An AVC perspective on source/channel separation

Anant Sahai
joint work with:
Mukul Agarwal    Sanjoy K. Mitter

EE 290S

Advanced Information Theory
Suppose the aliens landed . . .

- Your mission: reverse-engineer their communications technology
Suppose the aliens landed... 

- Your mission: reverse-engineer their communications technology
- What assumptions should you make?
Suppose the aliens landed…

- Your mission: reverse-engineer their communications technology
- What assumptions should you make?
Does information theory determine architecture?

Channel coding is an “optimal” interface.
Does information theory determine architecture?

- Channel coding is an “optimal” interface.
- Is it *the* unique interface?

Fig. 1 — Schematic diagram of a general communication system.
What is an interface?

Virtual "lossless transfer" of bits

An interface abstracts away the details
What is an interface?

- An interface abstracts away the details
  - Channel code promises only $P_e < \epsilon$
What is an interface?

Virtual "lossless transfer" of bits

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

- An interface abstracts away the details
  - Channel code promises only $P_e < \epsilon$
  - No promises about how errors occur.
An interface abstracts away the details

- Channel code promises only $P_e < \epsilon$
- No promises about how errors occur.

It reuses a solution to a different problem.

- Transfer lots of bits with Hamming Distortion 0
What is an interface?

- An interface abstracts away the details
  - Channel code promises only $P_e < \epsilon$
  - No promises about how errors occur.
- It reuses a solution to a different problem.
  - Transfer lots of bits with Hamming Distortion 0
  - Source code: **reduction** of one problem to another

---

**Fig. 1** — Schematic diagram of a general communication system.
Equivalence proved using mutual information characterizations of $R(D)$ and $C_{\text{Shannon}}$. 
Classical separation revisited

- Equivalence proved using mutual information characterizations of $R(D)$ and $C_{\text{Shannon}}$.

- Bidirectional reductions at the problem level.
So could we have a different interface?

Virtual "lossy transfer" for a different source

Reuse the solution to a different problem
So could we have a different interface?

Virtual "lossy transfer" for a different source

Reuse the solution to a different problem
- Channel code promises $\frac{1}{n} \sum_{i=1}^{n} \rho(X_i, \hat{X}_i) \leq D$
- Channel code promises $P_e < \epsilon$

![Schematic diagram of a general communication system.](image)
So could we have a different interface?

Virtual "lossy transfer" for a different source

Reuse the solution to a different problem

- Channel code promises \( \frac{1}{n} \sum_{i=1}^{n} \rho(X_i, \hat{X}_i) \leq D \)
- Channel code promises \( P_e < \epsilon \)
- No promises about how errors occur.
So could we have a different interface?

Virtual "lossy transfer" for a different source

- Reuse the solution to a different problem
  - Channel code promises $\frac{1}{n} \sum_{i=1}^{n} \rho(X_i, \hat{X}_i) \leq D$
  - Channel code promises $P_e < \epsilon$
  - No promises about how errors occur.

- AVC where “Jammer” has access to codeword $\vec{X}$
  - “Jammer” is constrained in what he can do
  - We are constrained to codewords that look like $P(X)$
Main Result: “Reverse Source Coding”

Suppose we have a family of black-box systems indexed by $\epsilon$ that can communicate streams from input alphabet $\{\mathcal{X}\}$ satisfying:

$$P\left(\frac{1}{n_\epsilon} \sum_{i=1}^{n_\epsilon} \rho(X_i, Y_i) > D\right) \leq \epsilon$$
Main Result: “Reverse Source Coding”

Suppose we have a family of black-box systems indexed by $\epsilon$ that can communicate streams from input alphabet $\{\mathcal{X}\}$ satisfying:

$$P\left(\frac{1}{n\epsilon} \sum_{i=1}^{n\epsilon} \rho(X_i, Y_i) > D\right) \leq \epsilon$$

- The distortion measure $\rho$ is non-negative and additive.
Main Result: “Reverse Source Coding”

