Modern Channel Coding Techniques in Source Coding Problems: An Overview

Joint work with: S. Verdú, S. Shamai, S. Seisig, K. Narayanan
Giuseppe Caire, Eurecom

May 23 – 25, 2005
3rd BEATS/WIP/CUBAN Workshop
Almost lossless data compression using block codes.

Applications: Slepian-Wolf, Bidirectional data exchange, Coding with limited feedback.

Applications to "robust" (lossy) source-channel coding.

Conclusions.
Theorem (Shannon–Macmillan) If \( \frac{u}{w} < \frac{H}{n} + \delta \), probability of block error \( \to 0 \).

\[
\begin{array}{cccccccc}
  n & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 2 \ 1 \\
  m & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 2 \ 1 \\
  u & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 2 \ 1 \\
\end{array}
\]
Block Huffman code for source with 2^7 values.

Optimum block source code: encode the 2^3 most probable source words.

Example: 7-to-3 code for bias coin.
Given the linearly encoded vector $z$, ML decoder chooses $\hat{u} = H(z)$ such that $Hu = z$.

Encoder: Given the linearly encoded vector $z = Hs$, ML decoder chooses $\hat{u} = H(z)$ such that $Hu = z$.

<table>
<thead>
<tr>
<th>Encoder: $z = Hs$</th>
<th>Decoder: $\hat{u} = H(z)$</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>s</th>
<th>t</th>
<th>g</th>
<th>t</th>
<th>h</th>
<th>i</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>H</td>
<td>h</td>
<td>u</td>
<td>k</td>
<td>y</td>
<td>h</td>
</tr>
<tr>
<td>i</td>
<td>t</td>
<td>o</td>
<td>n</td>
<td>h</td>
<td>u</td>
<td>t</td>
</tr>
<tr>
<td>e</td>
<td>g</td>
<td>f</td>
<td>a</td>
<td>r</td>
<td>n</td>
<td>m</td>
</tr>
<tr>
<td>x</td>
<td>a</td>
<td>r</td>
<td>n</td>
<td>$H$</td>
<td>z</td>
<td>i</td>
</tr>
<tr>
<td>n</td>
<td>$G$</td>
<td>y</td>
<td>l</td>
<td>e</td>
<td>g</td>
<td>n</td>
</tr>
<tr>
<td>u</td>
<td>m</td>
<td>r</td>
<td>t</td>
<td>k</td>
<td>x</td>
<td>i</td>
</tr>
<tr>
<td>r</td>
<td>w</td>
<td>c</td>
<td>y</td>
<td>f</td>
<td>g</td>
<td>s</td>
</tr>
<tr>
<td>y</td>
<td>i</td>
<td>t</td>
<td>w</td>
<td>b</td>
<td>a</td>
<td>o</td>
</tr>
<tr>
<td>h</td>
<td>e</td>
<td>x</td>
<td>o</td>
<td>h</td>
<td>r</td>
<td>l</td>
</tr>
<tr>
<td>o</td>
<td>i</td>
<td>t</td>
<td>a</td>
<td>c</td>
<td>h</td>
<td>e</td>
</tr>
<tr>
<td>d</td>
<td>e</td>
<td>x</td>
<td>a</td>
<td>x</td>
<td>s</td>
<td>i</td>
</tr>
<tr>
<td>p</td>
<td>w</td>
<td>y</td>
<td>x</td>
<td>b</td>
<td>a</td>
<td>e</td>
</tr>
<tr>
<td>a</td>
<td>m</td>
<td>b</td>
<td>b</td>
<td>n</td>
<td>e</td>
<td>h</td>
</tr>
<tr>
<td>s</td>
<td>g</td>
<td>l</td>
<td>r</td>
<td>c</td>
<td>l</td>
<td>m</td>
</tr>
<tr>
<td>p</td>
<td>d</td>
<td>e</td>
<td>z</td>
<td>i</td>
<td>p</td>
<td>h</td>
</tr>
<tr>
<td>u</td>
<td>n</td>
<td>t</td>
<td>z</td>
<td>i</td>
<td>p</td>
<td>h</td>
</tr>
</tbody>
</table>

Linear Block Source Codes
Coset partition and source typical set
Theorem (Elias, 1955; Csiszar-Korner, 1981)

If the source is stationary memoryless, the encoding matrix entries are independent, equiprobable on the alphabet, and
\[ q + H < \frac{u}{w} \]

If the source has sup-entropy rate \( H \), the average block error probability \( P_e \) is 0.

