Reliable Computation over Multiple-Access Channels

Bobak Nazer Michael Gastpar

Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

September 28, 2005

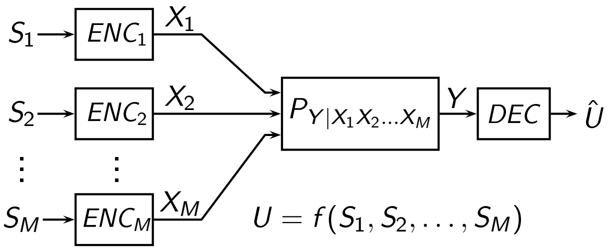
Allerton 2005

Outline

Sensor Networks and The Separation Theorem

- Typical sensor network problem: collect data at sensors and communicate the average to a central node
- Typical solution: distributed compression + multiple-access channel (MAC) protocol
- Motivated by Shannon's Separation Theorem: for point-to-point communication, compression can be separated from reliable communication
- This does *not* apply in general for networks.
- What about sending functions over channels? (ex: sums)

Problem Statement

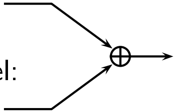


- M users each observe a source
- We want to reliably send a function of the sources, $U = f(S_1, S_2, \ldots, S_M)$ to a receiver, $P(\hat{U} \neq U) \rightarrow 0$
- We measure our performance by the *computation rate*, $\kappa =$ number of functions that get sent per channel use
- Is there a separation theorem for sending functions?

A Summation Channel

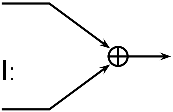
S_1 and S_2 are independent $\mathcal{B}(\frac{1}{2})$ sources.

Want to sent $U = S_1 \oplus S_2$ over this channel:

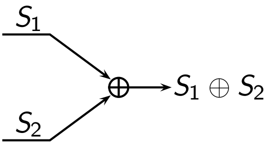


A Summation Channel

S_1 and S_2 are independent $\mathcal{B}(\frac{1}{2})$ sources.
Want to sent $U = S_1 \oplus S_2$ over this channel:



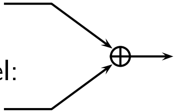
Uncoded



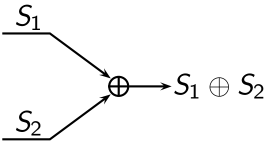
$$\kappa_{\text{COMP}} = 1$$

A Summation Channel

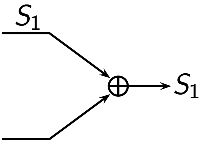
S_1 and S_2 are independent $\mathcal{B}(\frac{1}{2})$ sources.
Want to sent $U = S_1 \oplus S_2$ over this channel:



Uncoded



Separation

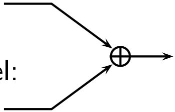


$$\kappa_{\text{COMP}} = 1$$

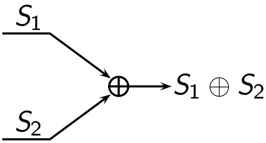
A Summation Channel

S_1 and S_2 are independent $\mathcal{B}(\frac{1}{2})$ sources.

Want to sent $U = S_1 \oplus S_2$ over this channel:

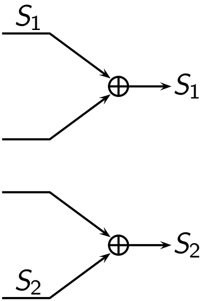


Uncoded



$\kappa_{\text{COMP}} = 1$

Separation

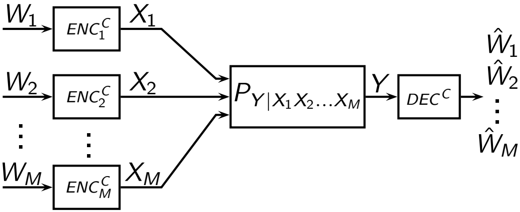


$\kappa_{\text{SEP}} = \frac{1}{2}$

A Summation Channel

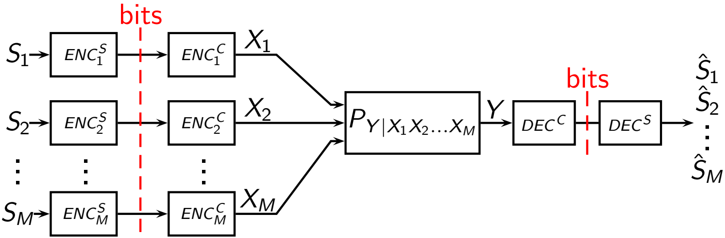
- We want to develop schemes that take advantage of the channel structure.
- Problem: How can we send functions reliably?
- Uncoded transmission will boost our performance at the cost of noise in the received signal.
- Can we get these gains and communicate losslessly?
- First, some background on separation.

