Integer-Forcing Source Coding

Or Ordentlich (MIT)
Joint work with Uri Erez (TAU)

July 6th, 2016
Wireless and Number Theory Workshop
York, England

The Communication Problem

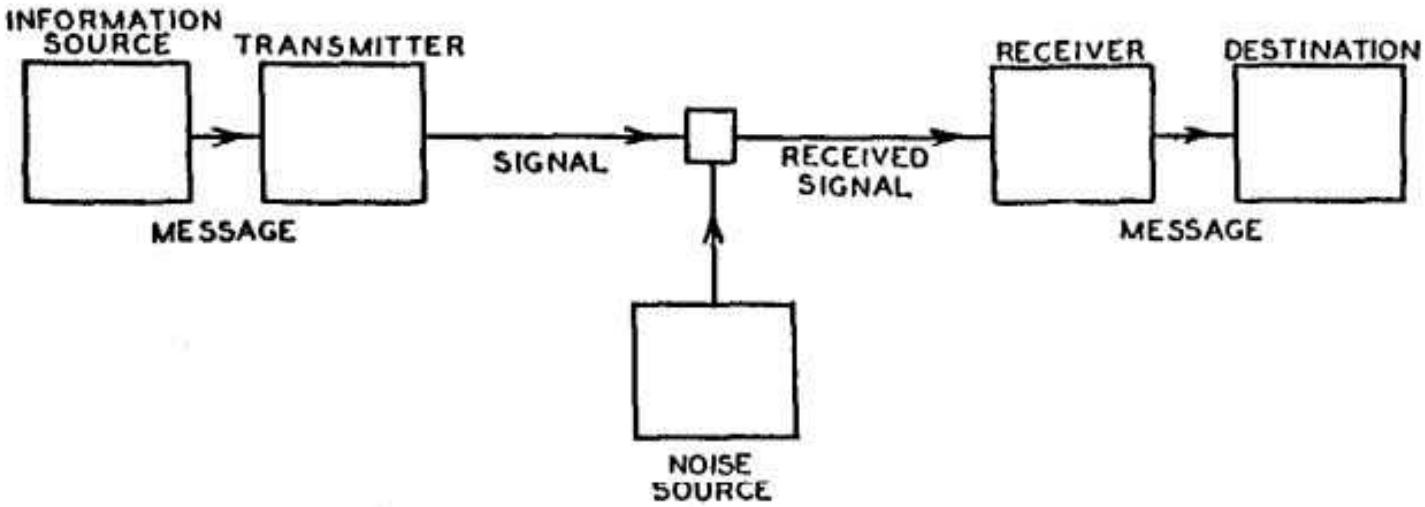


Fig. 1—Schematic diagram of a general communication system.

The Communication Problem

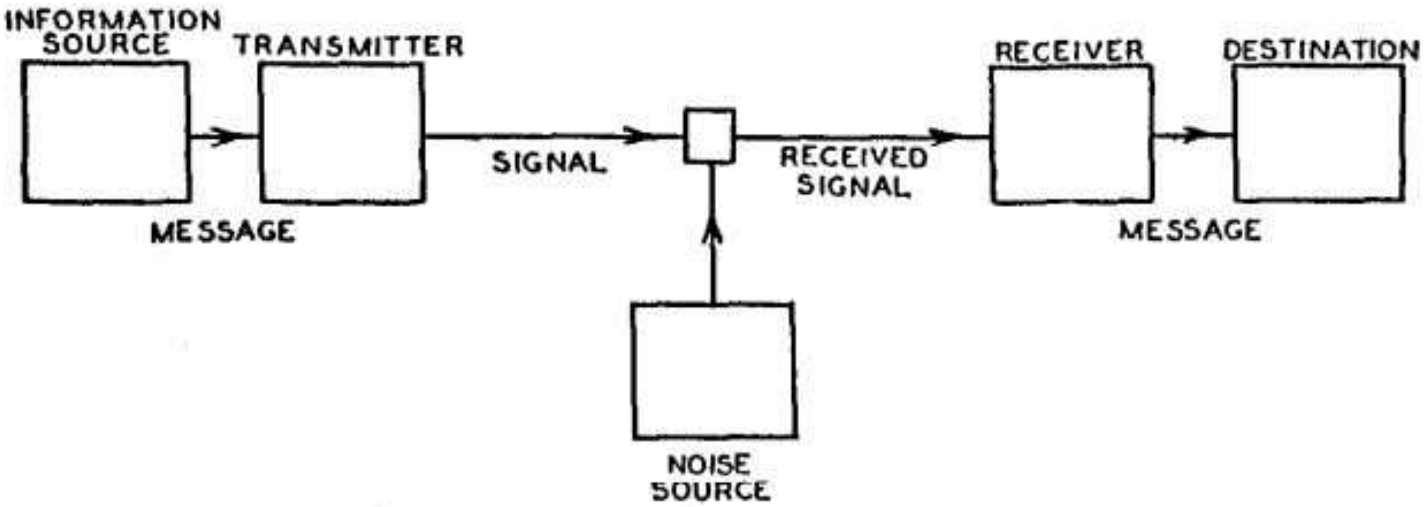


Fig. 1—Schematic diagram of a general communication system.

Source coding: representing an information source with minimum # bits

Channel coding: sending i.i.d. Bernoulli(1/2) bits over a noisy channel

The Communication Problem

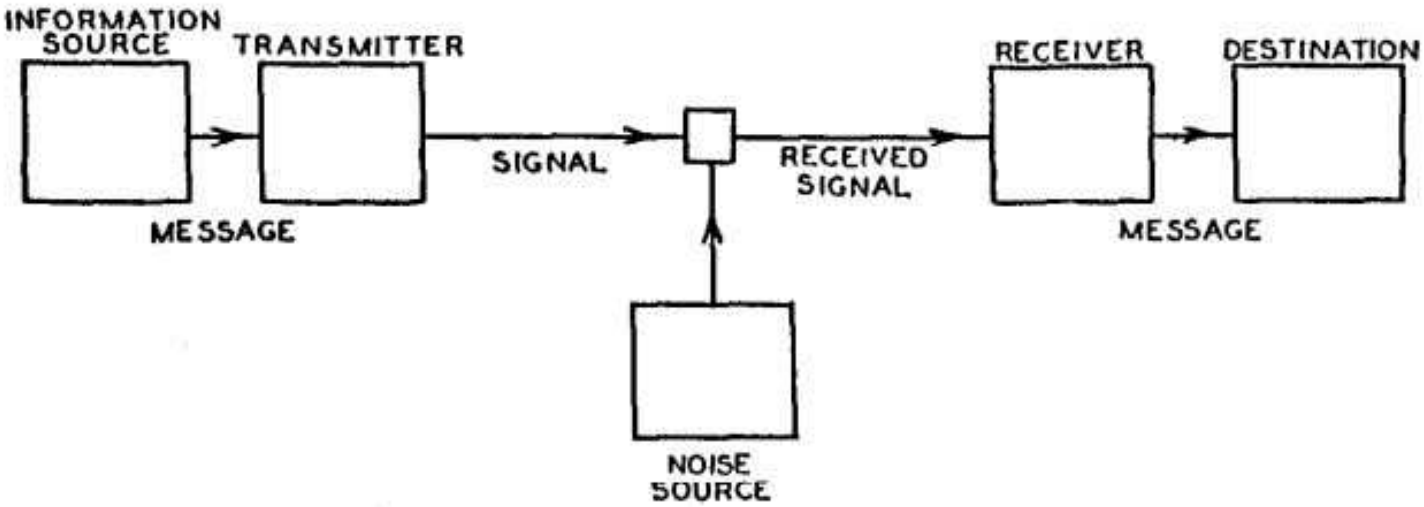


Fig. 1—Schematic diagram of a general communication system.

Source coding: representing an information source with minimum # bits
Channel coding: sending i.i.d. Bernoulli(1/2) bits over a noisy channel

Shannon

For i.i.d. information source and discrete memoryless channels there is no loss (asymptotically) in solving the source and channel coding separately

The Communication Problem

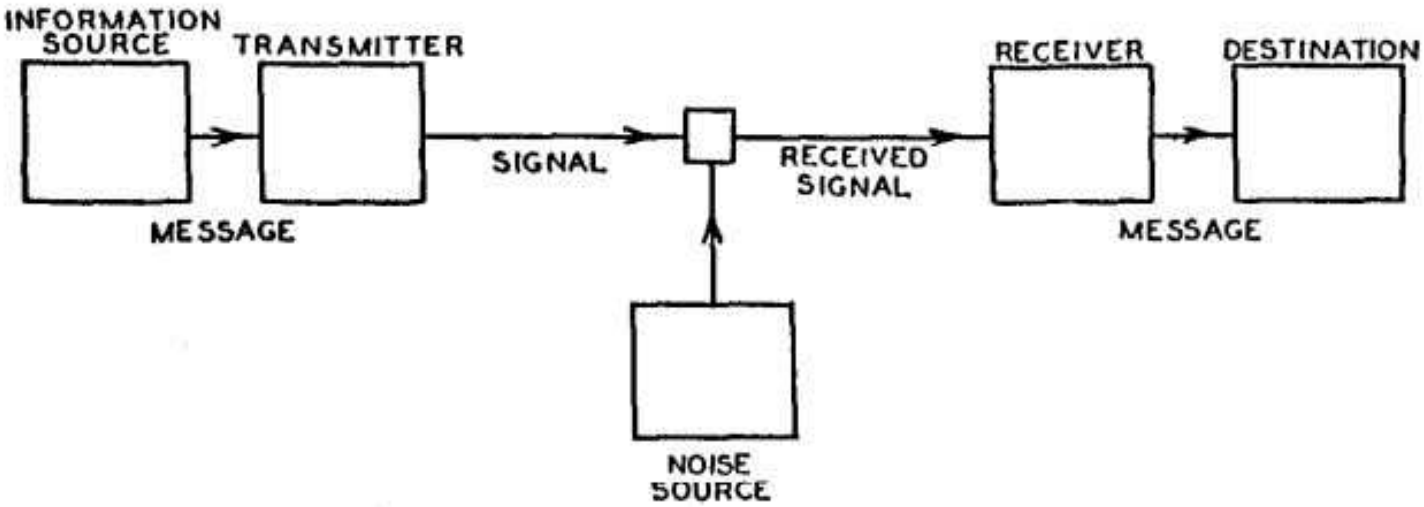


Fig. 1—Schematic diagram of a general communication system.

Source coding: representing an information source with minimum # bits

Channel coding: sending i.i.d. Bernoulli(1/2) bits over a noisy channel

Most talks in the workshop deal with channel coding

This talk is about source coding

The Communication Problem

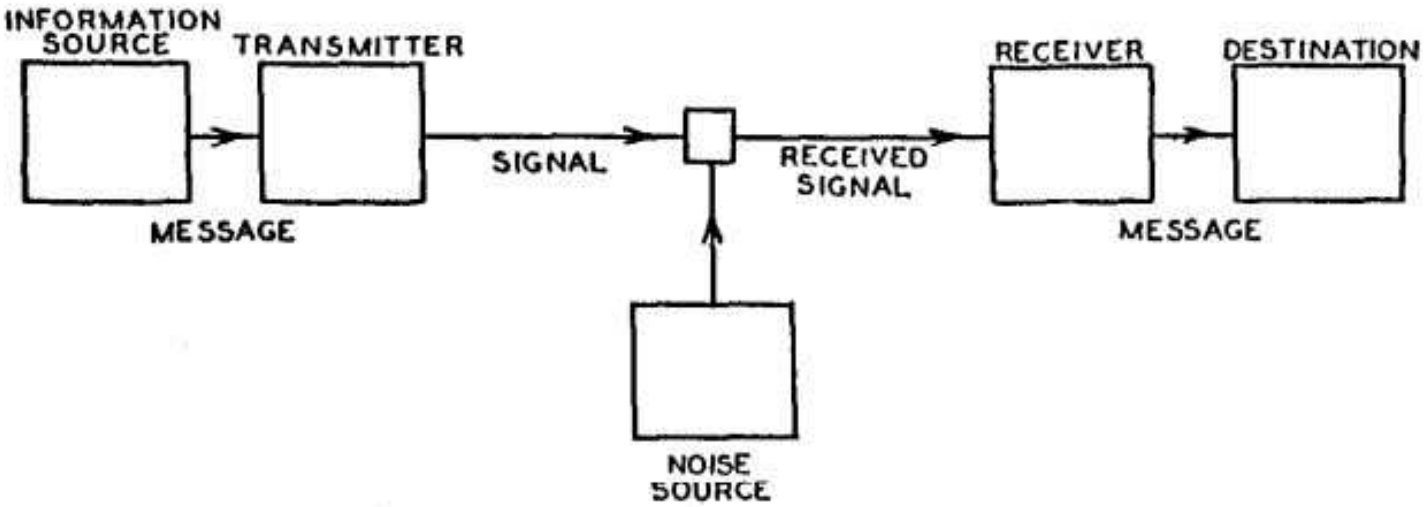


Fig. 1—Schematic diagram of a general communication system.

Source coding: representing an information source with minimum # bits

Channel coding: sending i.i.d. Bernoulli(1/2) bits over a noisy channel

watch out!

Channel coding: large rate is good

Source coding: small rate is good

Lossy Compression

Many information sources of interest are analog in nature

audio, video, pictures, EM waves

Lossy Compression

Many information sources of interest are analog in nature
audio, video, pictures, EM waves

There are many good reasons to represent them digitally
Resilience to noise/aging, cheap storage, fast access, efficient processing....