Theorem for general sources [Verdu-Shamai-Caire, 2003]:

If the source is stationary memoryless, the encoding matrix entries are independent, equiprobable on the alphabet, and
\[ q + H < \frac{u}{w} \]

If the source has sup-entropy rate \( H \), the average block error probability \( P_e \) is 0.
Additive-noised discrete channel:

\[ y = x + u \]

Linearchannelcodes (independent equiprobable parity-check matrix entries) achieve capacity.

Random linear codes (independent equiprobable parity-check matrix entries) achieve容量.

Same syndrome \( \mathbf{y} \).

It selects the most likely noise realization among those that lead to the same syndrome \( \mathbf{y} \).

The maximum likelihood decoder selects the codeword \( \mathbf{y} \).

Linear channel code with parity-check matrix \( \mathbf{H} \) such that \( \mathbf{H}^\mathsf{T} \mathbf{H} = \mathbf{I} \).

\[ \mathbf{n} + \mathbf{x} = \mathbf{y} \]

Additive-noise discrete channel.
Linear Block Channel Codes

as

Linear Block Source Codes

Source Encoder = Syndrome Computer:

\[ z = Hs \]

where \( H \) is the parity-check matrix of a channel code.

Maximum Likelihood Source Decoder: \( (z)H^B \) (syndrome decoder)

Block Error Probability = Block Error Probability of the channel code when \( y = x + u \) and \( u \) has the same distribution as \( s \).

Not a new idea! (Ancheta, 1976) tried syndrome coding for bias-polarizing coins \( d \)-biased with \( u = 15, 31 \) Hamming and BCH codes and could beat run-length coding.

Linear Block Source Codes

as

Linear Block Channel Codes
Optimum code: unique codeword for the $2^3$ most probable source words.

Hamming (7,4) parity-check matrix
<table>
<thead>
<tr>
<th>Source Knowledge Required at Decoder</th>
<th>Encoding Complexity</th>
<th>Decoding Complexity</th>
<th>Error Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal (u)O</td>
<td>Fixed (n^2)O</td>
<td>Exp(n)O</td>
<td></td>
</tr>
<tr>
<td>Lossless Fixed</td>
<td>Fixed (n)O</td>
<td>Exp(n)O</td>
<td></td>
</tr>
<tr>
<td>Variable Almost-Lossless LinearBlock</td>
<td>Fixed (n)O</td>
<td>Exp(n)O</td>
<td></td>
</tr>
<tr>
<td>Linear Block</td>
<td>Fixed (n)O</td>
<td>Exp(n)O</td>
<td></td>
</tr>
</tbody>
</table>

**Linear Block Source Codes vs. Lempel-Ziv**
Low-Density Parity-Check (LDPC) Encoding Matrices

Compressed Bits

Source Bits

1

m

I

u

I
Belief-Propagation Decoding (Binary Alphabet)

Apriori source log-ratios

\[ \log p_k = \log p_k; k \in \{1, 2, \ldots, n\} \]

\[ L_k = \prod \text{tanh} \left( z_{(i)}^{(t)} \right) \]

\[ \mathbf{H} = z \]

Checknode \( f \) Bitnodes \( e \)

\[ \mathbf{H} = \begin{cases} \{f\} - \mathbf{e} \in \mathbf{f} & \text{Checknode } f \rightarrow \text{Bitnodes } e \end{cases} \]

\[ \mathbf{J} = \begin{cases} \{u \cdots, 1\} \in \mathbf{f} & \frac{\mathbf{d}}{\mathbf{d} - \log} = \mathbf{J} \end{cases} \]

\[ \mathbf{J} = \mathbf{J} \]

\[ \mathbf{J} = \mathbf{J} \]

\[ \mathbf{J} = \mathbf{J} \]
\[ w, \cdots, 1 = f \quad \forall z = \underbrace{\sum_{\beta \in \mathcal{B}}}_{(t)\sim} x \]

Stop if

\[ \left\{ \forall \alpha \in \mathcal{A}_{(t)} \underbrace{\sum}_{(t)\sim} + \mathcal{J} \right\} \text{sign} = \underbrace{x}_{(t)\sim} \]

Bitnodes tentative decisions •
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).

