



Information Theory With Applications to Data Compression

Robert Bamler · Tutorial at IMPRS-IS Boot Camp 2024

While you're waiting:

If you brought a laptop (optional), please go to <https://bamler-lab.github.io/bootcamp24> and test if you can run the linked Google Colab notebook. You can also find the slides at this link.



Let's Debate

Slides and code available at:
<https://bamler-lab.github.io/bootcamp24>



1. Which of the following two messages contains more information?

- (a) "The instructor of this tutorial knows how to solve a quadratic equation."
- (b) "The instructor of this tutorial likes roller coasters."

2. Which of the following two pairs of quantities are more strongly correlated:

- (a) the *volumes* and *radii* of (spherical) glass marbles (of random sizes and colors)
- (b) the *volumes* and *masses* of glass marbles (of random sizes and colors)

So, What is Information Theory?



Information theory provides tools to analyze:

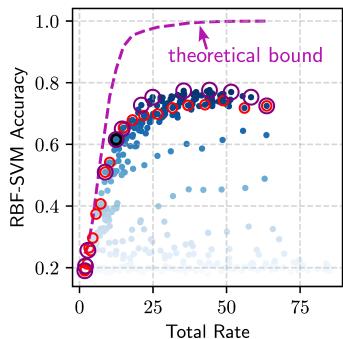
- ▶ the *quantity* (i.e., amount) of information in some data;
- ▶ more precisely, the amount of *novelty/surprisingness* of a piece of information w.r.t.:
 - (a) prior beliefs (e.g., an ML researcher probably knows high-school math); or
 - (b) a different piece of information (when quantifying correlations).

Information theory is oblivious to:

- ▶ the *quality* of a piece of information (e.g., its utility, urgency, or even truthfulness).
- ▶ how a piece of information is represented in the data, e.g.,
 - ▶ the volume and radius of a sphere are different representations of the same piece of information;
 - ▶ for a given neural network with known weights, its output cannot contain more information than its input.
- ▶ computational costs: compressed representations of the same information are sometimes easier but often harder to process than their uncompressed counterparts.

Where Are These Tools Useful?

Theoretical bounds for model performance



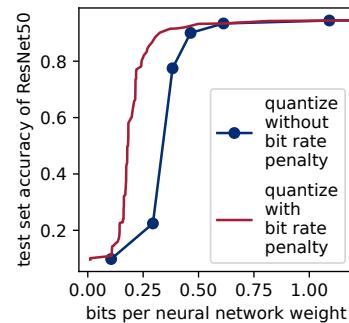
[Tim Xiao, RB, ICLR 2023]

Analyze abstract representation vectors

Metric	Dataset	Baseline
Specificity $MI(s; \ell) \uparrow$	Assist12	8.8
Consistency ⁻¹ $\mathbb{E}_{\ell_{\text{sub}}} MI(s^{\ell}; \ell_{\text{sub}}) \downarrow$	Assist12	12.1
Disentanglement $D_{\text{KL}}(s \ \ell) \uparrow$	Assist12	2.0
	Assist17	10.0
	Junyi15	13.0
	Assist17	6.0
	Junyi15	7.0
	Assist12	2.0
	Assist17	0.0
	Junyi15	5.0

[Hanqi Zhou, RB, C. M. Wu, Á. Tejero-Cantero, ICLR 2024]

Data Compression (“Source Coding”)



[Alexander Conzelmann, RB; coming soon]

My Promise for This Tutorial

Why?

What for?

Quantifying Information

[Shannon, *A Mathematical Theory of Communication*, 1948]