Standard Multiple-Access Problem



- M users with independent messages, $W_i \in \{1, \dots, 2^{nR_i}\}$
- Encoders possibly subject to cost constraints
- Capacity region completely known [Ahlswede 71, Liao 72]

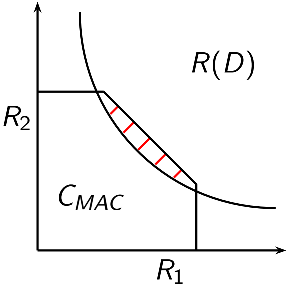
Standard Multiple-Access Problem



- M users with possibly correlated sources, S_i
- Must perfectly recover the sources
- Source encoders do distributed compression

Standard Multiple-Access Problem

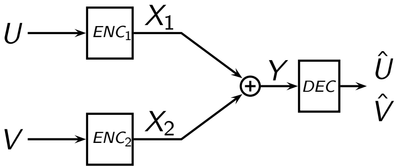
- Correlated sources, want each source perfectly
- Distributed compression rate region was completely characterized by Slepian and Wolf
- Separation: combine Slepian-Wolf source coding with MAC coding
- Is this optimal?



Standard Multiple-Access Problem

$P(U,V)$

V \ U	0	1
0	$\frac{1}{3}$	$\frac{1}{3}$
1	0	$\frac{1}{3}$

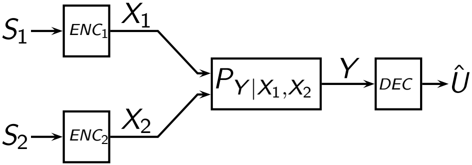


- Example from [Cover-El Gamal-Salehi 80]
- $\mathcal{X}_1, \mathcal{X}_2 = \{0,1\}$, $Y = X_1 + X_2$, $\mathcal{Y} = \{0,1,2\}$
- distributed compression requires $H(U,V) = \log_2 3$ bits $= 1.58$ bits
- $C_{MAC} = 1.5$ bits, so separation fails
- $X_1 = U$, $X_2 = V$ is perfectly decodable
- Separation not optimal for dependent sources

Motivating Example

$P(S_1, S_2)$

$S_2 \backslash S_1$	0	1
0	$\frac{1-p}{2}$	$\frac{p}{2}$
1	$\frac{p}{2}$	$\frac{1-p}{2}$



$$U = S_1 \oplus S_2, \quad H(U) = h_B(p)$$

- Must losslessly transmit the mod-2 sum, $U = S_1 \oplus S_2$, across a MAC
- Sources have uniform marginals, independent when $p = \frac{1}{2}$
- S_2 looks like S_1 passed through a BSC

Naive Source Coding: Slepian-Wolf

- Let's find the optimal separation scheme. Optimize source coding scheme first.
- Slepian-Wolf binning can be used to compress S_1 and S_2.

$$R_1 > h_B(p)$$
$$R_2 > h_B(p)$$
$$R_1 + R_2 > 1 + h_B(p)$$

- If the above constraints are satisfied, the source decoder can losslessly recover S_1 and S_2 and compute U.
- Not optimal! Wastefully sent the sources, S_1 and S_2, while only reconstructing the sum, U.

Optimal Source Coding: Körner-Marton

- Then what is the optimal source code?
- Idea: use a linear source code.
- Any iid $\mathcal{B}(p)$ source can be compressed to the entropy rate, $h_B(p)$, with a linear code
- This code can be written as a matrix $\mathbf{H}$

$$\mathbf{u} = [U(1)U(2)\cdots U(k)]$$
$$\mathbf{w} = \mathbf{u}\mathbf{H}$$

- Example: $[1\ 0] = [1\ 0\ 0]\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$

Optimal Source Coding: Körner-Marton

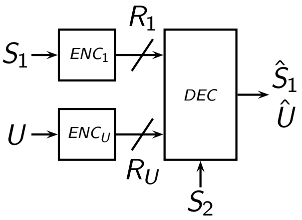
- Now use this source coding matrix at both terminals

$$\mathbf{w}_1 = \mathbf{s}_1\mathbf{H}$$
$$\mathbf{w}_2 = \mathbf{s}_2\mathbf{H}$$