• Arithmetic coding is easier adapted to source with memory (tree sources).

• Arithmetic coding encodes and decodes the source sequence sequentially (e.g., using the KT estimator).

• Arithmetic coding is still much simpler and faster than BP decoding.

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

Decoding at the Encoder

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

• Closed-loop iterative doping.

Choose "best" for the particular source realization.

Library of Parity-Check Matrices:

• the decompression algorithm.
Closed Loop Iterative Doping

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

\[ b_k = \arg \min_{k=1, \ldots, n} f_j(t_k)， j \in A_k \]

\[ k = \begin{cases} 1 & \text{if } x_k = 0 \\ 1 & \text{if } x_k = 1 \end{cases} \]

Almost lossless fixed-length version: Fixed number of iterations.

Lossless variable-length version (zero error probability):

Choose among the library of H's that leading to the shortest compressed length.

Dope and iterate until decoded word is correct.

Closed Loop Iterative Doping
Empirical Distribution of Doped Bits

Empirical bias of the doped symbols

Source bias

n=2000, p=0.1, d=100, c=8, irregular LDPC R=1/2
Biased Coin Experiment: 500-to-303 source codes for a biased coin.

Figure 1: Block error rate of 500-to-303 source codes for a biased coin.
Biased Coin Experiment: 500-to-303 code: Cont.

Closed-Loop Iterative Doping: 50 bits.

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

Arithmetic Coding: Nonuniversal, source exactly known.

Optimal Block Coding: (non-constructible) optimal code that assigns a unique codeword to the most probable 303 source realizations.

Information Spectrum: “Ideal” variable-length coding scheme where the length of the codeword assigned to $u_x \cdot \cdot \cdot u_x$ is equal to $\log_2 \left( \frac{u_x \cdot \cdot \cdot u_x}{P(x_1; \cdot \cdot \cdot; x_n)} \right)$. 

8(3,6) regular LDPCs (3 bit overhead).
Compression/Decompression Scheme:

- UniversalVersion

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
Figure 2: Histogram of redundancies; block length = 3000

Random ensemble of Markov sources
Random ensemble of Markov sources, block length = 10000

Figure 3: Histogram of redundancies; block length = 10000

- Gzip
- Bzip
- PPM
- NEW

Binary Markov Sources, k = 10000, max memory = 6

Redundancy (bit/symbol)
Figure 4: Histogram of redundancies; non-universal $p = 0.11$, blocklength = 2000.

Library of LDPC matrices vs. Fountain coding
Figure 5: Histogram of redundancies; non-universal, $d = 0.105$, blocklength = 5000, $p = 0.105$, $h(p) = 0.4846$. 

Redundancy (bit/symbol) vs. $k = 5000$. The source is Bernoulli with $p = 0.105$, $h(p) = 0.4846$. 

CLID versus open-loop incremental redundancy.
Applications: Slepian-Wolf data compression

Slepian-Wolf separate compression of correlated sources is one of the cornerstones 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).

Problem: ENC1 must encode $X$ at the conditional entropy without knowing the realization of $Y$.

Focus on the vertex $R_2$.

$H(X;Y) \leq R_1 + R_2$, $H(Y>X) \leq R_2$, $H(X|Y) \leq R_1$

Slepian-Wolf region:

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

Slepian-Wolf separate compression of correlated sources is one of the cornerstones of multiterminal information theory.
Slepian-Wolf random binning

Random binning: form a random code of size $2^{(\epsilon + \mathbb{H}(X|Y))w}$, and randomly partition it into $2^{(\epsilon + \mathbb{H}(X)w)}$ bins of size $2^{(\epsilon + \mathbb{H}(Y)w)}$.