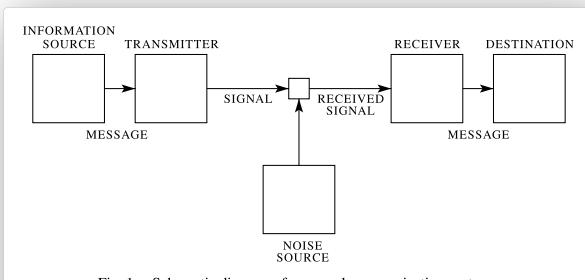
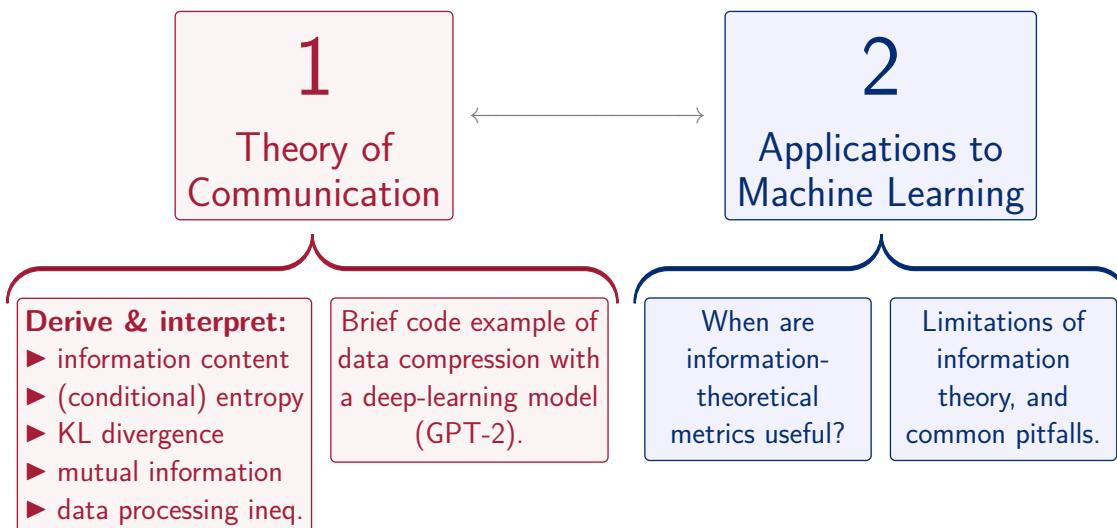


Fig. 1—Schematic diagram of a general communication system.

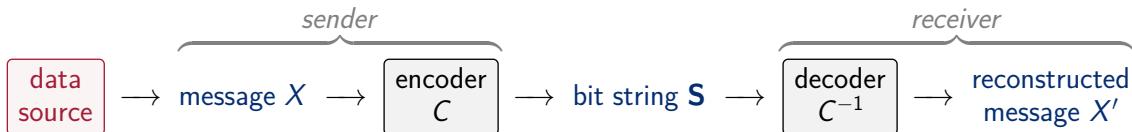
Def. “information content of a message”:

The **minimum number of bits** that you would have to transmit over a noise-free channel in order to communicate the message, *assuming an optimal encoder and decoder*.

- ▶ What does “optimal” mean?
- ▶ You don’t actually have to construct an optimal encoder & decoder to calculate this number.



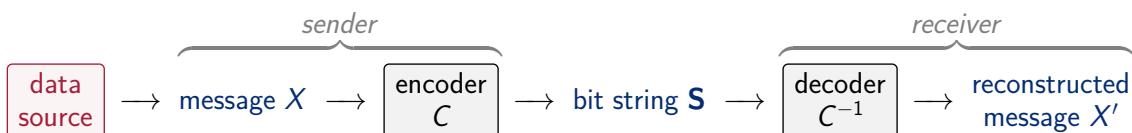
Data Compression: Precise Problem Setup



Assumptions:

- ▶ the bit string \mathbf{S} is sent over a *noise free* channel (we won't cover *channel coding*);
- ▶ *lossless* compression: we require that $X' = X$;
- ▶ \mathbf{S} may have a different length $|\mathbf{S}|$ for different messages: $\mathbf{S} \in \{0,1\}^* := \bigcup_{n=0}^{\infty} \{0,1\}^n$;
 - ▶ But: the encoder must *not* encode any information in the *length* of \mathbf{S} alone (see next slide).
- ▶ Before the sender sees the message, sender and receiver can communicate arbitrarily much for free in order to agree on a *code* $C : \text{message space } \mathcal{X} \rightarrow \{0,1\}^*$.
- ▶ **Goal:** find a valid code C that minimizes the *expected bit rate* $\mathbb{E}_{P_{\text{data source}}(X)}[|C(X)|]$.