Lossy Compression

Many information sources of interest are analog in nature
audio, video, pictures, EM waves

There are many good reasons to represent them digitally
Resilience to noise/aging, cheap storage, fast access, efficient processing....

Analog-to-Digital Conversion

- Continuous-to-discrete (sampling)
  Can represent any band-limited signal by a discrete sequence (Nyquist's Theorem)
- Quantization of samples to a finite-alphabet (w.l.o.g. bits)
  Storage is finite, so average number of bits/sample is limited

Lossy Compression

Many information sources of interest are analog in nature
audio, video, pictures, EM waves

There are many good reasons to represent them digitally
Resilience to noise/aging, cheap storage, fast access, efficient processing....

Analog-to-Digital Conversion

- Continuous-to-discrete (sampling)
  Can represent any band-limited signal by a discrete sequence (Nyquist's Theorem)
- Quantization of samples to a finite-alphabet (w.l.o.g. bits)
  Storage is finite, so average number of bits/sample is limited

Something is always lost in the conversion from analog-to-digital

Lossy Compression

Many information sources of interest are analog in nature
audio, video, pictures, EM waves

There are many good reasons to represent them digitally
Resilience to noise/aging, cheap storage, fast access, efficient processing....

Analog-to-Digital Conversion

- Continuous-to-discrete (sampling)
  Can represent any band-limited signal by a discrete sequence (Nyquist's Theorem)
- Quantization of samples to a finite-alphabet (w.l.o.g. bits)
  Storage is finite, so average number of bits/sample is limited

Something is always lost in the conversion from analog-to-digital

The more bits we allocate for storing the signal, the smaller the loss

Rate-Distortion Theory

- We can restrict attention to discrete signals
- Let $x[k]$, $k = 1, \ldots, n$, be the discrete-time sampled signal

Rate-Distortion Theory

- We can restrict attention to discrete signals
- Let $x[k]$, $k = 1, \ldots, n$, be the discrete-time sampled signal

Assume further we have nR bits for storing $\mathbf{x} = (x[1], \ldots, x[n])$

Rate-Distortion Theory

- We can restrict attention to discrete signals
- Let $x[k]$, $k = 1, \ldots, n$, be the discrete-time sampled signal

Assume further we have nR bits for storing $\mathbf{x} = (x[1], \ldots, x[n])$

This is done using two functions:

- Encoder: $\mathcal{E} : \mathbb{R}^n \mapsto \{1, \ldots, 2^{nR}\}$
- Decoder: $\mathcal{D} : \{1, \ldots, 2^{nR}\} \mapsto \mathbb{R}^n$

Rate-Distortion Theory

- We can restrict attention to discrete signals
- Let $x[k]$, $k = 1, \ldots, n$, be the discrete-time sampled signal

Assume further we have nR bits for storing $\mathbf{x} = (x[1], \ldots, x[n])$

This is done using two functions:

- Encoder: $\mathcal{E} : \mathbb{R}^n \mapsto \{1, \ldots, 2^{nR}\}$
- Decoder: $\mathcal{D} : \{1, \ldots, 2^{nR}\} \mapsto \mathbb{R}^n$

We would like $\mathbf{x}$ and $\hat{\mathbf{x}} \triangleq \mathcal{D}(\mathcal{E}(\mathbf{x}))$ to be as close as possible

Rate-Distortion Theory

- We can restrict attention to discrete signals
- Let $x[k]$, $k = 1, \ldots, n$, be the discrete-time sampled signal

Assume further we have nR bits for storing $\mathbf{x} = (x[1], \ldots, x[n])$

This is done using two functions:

- Encoder: $\mathcal{E} : \mathbb{R}^n \mapsto \{1, \ldots, 2^{nR}\}$
- Decoder: $\mathcal{D} : \{1, \ldots, 2^{nR}\} \mapsto \mathbb{R}^n$

We would like $\mathbf{x}$ and $\hat{\mathbf{x}} \triangleq \mathcal{D}(\mathcal{E}(\mathbf{x}))$ to be as close as possible

Need to specify distortion metric. We will use squared loss

$$D \triangleq \frac{1}{n} \sum_{k=1}^{n} (x[k] - \hat{x}[k])^2$$

Rate-Distortion Theory

- We can restrict attention to discrete signals
- Let $x[k]$, $k = 1, \ldots, n$, be the discrete-time sampled signal

Assume further we have nR bits for storing $\mathbf{x} = (x[1], \ldots, x[n])$

This is done using two functions:

- Encoder: $\mathcal{E} : \mathbb{R}^n \mapsto \{1, \ldots, 2^{nR}\}$
- Decoder: $\mathcal{D} : \{1, \ldots, 2^{nR}\} \mapsto \mathbb{R}^n$

We would like $\mathbf{x}$ and $\hat{\mathbf{x}} \triangleq \mathcal{D}(\mathcal{E}(\mathbf{x}))$ to be as close as possible

Need to specify distortion metric. We will use squared loss

$$D \triangleq \frac{1}{n} \sum_{k=1}^{n} (x[k] - \hat{x}[k])^2$$

Goal: design $\mathcal{E}$ and $\mathcal{D}$ to minimize D

Rate-Distortion Theory

When designing $\mathcal{E}$ and $\mathcal{D}$ the signal $\mathbf{x}$ is not known

Rate-Distortion Theory

When designing $\mathcal{E}$ and $\mathcal{D}$ the signal $\mathbf{x}$ is not known

Need to assume distribution $\mathbf{x} \sim P$ and design w.r.t. P

Rate-Distortion Theory

When designing $\mathcal{E}$ and $\mathcal{D}$ the signal $\mathbf{x}$ is not known

Need to assume distribution $\mathbf{x} \sim P$ and design w.r.t. P

Goal is to minimize

$$D = \frac{1}{n}\mathbb{E}_{\mathbf{x}\sim P}\sum_{k=1}^{n}\left(x[n] - \hat{x}[n]\right)^2$$

where $\hat{\mathbf{x}} = \mathcal{D}\left(\mathcal{E}\left(\mathbf{x}\right)\right)$.

Rate-Distortion Theory

When designing $\mathcal{E}$ and $\mathcal{D}$ the signal $\mathbf{x}$ is not known

Need to assume distribution $\mathbf{x} \sim P$ and design w.r.t. P

Goal is to minimize

$$D = \frac{1}{n}\mathbb{E}_{\mathbf{x}\sim P}\sum_{k=1}^{n}\left(x[n] - \hat{x}[n]\right)^2$$

where $\hat{\mathbf{x}} = \mathcal{D}\left(\mathcal{E}\left(\mathbf{x}\right)\right)$.

Common assumption: $\mathbf{x}$ is a random vector with i.i.d. entries

Rate-Distortion Theory

When designing $\mathcal{E}$ and $\mathcal{D}$ the signal $\mathbf{x}$ is not known

Need to assume distribution $\mathbf{x} \sim P$ and design w.r.t. P

Goal is to minimize

$$D = \frac{1}{n}\mathbb{E}_{\mathbf{x}\sim P}\sum_{k=1}^{n}\left(x[n] - \hat{x}[n]\right)^2$$

where $\hat{\mathbf{x}} = \mathcal{D}\left(\mathcal{E}\left(\mathbf{x}\right)\right)$.

Common assumption: $x[n] \sim \mathcal{N}(0, \sigma^2)$ i.i.d.

Rate-Distortion Theory

When designing $\mathcal{E}$ and $\mathcal{D}$ the signal $\mathbf{x}$ is not known

Need to assume distribution $\mathbf{x} \sim P$ and design w.r.t. P

Goal is to minimize

$$D = \frac{1}{n}\mathbb{E}_{\mathbf{x}\sim P}\sum_{k=1}^{n}(x[n]-\hat{x}[n])^2$$

where $\hat{\mathbf{x}} = \mathcal{D}(\mathcal{E}(\mathbf{x}))$.

Common assumption: $x[n] \sim \mathcal{N}(0,\sigma^2)$ i.i.d.

Rate-distortion theorem (Shannon)

The minimum number of bits/sample for attaining avg. distortion D is

$$R(D) = \frac{1}{2}\log\left(\frac{\sigma^2}{D}\right)$$

Scalar Uniform Quantization 1

Achieving optimal $R(D)$ requires $n \to \infty$

Scalar Uniform Quantization 1

Achieving optimal $R(D)$ requires $n \to \infty$

In practice, delay and computational complexity are limited

Scalar Uniform Quantization 1

Achieving optimal $R(D)$ requires $n \to \infty$

In practice, delay and computational complexity are limited

Ideally, we would like to work with $n = 1$
This is also the case for Analog-to-Digital Convertors (ADC)

Scalar Uniform Quantization 1

Achieving optimal $R(D)$ requires $n \to \infty$

In practice, delay and computational complexity are limited

Ideally, we would like to work with $n = 1$
This is also the case for Analog-to-Digital Convertors (ADC)

Simplest choice is a uniform scalar quantizer

Scalar Uniform Quantization 1

Achieving optimal $R(D)$ requires $n \to \infty$

In practice, delay and computational complexity are limited

Ideally, we would like to work with $n = 1$
This is also the case for Analog-to-Digital Convertors (ADC)

Simplest choice is a uniform scalar quantizer

$$Q(x) = \begin{cases} (\Delta - 1)/2 & \text{if } \Delta/2 < x \\ \text{round}(x) & \text{if } -\Delta/2 \leq x \leq \Delta/2 \\ -(\Delta - 1)/2 & \text{if } x < -\Delta/2 \end{cases}$$

Scalar Uniform Quantization 1

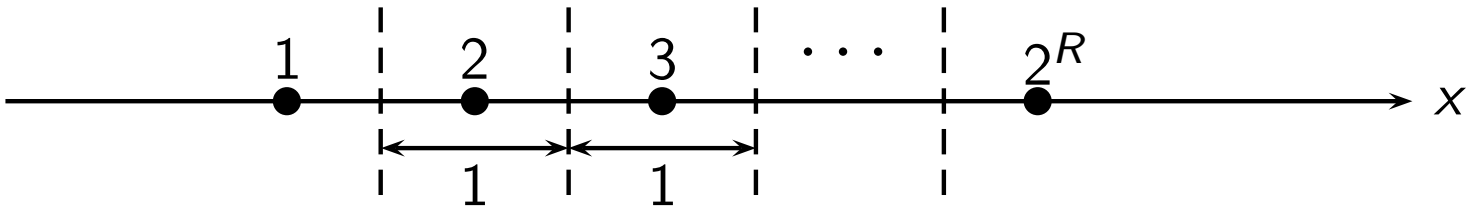
Achieving optimal $R(D)$ requires $n \to \infty$

In practice, delay and computational complexity are limited

Ideally, we would like to work with $n = 1$
This is also the case for Analog-to-Digital Convertors (ADC)

Simplest choice is a uniform scalar quantizer

$Q(x)$:



Scalar Uniform Quantization 1

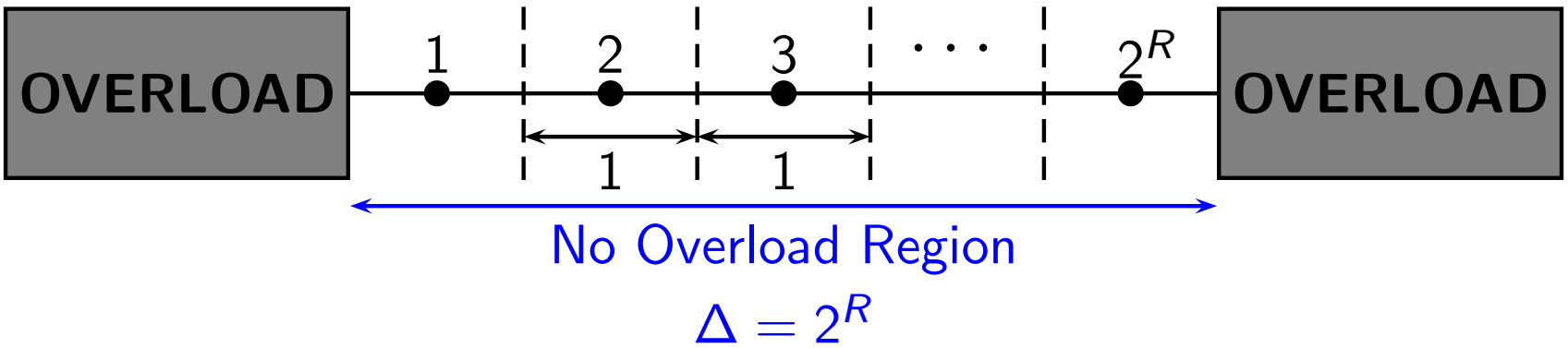
Achieving optimal $R(D)$ requires $n \to \infty$

In practice, delay and computational complexity are limited

Ideally, we would like to work with $n = 1$
This is also the case for Analog-to-Digital Convertors (ADC)

Simplest choice is a uniform scalar quantizer

$Q(x)$:



Scalar Uniform Quantization 1

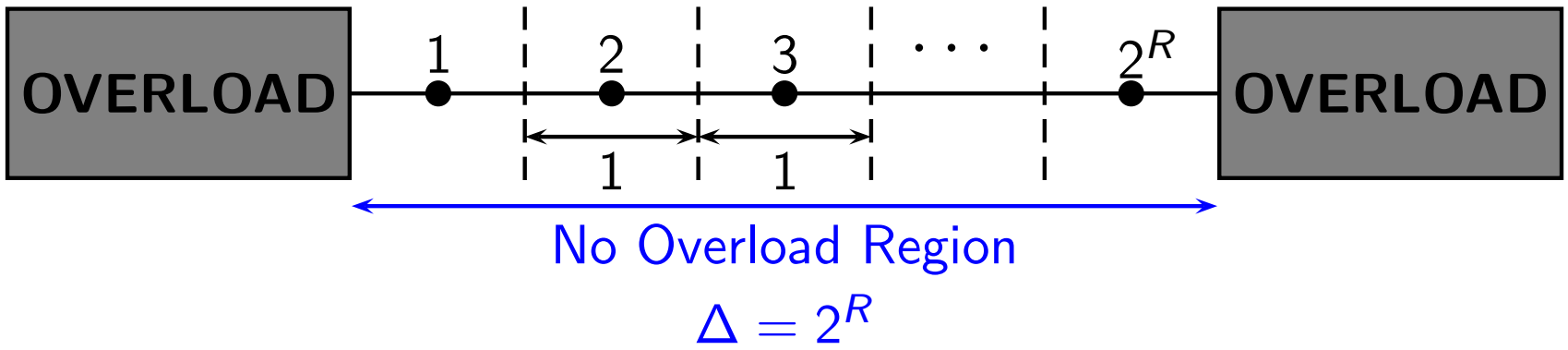
Achieving optimal $R(D)$ requires $n \to \infty$

In practice, delay and computational complexity are limited

Ideally, we would like to work with $n = 1$
This is also the case for Analog-to-Digital Convertors (ADC)

Simplest choice is a uniform scalar quantizer

$Q(x)$:



Dynamic range$= [-\Delta/2, \Delta/2]$
Overload probability$= \Pr(|X| > \Delta/2) = 2\mathrm{erfc}(\Delta/(2\sigma))$

Dithered quantization

The quantization error $e = Q(x) - x$ is a deterministic function of x

Dithered quantization

The quantization error $e = Q(x) - x$ is a deterministic function of x

complicates performance analysis

Dithered quantization

The quantization error $e = Q(x) - x$ is a deterministic function of x

complicates performance analysis

This can be mitigated by adding randomization to the quantizer

Dithered quantization

The quantization error $e = Q(x) - x$ is a deterministic function of x

complicates performance analysis

This can be mitigated by adding randomization to the quantizer

In practice, this randomization is seldom needed

Dithered quantization

The quantization error $e = Q(x) - x$ is a deterministic function of x

complicates performance analysis

This can be mitigated by adding randomization to the quantizer

In practice, this randomization is seldom needed

Let $u \sim \mathrm{Uniform}\left(-\frac{1}{2}, \frac{1}{2}\right)$, $u \perp\!\!\!\perp x$

Dithered quantization

The quantization error $e = Q(x) - x$ is a deterministic function of x

complicates performance analysis

This can be mitigated by adding randomization to the quantizer

In practice, this randomization is seldom needed

Let $u \sim \mathrm{Uniform}\left(-\frac{1}{2}, \frac{1}{2}\right)$, $u \perp\!\!\!\perp x$
Set $\hat{x} = Q(x + u) - u$

Dithered quantization

The quantization error $e = Q(x) - x$ is a deterministic function of x

complicates performance analysis

This can be mitigated by adding randomization to the quantizer

In practice, this randomization is seldom needed

Let $u \sim \mathrm{Uniform}\left(-\frac{1}{2}, \frac{1}{2}\right)$, $u \perp\!\!\!\perp x$
Set $\hat{x} = Q(x + u) - u$

Assuming no overload, we have

$$e = \hat{x} - x = Q(x + u) - (x + u) = \mathrm{round}(x + u) - (x + u)$$

Dithered quantization

The quantization error $e = Q(x) - x$ is a deterministic function of x

complicates performance analysis

This can be mitigated by adding randomization to the quantizer

In practice, this randomization is seldom needed

Let $u \sim \mathrm{Uniform}\left(-\frac{1}{2}, \frac{1}{2}\right)$, $u \perp\!\!\!\perp x$
Set $\hat{x} = Q(x + u) - u$

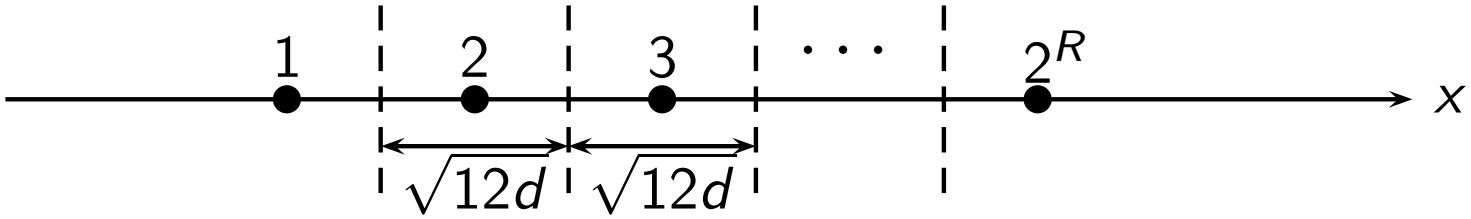
Assuming no overload, we have

$$e = \hat{x} - x = Q(x + u) - (x + u) = \mathrm{round}(x + u) - (x + u)$$

It is easy to see that $\mathrm{round}(x + u) - (x + u)$ is $\mathrm{Uniform}\left(-\frac{1}{2}, \frac{1}{2}\right)$ and $\perp\!\!\!\perp x$

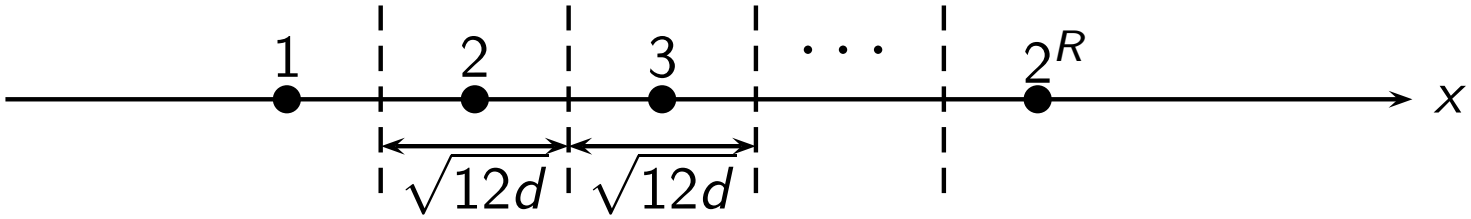
Scalar Uniform Quantization 2

$Q(x)$:

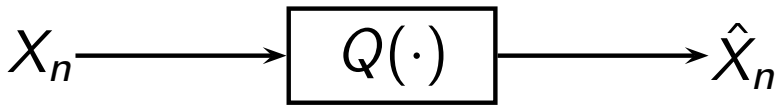


Scalar Uniform Quantization 2

$Q(x)$:

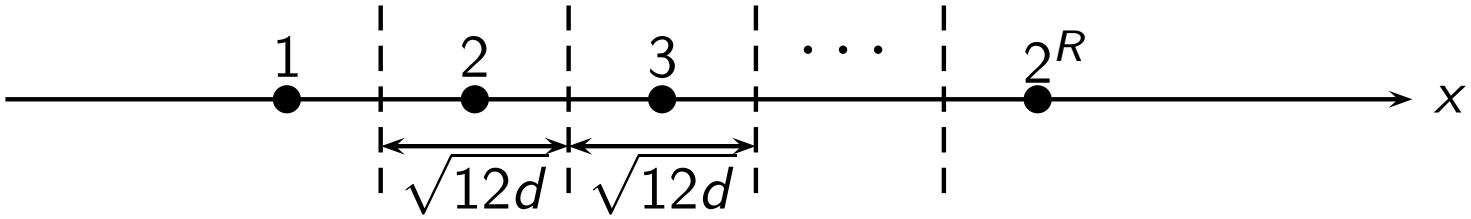


Under no overload + dithered quantization

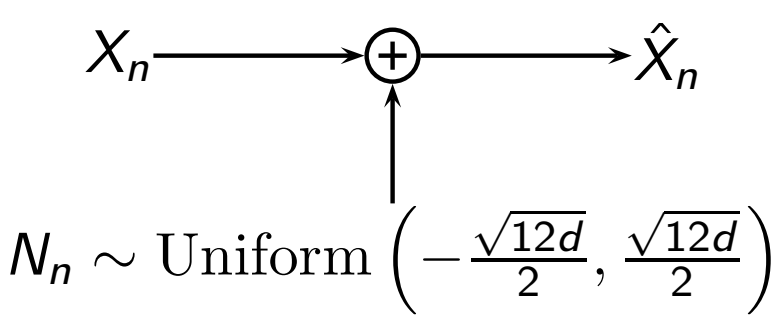


Scalar Uniform Quantization 2

$Q(x)$:



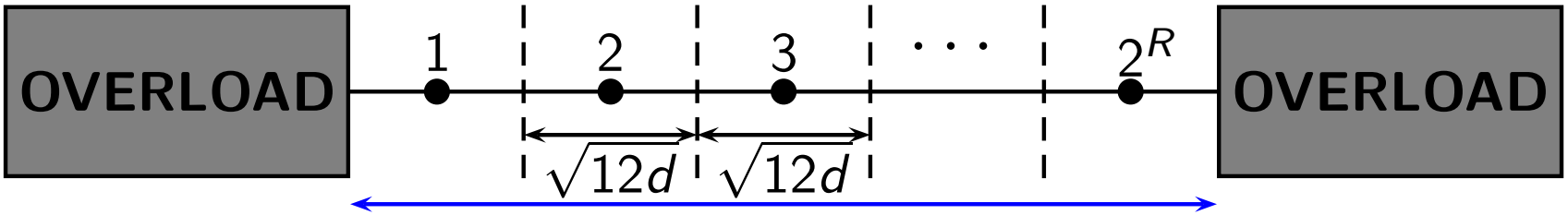
Under no overload + dithered quantization



$$\mathbb{E}\left(\hat{X}_n - X_n\right)^2 = d$$

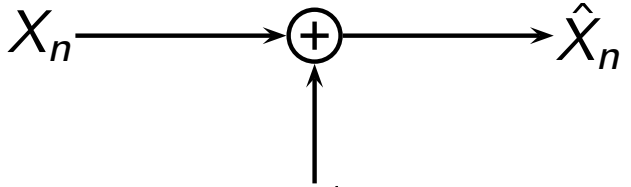
Scalar Uniform Quantization 2

$Q(x)$:



$$\Delta = 2^R \sqrt{12d}$$

Under no overload + dithered quantization

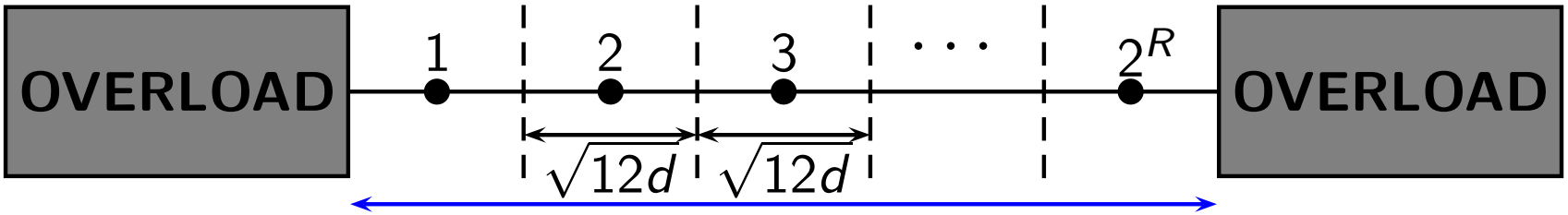


$$\mathbb{E}\left(\hat{X}_n - X_n\right)^2 = d$$

$$\Pr(\text{OL}) = 2\text{erfc}\left(\frac{\Delta}{2\sigma}\right) \leq e^{-\frac{\Delta^2}{8\sigma^2}} = e^{-\frac{3}{2}\cdot 2^{2\left(R-\frac{1}{2}\log\left(\frac{\sigma^2}{d}\right)\right)}}$$

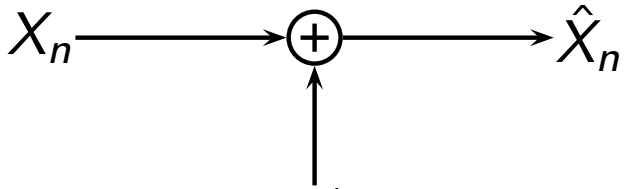
Scalar Uniform Quantization 2

$Q(x)$:



$$\Delta = 2^R \sqrt{12d}$$

Under no overload + dithered quantization



$$\mathbb{E}\left(\hat{X}_n - X_n\right)^2 = d$$

$$N_n \sim \text{Uniform}\left(-\frac{\sqrt{12d}}{2}, \frac{\sqrt{12d}}{2}\right)$$

$$\Pr(\text{OL}) = 2\text{erfc}\left(\frac{\Delta}{2\sigma}\right) \leq e^{-\frac{\Delta^2}{8\sigma^2}} = e^{-\frac{3}{2}\cdot 2^{2\delta}}$$

Scalar Uniform Quantization 2

Conclusion:

For $x \sim \mathcal{N}(0, \sigma^2)$, a scalar uniform quantizer with rate

$$R = \delta + \underbrace{\frac{1}{2} \log \left(\frac{\sigma^2}{d} \right)}_{\text{Shannon}}$$

achieves distortion $\approx d$ whenever overload does not occur, and

$$\Pr(\mathrm{OL}) \leq e^{-\frac{3}{2} 2^{2\delta}}$$

Scalar Uniform Quantization 2

Conclusion:

For $x \sim \mathcal{N}(0, \sigma^2)$, a scalar uniform quantizer with rate

$$R = \delta + \underbrace{\frac{1}{2} \log \left(\frac{\sigma^2}{d} \right)}_{\text{Shannon}}$$

achieves distortion $\approx d$ whenever overload does not occur, and

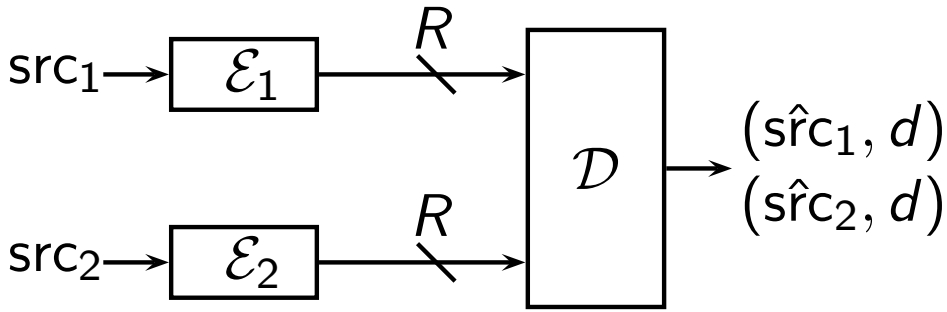
$$\Pr(\mathrm{OL}) \leq e^{-\frac{3}{2} 2^{2\delta}}$$

Main Goal:

Can we get close-to-optimal performance using scalar quantizers also when the source is a Gaussian *vector* $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$?

