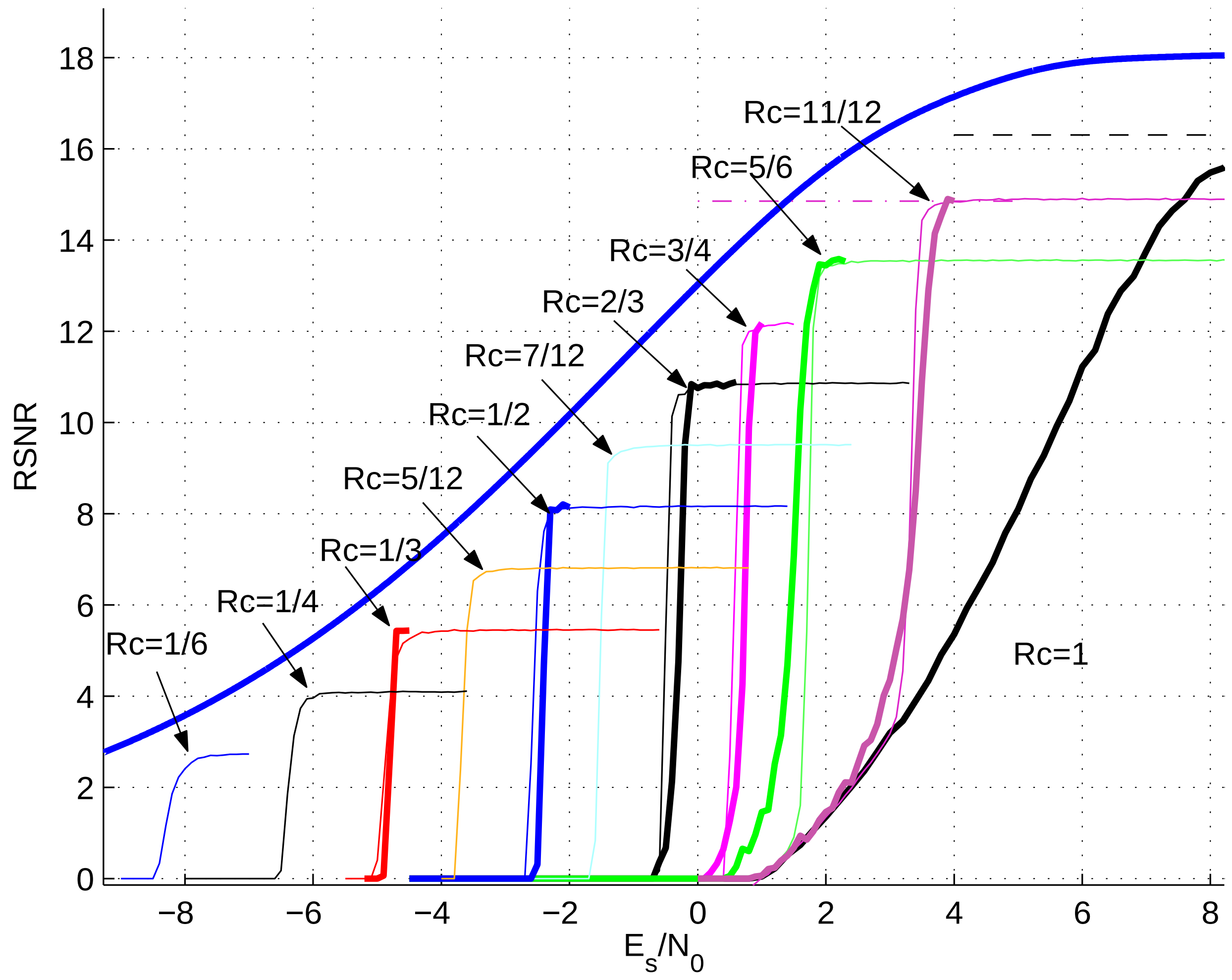
Option 1: $\approx$ optimal quantization with clever indexing