Suppose we have a family of black-box systems indexed by $\epsilon$ that can communicate streams from input alphabet $\{\mathcal{X}\}$ satisfying:

$$P\left(\frac{1}{n}\sum_{i=1}^{n\epsilon} \rho(X_i, Y_i) > D\right) \leq \epsilon$$

- The distortion measure $\rho$ is non-negative and additive.
- The probability measure $P$ is for $\{X_i\}$ iid according to $P(x)$.
Main Result: “Reverse Source Coding”

Suppose we have a family of black-box systems indexed by $\epsilon$ that can communicate streams from input alphabet $\{X\}$ satisfying:

$$P\left(\frac{1}{n_\epsilon} \sum_{i=1}^{n_\epsilon} \rho(X_i, Y_i) > D\right) \leq \epsilon$$

- The distortion measure $\rho$ is non-negative and additive.
- The probability measure $P$ is for $\{X_i\}$ iid according to $P(x)$

Then, assuming access to common-randomness at both the encoder and decoder, it is possible to choose $\epsilon$ and reliably communicate $n_\epsilon R$ bits at all rates $R < R(D)$ bits per source symbol over the black-box system so that the probability of error is as small as desired.
Finite alphabet case

- Random encoder

- “Nearest typical neighbor” decoder

- Error events:
Finite alphabet case

- Random encoder
  - Randomly draw $2^{nR}$ length $n$ codewords iid according to $P_X$ using common randomness.

- “Nearest typical neighbor” decoder

- Error events:
Finite alphabet case

- Random encoder
  - Randomly draw $2^{nR}$ length $n$ codewords iid according to $P_X$ using common randomness.
  - Transmit the one corresponding to the message.
- “Nearest typical neighbor” decoder

- Error events:
Finite alphabet case

- Random encoder
  - Randomly draw $2^{nR}$ length $n$ codewords iid according to $P_X$ using common randomness.
  - Transmit the one corresponding to the message.
- “Nearest typical neighbor” decoder
  - Define the typical codeword set $C_R$ to be codewords with type $P_X \pm \epsilon'$ for small $\epsilon'$.

Error events:
Finite alphabet case

- Random encoder
  - Randomly draw $2^n R$ length $n$ codewords iid according to $P_X$ using common randomness.
  - Transmit the one corresponding to the message.

- “Nearest typical neighbor” decoder
  - Define the typical codeword set $C_R$ to be codewords with type $P_X \pm \epsilon'$ for small $\epsilon'$.
  - Let $y_1^n$ be the received string.

- Error events:
Finite alphabet case

- Random encoder
  - Randomly draw $2^nR$ length $n$ codewords iid according to $P_X$ using common randomness.
  - Transmit the one corresponding to the message.
- “Nearest typical neighbor” decoder
  - Define the typical codeword set $C_R$ to be codewords with type $P_X \pm \epsilon'$ for small $\epsilon'$.
  - Let $y_1^n$ be the received string.
  - Decode to $\hat{X}_1^n \in C_R$ closest to $y_1^n$ in a $\rho$-distortion sense.
- Error events:
Finite alphabet case

Random encoder
- Randomly draw $2^{nR}$ length $n$ codewords iid according to $P_X$ using common randomness.
- Transmit the one corresponding to the message.

“Nearest typical neighbor” decoder
- Define the typical codeword set $C_R$ to be codewords with type $P_X \pm \epsilon'$ for small $\epsilon'$.
- Let $y_1^n$ be the received string.
- Decode to $\hat{X}_1^n \in C_R$ closest to $y_1^n$ in a $\rho$-distortion sense.

Error events:
- Atypical codeword $X_1^n$. Probability tends to 0.
Finite alphabet case

- Random encoder
  - Randomly draw $2^nR$ length $n$ codewords iid according to $P_X$ using common randomness.
  - Transmit the one corresponding to the message.

- “Nearest typical neighbor” decoder
  - Define the typical codeword set $C_R$ to be codewords with type $P_X \pm \epsilon'$ for small $\epsilon'$.
  - Let $y_1^n$ be the received string.
  - Decode to $\hat{X}_1^n \in C_R$ closest to $y_1^n$ in a $\rho$-distortion sense.