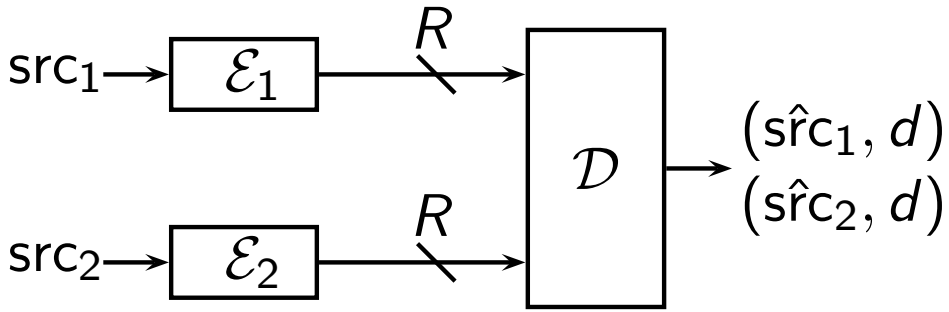
Quantization of Gaussian Vectors

$$\begin{bmatrix} \mathrm{src}_1 \\ \mathrm{src}_2 \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \sigma^2 & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 \end{bmatrix}\right)$$



Quantization of Gaussian Vectors

$$\begin{bmatrix} \text{src}_1 \\ \text{src}_2 \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \sigma^2 & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 \end{bmatrix}\right)$$



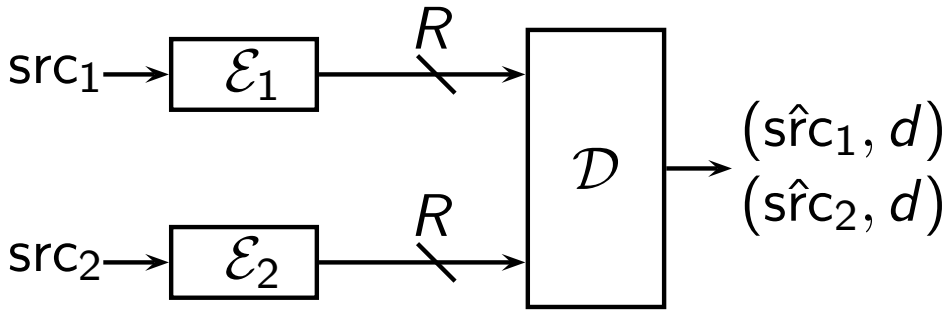
Naive Approach (typically used in practice)

Ignore correlation and use scalar quantizers with $\Delta \propto \sigma$ so that with high probability both $\text{src}_1, \text{src}_2 \in [-\frac{\Delta}{2}, \frac{\Delta}{2}]$

$$\Rightarrow R \propto \frac{1}{2} \log\left(\frac{\sigma^2}{d}\right)$$

Quantization of Gaussian Vectors

$$\begin{bmatrix} \mathrm{src}_1 \\ \mathrm{src}_2 \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}\right)$$



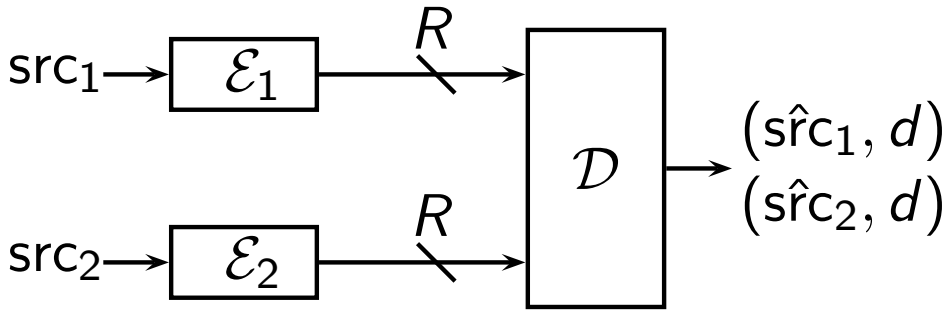
Naive Approach (typically used in practice)

Ignore correlation and use scalar quantizers with $\Delta \propto \sigma$ so that with high probability both $\mathrm{src}_1, \mathrm{src}_2 \in [-\frac{\Delta}{2}, \frac{\Delta}{2}]$

$$\Rightarrow R \propto \frac{1}{2} \log\left(\frac{\sigma^2}{d}\right)$$

Quantization of Gaussian Vectors

$$\begin{bmatrix} \text{src}_1 \\ \text{src}_2 \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \sigma^2 & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 \end{bmatrix}\right)$$



Naive Approach (typically used in practice)

Ignore correlation and use scalar quantizers with $\Delta \propto \sigma$ so that with high probability both $\text{src}_1, \text{src}_2 \in [-\frac{\Delta}{2}, \frac{\Delta}{2}]$

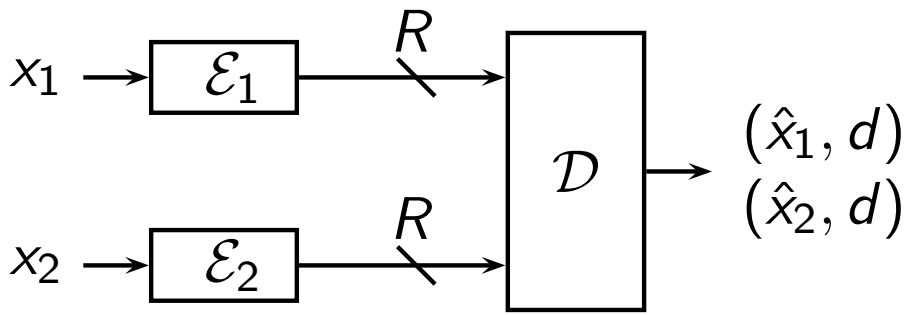
$$\Rightarrow R \propto \frac{1}{2} \log\left(\frac{\sigma^2}{d}\right)$$

Fundamental IT Limits (Berger-Tung, Wagner et al)

With unlimited delay and computational complexity, it is possible to achieve* $R \approx \frac{1}{2} \cdot \frac{1}{2} \log\det\left(\mathbf{I} + \frac{1}{d} K_{xx}\right)$

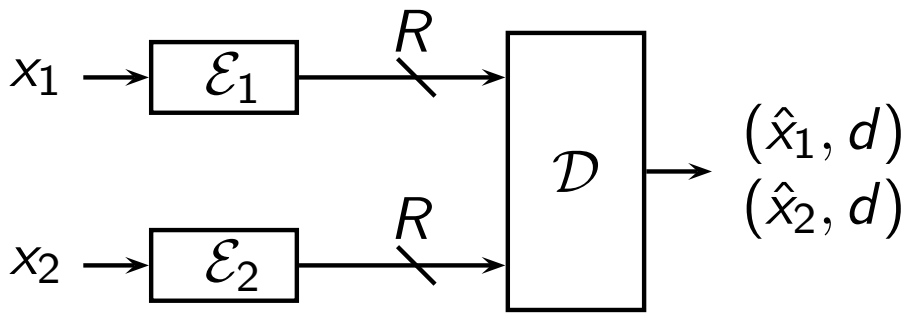
Goal 1 - Universal Quantization

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$$



Goal 1 - Universal Quantization

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$$

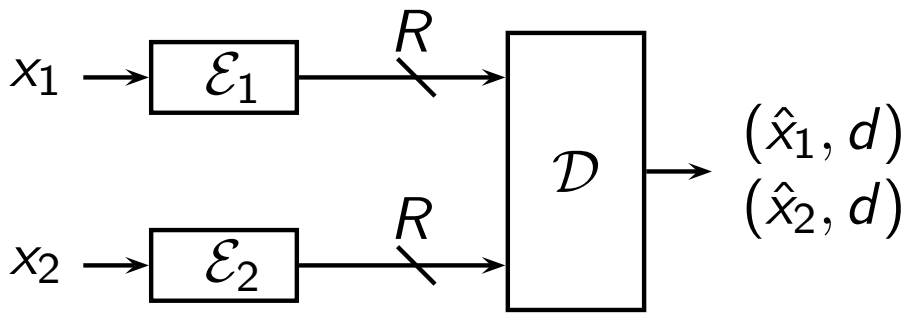


Goal:

- Simple, identical, universal, non-cooperating quantizers $\mathcal{E}_1, \mathcal{E}_2$
- Simple decoder $\mathcal{D}$ that can depend on $\mathbf{K}_{xx}$
- Good performance for all $\mathbf{K}_{xx}$ with the same $\log\det\left(\mathbf{I} + \frac{1}{d}\mathbf{K}_{xx}\right)$

Goal 1 - Universal Quantization

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$$



Goal:

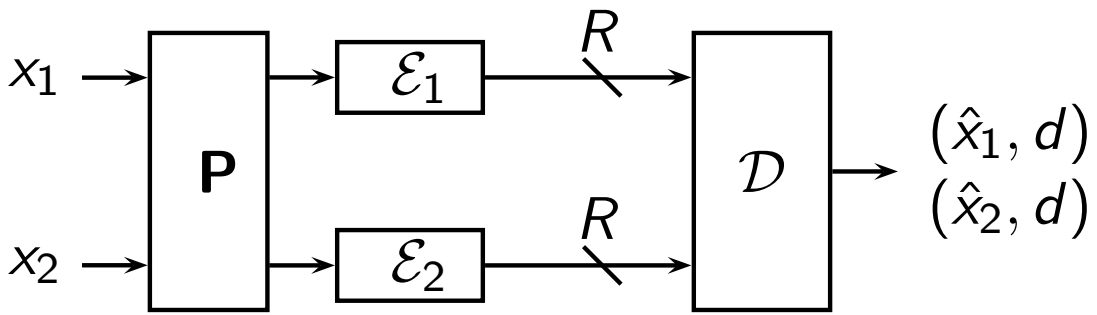
- Simple, identical, universal, non-cooperating quantizers $\mathcal{E}_1, \mathcal{E}_2$
- Simple decoder $\mathcal{D}$ that can depend on $\mathbf{K}_{xx}$
- Good performance for all $\mathbf{K}_{xx}$ with the same $\log\det\left(\mathbf{I} + \frac{1}{d}\mathbf{K}_{xx}\right)$

Extreme cases:

$$\mathbf{K}_{xx}^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{K}_{xx}^2 = \begin{bmatrix} a & 0 \\ 0 & 0 \end{bmatrix}, \text{ and } \mathbf{K}_{xx}^3 = \begin{bmatrix} b & b \\ b & b \end{bmatrix}$$

Goal 1 - Universal Quantization

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$$



Goal:

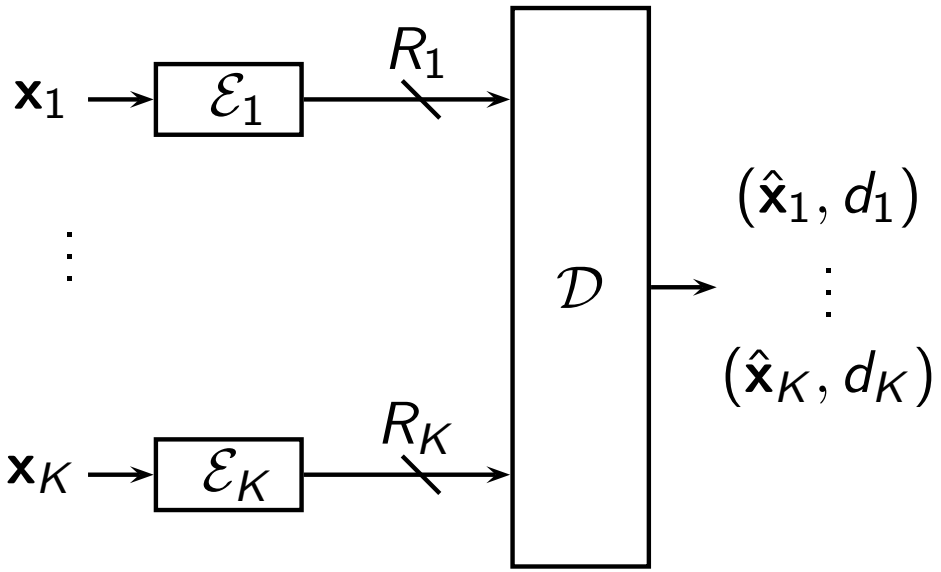
- Simple, identical, universal, non-cooperating quantizers $\mathcal{E}_1, \mathcal{E}_2$
- Simple decoder $\mathcal{D}$ that can depend on $\mathbf{K}_{xx}$
- Good performance for all $\mathbf{K}_{xx}$ with the same $\log\det\left(\mathbf{I} + \frac{1}{d}\mathbf{K}_{xx}\right)$