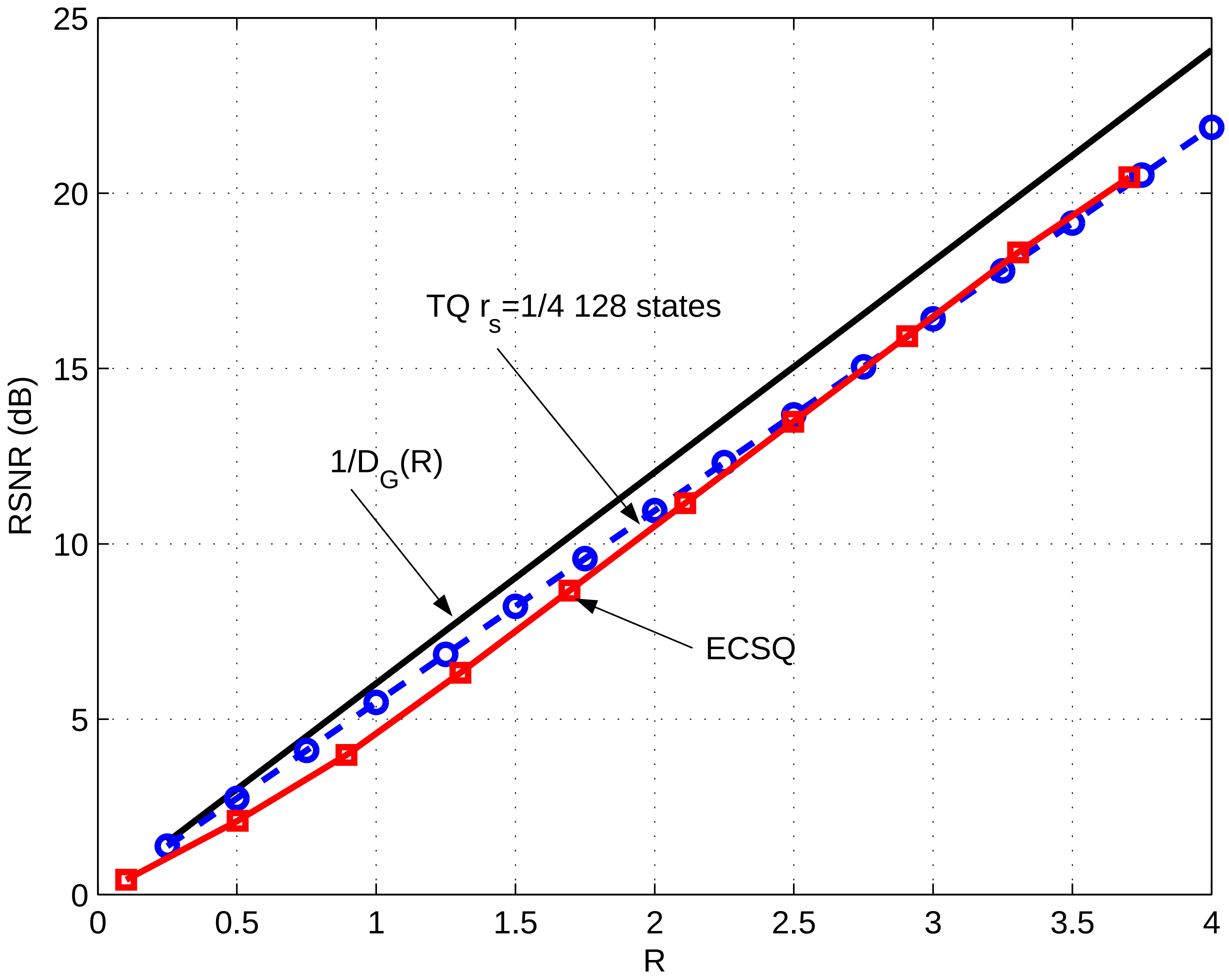
Option 2: Suboptimal quantization with entropy coding