- Error events:
  - Atypical codeword $X_1^n$. Probability tends to 0.
  - Excess average distortion ($> D$) on true codeword. Probability tends to 0 by assumption.
Finite alphabet case

- Random encoder
  - Randomly draw $2^{nR}$ length $n$ codewords iid according to $P_X$ using common randomness.
  - Transmit the one corresponding to the message.

- “Nearest typical neighbor” decoder
  - Define the typical codeword set $C_R$ to be codewords with type $P_X \pm \epsilon'$ for small $\epsilon'$.
  - Let $y_1^n$ be the received string.
  - Decode to $\hat{X}_1^n \in C_R$ closest to $y_1^n$ in a $\rho$-distortion sense.

- Error events:
  - Atypical codeword $X_1^n$. Probability tends to 0.
  - Excess average distortion ($> D$) on true codeword. Probability tends to 0 by assumption.
  - There exists a false codeword has average distortion $< D$
Suppose $y_1^n$ has type $q_Y$. Consider $z_1^n \in C_R$ that is false. Let $q_{XY}$ be the resulting joint type.
Suppose $y^n_1$ has type $q_Y$. Consider $z^n_1 \in C_R$ that is false. Let $q_{XY}$ be the resulting joint type.

$q_X \in p_X \pm \epsilon'$
Probability of false codeword being close

- Suppose $y^n_1$ has type $q_Y$. Consider $z^n_1 \in C_R$ that is false. Let $q_{XY}$ be the resulting joint type.
- $q_X \in p_X \pm \epsilon'$
- $E_{q_{XY}}[\rho(X, Y)] \leq D$ if an error.
Probability of false codeword being close

- Suppose $y_1^n$ has type $q_Y$. Consider $z_1^n \in C_R$ that is false. Let $q_{XY}$ be the resulting joint type.
- $q_X \in p_X \pm \epsilon'$
- $E_{q_{XY}}[\rho(X, Y)] \leq D$ if an error.

Sort $y_1^n$ and correspondingly shuffle codeword $z_1^n$
Probability of false codeword being close

- Suppose \(y^n_i\) has type \(q_Y\). Consider \(z^n_i \in C_R\) that is false. Let \(q_{XY}\) be the resulting joint type.
- \(q_X \in p_X \pm \epsilon'\)
- \(E_{q_{XY}}[\rho(X, Y)] \leq D\) if an error.

Sort \(y^n_i\) and correspondingly shuffle codeword \(z^n_i\)

\[
P(Z^n_1 \text{ collides}) \leq \prod_j 2^{-nq_Y(j)D(q_{X|Y=j}\|p_X)} = 2^{-nD(q_{XY}\|p_Xq_Y)}
\]
Bounding probability of collision

- Union bound over $2^{nR}$ codewords and $(n + 1)|\mathcal{X}||\mathcal{Y}|$ possible joint types $q_{XY}$.
- Minimize divergence $D(q_{XY}||p_X q_Y)$ over set of joint types $q_{XY}$ satisfying:
  
  - $E_{q_{XY}}[\rho(X, Y)] \leq D$
  - $q_X \in p_X \pm \epsilon'$
Bounding probability of collision

- Union bound over $2^{nR}$ codewords and $(n + 1)|\mathcal{X}||\mathcal{Y}|$ possible joint types $q_{XY}$.
- Minimize divergence $D(q_{XY}||p_Xq_Y)$ over set of joint types $q_{XY}$ satisfying:
  
  ▶ $E_{q_{XY}}[\rho(X, Y)] \leq D$
  
  ▶ $q_X \in p_X \pm \epsilon'$

- Key step:
  
  $D(q_{XY}||p_Xq_Y) = D(q_X||p_X) + D(q_{XY}||q_Xq_Y) \leq D(q_{XY}||q_Xq_Y) = I_q(X; Y)$
Bounding probability of collision