Extreme cases:

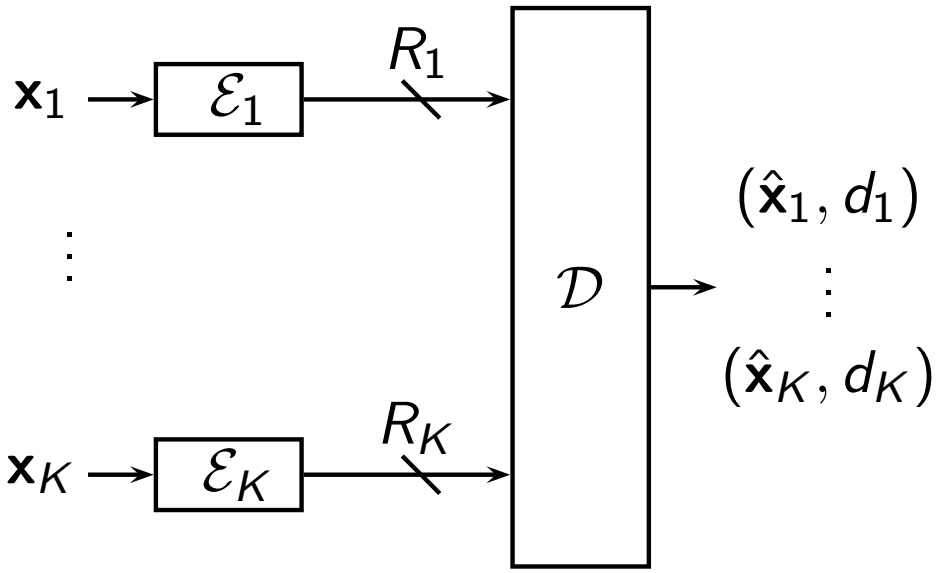
$$\mathbf{K}_{xx}^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{K}_{xx}^2 = \begin{bmatrix} a & 0 \\ 0 & 0 \end{bmatrix}, \text{ and } \mathbf{K}_{xx}^3 = \begin{bmatrix} b & b \\ b & b \end{bmatrix}$$

Willing to apply a **universal** linear transformation before quantization

Goal 2 - Distributed Lossy Compression

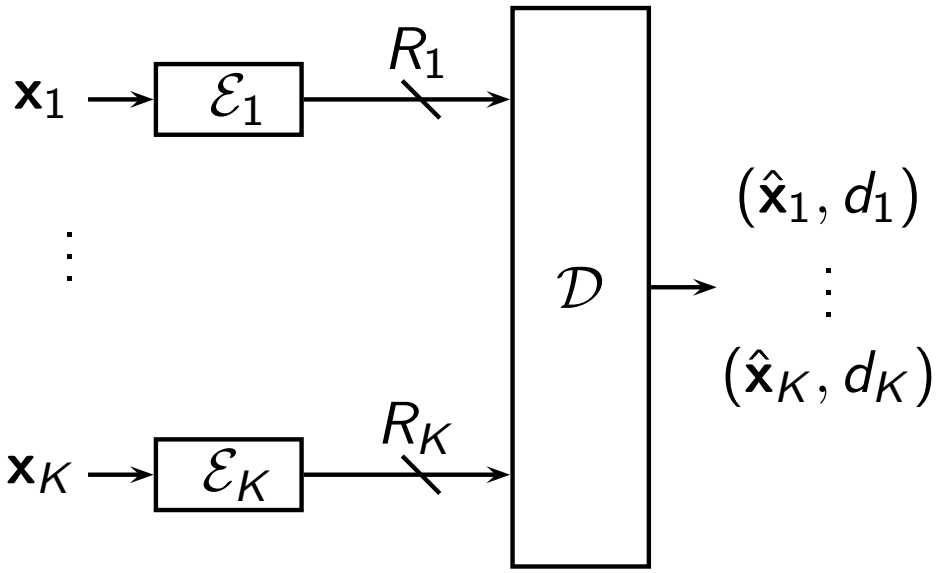


Goal 2 - Distributed Lossy Compression



- Fundamental limits understood in some cases
- Inner and outer bounds known

Goal 2 - Distributed Lossy Compression

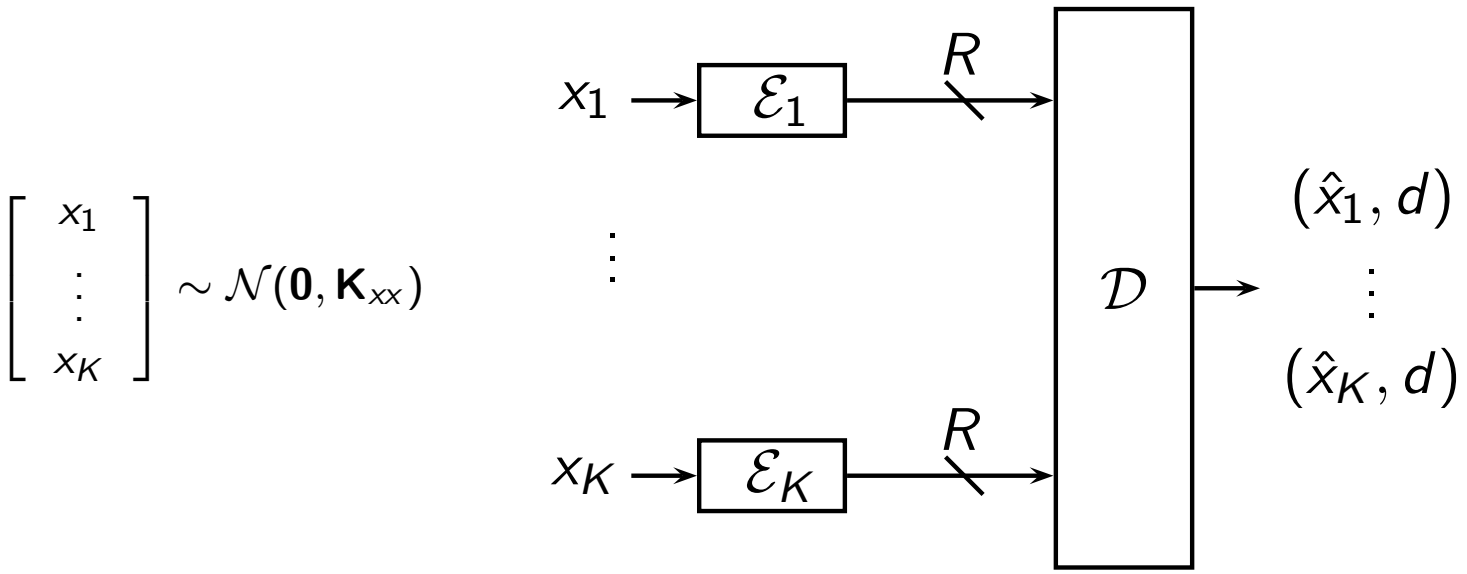


- Fundamental limits understood in some cases
- Inner and outer bounds known

Some applications require

- Extremely simple encoders/decoder
- $n = 1$

Goal 2 -Distributed Lossy Compression



We restrict attention to:

- Gaussian sources $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$
- One-shot compression - block length is 1
- Symmetric rates $R_1 = \cdots = R_K = R$
- Symmetric distortions $d_1 = \cdots = d_K = d$
- MSE distortion measure: $E(x_k - \hat{x}_k)^2 \leq d$

Goal and Means

Goal

- Simple encoders: uniform scalar quantizers
- Decoupled decoding
- Performance close to best known inner bounds (Berger-Tung)

Goal and Means

Goal

- Simple encoders: uniform scalar quantizers
- Decoupled decoding
- Performance close to best known inner bounds (Berger-Tung)

Binning:

- Well understood for large blocklengths, less for short blocks
- Requires sophisticated joint decoding techniques

Goal and Means

Goal

- Simple encoders: uniform scalar quantizers
- Decoupled decoding
- Performance close to best known inner bounds (Berger-Tung)

Binning:

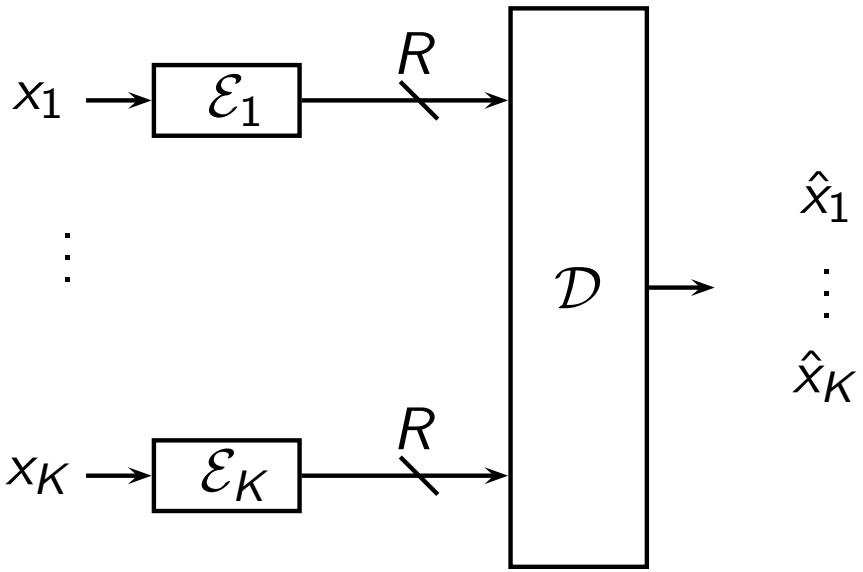
- Well understood for large blocklengths, less for short blocks
- Requires sophisticated joint decoding techniques

Scalar Modulo

- A simple 1-D binning operation
- Allows for efficient decoding using integer-forcing

