State-Dependent DMC with a Causal Helper

Ligong Wang

Joint work with Amos Lapidoth

Information Theory and Tapas Workshop

Madrid, 27 Jan. 2023

State-dependent DMC

- Input, output, and state alphabets are $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{S}$
- State sequence is IID according to P_S
- Given input $X = x$ and state $S = s$, output equals y with probability

$$W(y|x, s)$$

Capacity depends on whether or not channel-state information (CSI) is available, to whom, and how.

Causal CSI: Shannon's classic result

At time i, encoder knows s^i, so time-i input is produced via

$$x_i = f_i(m, s^i)$$

where m denotes the message.

Causal CSI: Shannon's classic result

At time i, encoder knows s^i, so time-i input is produced via

$$x_i = f_i(m, s^i)$$

where m denotes the message.

Theorem [Shannon '58]

Capacity with causal CSI at Transmitter is the maximum of

$$I(U;Y)$$

over joint distributions of the form

$$P_S(s)P_U(u)P_{X|US}(x|u,s)W(y|x,s).$$

Without loss of optimality, x can be chosen as a deterministic function of (u,s).

How this capacity is achieved

- u is a mapping (or "strategy") that maps s to x
- Capacity is that of the "super channel" with input u and output y
- Notice that, at time i, the optimal encoder only uses s_i and ignores s^{i-1}:

$$x_i = f_i(m, s_i).$$

How this capacity is achieved

- u is a mapping (or "strategy") that maps s to x
- Capacity is that of the "super channel" with input u and output y
- Notice that, at time i, the optimal encoder only uses s_i and ignores s^{i-1}:

$$x_i = f_i(m, s_i).$$

Related to the above: strictly causal CSI, where

$$x_i = f_i(m, s^{i-1}),$$

does not increase capacity.

This work: Imperfect causal CSI via Helper

At time i:

- Helper observes s^i and produces $t_i \in \mathcal{T}$:

$$t_i = h_i(s^i)$$

  (No additional constraint on Helper except that $\mathcal{T}$ is fixed, with $|\mathcal{T}| < |\mathcal{S}|$)

- Transmitter sees t^i and produces input

$$x_i = f_i(m, t^i)$$

This work: Imperfect causal CSI via Helper

At time i:

- Helper observes s^i and produces $t_i \in \mathcal{T}$:

$$t_i = h_i(s^i)$$

  (No additional constraint on Helper except that $\mathcal{T}$ is fixed, with $|\mathcal{T}| < |\mathcal{S}|$)

- Transmitter sees t^i and produces input

$$x_i = f_i(m, t^i)$$

Main question: Is it optimal to choose $t_i = h_i(s_i)$ and $x_i = f_i(m, t_i)$?

This work: Imperfect causal CSI via Helper

At time i:

- Helper observes s^i and produces $t_i \in \mathcal{T}$:

$$t_i = h_i(s^i)$$

(No additional constraint on Helper except that $\mathcal{T}$ is fixed, with $|\mathcal{T}| < |\mathcal{S}|$)

- Transmitter sees t^i and produces input

$$x_i = f_i(m, t^i)$$

Main question: Is it optimal to choose $t_i = h_i(s_i)$ and $x_i = f_i(m, t_i)$?

We shall also consider some variants of the above setting.

Some related works

- Rosenzweig, Steinberg, and Shamai '05:

  Full CSI at Receiver and quantized CSI at Transmitter

  (In the causal case, $t_i = h_i(s_i)$ is assumed as part of the setup)

- Steinberg '08: Full CSI at Transmitter and quantized CSI at Receiver

- Bross, Lapidoth, and Marti '20 (two papers):

  Additive-noise channels with quantized CSI at Transmitter or at Receiver

Scalar-quantization lower bound

- Helper: $t_i = g(s_i)$
- Encoder: Shannon strategies treating t as the effective state: $x_i = u_i(t_i)$.