Encoding: assuming $x$ typical, output the index of the bin that contains $x$.

Decoding: the decoder looks into the bin for the $x$ jointly typical with $y$.

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

Encoding: assuming $x$ typical, output the index of the bin that contains $x$. 

Decoding: the decoder looks into the bin for the $x$ jointly typical with $y$. 

For any $\epsilon > 0$ and sufficiently large $n$ the probability of encoding and decoding errors vanishes.
Slepian-Wolf compression is a channel coding problem (with small block error rate).

S-W is as hard as achieving channel capacity for sufficiently large $u$.

For any $\epsilon > 0$ there exist matrices $H^\dagger$ such that $P_e < \epsilon$.

If $m = n > H(X|Y)$, for any $\epsilon > 0$ there exist matrices $H^\dagger$ such that $P_e < \epsilon$.

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

Assume $X = \mathcal{F}_q$.

S-W operating at $(X|X)H = H^\dagger$ can be implemented optimally by linear block codes (Wyner scheme).

Let $H^\dagger_x = \mathcal{F}_q$ and sends it to $B$.

$B$ has $Y$ and $s_1$ and reconstructs $x$ by ML decoding:

$\max_{x:Y|x=H^\dagger_x} P = \mathcal{X}$ and sends $s_1$ to $B$.

Assume $s_1$ operating at $H^\dagger_x = \mathcal{F}_q$.

Implement optimally by linear block codes (Wyner scheme).

$(X|X)H = H^\dagger$ can be implemented optimally by linear block codes (Wyner scheme).
Applications: bidirectional data-exchange

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

---

Applications: bidirectional data-exchange
Bidirectional data exchange rate region

Proposition:
\[ R_1 \] \[ H(Y|X) \]

Achievability: use Slepian-Wolf in both directions.

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

Reverse direction.
The Slepian-Wolf and data exchange regions

Achievable by bidirectional data exchange

\[ H(Y_j | X) \]
Scenario: Multicast over heterogeneous networks

A is a server that sends a source file x to a client B via an unreliable high-throughput forward link (e.g., DVB-S, DVB-T, DAB). After demodulation/decoding, B has y = x + e, a corrupted version of x.

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

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

A and B are also connected by a bidirectional reliable data link (e.g., GPRS, UMTS, Wired Internet).
Example: BEC “correlation” (one-directional)

\[ d = (X|X)H \]

Suppose that \( X, Y \in \mathcal{F}_n \) are linked by a BEC with erasure probability \( p \).

Using a Fountain Code, \( A \) produces \( m > np \) parity bits \( s^1 \) and sends them to \( B \) via the reliable link.

With high probability, \( B \) is able to decode \( X \) since it has received \( u(d - 1) \) symbols via \( Y \) and additional \( du < m \) symbols via \( s^1 \).

Communication takes place at rate slightly above \( H(X|Y) = p \) in direction \( A \rightarrow B \) and zero in the reverse direction.

\( d \equiv X \) and zero in the reverse direction.

Problem: very heavy demand on the performance of the Fountain Code.

\[ u(d - 1) \approx \]

\[ d u < m \]

and focus on \( A \rightarrow B \).
In the BEC “correlation” case, we achieve the S-W rates in both directions. The bidirectional link affords significant simplification. Encoding the erasure pattern using standard data compression and sends it to B. 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 takes place at rate $H(Y|X) = h(p)$ in direction B! A and $H(X|Y) = p$ in direction A! For data exchange and avoid complicated coding and decoding.

Example: BEC “correlation” (bi-directional)
Low-complexity bi-directional CLID algorithm.

Step 1. A sends $s_1 = Hx$ to B, and B sends $s_2 = Hy$ to A.

Step 2. Both terminals have $s^r$, they run locally exact copies of the BP decoder and use CLID over the bidirectional link.

$A - x \rightarrow H = z^r s - st \leftarrow s$.

At each CLID round, $p$ symbols of $x$ are exchanged in direction A to B, and $p$ symbols of $y$ in the same positions are exchanged in direction B to A.