What's a "Valid Code"? (Unique Decodability)



Recall:

- ▶ The bit string $\mathbf{S} = C(X) \in \{0,1\}^*$ can have different lengths for different messages X .
- ▶ We want to interpret its length $|\mathbf{S}|$ as the *amount of information* in the message X .
 - ▶ Seems to make sense: if the sender sends, e.g., a bit string of length 3 to the receiver, then they can't communicate more than 3 bits of information ...
 - ▶ ... unless the fact that $|\mathbf{S}| = 3$ already communicates some information. **We want to forbid this.**
- ▶ **Add additional requirement:** C must be *uniquely decodable*:
 - ▶ Sender may concatenate the encodings of several messages: $\mathbf{S} := C(X_1) \parallel C(X_2) \parallel C(X_3) \parallel \dots$
 - ▶ Upon receiving \mathbf{S} , the receiver must still be able to detect where each part ends.

Source Coding Theorem

Theorem (Shannon, 1949): Consider a data source $P(X)$ over a discrete message space \mathcal{X} .

- **The bad news:** in expectation, lossless compression can't beat the entropy:

$$\forall \text{ uniquely decodable codes } C: \mathbb{E}_P[|C(X)|] \geq \mathbb{E}_P[-\log_2 P(X)] =: H_P(X).$$

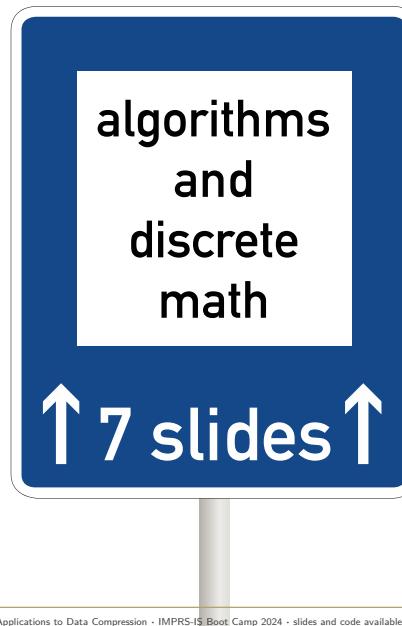
- **The good news:** but one can get quite close (and not just in expectation):

\exists uniquely decodable code C :

$$\begin{aligned} \forall \text{ messages } x \in \mathcal{X}: & |C(x)| < -\log_2 P(X=x) + 1. \\ (\Rightarrow \mathbb{E}_P[|C(X)|] & < H_P(X) + 1) \end{aligned}$$

- Also, we can keep the total overhead < 1 bit even when encoding *several* messages.

\Rightarrow *$-\log_2 P(X=x)$ is the contribution of message x to the bit rate of an optimal code when we amortize over many messages. It is called “information content of x ”.*



The Kraft-McMillan Theorem [Kraft, 1949; McMillan, 1956]

- (a) \forall uniquely decodable codes $C : \mathcal{X} \rightarrow \{0, 1\}^*$ over some message space \mathcal{X} :

$$\sum_{x \in \mathcal{X}} 2^{-|C(x)|} \leq 1 \quad (\text{"Kraft inequality"}).$$

Interpretation: we have a finite budget of “shortness” for bit strings:

- Interpret $2^{-|C(x)|}$ as the “shortness” of bit string $C(x)$.
- The sum of all “shortnesses” must not exceed 1.
 \Rightarrow If we shorten one bit string then we may have to make another bit string longer so that we don't exceed our “shortness budget”.

- (b) \forall functions $\ell : \mathcal{X} \rightarrow \mathbb{N}$ that satisfy the Kraft inequality (i.e., $\sum_{x \in \mathcal{X}} 2^{-\ell(x)} \leq 1$):
 \exists uniquely decodable code C_ℓ with $|C_\ell(x)| = \ell(x) \ \forall x \in \mathcal{X}$.

Why is this theorem useful? $\Rightarrow \min \mathbb{E}_P[\text{bit rate}] = \min_{\substack{C: \text{uniq.} \\ \text{decodable}}} \mathbb{E}_P[|C(X)|] = \min_{\substack{C: \text{satisfies} \\ \text{Kraft ineq.}}} \mathbb{E}_P[|C(X)|]$

Preparations for Proof of KM Theorem

Definition: For a code $C : \mathcal{X} \rightarrow \{0,1\}^*$, define

$$C^* : \mathcal{X}^* \rightarrow \{0,1\}^*, \quad C^*((x_1, x_2, \dots, x_k)) := C(x_1) \parallel C(x_2) \parallel \dots \parallel C(x_k).$$

(Thus: C is uniquely decodable $\iff C^*$ is injective)

Lemma:

- let: $\begin{cases} C \text{ be a uniquely decodable code over } \mathcal{X}; \\ n \in \mathbb{N}_0; \\ Y_n := \{\mathbf{x} \in \mathcal{X}^* \text{ with } |C^*(\mathbf{x})| = n\}. \end{cases}$
- then: $|Y_n| \leq 2^n$.