This achieves rate

$$I(U;Y)$$

with joint distribution

$$P_S(s)P_U(u)P_{T|S}(t|s)P_{X|UT}(x|u,t)W(y|x,s)$$

which satisfies three conditions:

$$(U,T) -\circ- (X,S) -\circ- Y$$
$$S -\circ- (U,T) -\circ- X$$
$$U \perp\!\!\!\perp (S,T).$$

(It's optimal to choose both $P_{T|S}$ and $P_{X|UT}$ to be deterministic.)

An upper bound

Define for every i

$$U_i \triangleq (M, T^{i-1}, Y^{i-1}).$$

Then

$$\begin{aligned} n(R-\epsilon) &\leq I(M; Y^n) \\ &= \sum_{i=1}^{n} I(M; Y_i | Y^{i-1}) \\ &\leq \sum_{i=1}^{n} I(M, Y^{i-1}; Y_i) \\ &\leq \sum_{i=1}^{n} I(U_i; Y_i). \end{aligned}$$

Let's check the conditions

Recall $U_i \triangleq (M, T^{i-1}, Y^{i-1})$.

$$(U_i, T_i) = (M, T^i, Y^{i-1}) \multimap\!\!- (X_i, S_i) \multimap\!\!- Y_i \quad \checkmark$$

Let's check the conditions

Recall $U_i \triangleq (M, T^{i-1}, Y^{i-1})$.

$$(U_i, T_i) = (M, T^i, Y^{i-1}) \multimap (X_i, S_i) \multimap Y_i \quad \checkmark$$
$$S_i \multimap (M, T^i, Y^{i-1}) \multimap X_i \quad \checkmark$$

Let's check the conditions

Recall $U_i \triangleq (M, T^{i-1}, Y^{i-1})$.

$$(U_i, T_i) = (M, T^i, Y^{i-1}) \multimap\!\!- (X_i, S_i) \multimap\!\!- Y_i \quad \checkmark$$
$$S_i \multimap\!\!- (M, T^i, Y^{i-1}) \multimap\!\!- X_i \quad \checkmark$$

$$U_i = (M, T^{i-1}, Y^{i-1}) \not\perp\!\!\!\perp (S_i, T_i) \quad \times$$

For example, both T and S binary, $T_{i-1} = S_{i-1}$ and $T_i = S_i \oplus S_{i-1}$

Let's check the conditions

Recall $U_i \triangleq (M, T^{i-1}, Y^{i-1})$.

$$(U_i, T_i) = (M, T^i, Y^{i-1}) \multimap (X_i, S_i) \multimap Y_i \quad \checkmark$$
$$S_i \multimap (M, T^i, Y^{i-1}) \multimap X_i \quad \checkmark$$

$$U_i = (M, T^{i-1}, Y^{i-1}) \not\perp\!\!\!\perp (S_i, T_i) \quad \times$$

For example, both T and S binary, $T_{i-1} = S_{i-1}$ and $T_i = S_i \oplus S_{i-1}$

We only have

$$U_i \perp\!\!\!\perp S_i$$

i.e., joint distribution looks like

$$P_S(s) P_U(u) P_{T|US}(t|u,s) P_{X|UT}(x|u,t) W(y|x,s)$$

Why the gap?

- Can we prove a tighter upper bound?

Why the gap?

- Can we prove a tighter upper bound?
- Could it be that $\max I(U;Y)$ over the two types of joint distributions is in fact the same?

  I.e. once you optimize, “maybe” you actually want $U \perp\!\!\!\perp T$?

  The example $T_{i-1} = S_{i-1}$ and $T_i = S_i \oplus S_{i-1}$ was obviously a silly scheme.

Why the gap?

- Can we prove a tighter upper bound?

- Could it be that $\max I(U;Y)$ over the two types of joint distributions is in fact the same?

  I.e. once you optimize, "maybe" you actually want $U \perp\!\!\!\perp T$?

  The example $T_{i-1} = S_{i-1}$ and $T_i = S_i \oplus S_{i-1}$ was obviously a silly scheme.

- If neither works, then we need to think about a better lower bound, but

Why on earth would Helper want to tell Transmitter anything about S^{i-1} at time i?