- Union bound over $2^{nR}$ codewords and $(n + 1)|\mathcal{X}|\mathcal{Y}$ possible joint types $q_{XY}$.
- Minimize divergence $D(q_{XY}||p_{X}q_{Y})$ over set of joint types $q_{XY}$ satisfying:
  - $E_{q_{XY}}[\rho(X, Y)] \leq D$
  - $q_X \in p_X \pm \epsilon'$
- Key step:
  $D(q_{XY}||p_{X}q_{Y}) = D(q_{X}||p_{X}) + D(q_{XY}||q_{X}q_{Y}) \leq D(q_{XY}||q_{X}q_{Y}) = I_q(X; Y)$
- Minimizing $I_q(X; Y)$ gives $R(D)$ when $\epsilon' \to 0$
Bounding probability of collision

- Union bound over $2^{nR}$ codewords and $(n + 1)|\mathcal{X}||\mathcal{Y}|$ possible joint types $q_{XY}$.
- Minimize divergence $D(q_{XY}||p_Xq_Y)$ over set of joint types $q_{XY}$ satisfying:
  - $E_{q_{XY}}[\rho(X, Y)] \leq D$
  - $q_X \in p_X \pm \epsilon'$
- Key step:
  $$D(q_{XY}||p_Xq_Y) = D(q_X||p_X) + D(q_{XY}||q_Xq_Y) \leq D(q_{XY}||q_Xq_Y) = I_q(X; Y)$$
- Minimizing $I_q(X; Y)$ gives $R(D)$ when $\epsilon' \to 0$
- If $R < R(D)$, collision probability $\to 0$ with $n$. 

Anant Sahai (290S)  Inverse Rate Distortion  Nov 15, 2006  12 / 19
Can we reduce the randomization requirements?

- Under the current promises
  - Jammer can completely corrupt a tiny fraction of inputs
Can we reduce the randomization requirements?

- Under the current promises
  - Jammer can completely corrupt a tiny fraction of inputs
  - Is this what we want to allow?
Can we reduce the randomization requirements?

- Under the current promises
  - Jammer can completely corrupt a tiny fraction of inputs
  - Is this what we want to allow?

- Let’s renegotiate!
  - Have to be reasonable
  - Allow Jammer to kill before seeing the codeword. (channel noise)
  - Otherwise require good treatment of typical sources.
Can we reduce the randomization requirements?

- Under the current promises
  - Jammer can completely corrupt a tiny fraction of inputs
  - Is this what we want to allow?
- Let’s renegotiate!
  - Have to be reasonable
  - Allow Jammer to kill before seeing the codeword. (channel noise)
  - Otherwise require good treatment of typical sources.
- Use list-decoding
  - A distortion ball need not contain too many codewords
Extending to real vector alphabets

- Compact support and difference-distortion
  - Same codebook construction
Extending to real vector alphabets

- Compact support and difference-distortion
  - Same codebook construction
  - Pick a fine enough quantization $\Delta$
Extending to real vector alphabets

- Compact support and difference-distortion
  - Same codebook construction
  - Pick a fine enough quantization $\Delta$
  - Reduces to finite alphabet case at decoder with a small factor increase in distortion.
Extending to real vector alphabets

- **Compact support and difference-distortion**
  - Same codebook construction
  - Pick a fine enough quantization $\Delta$
  - Reduces to finite alphabet case at decoder with a small factor increase in distortion.

- **Unbounded support**
  - New codebook construction:
    - View $F_X(x) = (1 - \delta)F_{\tilde{X}}(x) + \delta F_{\tilde{X}}(x)$ where $\tilde{X}$ has compact support.
Extending to real vector alphabets

- **Compact support and difference-distortion**
  - Same codebook construction
  - Pick a fine enough quantization $\Delta$
  - Reduces to finite alphabet case at decoder with a small factor increase in distortion.

- **Unbounded support**
  - New codebook construction:
    - View $F_X(x) = (1 - \delta)F_{\tilde{X}}(x) + \delta F_{\bar{X}}(x)$ where $\tilde{X}$ has compact support.
    - First flip $n$ iid unfair coins with $P(\text{head}) = \delta$.
    - Mark positions as dirty or clean.
Extending to real vector alphabets

- Compact support and difference-distortion
  - Same codebook construction
  - Pick a fine enough quantization $\Delta$
  - Reduces to finite alphabet case at decoder with a small factor increase in distortion.