Both terminals can compute the value of $e$ in the doped positions. A, such that both terminals can compute the value of $e$ in the doped positions.
This scheme achieves rate of the scheme.}

\[ R = \frac{\log(1 + P_0)}{\log(1 + P)} \]

result of channel decoding with sufficient interleaving (e.g., multicast of compressed video).

Remark: this is the case where \( x \) is a compressed source and \( y \) is the

In general, \( \max_{x, y} H(x|y) \) is the non-redundant and \( x = y \) is i.i.d.

unless and are \( x \) and \( y \) non-redundant and \( x = y \) is i.i.d.

Remark: this is the case where \( x \) is a compressed source and \( y \) is the

In general, \( \max_{x, y} H(x|y) \) is the non-redundant and \( x = y \) is i.i.d.

unless and are \( x \) and \( y \) non-redundant and \( x = y \) is i.i.d.

Remark: this is the case where \( x \) is a compressed source and \( y \) is the

In general, \( \max_{x, y} H(x|y) \) is the non-redundant and \( x = y \) is i.i.d.

unless and are \( x \) and \( y \) non-redundant and \( x = y \) is i.i.d.

Remark: this is the case where \( x \) is a compressed source and \( y \) is the

In general, \( \max_{x, y} H(x|y) \) is the non-redundant and \( x = y \) is i.i.d.

unless and are \( x \) and \( y \) non-redundant and \( x = y \) is i.i.d.

Remark: this is the case where \( x \) is a compressed source and \( y \) is the

In general, \( \max_{x, y} H(x|y) \) is the non-redundant and \( x = y \) is i.i.d.

unless and are \( x \) and \( y \) non-redundant and \( x = y \) is i.i.d.

Remark: this is the case where \( x \) is a compressed source and \( y \) is the

In general, \( \max_{x, y} H(x|y) \) is the non-redundant and \( x = y \) is i.i.d.

unless and are \( x \) and \( y \) non-redundant and \( x = y \) is i.i.d.
Let $x$ be a binary i.i.d. uniform vector and $y = x + e$, with $e$ i.i.d. Bernoulli.

At each CLID round, a block of $p$ CLID bits are sent. We considered the redundancy w.r.t. the S-W limit. The compression coding rate is 0.0583, corresponding to 28% redundancy w.r.t. the S-W limit.

\[
\begin{align*}
8 &\cdot 1,5x_{107} + 0.5x_{108} = (x)^d \\
1 &\cdot 2x_{131} + 0.3x_{178} + 0.3x_{30} + 0.3x_{9} + 0.0386x_{30} + 0.0005
\end{align*}
\]

Linear programming:

We use an irregular LDPC ensemble with block length $u = 60000$ and degree sequences optimized using approximated density evolution and we used an irregular LDPC ensemble with block length $u = 60000$ and degree sequences optimized using approximated density evolution and

\[
\begin{align*}
\ell &\cdot 0.005, \text{ yielding } d = 0.2957x_{31} + 0.3178x_{30} + 0.3479x_{2} + 0.0005
\end{align*}
\]

Let $x$ be a binary i.i.d. uniform vector and $y = x + e$, with $e$ i.i.d. Bernoulli.

Numerical experiment

\[
\begin{align*}
\text{cases } 1 &\cdot 10, 70, 100, 200.
\end{align*}
\]
Algorithm I, $H(X_i - Y_i) = 0.0454$, block length 60000
Rate cumulative distribution (detail)
Histogram of the number of protocol CLID rounds, $d = 1$.

Algorithm 1, ensemble 1, $d = 1$. $0 \leq p \leq 1$. 

$\Rightarrow$
Histogram of the number of protocol CLID rounds,

Algorithm I, ensemble 1, 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 $y_1^{k-1}$.

Burnashev’s exponent:

$$E(f(R)) = D\left(1 - \frac{R}{C}\right)$$

where $C$ is the channel capacity and $D$ is:

$$D(P_{Y|X}=a, P_{Y|X}=b) = -\frac{1}{2} \log \left( \frac{1}{2} \right)$$

Applications: coding with limited feedback
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 $q$ and $r$ are chosen to achieve $\Delta$. 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.