A variant (special case): S is part of the output

i.e. CSI is available to Receiver (causality plays no role).

(For clarity, Y denotes the part of output without S.)

Lower bound on capacity becomes

$$I(U;Y,S) = I(U;Y|S) = I(X;Y|S)$$

for joint distribution

$$P_S(s)P_{T|S}(t|s)P_{X|T}(x|t)W(y|x,s)$$

where it's again optimal to choose $P_{T|S}$ to be deterministic.

S known to Receiver: Upper bound

Define

$$U_i \triangleq (M, Y^{i-1}), \qquad V_i \triangleq S^{i-1}$$

Then

$$\begin{aligned} n(R-\epsilon) &\leq I(M; Y^n, S^n) \\ &= \sum_{i=1}^{n} I(M; Y_i, S_i | Y^{i-1}, S^{i-1}) \\ &\leq \sum_{i=1}^{n} I(M, Y^{i-1}; Y_i, S_i | S^{i-1}) \\ &= \sum_{i=1}^{n} I(M, Y^{i-1}; Y_i | S^{i-1}, S_i) \\ &= \sum_{i=1}^{n} I(U_i; Y_i | V_i, S_i) \end{aligned}$$

S known to Receiver: Upper bound (contd.)

So capacity is upper-bounded by the maximum of

$$I(U;Y|V,S)$$

over distributions of the form

$$P_V(v)P_S(s)P_{U|V}(u|v)P_{T|SV}(t|s,v)P_{X|TUV}(x|t,u,v)W(y|x,s)$$

S known to Receiver: Upper bound (contd.)

So capacity is upper-bounded by the maximum of

$$I(U;Y|V,S)$$

over distributions of the form

$$P_V(v)P_S(s)P_{U|V}(u|v)P_{T|SV}(t|s,v)P_{X|TUV}(x|t,u,v)W(y|x,s)$$

Observation: V can be removed via maximization over $V=v$, yielding

$$I(U;Y|S) = I(X;Y|S)$$

over

$$P_S(s)P_{T|S}(t|s)P_{X|T}(x|t)W(y|x,s)$$

which coincides with lower bound.

S known to Receiver: Result

Theorem

When states are known to the Receiver, capacity is given by

$$\max I(X;Y|S)$$

where $\max$ *is over joint distribiutions of the form*

$$P_S(s)P_{T|S}(t|s)P_{X|T}(x|t)W(y|x,s).$$

S known to Receiver: Result

Theorem

When states are known to the Receiver, capacity is given by

$$\max I(X;Y|S)$$

where $\max$ *is over joint distribiutions of the form*

$$P_S(s)P_{T|S}(t|s)P_{X|T}(x|t)W(y|x,s).$$

Note: in this variant it is optimal to choose

$$t_i = h(s_i), \qquad x_i = f_i(m, t_i).$$

Another variant: Helper knows message M

That is, at time i, Helper produces

$$t_i = h_i(m, s^i)$$

Another variant: Helper knows message M

That is, at time i, Helper produces

$$t_i = h_i(m, s^i)$$

Our upper bound in the original setting continues to hold in this case:

$$\text{capacity} \le \max I(U;Y)$$

over

$$P_S(s)P_U(u)P_{T|US}(t|u,s)P_{X|UT}(x|u,t)W(y|x,s).$$

Helper knows M: Achievability

- Codebook: $\{u^n(m), m \in \mathcal{M}\}$ all generated IID according to P_U.
- Helper knows m and hence also $u^n(m)$; it generates

$$t_i = h(u_i, s_i)$$

  Now t_i does depend on u_i!

- Encoder generates

$$x_i = f(u_i, t_i)$$

Helper knows M: Result

Theorem

When the message is known to the helper, capacity is given by

$$\max I(U;Y)$$

over joint distributions of the form

$$P_S(s)P_U(u)P_{T|US}(t|u,s)P_{X|UT}(x|u,t)W(y|x,s).$$

Helper knows M: Result

Theorem

When the message is known to the helper, capacity is given by

$$\max I(U;Y)$$

over joint distributions of the form

$$P_S(s)P_U(u)P_{T|US}(t|u,s)P_{X|UT}(x|u,t)W(y|x,s).$$

Again, it is optimal to let t_i not depend on s^{i-1}!