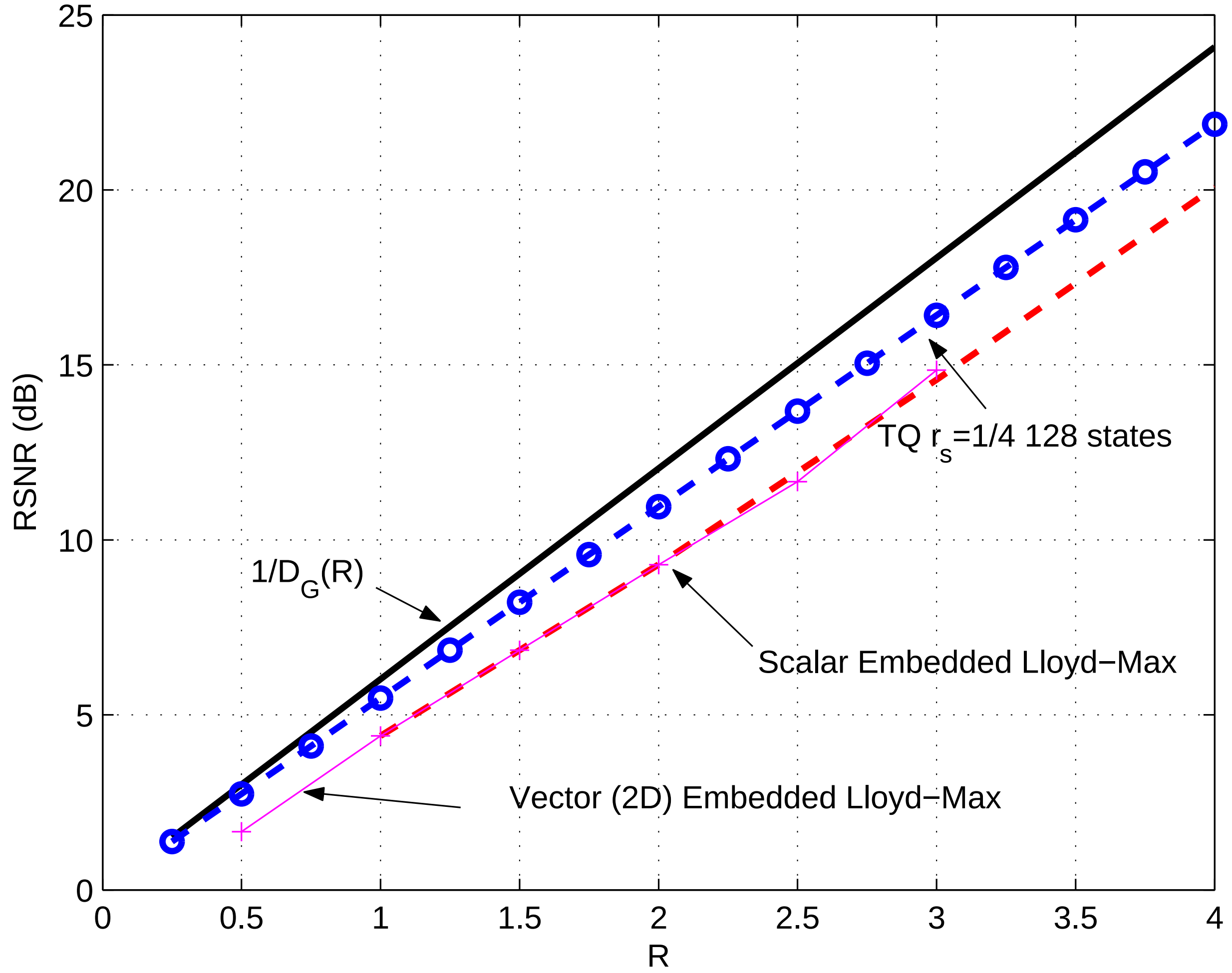
Option 1: Multistage TCQ + standard LDPC coding



Option 2: ECSQ + LDPC coding for data compression



Comparison with successive refinement scalar quantization



Design tradeoffs

- Multistage TCQ based on binary CC (typically, rate 1/4, 128 states) yields an efficient successive refinement quantization scheme, yet, suboptimal at high rates.
- Binary CC yields directly a non-catastrophic binary indexing (directly following from non-catastropic property of the CC encoder).
- Soft-reconstruction (softbits) yields just minor improvement w.r.t. hard reconstruction.
- Entropy-Constrained Scalar Quantization is much simpler but not generally succesively refinement.
- It requires careful mapping of M-ary indices onto binary: we have investigated multi-level coding, by exploiting the entropy chain rule:

$$H(\mu) = H(b_1) + H(b_2|b_1) + \cdots H(b_m|b_{m-1}, \ldots, b_1)$$

- Pleasant reminiscence of the “bit-plane” structure of practical source coding schemes (e.g., JPEG2000).

- It is expected that ECTCQ $+$ carefully designed LDPC yields the best performance (to be optimized and demonstrated ...).

Examples [On-going work!]