- $\mathbf{w}_1$ and $\mathbf{w}_2$ are provided to the decoder. Neither source can be reconstructed but U comes through perfectly by decoding from the mod-2 sum of the compressed bits $\mathbf{w} = \mathbf{w}_1 \oplus \mathbf{w}_2 = (\mathbf{s}_1 \oplus \mathbf{s}_2)\mathbf{H} = \mathbf{u}\mathbf{H}$
- Körner-Marton scheme achieves:

$$R_1 > h_B(p)$$
$$R_2 > h_B(p)$$

Optimal Source Coding: Körner-Marton

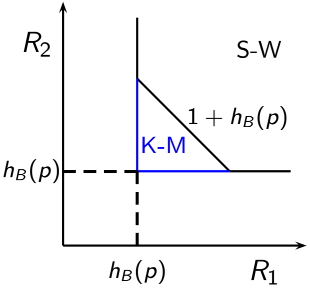


- Converse: S_1 and U must be perfectly reconstructed at the decoder. S_2 is available at the decoder so recovering one variable gives the other for free. A rate of at least $H(U) = h_B(p)$ is needed.

$$R_1 + R_U \geq h_B(p)$$
$$R_2 + R_U \geq h_B(p)$$

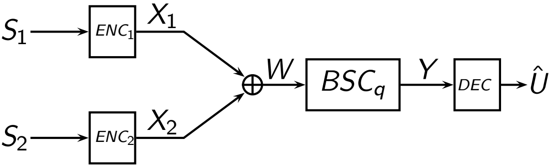
- Set $R_U = 0$.

Optimal Source Coding: Körner-Marton



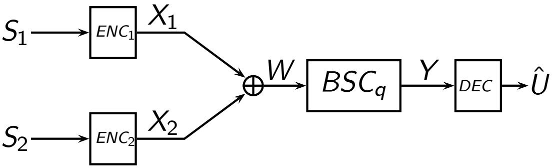
- If S_1 and S_2 are independent, $R_1 \geq 1$, $R_2 \geq 1$ required.
- Independence $\Rightarrow$ Slepian-Wolf and Körner-Marton are equivalent

An Appropriately Matched MAC



- Channel takes a mod-2 sum of the inputs: $Y = X_1 \oplus X_2$
- This is followed by a BSC with crossover probability q
- Sum rate capacity, $C_{MAC} = 1 - h_B(q)$
- Capacity region is a simplex, time-sharing is optimal

Performance Metric Bounds



- Recall performance metric: *computation rate*, $\kappa =$ number of U's (functions) sent per channel symbol
- Lower bound: Send each source individually over the MAC. Gives $\kappa = \frac{1-h_B(q)}{2}$.
- Upper bound: joint encoder. Gives $\kappa = \frac{1-h_B(q)}{h_B(p)}$.
- How well does separation do?

Separation-Based Scheme

- Use Körner-Marton scheme to compress U:

$$R_{S_1} + R_{S_2} > 2h_B(p)$$

- Then use a MAC channel code:

$$R_{X_1} + R_{X_2} < 1 - h_B(q)$$

- Achieves any computation rate satisfying:

$$\kappa_{SEP} < \frac{1 - h_B(q)}{2h_B(p)}$$

- Can we do better?

Linear Channel Code for the BSC

- Random coding does not take advantage of the channel structure
- We need a linear channel code and a linear source code
- For any BSC, there is a linear channel code that can achieve capacity.
- Any linear channel code can be written as a generator matrix: $\mathbf{G}$

$$\mathbf{x} = \mathbf{w}\mathbf{G}$$

- Example: $[1\ 1\ 0] = [1\ 1] \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

Computation Coding

Theorem

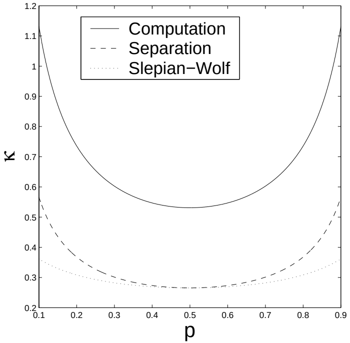
There exists a linear code that can approach the computation rate $\kappa = \frac{1-h_B(q)}{h_B(p)}$. *This is the best available computation rate for lossless transmission of* $U = S_1 \oplus S_2$ *over this channel.*