Helper knows M: An example

- State contains two independent uniform bits:

$$S = (S^{(0)}, S^{(1)})$$

- Helper has one bit to use:

$$\mathcal{T} = \{0, 1\}$$

- Input contains two bits

$$X = (A, B), \qquad A, B \in \{0, 1\}$$

- Output is also two bits

$$Y = (A, B \oplus S^{(A)})$$

Helper knows M example: Solution

We can send two information bits (k, ℓ) as follows:

- Help is $T = S^{(k)}$
- Transmitter sends $(k, \ell \oplus T)$
- Output is then (k, ℓ) (no decoding needed)

It is easy to prove that the above is optimal, so

$$\text{capacity} = 2 \quad \text{bits.}$$

(We can also use the capacity formula to get this.)

Same example when Helper does *not* know M

First consider scalar quantizer (independent of M).

Same example when Helper does *not* know M

First consider scalar quantizer (independent of M).

For an upper bound, reveal T to Receiver.

Given $T = t$, we have a "sum channel":

- Channel 1: $X = (0, B)$, $Y = (0, B \oplus S^{(0)})$; capacity is $1 - H(S^{(0)}|T = t)$
- Channel 2: $X = (1, B)$, $Y = (1, B \oplus S^{(1)})$; capacity is $1 - H(S^{(1)}|T = t)$

Same example when Helper does *not* know M

First consider scalar quantizer (independent of M).

For an upper bound, reveal T to Receiver.

Given $T = t$, we have a "sum channel":

- Channel 1: $X = (0, B)$, $Y = (0, B \oplus S^{(0)})$; capacity is $1 - H(S^{(0)}|T = t)$
- Channel 2: $X = (1, B)$, $Y = (1, B \oplus S^{(1)})$; capacity is $1 - H(S^{(1)}|T = t)$

Capacity of the sum channel is given by

$$\log\left(2^{1-H(S^{(0)}|T=t)} + 2^{1-H(S^{(1)}|T=t)}\right)$$

Same example when Helper does *not* know M (contd.)

$$\begin{aligned}
&\log\left(2^{1-H(S^{(0)}|T=t)} + 2^{1-H(S^{(1)}|T=t)}\right) \\
&\quad \le \log\left(2 - H(S^{(0)}|T=t) + 2 - H(S^{(1)}|T=t)\right) \\
&\quad \le \log\left(4 - H(S^{(0)}, S^{(1)}|T=t)\right)
\end{aligned}$$

Averaging over T and noting that $\log$ is concave, we have that capacity—with a scalar-quantization Helper who doesn't know M—is at most

$$\log\left(4 - H(S^{(0)}, S^{(1)}|T)\right) \le \log 3$$

(This bound is tight: It can be achieved when Helper always sends $T = S^{(0)}$).

Example when Helper is non-scalar (does not know M)

Allow Helper to be noncausal, and provide T to both transmitter and receiver.

First consider $t = h(s_1, s_2)$. Given $T = t$, capacity of the "sum channel" is

$$\log\left(2^{2-H(S_1^{(0)},S_2^{(0)}|T=t)} + 2^{2-H(S_1^{(0)},S_2^{(1)}|T=t)} + 2^{2-H(S_1^{(1)},S_2^{(0)}|T=t)} + 2^{2-H(S_1^{(1)},S_2^{(1)}|T=t)}\right)$$

$$= \log\Big(2^{1-H(S_1^{(0)}|T=t)} \cdot 2^{1-H(S_2^{(0)}|S_1^{(0)},T=t)} + 2^{1-H(S_1^{(0)}|T=t)} \cdot 2^{1-H(S_2^{(1)}|S_1^{(0)},T=t)}$$