- Unbounded support
  - New codebook construction:
    - View $F_X(x) = (1 - \delta) F_{\tilde{X}}(x) + \delta F_{\bar{X}}(x)$ where $\tilde{X}$ has compact support.
    - First flip $n$ iid unfair coins with $P(\text{head}) = \delta$.
    - Mark positions as dirty or clean.
    - Draw codewords from $\tilde{X}$ in clean positions and $\bar{X}$ in dirty ones.
Extending to real vector alphabets

- Compact support and difference-distortion
  - Same codebook construction
  - Pick a fine enough quantization $\Delta$
  - Reduces to finite alphabet case at decoder with a small factor increase in distortion.

- Unbounded support
  - New codebook construction:
    - View $F_X(x) = (1 - \delta)F_{\tilde{X}}(x) + \delta F_{\bar{X}}(x)$ where $\tilde{X}$ has compact support.
    - First flip $n$ iid unfair coins with $P(\text{head}) = \delta$.
    - Mark positions as dirty or clean.
    - Draw codewords from $\tilde{X}$ in clean positions and $\bar{X}$ in dirty ones.
  - Modified decoding rule:
    - Declare error if fewer than $(1 - 2\delta)n$ clean positions.
    - Restrict codebook to clean positions only for decoding purposes.
Extending to real vector alphabets

- Compact support and difference-distortion
  - Same codebook construction
  - Pick a fine enough quantization $\Delta$
  - Reduces to finite alphabet case at decoder with a small factor increase in distortion.

- Unbounded support
  - New codebook construction:
    - View $F_X(x) = (1 - \delta)F_{\tilde{X}}(x) + \delta F_{\bar{X}}(x)$ where $\tilde{X}$ has compact support.
    - First flip $n$ iid unfair coins with $P(\text{head}) = \delta$.
    - Mark positions as dirty or clean.
    - Draw codewords from $\tilde{X}$ in clean positions and $\bar{X}$ in dirty ones.
  - Modified decoding rule:
    - Declare error if fewer than $(1 - 2\delta)n$ clean positions.
    - Restrict codebook to clean positions only for decoding purposes.
  - Increases distortion by a factor of $\frac{1 + 2\delta}{1 - 2\delta}$. 

Anant Sahai (290S)  
Inverse Rate Distortion  
Nov 15, 2006 14 / 19
Extending to real vector alphabets

- Compact support and difference-distortion
  - Same codebook construction
  - Pick a fine enough quantization $\Delta$
  - Reduces to finite alphabet case at decoder with a small factor increase in distortion.

- Unbounded support
  - New codebook construction:
    - View $F_X(x) = (1 - \delta)F_{\tilde{X}}(x) + \delta F_{\bar{X}}(x)$ where $\tilde{X}$ has compact support.
    - First flip $n$ iid unfair coins with $P(\text{head}) = \delta$.
    - Mark positions as dirty or clean.
    - Draw codewords from $\tilde{X}$ in clean positions and $\bar{X}$ in dirty ones.
  - Modified decoding rule:
    - Declare error if fewer than $(1 - 2\delta)n$ clean positions.
    - Restrict codebook to clean positions only for decoding purposes.
  - Increases distortion by a factor of $\frac{1 + 2\delta}{1 - 2\delta}$.
  - $\lim_{\Delta \to 0} \lim_{\delta \to 0} R_{\Delta, \delta} \left( D \frac{1 + 2\delta}{1 - 2\delta} \right) = R(D)$
Conditional Rate-Distortion

- Assume “coverstory” $V^n_1$ drawn according to $P(V)$ is known to all parties: encoder, decoder, and attacker.
- All $R < R_{X|V}(D)$ are achievable.
Assume “coverstory” $V_1^n$ drawn according to $P(V)$ is known to all parties: encoder, decoder, and attacker.

All $R < R_{X|V}(D)$ are achievable.

- Encoder draws conditionally using $P(X|V)$. 