- If "deny" is decoded, the whole block is re-transmitted.
feedback symbols over forward channel symbols.

A problem of practical interest: devise schemes that attain both high reliability and low complexity for a given fixed feedback rate $R_f$ (ratio of feedback symbols over forward channel symbols).

- Essentially based on ACK/NACK: performance dominated by the ability of the decoder to detect its error status.
- Hybrid-ARQ
- ARQ

Practical (limited) feedback schemes
Proposed scheme

Serial concatenation of two LDPC codes: outer coding $x^1$, inner $x^2$

Caire, Shamai, Verdu 2005.

Neither knows $e^1$. The encoder knows $x^1$ and $z^1$ and the decoder knows $x^1$ and $z^1$, but

$z^1 \neq 0$, $z^1$ is sent over the feedback link. If $z^1 = 0$, the receiver declares the subvector of $x^1$ as the output. If $e^1 \sim \text{Bernoulli}(d)$,

Main feedback: the decoder computes the syndrome $H z^1 = H x^1$. If $z^1 = 0$, the receiver declares the subvector of $x^1$ as the output. If $z^1 \neq 0$, $z^1$ is sent over the feedback link.

Step 1: upon reception of the channel output $y$, the decoder runs BP over the joint Tanner graph and obtains $\hat{x}^1$, such that, assuming interleaving,

Coding $x^1 \leftarrow x^2$. Caire, Shamai, Verdu 2005.
Step 2: Both encoder and decoder run BP simultaneously on the Tanner graph of the outer code with syndrome equal to $z_1$, and variable nodes associated to the elements of $e_1$.

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 $\ell$ be the position of the least reliable symbol at a given doping iteration, $\ell \in \mathbb{Z}_1$.

Both encoder and decoder have the same noisy observation of $e_1$ and

\[ J_{\ell}(z_1) \log J_{\ell}(z_1) \]

can update their BP algorithm by incorporating the additional message and feeds it back to the encoder via the noiseless feedback link.

$J_{\ell}(x_1) - J_{\ell}(x_1) = J_{\ell}(x_1)$ and $J_{\ell}(e_1) = J_{\ell}(x_1)$.

The decoder computes the MAP estimates through the forward channel which outputs then the encoder sends $x_1$ through the forward channel which originates.

Let $\ell$ be the position of the least reliable symbol at a given doping iteration, $\ell \in \mathbb{Z}_1$.

\[ J_{\ell}(z_1) \log J_{\ell}(z_1) \]

for the $\ell$-th variable node.

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”.

Both encoder and decoder have the same noisy observation of $e_1$ and

\[ J_{\ell}(z_1) \log J_{\ell}(z_1) \]

can update their BP algorithm by incorporating the additional message and feeds it back to the encoder via the noiseless feedback link.

$J_{\ell}(x_1) - J_{\ell}(x_1) = J_{\ell}(x_1)$ and $J_{\ell}(e_1) = J_{\ell}(x_1)$.

The decoder computes the MAP estimates through the forward channel which outputs then the encoder sends $x_1$ through the forward channel which originates.

Let $\ell$ be the position of the least reliable symbol at a given doping iteration, $\ell \in \mathbb{Z}_1$.

\[ J_{\ell}(z_1) \log J_{\ell}(z_1) \]

for the $\ell$-th variable node.
The BP algorithms at the encoder and decoder remain synchronized.

The decoding process stops at an iteration when its guess satisfies the parity-check equations, i.e.,

\[ z_1^1 = H^1 \]
Design options

The outer code must be optimized for a BSC with probability of error $p$. This choice yields $r_1 < 1 - h(p)$, with some gap-to-capacity $\epsilon > 0$.