$$+ 2^{1-H(S_1^{(1)}|T=t)} \cdot 2^{1-H(S_2^{(0)}|S_1^{(1)},T=t)} + 2^{1-H(S_1^{(1)}|T=t)} \cdot 2^{1-H(S_2^{(1)}|S_1^{(1)},T=t)}\Big)$$

$$\leq \log\Big(2^{1-H(S_1^{(0)}|T=t)} \cdot 2^{1-H(S_2^{(0)}|S_1^{(0)},S_1^{(1)},T=t)} + 2^{1-H(S_1^{(0)}|T=t)} \cdot 2^{1-H(S_2^{(1)}|S_1^{(0)},S_1^{(1)},T=t)}$$

$$+ 2^{1-H(S_1^{(1)}|T=t)} \cdot 2^{1-H(S_2^{(0)}|S_1^{(0)},S_1^{(1)},T=t)} + 2^{1-H(S_1^{(1)}|T=t)} \cdot 2^{1-H(S_2^{(1)}|S_1^{(0)},S_1^{(1)},T=t)}\Big)$$

$$= \log\left(2^{1-H(S_1^{(0)}|T=t)} + 2^{1-H(S_1^{(1)}|T=t)}\right) + \log\left(2^{1-H(S_2^{(0)}|S_1,T=t)} + 2^{1-H(S_2^{(1)}|S_1,T=t)}\right)$$

$$\leq \log\left(4 - H(S_1|T=t)\right) + \log\left(4 - H(S_2|S_1, T=t)\right) \leq 2\log\left(4 - \frac{1}{2}H(S_1, S_2|T=t)\right)$$

Averaging over T we again get $\log 3$ per channel use.

The proof can be generalized to $t = h(s^n)$ for any finite n.

Example: Summary

Theorem

In the above example, when Helper knows the message,

$$\text{capacity} = 2 \text{ bits.}$$

When Helper does not know the message,

$$\text{capacity} = \log 3.$$

Both equations hold irrespectively of whether Helper is causal or noncausal. Furthermore, they hold irrespectively of whether or not the help is also given to the Receiver.

Example: Summary

Theorem

In the above example, when Helper knows the message,

$$\text{capacity} = 2 \text{ bits.}$$

When Helper does not know the message,

$$\text{capacity} = \log 3.$$

Both equations hold irrespectively of whether Helper is causal or noncausal. Furthermore, they hold irrespectively of whether or not the help is also given to the Receiver.

We have learned from this example:

1. Helper knowing the message M can indeed make a difference.
2. There is indeed a gap between our lower and upper bounds in the original setting (where Helper does not know M).

Let's go back to the original problem

Recall our upper bound was

$$\frac{1}{n}\sum_{i=1}^{n} I(U_i;Y_i)$$

with $U_i \triangleq (M, T^{i-1}, Y^{i-1})$. The "problem" was (unlike in the lower bound)

$$U_i \not\!\perp\!\!\!\perp (S_i, T_i)$$

Let's go back to the original problem

Recall our upper bound was

$$\frac{1}{n}\sum_{i=1}^{n} I(U_i;Y_i)$$

with $U_i \triangleq (M, T^{i-1}, Y^{i-1})$. The "problem" was (unlike in the lower bound)

$$U_i \not\perp\!\!\!\perp (S_i, T_i)$$

Roughly speaking, possible dependence between U_i and T_i comes via T^{i-1}.

Let's go back to the original problem

Recall our upper bound was

$$\frac{1}{n}\sum_{i=1}^{n} I(U_i;Y_i)$$

with $U_i \triangleq (M, T^{i-1}, Y^{i-1})$. The "problem" was (unlike in the lower bound)

$$U_i \not\perp\!\!\!\perp (S_i, T_i)$$

Roughly speaking, possible dependence between U_i and T_i comes via T^{i-1}.

We want to find a situation where it is useful to convey T^{i-1} to the Receiver.

WARNING

The following example may cause headache to some audiences.