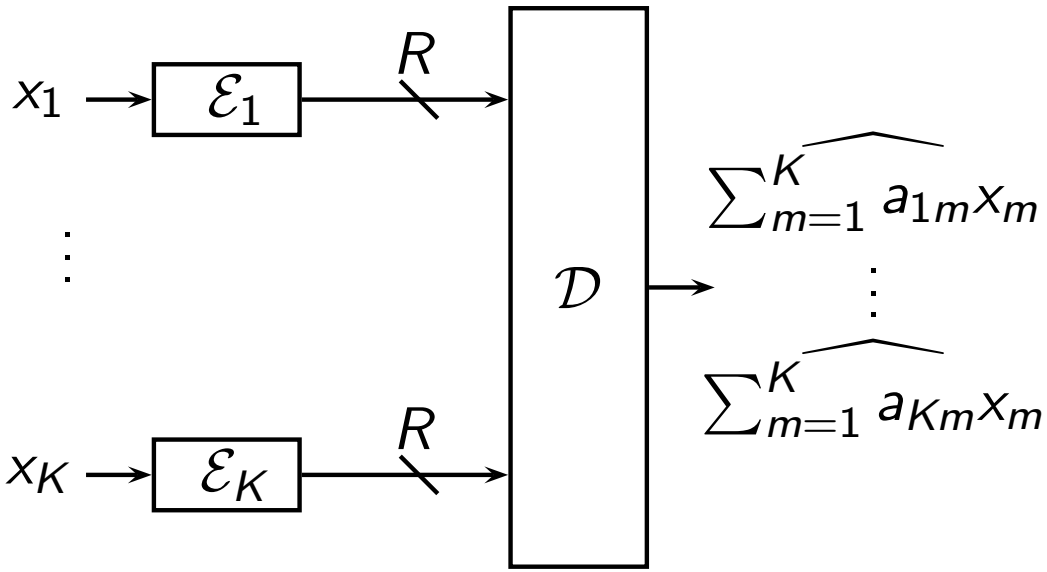
Integer-Forcing Source Coding: Overview

Basic Idea: Rather than solving the problem



Integer-Forcing Source Coding: Overview

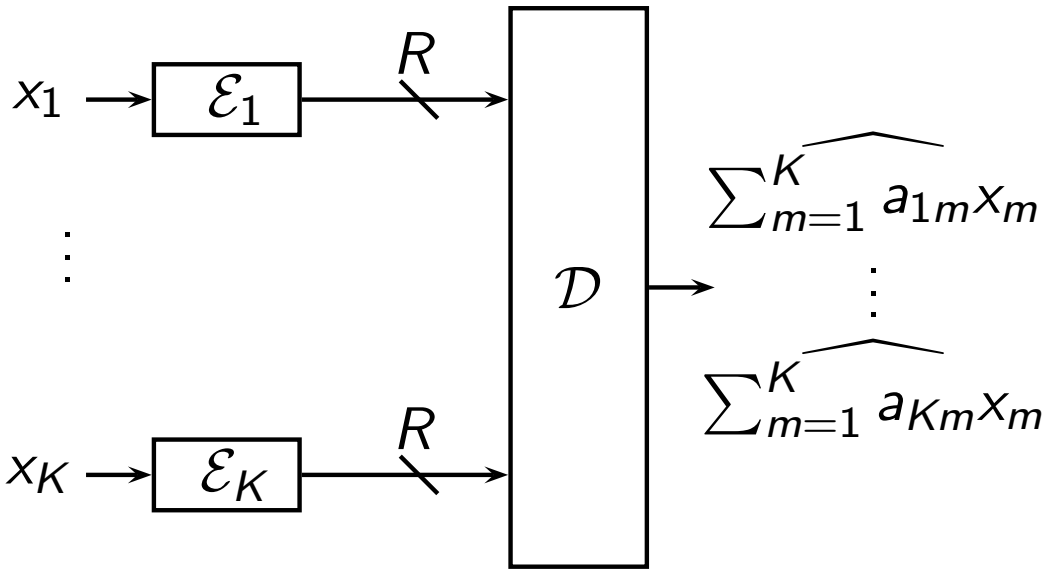
First solve



and then invert equations to get $\hat{x}_1, \ldots, \hat{x}_K$

Integer-Forcing Source Coding: Overview

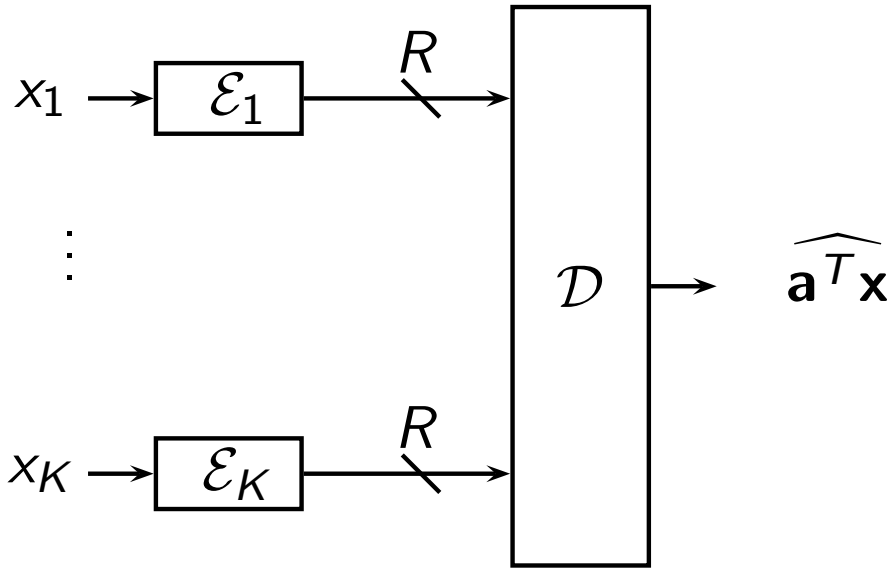
First solve



and then invert equations to get $\hat{x}_1, \ldots, \hat{x}_K$

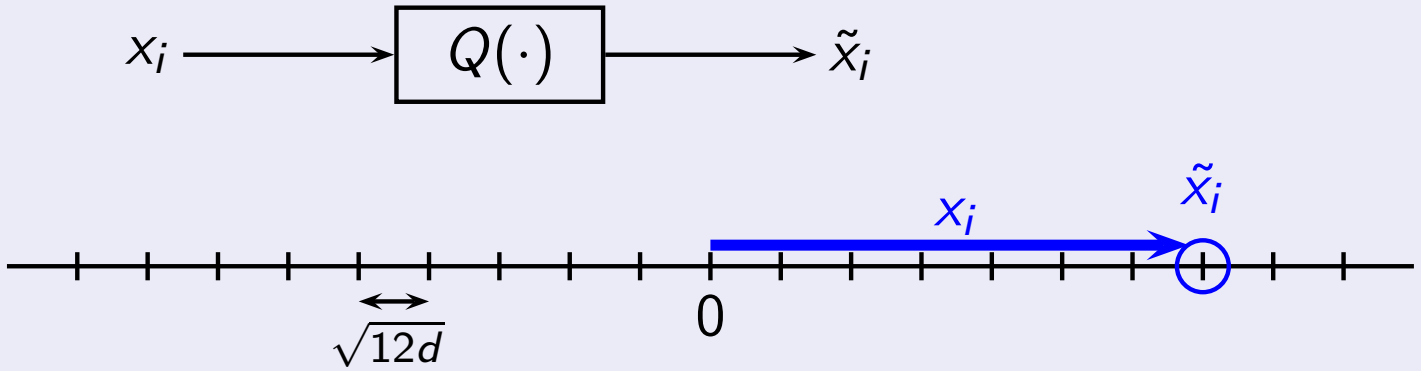
- Problem reduces to simultaneous distributed compression of K linear combinations
- Can be efficiently solved with small rates for certain choices of coefficients
- Equation coefficients can be chosen to optimize performance

Distributed Compression of Integer Linear Combination



Distributed Compression of Integer Linear Combination

Scalar Quantization



- High resolution/dithered quantization:

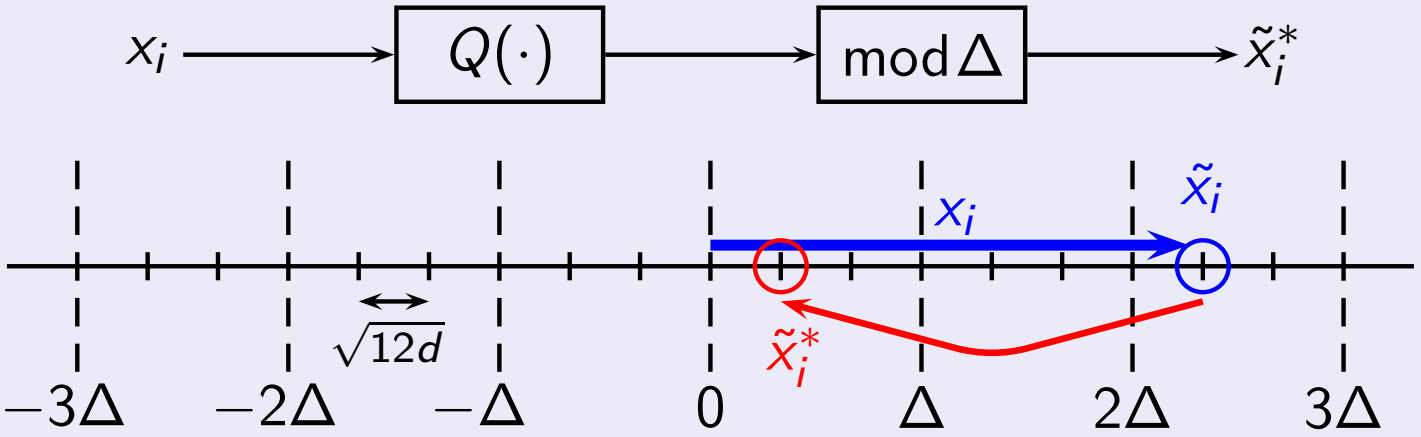
$$\tilde{x}_i = x_i + u_i$$

where $u_i \sim \text{Uniform}\left(\left[-\frac{\sqrt{12d}}{2}, \frac{\sqrt{12d}}{2}\right)\right)$, $u_i \perp\!\!\!\perp x_i$

- $\mathbb{E}(\tilde{x}_i - x_i)^2 = d$

Distributed Compression of Integer Linear Combination

Modulo Scalar Quantization



- $\Delta = 2^R\sqrt{12d} \Longrightarrow$ Compression rate is R
- High resolution/dithered quantization:

$$\tilde{x}_i^* = [x_i + u_i]^*$$

Distributed Compression of Integer Linear Combination

Encoders

Each encoder is a modulo scalar quantizer with rate R : produces $\tilde{x}_k^*$

Distributed Compression of Integer Linear Combination

Encoders

Each encoder is a modulo scalar quantizer with rate R : produces $\tilde{x}_k^*$

Simple modulo property

For any set of integers $a_1, \ldots, a_K$ and real numbers $\tilde{x}_1, \ldots, \tilde{x}_K$

$$\left[\sum_{k=1}^{K} a_k \tilde{x}_k\right]^* = \left[\sum_{k=1}^{K} a_k \tilde{x}_k^*\right]^*$$

Distributed Compression of Integer Linear Combination

Encoders

Each encoder is a modulo scalar quantizer with rate R : produces $\tilde{x}_k^*$

Simple modulo property

For any set of integers $a_1, \ldots, a_K$ and real numbers $\tilde{x}_1, \ldots, \tilde{x}_K$

$$\left[\sum_{k=1}^{K} a_k \tilde{x}_k\right]^* = \left[\sum_{k=1}^{K} a_k \tilde{x}_k^*\right]^*$$

Decoder

- Gets: $\tilde{x}_1^*, \ldots, \tilde{x}_K^*$
- Outputs:

$$\widehat{\mathbf{a}^T \mathbf{x}} = \left[\sum_{k=1}^{K} a_k \tilde{x}_k^*\right]^* = \left[\sum_{k=1}^{K} a_k \tilde{x}_k\right]^* = \left[\mathbf{a}^T(\mathbf{x}+\mathbf{u})\right]^*$$

Compression of Integer Linear Combination - P_e

$$\widehat{\mathbf{a}^T\mathbf{x}} = \left[\mathbf{a}^T(\mathbf{x}+\mathbf{u})\right]^*$$

$$\widehat{\mathbf{a}^T\mathbf{x}} = \begin{cases} \mathbf{a}^T\mathbf{x} + \mathbf{a}^T\mathbf{u} & \text{if } \mathbf{a}^T(\mathbf{x}+\mathbf{u}) \in \left[-\frac{\Delta}{2}, \frac{\Delta}{2}\right) \\ \text{error} & \text{otherwise} \end{cases}$$

- P_e is small if $\frac{\Delta}{\sqrt{\mathrm{Var}\left(\mathbf{a}^T(\mathbf{x}+\mathbf{u})\right)}}$ is large
- Δ grows exponentially with R

Compression of Integer Linear Combination - P_e

$$\widehat{\mathbf{a}^T\mathbf{x}} = \left[\mathbf{a}^T(\mathbf{x}+\mathbf{u})\right]^*$$

$$\widehat{\mathbf{a}^T\mathbf{x}} = \begin{cases} \mathbf{a}^T\mathbf{x} + \mathbf{a}^T\mathbf{u} & \text{if } \mathbf{a}^T(\mathbf{x}+\mathbf{u}) \in \left[-\frac{\Delta}{2}, \frac{\Delta}{2}\right) \\ \text{error} & \text{otherwise} \end{cases}$$

- P_e is small if $\frac{\Delta}{\sqrt{\mathrm{Var}\left(\mathbf{a}^T(\mathbf{x}+\mathbf{u})\right)}}$ is large
- Δ grows exponentially with R

$$P_e \leq 2\exp\left\{-\frac{3}{2}2^{2\left(R-\frac{1}{2}\log\left(\frac{\mathbf{a}^T(\mathbf{K_{xx}}+d\mathbf{I})\mathbf{a}}{d}\right)\right)}\right\}$$

Compression of Integer Linear Combination - P_e

$$\widehat{\mathbf{a}^T\mathbf{x}} = \left[\mathbf{a}^T(\mathbf{x}+\mathbf{u})\right]^*$$

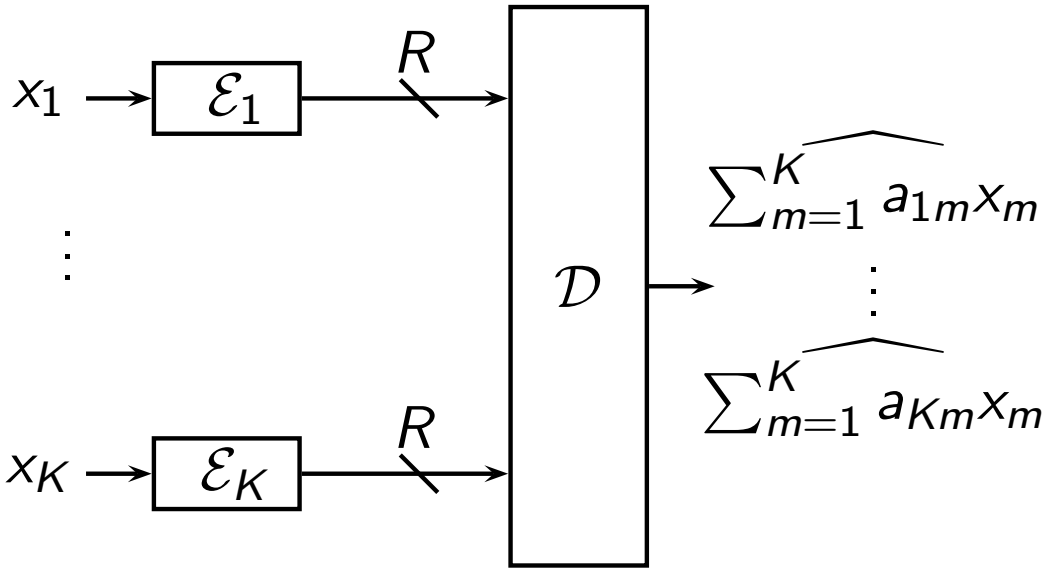
$$\widehat{\mathbf{a}^T\mathbf{x}} = \begin{cases} \mathbf{a}^T\mathbf{x} + \mathbf{a}^T\mathbf{u} & \text{if } \mathbf{a}^T(\mathbf{x}+\mathbf{u}) \in \left[-\frac{\Delta}{2}, \frac{\Delta}{2}\right) \\ \text{error} & \text{otherwise} \end{cases}$$