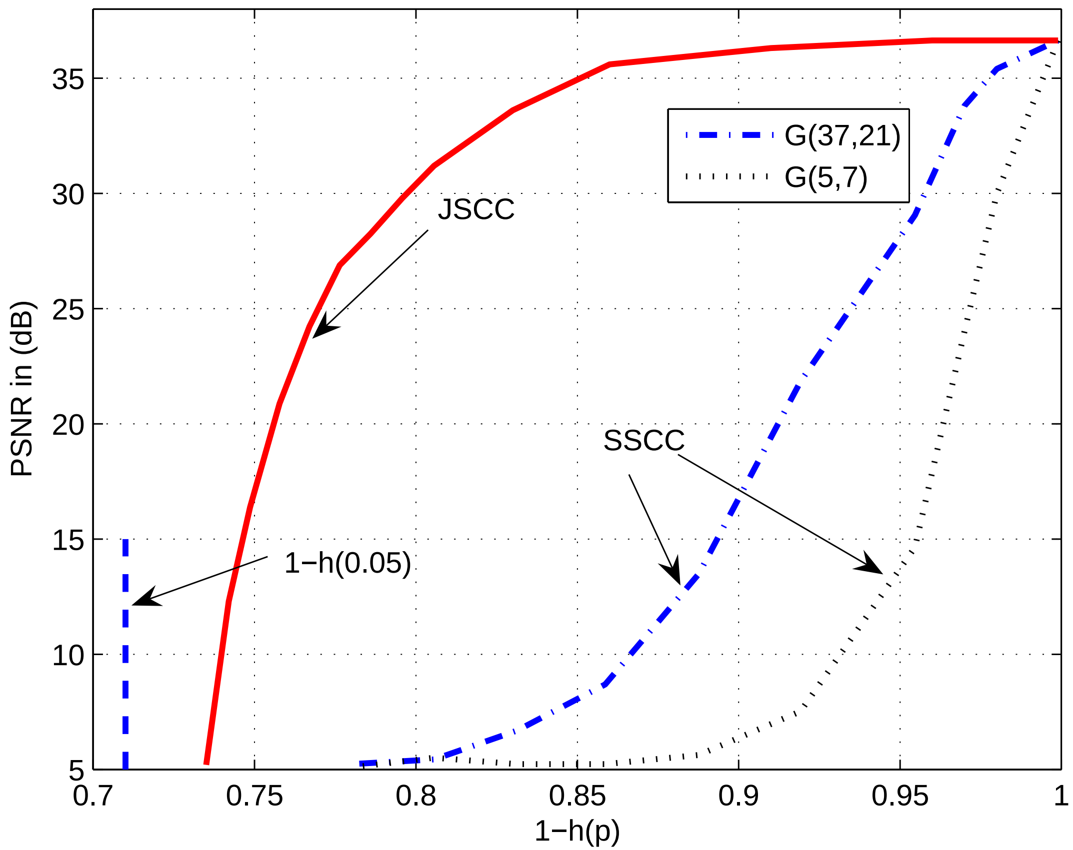
- Concatenation of predictive quantization for images, followed by adaptive probability estimation/identification and compression by multi-level Turbo Codes [Kim, Sesia, Ramstad and Caire, 2005].

- In this work we exploit the fact that the quantized source after predictive quantization looks like a non-stationary but independent M discrete source.

- Concatenation of Wavelet transform, quantization and Markov (context) model estimation of JPEG2000 with multi-level Turbo-Codes [on-going work with M. Fresia].

- In this work we exploit the fact that JPEG2000 models the bitplanes as non-stationary binary Markov-Chains, whose structure can be incorporated into the BP decoder.

A numerical experiment



Conclusions

- Data compression via linear coding $\mathbf{z} = \mathbf{H}\mathbf{x}$ is theoretically an optimal fixed-to-fixed length data compression scheme.
- Because of complexity, we use sparse $\mathbf{H}$ and BP decoding.
- Good performances in the non-universal and universal case can be obtained by decoding at the encoder, using a library of matrices, CLID, BWT, MDL probability model estimation.
- Although competing with state-of-the art is very hard for the pure data (lossless) compression case, linear block coding has unique “non-catastrophic” features that make it a key building block for robust JSCC.
- Distributed data compression (Slepian-Wolf) is a channel coding problem, therefore there is no alternative.

Open problems

- Open problem 1: using modern (sparse-graph based) techniques for lossy compression. This is a totally different approach, linear maps cannot be optimal for lossy coding!.
- Preliminary results in special cases: Zecchina, Mezard et al., building on the analogy with the K-SAT problem and using techniques such as Survey Propagation.
- Open problem 2: lossy transmission for a compound/broadcast channel, achievable distortion region.
- Very important for future DAB/DVB, and MM broadcasting for 3G+.
- Best known performances are obtained by Hybrid analog-digital schemes.