Example where scalar-quantization Helper is suboptimal

- State is two uniform bits as in previous example: $S = (S^{(0)}, S^{(1)})$
- Input has three parts: $X = (A, B, C)$

  where A and B are binary, while $|\mathcal{C}| = 2^\eta$
- Output also has three parts: $Y = (A', D^{(0)}, D^{(1)})$

  where A' is binary, while $D^{(0)}$ and $D^{(1)}$ each contains η bits
- Helper has one bit: $\mathcal{T} = \{0, 1\}$

Example where scalar-quantization Helper is suboptimal

- State is two uniform bits as in previous example: $S = (S^{(0)}, S^{(1)})$
- Input has three parts: $X = (A, B, C)$
  where A and B are binary, while $|\mathcal{C}| = 2^\eta$
- Output also has three parts: $Y = (A', D^{(0)}, D^{(1)})$
  where A' is binary, while $D^{(0)}$ and $D^{(1)}$ each contains η bits
- Helper has one bit: $\mathcal{T} = \{0, 1\}$

Channel law is:

- If $B \neq S^{(A)}$, then $Y \perp\!\!\!\perp (X, S)$ and uniform over its alphabet
- If $B = S^{(A)}$, then

$$A' = A, \qquad D^{(B)} = C, \qquad D^{(B\oplus 1)} \perp\!\!\!\perp X.$$

Example where scalar-quantization Helper is suboptimal

- State is two uniform bits as in previous example: $S = (S^{(0)}, S^{(1)})$
- Input has three parts: $X = (A, B, C)$

  where A and B are binary, while $|\mathcal{C}| = 2^\eta$
- Output also has three parts: $Y = (A', D^{(0)}, D^{(1)})$

  where A' is binary, while $D^{(0)}$ and $D^{(1)}$ each contains η bits
- Helper has one bit: $\mathcal{T} = \{0, 1\}$

Channel law is:

- If $B \neq S^{(A)}$, then $Y \perp\!\!\!\perp (X, S)$ and uniform over its alphabet
- If $B = S^{(A)}$, then

$$A' = A, \qquad D^{(B)} = C, \qquad D^{(B \oplus 1)} \perp\!\!\!\perp X.$$

You really want B to equal $S^{(A)}$...

Coding scheme for this example with non-scalar quantizer

Channel law:

- If $B \neq S^{(A)}$, then $Y \perp\!\!\!\perp (X, S)$
- If $B = S^{(A)}$, then $A' = A$, $D^{(B)} = C$, $D^{(B \oplus 1)} \perp\!\!\!\perp X$.

Let $T_0 \triangleq 0$. At time i,

$$\begin{aligned} T_i &= S_i^{(T_{i-1})} \\ A_i &= T_{i-1} \\ B_i &= T_i \\ C_i &\text{ carries } \eta \text{ information bits} \end{aligned}$$

Coding scheme for this example with non-scalar quantizer

Channel law:

- If $B \neq S^{(A)}$, then $Y \perp\!\!\!\perp (X, S)$
- If $B = S^{(A)}$, then $A' = A$, $D^{(B)} = C$, $D^{(B \oplus 1)} \perp\!\!\!\perp X$.

Let $T_0 \triangleq 0$. At time i,

$$\begin{aligned} T_i &= S_i^{(T_{i-1})} \\ A_i &= T_{i-1} \\ B_i &= T_i \\ C_i & \text{ carries } \eta \text{ information bits} \end{aligned}$$

How this scheme works:

- Always guaranteed that $B = S^{(A)}$
- At time i, Receiver learns B_{i-1}, so it can recover C_{i-1} from $D_i^{(B_{i-1})}$
  $\Longrightarrow$ we achieve η bits per channel use

Example: Scalar-quantization Helper is always worse

Channel law:

- If $B \neq S^{(A)}$, then $Y \perp\!\!\!\perp (X, S)$
- If $B = S^{(A)}$, then $A' = A$, $D^{(B)} = C$, $D^{(B \oplus 1)} \perp\!\!\!\perp X$.

First consider $T_i = S_i^{(0)}$.

- We must avoid $B \neq S^{(A)}$, so we must always choose $A = 0$ and $B = T$.
- $D^{(0)}$ or $D^{(1)}$ equals C, but Receiver does not know which one.