Assume “coverstory” $V_1^n$ drawn according to $P(V)$ is known to all parties: encoder, decoder, and attacker.

All $R < R_{X|V}(D)$ are achievable.

- Encoder draws conditionally using $P(X|V)$.
- Codeword typicality defined similarly (include $V_1^n$)
Assume “coverstory” $V_1^n$ drawn according to $P(V)$ is known to all parties: encoder, decoder, and attacker.

All $R < R_{X|V}(D)$ are achievable.

- Encoder draws conditionally using $P(X|V)$.
- Codeword typicality defined similarly (include $V_1^n$).
- Nearest-conditionally-typical codeword decoding.
Assume “coverstory” $V_1^n$ drawn according to $P(V)$ is known to all parties: encoder, decoder, and attacker.

All $R < R_{X|V}(D)$ are achievable.

- Encoder draws conditionally using $P(X|V)$.
- Codeword typicality defined similarly (include $V_1^n$).
- Nearest-conditionally-typical codeword decoding
- Parallel proof
Application: unstable Markov Processes: $R(D)$

\[ X_{t+1} = AX_t + W_t \text{ where } A > 1 \]
Accumulation: Look at \( \{X_{kn}\} \)
- Can embed \( R_1 < n \log_2 A \) bits per symbol
- These bits are recovered with anytime reliability if black-box has finite error moments.
Unstable Markov Processes: two kinds of information

- Accumulation: Look at \( \{X_{kn}\} \)
  - Can embed \( R_1 < n \log_2 A \) bits per symbol
  - These bits are recovered with anytime reliability if black-box has finite error moments.

- Dissipation: Look at \( \{X_{kn-1}^{kn-1} | X_{kn}\} \)
Unstable Markov Processes: two kinds of information

- **Accumulation:** Look at $\{X_{kn}\}$
  - Can embed $R_1 < n \log_2 A$ bits per symbol
  - These bits are recovered with anytime reliability if black-box has finite error moments.

- **Dissipation:** Look at $\{X_{k(n-1)+1}^{kn-1} | X_{kn}\}$
  - Can be transformed to look iid
  - Fall under our results
Unstable Markov Processes: two kinds of information

- **Accumulation:** Look at \( \{X_{kn}\} \)
  - Can embed \( R_1 < n \log_2 A \) bits per symbol
  - These bits are recovered with anytime reliability if black-box has finite error moments.

- **Dissipation:** Look at \( \{X_{kn-1}^{k(n-1)+1} | X_{kn}\} \)
  - Can be transformed to look iid
  - Fall under our results
  - Can embed \( R_2 < R(D) - \log_2 A \) bits per symbol
Unstable Markov Processes: two kinds of information

- **Accumulation:** Look at $\{X_{kn}\}$
  - Can embed $R_1 < n \log_2 A$ bits per symbol
  - These bits are recovered with anytime reliability if black-box has finite error moments.
- **Dissipation:** Look at $\{X_{k(n-1)+1}^{kn-1} | X_{kn}\}$
  - Can be transformed to look iid
  - Fall under our results
  - Can embed $R_2 < R(D) - \log_2 A$ bits per symbol

- Two-tiered nature of information-flow proved by direct reduction.
Conclusions and open problems

Traditional point-to-point source-channel separation is a consequence of a problem-level equivalence that can be proved using direct reductions in both directions.
Conclusions and open problems

- Can the gap between $R_{seq}(D)$ and $R(D)$ be used to carry information?
  - Yes, but nearest neighbor alone won’t do it.

Traditional point-to-point source-channel separation is a consequence of a problem-level equivalence that can be proved using direct reductions in both directions.
Conclusions and open problems

Traditional point-to-point source-channel separation is a consequence of a problem-level equivalence that can be proved using direct reductions in both directions.

- Can the gap between $R_{\text{seq}}(D)$ and $R(D)$ be used to carry information?
  - Yes, but nearest neighbor alone won’t do it.

- Open problems and project ideas
  - Can Wyner-Ziv be covered?
  - Is there another reason to suspect that “digital communication” is fundamental?