Hence, we get $r_2 = 1$ and $r_1 = C$ with some gap-to-capacity $\epsilon > 0$. Choosing a trivial inner code and an outer code optimized for the forward channel, we get $r_2 = 1$ and $r_1 = C$ with some gap-to-capacity $\epsilon > 0$. This choice yields $R_f = \frac{C}{p} + \frac{p}{1 - p}$. Letting $u$ be small (doping redundancy),

\[
\mathcal{L} + (1 - 1)u \approx \frac{\mathcal{P} + u}{\mathcal{P} + (1 - 1)u} = \mathcal{H}
\]

The expected number of doped bits, the resulting feedback rate is given by

Letting $d = 1 + \frac{1}{u}$, the resulting feedback rate is given by

\[
(d) - 1 > 1 - 1
\]

Hence, the outer code must be optimized for a BSC with probability of error $p$.
Choosing an inner code optimized for the forward channel yields

\[ u + (\varepsilon + (d)\eta)(\varepsilon - \mathcal{C}) = f_R \]

which can be made as small as desired.

This yields

\[ \varepsilon \approx \varepsilon - (d)\eta - \mathcal{C} = \varepsilon - \varepsilon - \mathcal{C} = \varepsilon - \mathcal{C} \]

Choosing an inner code optimized for the forward channel yields.
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.

**Robust joint source-channel coding**

- **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 source reconstructed sequence.
Option 1: optimal quantization with clever indexing
Option 2: Suboptimal quantization with entropy coding
Option 1: Multistage TCA + standard LDPC coding
Option 2: ECSQ + LDPC coding for data compression
Comparison with successive refinement scalar quantization

Scalar Embedded Lloyd–Max
Vector (2D) Embedded Lloyd–Max
$T_0 = 1/4$ 128 states
Design tradeoffs

Investigated multi-level coding by exploiting the entropy chain rule:

\[
(\mathcal{I}_q \cdot \cdots \cdot \mathcal{I}_q^{w-q}) H \cdots + (\mathcal{I}_q^{w}) H + (\mathcal{I}_q) H = (\eta | H
\]

- Requires careful mapping of M-ary indices onto binary: we have
- Generally successfully refinement.
- Entropy-Constrained Scalar Quantization is much simpler but not
- Reconstruction.
- Soft-reconstruction (softbits) yields just minor improvement w.r.t. hard
- Following from non-catastrophic property of the CC encoder.
- Binary CC yields directly a non-catastrophic binary indexing (directly at high rates.
- Multistage TCQ based on binary CC (typically, rate 1/4; 128 states) yields
- Multistage TCQ based on binary CC (typically, rate 1/4; 128 states) yields

Design tradeoffs

Binary CC yields directly a non-catastrophic binary indexing (directly at high rates.

Multistage TCQ based on binary CC (typically, rate 1/4; 128 states) yields

Binary CC yields directly a non-catastrophic binary indexing (directly at high rates.

Multistage TCQ based on binary CC (typically, rate 1/4; 128 states) yields
Pleasant reminiscence of the "bit-plane" structure of practical source coding schemes (e.g., JPEG2000).

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

Pleasant reminiscence of the "bit-plane" structure of practical source coding schemes (e.g., JPEG2000).
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 JPEG2000 models the bitplanes incorporated into the BP decoder.

- Concatenation of Wavelet transform, quantization, and Markov (context) source 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 inferred after predictive quantization looks like a non-stationary but independent discrete source.
A numerical experiment
Data compression via linear coding $z = Hx$ is theoretically optimal.

- Distributed data compression (Slepian-Wolf) is a channel coding problem; therefore, there is no alternative.
- Data compression via linear coding $z = Hx$ is theoretically optimal.
- Because of complexity, we use sparse $H$ and BP decoding.
- Although competing with state-of-the-art is very hard for the pure data compression, BWT, MDL, probability model estimation.
- Distributed data compression (Slepian-Wolf) is a channel coding problem; therefore, there is no alternative.
- Data compression via linear coding $z = Hx$ is theoretically optimal.

Conclusions
Open problems

1. Using modern (sparse-graph based) techniques for lossy compression. This is a totally different approach, linear maps cannot be optimal for lossy coding.

2. Lossy transmission for a compound/broadcast channel.

Best known performances are obtained by Hybrid analog-digital schemes.

Very important for future DAB/DVB, and MM broadcasting for 3G+.

Preliminary results in special cases: Zecchina, Mézard, et al., building.

Survey Propagation. Preliminary results in special cases: Zecchina, Mézard, et al., building.