  For large η, the achieved rate $\approx \eta - 1$ bits.

Example: Scalar-quantization Helper is always worse

Channel law:

- If $B \neq S^{(A)}$, then $Y \perp\!\!\!\perp (X, S)$
- If $B = S^{(A)}$, then $A' = A$, $D^{(B)} = C$, $D^{(B \oplus 1)} \perp\!\!\!\perp X$.

First consider $T_i = S_i^{(0)}$.

- We must avoid $B \neq S^{(A)}$, so we must always choose $A = 0$ and $B = T$.
- $D^{(0)}$ or $D^{(1)}$ equals C, but Receiver does not know which one.

  For large η, the achieved rate $\approx \eta - 1$ bits.

Other scalar quantizers are even worse. We can list all possible scalar quantizers and upper-bound the rate that is achievable with each quantizer. Proof omitted.

Some reflections

- Why on earth would Helper want to tell Transmitter anything about S^{i-1} at time i?

Some reflections

- Why on earth would Helper want to tell Transmitter anything about S^{i-1} at time i?

  Perhaps it doesn't want to tell Transmitter anything "about" S^{i-1}.

  But WHICH PART of S_i to convey can depend on S^{i-1}.

Some reflections

- Why on earth would Helper want to tell Transmitter anything about S^{i-1} at time i?

  Perhaps it doesn't want to tell Transmitter anything "about" S^{i-1}.

  But WHICH PART of S_i to convey can depend on S^{i-1}.

- In fact, giving S^{i-1} to Transmitter doesn't change anything!

Some reflections

- Why on earth would Helper want to tell Transmitter anything about S^{i-1} at time i?

  Perhaps it doesn't want to tell Transmitter anything "about" S^{i-1}.

  But WHICH PART of S_i to convey can depend on S^{i-1}.

- In fact, giving S^{i-1} to Transmitter doesn't change anything!

- In the example, T_i wants to depend on S^{i-1} because Transmitter wants to tell Receiver something about S^{i-1},

Some reflections

- Why on earth would Helper want to tell Transmitter anything about S^{i-1} at time i?

  Perhaps it doesn't want to tell Transmitter anything "about" S^{i-1}.

  But WHICH PART of S_i to convey can depend on S^{i-1}.

- In fact, giving S^{i-1} to Transmitter doesn't change anything!

- In the example, T_i wants to depend on S^{i-1} because Transmitter wants to tell Receiver something about S^{i-1},

  which is why scalar quantization is optimal when Receiver knows S^n!

Some reflections

- Why on earth would Helper want to tell Transmitter anything about S^{i-1} at time i?

  Perhaps it doesn't want to tell Transmitter anything "about" S^{i-1}.

  But WHICH PART of S_i to convey can depend on S^{i-1}.

- In fact, giving S^{i-1} to Transmitter doesn't change anything!

- In the example, T_i wants to depend on S^{i-1} because Transmitter wants to tell Receiver something about S^{i-1},

  which is why scalar quantization is optimal when Receiver knows S^n!

- This can be generalized to a block-Markov scheme; details omitted.

Summary

- Main finding: for a DMC with a memoryless state sequence,

  scalar-quantization Helper + Shannon strategy at Transmitter

  need not be optimal

Summary

- Main finding: for a DMC with a memoryless state sequence,

  scalar-quantization Helper + Shannon strategy at Transmitter

  need not be optimal

- They become optimal if states are revealed to Receiver

Summary

- Main finding: for a DMC with a memoryless state sequence,

  scalar-quantization Helper + Shannon strategy at Transmitter

  need not be optimal

- They become optimal if states are revealed to Receiver

- If Helper knows the message, then (message-dependent) scalar quantizer is optimal

Summary

- Main finding: for a DMC with a memoryless state sequence,

  scalar-quantization Helper + Shannon strategy at Transmitter

  need not be optimal
- They become optimal if states are revealed to Receiver
- If Helper knows the message, then (message-dependent) scalar quantizer is optimal
- Helper knowing the message increases capacity

Thank you!