- *Achievability.* Choose $\mathbf{G}$ for the BSC and $\mathbf{H}$ for compressing U to entropy. Set

$$\mathbf{x}_1 = \mathbf{s}_1\mathbf{H}\mathbf{G}$$
$$\mathbf{x}_2 = \mathbf{s}_2\mathbf{H}\mathbf{G}$$

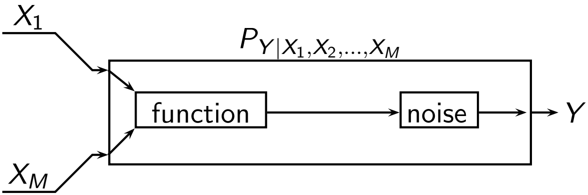
- After the channel, it looks as if U was jointly encoded.
- *Converse.* Relax to joint encoding of U. By the data processing inequality, $I(U;\hat{U}) \leq I(X_1, X_2; Y)$.

Separation vs. Joint Source-Channel Coding



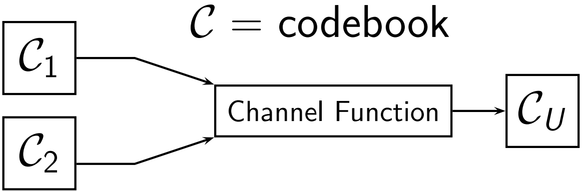
- Our scheme dominates separation by a factor of 2, *even when the sources are independent.*
- We can get the benefits of uncoded transmission and maintain reliable communication.

Computation Codes: Desired Properties



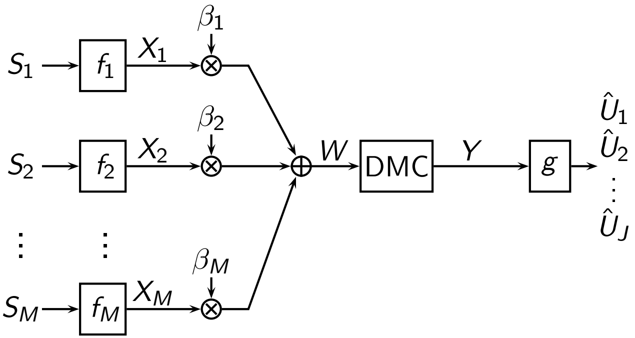
- Idea: we can very efficiently send functions close to the "natural function" of a MAC.
- Channel is a function followed by noise.
- U is the same function of the sources.

Computation Codes: Desired Properties



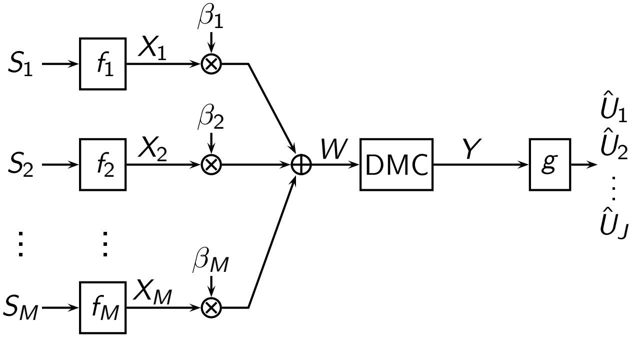
- After the channel function, output looks like a codebook for joint encoding of U for a point-to-point channel.
- Purely random codes do not work. We need structure (ex: linear codes).
- Codebooks may not be valid for recovering individual sources *even before noise.*
- Let's consider channels that sum.

Discrete Linear Multiple-Access Channels



- $S_i \in \mathcal{X}$ where $\mathcal{X}$ is a Galois field, $i \in \{1, \ldots, M\}$
- M block encoders, f_i, and a block decoder, g
- $W = \sum_{i=1}^{M} \beta_i S_i$ where $\beta_i \in \mathcal{X} \setminus \{0\}$

Discrete Linear Multiple-Access Channels



- Would like to losslessly communicate $U_1, U_2, \ldots, U_J$
- $U_j = \sum_{i=1}^{M} \alpha_{ji} S_i \qquad \alpha_{ji} \in \mathcal{X}, \quad j = 1, 2, \ldots, J$
- $U_1, U_2, \ldots, U_J$ can be correlated

Computation Coding for Discrete Linear MACs

Theorem

There is a linear code that can approach the rate $\kappa == \frac{C_{MAC}}{H(U_1, U_2, \ldots, U_J)}$. This is the maximum computation rate for lossless transmission of $U_1, U_2, \ldots, U_J$ over a symmetric discrete linear MAC.