- P_e is small if $\frac{\Delta}{\sqrt{\mathrm{Var}\left(\mathbf{a}^T(\mathbf{x}+\mathbf{u})\right)}}$ is large
- Δ grows exponentially with R

$$P_e \leq 2\exp\left\{-\frac{3}{2}2^{2\left(R-\frac{1}{2}\log\left(\frac{\mathbf{a}^T(\mathbf{K_{xx}}+d\mathbf{I})\mathbf{a}}{d}\right)\right)}\right\}$$

For $\mathbf{a}$ with small $\mathrm{Var}\left(\mathbf{a}^T(\mathbf{x}+\mathbf{u})\right)$ we can take small R

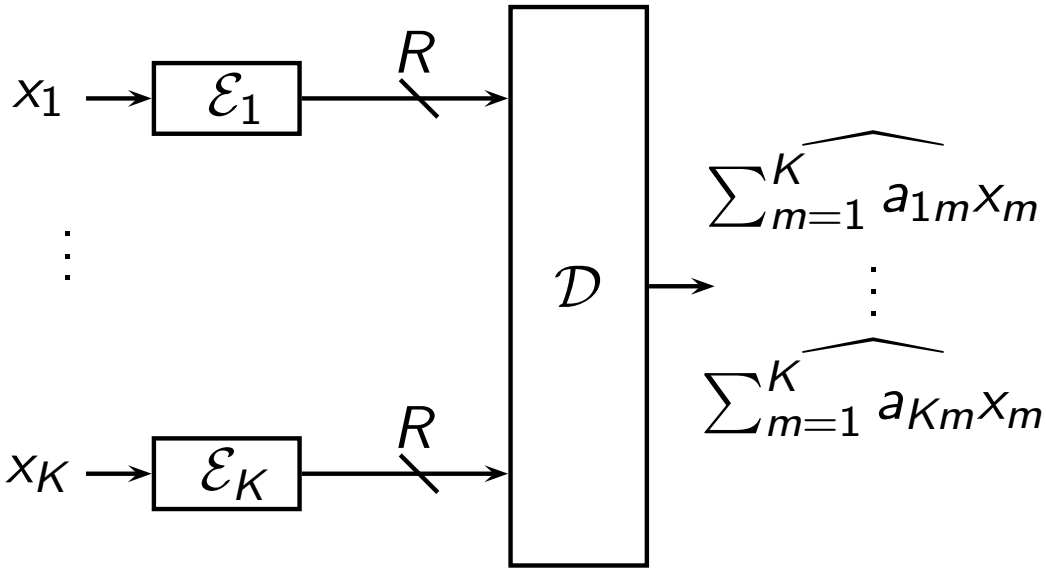
Integer-Forcing Source Coding



- Need to estimate K linearly independent integer linear combinations
- If all combinations estimated without error, can compute

$$\hat{\mathbf{x}} = \mathbf{A}^{-1}\widehat{\mathbf{Ax}} = \mathbf{A}^{-1}(\mathbf{Ax} + \mathbf{Au}) = \mathbf{x} + \mathbf{u}$$

Integer-Forcing Source Coding



- Need to estimate K linearly independent integer linear combinations
- If all combinations estimated without error, can compute

$$\hat{\mathbf{x}} = \mathbf{A}^{-1}\widehat{\mathbf{Ax}} = \mathbf{A}^{-1}(\mathbf{Ax} + \mathbf{Au}) = \mathbf{x} + \mathbf{u}$$

$$P_e \leq 2K \exp\left\{ -\frac{3}{2} 2^{2\left(R - \frac{1}{2}\log\left(\frac{\max_{m=1,\ldots,K} \mathbf{a}_m^T(\mathbf{K_{xx}} + d\mathbf{I})\mathbf{a}_m}{d} \right) \right)} \right\}$$

Integer-Forcing Source Coding - Performance

Let

$$R_{\mathrm{IF}}(\mathbf{A}, d) \triangleq \frac{1}{2} \log \left(\max_{m=1,\ldots,K} \mathbf{a}_m^T \left(\mathbf{I} + \frac{1}{d} \mathbf{K}_{\mathbf{xx}} \right) \mathbf{a}_m \right)$$

Integer-Forcing Source Coding - Performance

Let

$$R_{\mathrm{IF}}(\mathbf{A}, d) \triangleq \frac{1}{2} \log \left(\max_{m=1,\ldots,K} \mathbf{a}_m^T \left(\mathbf{I} + \frac{1}{d} \mathbf{K}_{\mathbf{xx}} \right) \mathbf{a}_m \right)$$

Theorem

Let $R = R_{\mathrm{IF}}(\mathbf{A}, d) + \delta$. IF source coding produces estimates with average MSE distortion d for all $x_1, \ldots, x_K$ with probability $> 1 - 2K \exp\left\{ -\frac{3}{2} 2^{2\delta} \right\}$

Integer-Forcing Source Coding - Performance

Let

$$R_{\mathrm{IF}}(\mathbf{A}, d) \triangleq \frac{1}{2} \log \left(\max_{m=1,\ldots,K} \mathbf{a}_m^T \left(\mathbf{I} + \frac{1}{d} \mathbf{K}_{\mathbf{xx}} \right) \mathbf{a}_m \right)$$

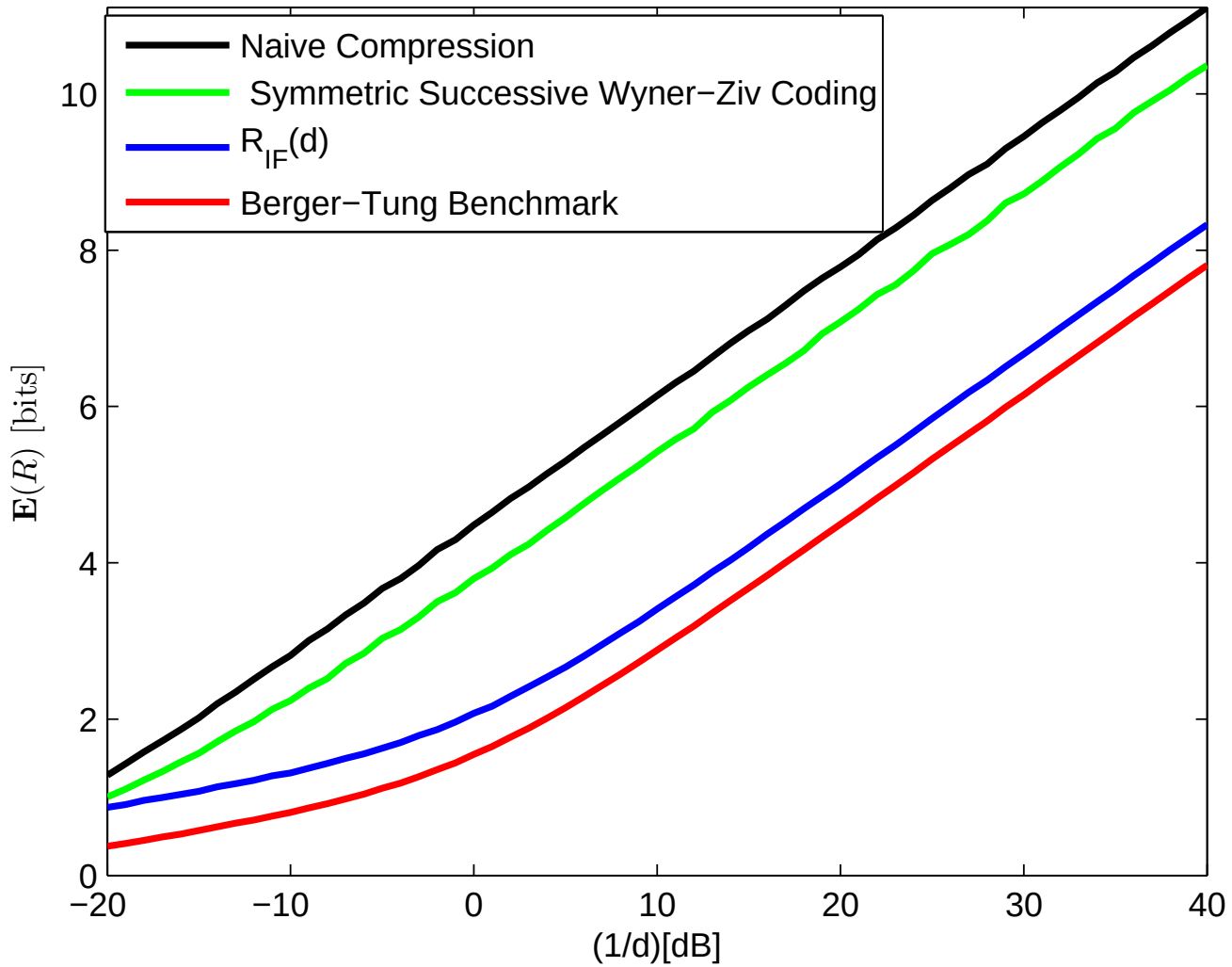
Theorem

Let $R = R_{\mathrm{IF}}(\mathbf{A}, d) + \delta$. IF source coding produces estimates with average MSE distortion d for all $x_1, \ldots, x_K$ with probability $> 1 - 2K \exp\left\{ -\frac{3}{2} 2^{2\delta} \right\}$

Can minimize compression rate by minimizing $R_{\mathrm{IF}}(\mathbf{A}, d)$ w.r.t. $\mathbf{A}$

Integer-Forcing Source Coding: Example

$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{K_{xx}})$, $\mathbf{K_{xx}} = \mathbf{I} + \mathrm{SNR}\mathbf{H}\mathbf{H}^T$, $\mathrm{SNR} = 20\mathrm{dB}$ and $\mathbf{H} \in \mathbb{R}^{8\times 2}$



Back to Motivation 1

How close is $R_{\mathrm{IF}}(d)$ to the optimal performance?

- Usually very close to the performance of the Berger-Tung inner bound.
- But... the gap can be arbitrarily large.

Back to Motivation 1

How close is $R_{\text{IF}}(d)$ to the optimal performance?

- Usually very close to the performance of the Berger-Tung inner bound.
- But... the gap can be arbitrarily large.

However, if we change the setting...

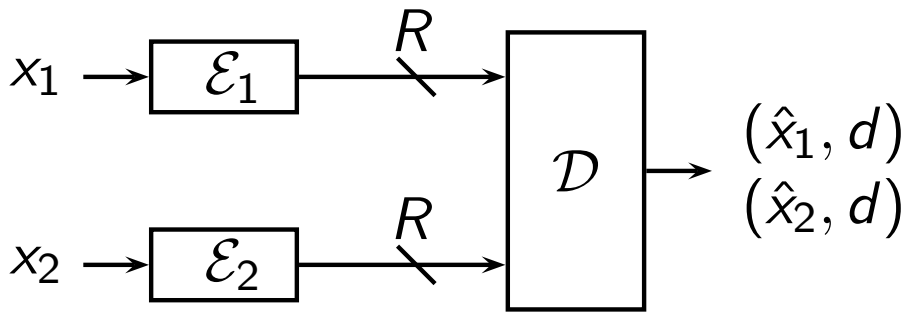
this obstacle can be overcome.

Back to Motivation 1

How close is $R_{\mathrm{IF}}(d)$ to the optimal performance?

- Usually very close to the performance of the Berger-Tung inner bound.
- But... the gap can be arbitrarily large.

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$$

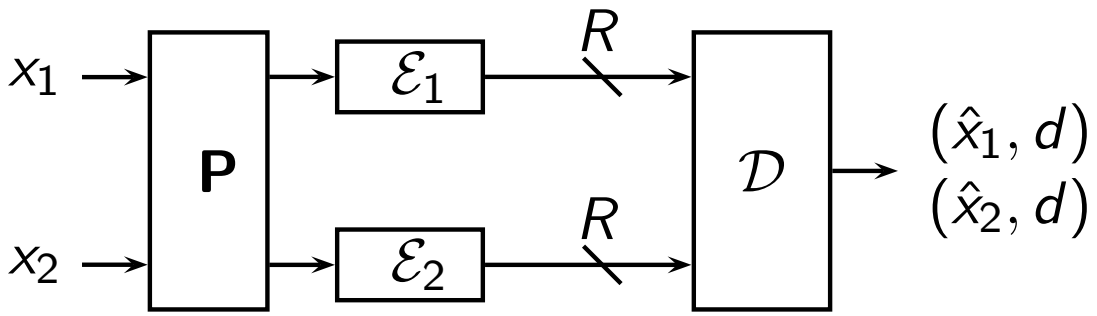


Back to Motivation 1

How close is $R_{\mathrm{IF}}(d)$ to the optimal performance?

- Usually very close to the performance of the Berger-Tung inner bound.
- But... the gap can be arbitrarily large.

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$$

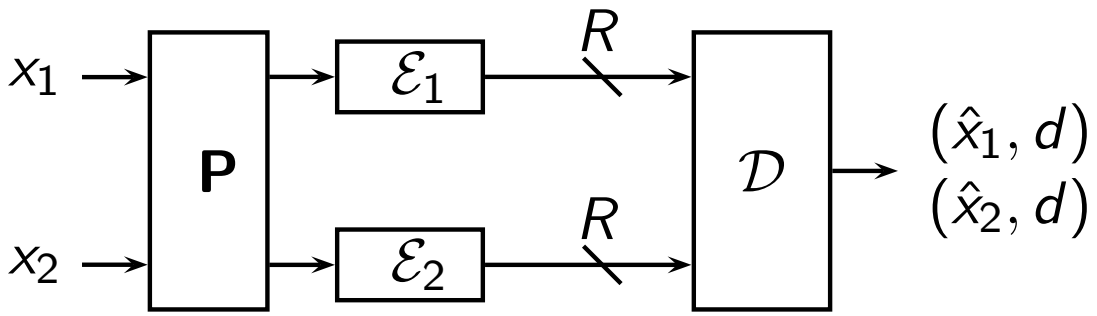


Back to Motivation 1

How close is $R_{\mathrm{IF}}(d)$ to the optimal performance?

- Usually very close to the performance of the Berger-Tung inner bound.
- But... the gap can be arbitrarily large.

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$$



Requirements

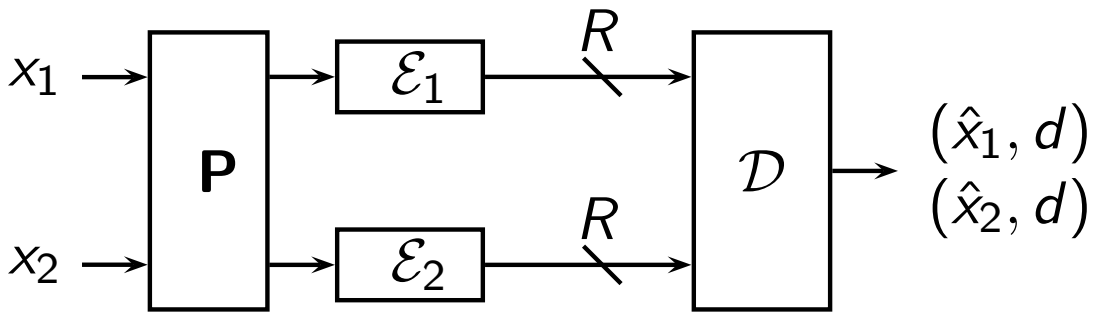
- Universal precoding matrix $\mathbf{P}$ (does not depend on $\mathbf{K}_{xx}$)
- $R_{\mathrm{IF}}(d) \leq \text{const} + \frac{1}{2K} \log\det(\mathbf{I} + \frac{1}{d}\mathbf{K}_{xx})$ for all $\mathbf{K}_{xx}$

Back to Motivation 1

How close is $R_{\mathrm{IF}}(d)$ to the optimal performance?

- Usually very close to the performance of the Berger-Tung inner bound.
- But... the gap can be arbitrarily large.

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$$

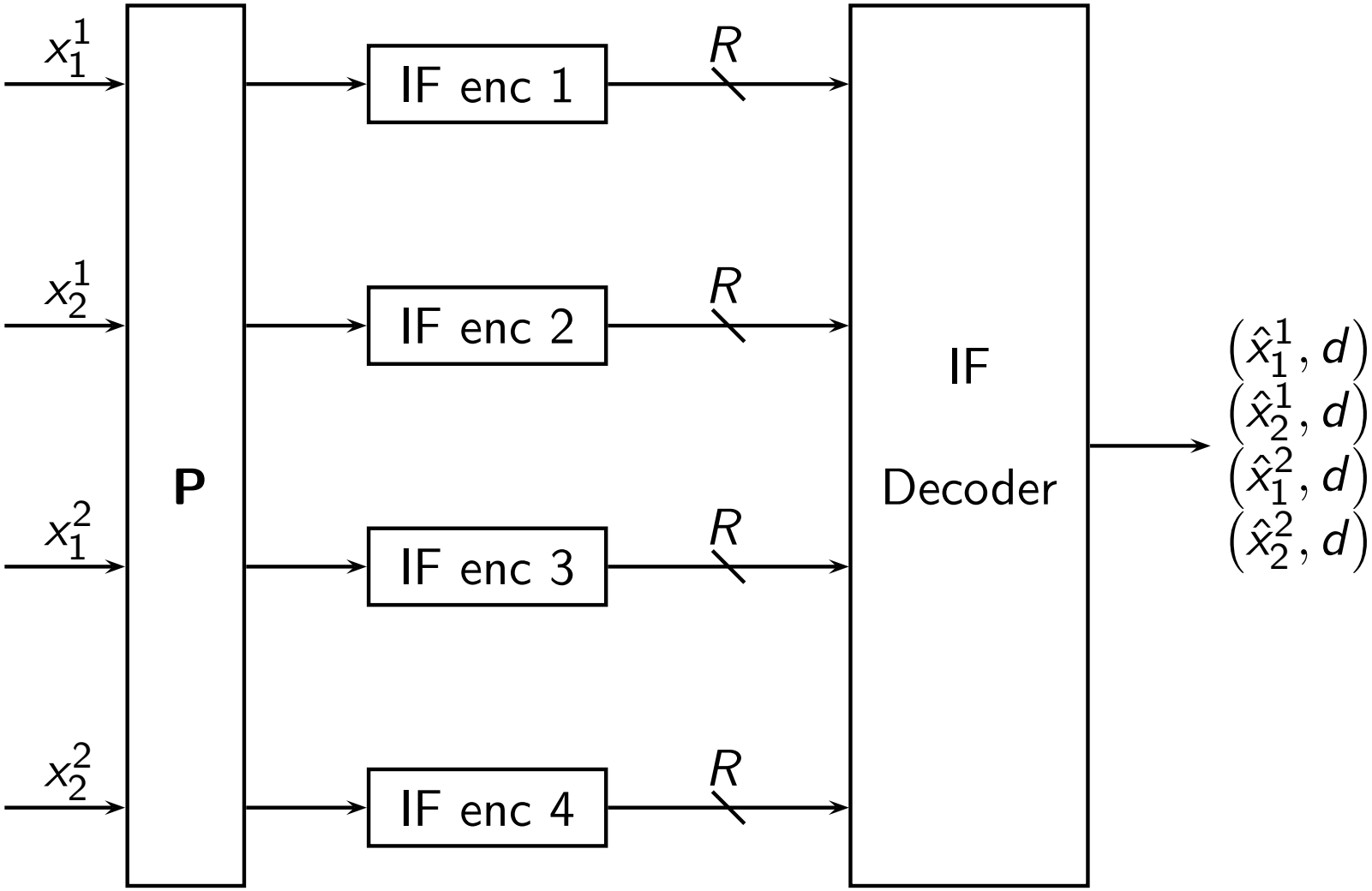


Requirements

- Universal precoding matrix $\mathbf{P}$ (does not depend on $\mathbf{K}_{xx}$)
- $R_{\mathrm{IF}}(d) \leq \text{const} + \frac{1}{2K} \log\det(\mathbf{I} + \frac{1}{d}\mathbf{K}_{xx})$ for all $\mathbf{K}_{xx}$

Price of universality - need to jointly encode K realizations

Space-Time Source Coding



Space-Time Source Coding - Performance Guarantees

Let $\mathbf{P}$ be a generating matrix of a "perfect" linear dispersion space-time code, with minimum det $\delta_{\min}(\mathcal{C}_{\infty}^{\mathrm{ST}})$

Theorem

For any source with covariance matrix $\mathbf{K_{xx}}$, the rate-distortion function of space-time integer-forcing source coding with precoding matrix $\mathbf{P}$ is bounded by

$$R_{\mathrm{IF}}(d) < \frac{1}{2K} \log\det\left(\mathbf{I} + \frac{1}{d}\mathbf{K_{xx}}\right) + \Gamma\left(K, \delta_{\min}(\mathcal{C}_{\infty}^{\mathrm{ST}})\right)$$

where $\Gamma\left(K, \delta_{\min}(\mathcal{C}_{\infty}^{\mathrm{ST}})\right) \triangleq 2K^2 \log(2K^2) + K \log \frac{1}{\delta_{\min}(\mathcal{C}_{\infty}^{\mathrm{ST}})}$

Remark: For $K = 2$ the golden-code precoding matrix has $\delta_{\min}(\mathcal{C}_{\infty}^{\mathrm{ST}}) = 1/5$

Example

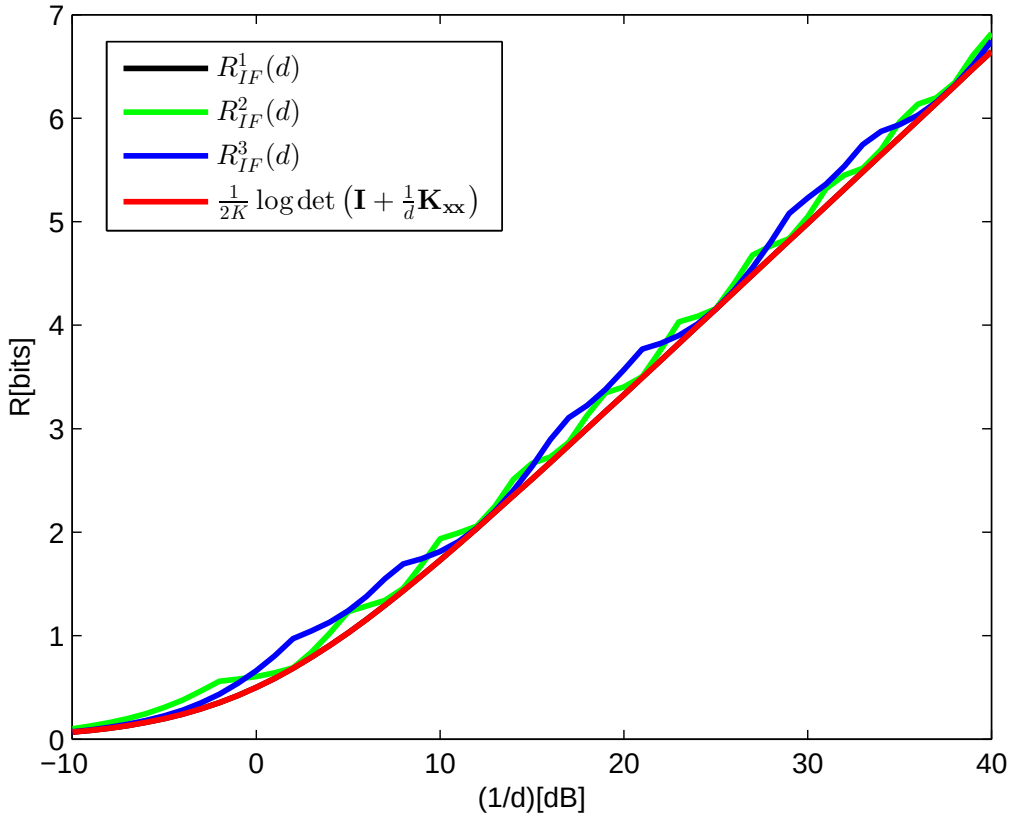
$$\mathbf{K}_{xx}^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{K}_{xx}^2 = \begin{bmatrix} a & 0 \\ 0 & 0 \end{bmatrix}, \text{ and } \mathbf{K}_{xx}^3 = \begin{bmatrix} b & b \\ b & b \end{bmatrix}$$

$$\frac{1}{2K} \log \left(\mathbf{I} + \frac{1}{d} \mathbf{K}_{xx}^1 \right) = \frac{1}{2K} \log \left(\mathbf{I} + \frac{1}{d} \mathbf{K}_{xx}^2 \right) = \frac{1}{2K} \log \left(\mathbf{I} + \frac{1}{d} \mathbf{K}_{xx}^3 \right)$$

Example

$$\mathbf{K}_{xx}^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{K}_{xx}^2 = \begin{bmatrix} a & 0 \\ 0 & 0 \end{bmatrix}, \text{ and } \mathbf{K}_{xx}^3 = \begin{bmatrix} b & b \\ b & b \end{bmatrix}$$

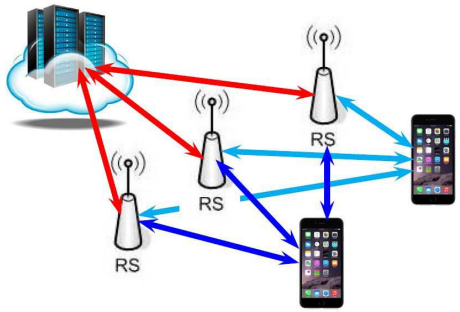
$$\frac{1}{2K} \log \left(\mathbf{I} + \frac{1}{d} \mathbf{K}_{xx}^1 \right) = \frac{1}{2K} \log \left(\mathbf{I} + \frac{1}{d} \mathbf{K}_{xx}^2 \right) = \frac{1}{2K} \log \left(\mathbf{I} + \frac{1}{d} \mathbf{K}_{xx}^3 \right)$$



Further Applications of Integer-Forcing Source Coding:

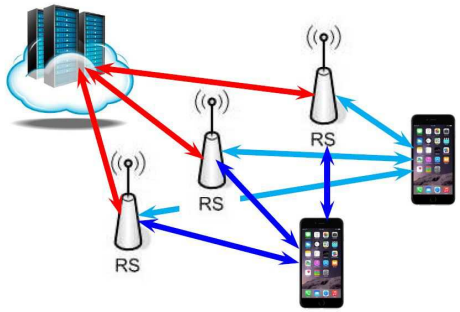
Further Applications of Integer-Forcing Source Coding:

Compress-and-forward for relay networks (C-RAN)

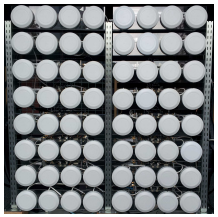


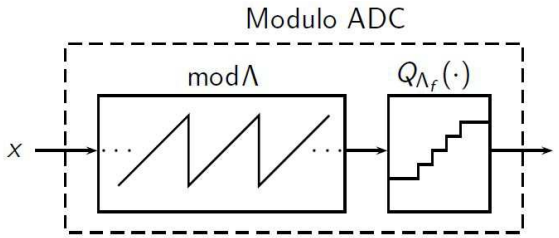
Further Applications of Integer-Forcing Source Coding:

Compress-and-forward for relay networks (C-RAN)



Analog-to-digital converters





Thanks for your attention!