Converse. Relax to joint encoding of $\mathbf{U} = (U_1, U_2, \ldots, U_J)$. By the data processing inequality, $I(\mathbf{U}; \hat{\mathbf{U}}) \leq I(X_1, X_2, \ldots, X_M; Y)$.

$$\min_{p(\hat{\mathbf{U}}|\mathbf{U})} I(\mathbf{U}; \hat{\mathbf{U}}) = H(U_1, U_2, \ldots, U_J)$$

.

Linear Source and Channel Coding

- Any source can be compressed to its entropy rate with a linear code over a Galois field.
- This linear code can be written as a matrix $\mathbf{H}$
- For any symmetric DMC whose input alphabet is a Galois field, there is a linear code that achieves capacity.
- This linear code can be written as a generator matrix $\mathbf{G}$

Computation Coding: Achievability

- Coding scheme

$$\mathbf{A_i} = \begin{bmatrix} \alpha_{1i}\mathbb{I}^{k_1\times k_1} & & & \mathbf{0} \\ & \alpha_{2i}\mathbb{I}^{k_2\times k_2} & & \\ & & \ddots & \\ \mathbf{0} & & & \alpha_{Ji}\mathbb{I}^{k_J\times k_J} \end{bmatrix}$$

- Choose $\mathbf{H}$ for joint compression of $(U_1, U_2, \ldots, U_J)$
- Choose $\mathbf{G}$ for achieving capacity over the symmetric DMC
- Set $\mathbf{B_i} = \beta_i^{-1}\mathbb{I}^{n\times n}$
- $\mathbf{x_i} = \mathbf{A_i s_i H G B_i}$
- $\mathbf{s_i} = [(S[1]S[2]\cdots S[k])]$

Computation Coding: Achievability

Corollary

The rate $\kappa = \frac{I(W;Y)}{H(U_1,U_2,\ldots,U_J)}$ *where* $p(W)$ *is uniform is achievable over any discrete linear MAC.*

- Linear codes result in a uniform input distribution to the channel.
- Completely linear codes can get us very high (and sometimes optimal) performance. How does separation do?

Separation over Discrete Linear MACs

- Körner-Marton scheme does not generalize
- *Example*: Let S_1 and S_2 be independent random variables on GF(3) with mod-3 sum $U = S_1 \oplus S_2$. Their pdfs are given by the following table:

V	$P(S_1 = V)$	$P(S_2 = V)$	$P(U = V)$
0	0.5	0	0.25
1	0.5	0.5	0.25
2	0	0.5	0.5

- $H(S_1) = H(S_2) = 1$ and $H(U) = 1.5$
- In this case, sending sources makes more sense
- No converse though

Rate Gains

- We can't give the separation scheme in general.
- Still for most situations of interest our scheme achieves a computation rate, κ, M times larger than separation-based schemes.
- Our rate gain is proportional to the number of users.
- This suggests that even in mismatched cases, large gains are possible.
- Cost of maintaining synchronization is worthwhile.

Unmatched Channels

- Ideally, want to optimally transmit any function over any channel. This is hard.
- *Example*: S_1 and S_2 independent $\mathcal{B}(\frac{1}{2})$ processes. Want to send $U = S_1 \oplus S_2$. Channel is a real addition, $W = S_1 + S_2$, followed by symmetric noise.

$$P_{Y|W} = \begin{pmatrix} 1-2\epsilon & \epsilon & \epsilon \\ \epsilon & 1-2\epsilon & \epsilon \\ \epsilon & \epsilon & 1-2\epsilon \end{pmatrix}$$

- $\epsilon = 0.1$ gives $\kappa_{SEP} = 0.3104$.
- Interpreting 2 as 0 at the channel output, we can achieve $\kappa_{COMP} = 0.3973$.

Future Work

- Linear codes for sending sums over the Gaussian MAC
- Bounds for sending mismatched functions over channels

Conclusions

- Large gains are possible for reliable computation with joint source-channel codes.
- Very simple codes, such as linear codes can completely achieve these gains.
- Even mismatched functions can benefit.

Acknowledgements

We would like to thank Anand Sarwate and Krish Eswaran for helpful discussions.