Proof:

Proof of Part (a) of KM Theorem

Lemma (reminder): $|Y_n| \leq 2^n$ where $Y_n := \{\mathbf{x} \in \mathcal{X}^* \text{ with } |C^*(\mathbf{x})| = n\}$, C uniq. dec.

Claim (reminder): C is uniquely decodable $\implies \sum_{x \in \mathcal{X}} 2^{-|C(x)|} \leq 1$.

(i) if \mathcal{X} is finite:

(ii) if \mathcal{X} is countably infinite:

Proof of Part (b) of KM Theorem

Claim (reminder): $\sum_{x \in \mathcal{X}} 2^{-\ell(x)} \leq 1 \implies \exists \text{ uniq. dec. code } C_\ell \text{ with } |C_\ell(x)| = \ell(x) \ \forall x \in \mathcal{X}$.

Algorithm 1: Construction of C_ℓ .

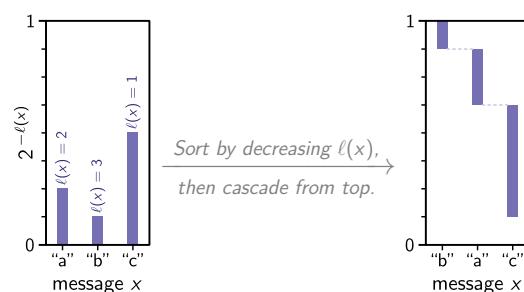
Initialize $\xi \leftarrow 1$;

for $x \in \mathcal{X}$ in order of nonincreasing $\ell(x)$ do

 Update $\xi \leftarrow \xi - 2^{-\ell(x)}$;

 Write $\xi \in [0, 1)$ in binary: $\xi = (0.??? \dots)_2$;

 Set $C_\ell(x)$ to the first $\ell(x)$ bits after the “0.”
(pad with trailing zeros if necessary);



Claim: the resulting code C_ℓ is uniquely decodable.

- We even show: C_ℓ is prefix free: $\forall x \in \mathcal{X}: C_\ell(x)$ is not the beginning of any $C_\ell(x')$, $x' \neq x$.
- Formalization of this proof: see solutions to Problem 2.1 on this problem set:
<https://robamler.github.io/teaching/compress23/problem-set-02-solutions.zip>

Example: Sum of Two Fair 3-Sided Dice

x	possible throws	$P(X=x)$	$\ell(x)$	$C_\ell(x)$
2	■ ■	1/9	3	
3	■ ■ ■, ■ ■ ■	2/9	2	
4	■ ■ ■ ■, ■ ■ ■ ■, ■ ■ ■ ■	1/3	2	
5	■ ■ ■ ■ ■, ■ ■ ■ ■ ■	2/9	2	
6	■ ■ ■ ■ ■ ■	1/9	3	

- ▶ Check if ℓ satisfies Kraft inequality: $\sum_{x \in \mathcal{X}} 2^{-\ell(x)} =$
- ▶ **Question:** how should we choose $\ell : \mathcal{X} \rightarrow \mathbb{N}$ for a given model P of the data source?
 - ▶ **typical goal:** minimize the expected bit rate $\mathbb{E}_P[\ell(X)]$ (with the constraint $\sum_{x \in \mathcal{X}} 2^{-\ell(x)} \leq 1$).
 - ▶ **optimally:** by *Huffman coding* (comp. cost $\propto |\mathcal{X}| \log |\mathcal{X}|$, i.e., exponential in message length).
 - ▶ **near optimally:** via *information content*; \rightarrow bounds on optimal $\mathbb{E}_P[\ell(X)]$ (next slide).

Optimal Choice of Target Length $\ell : \mathcal{X} \rightarrow \mathbb{N}$

- ▶ **Constrained optimization problem:**
 - ▶ Minimize $\mathbb{E}_P[\ell(X)] = \sum_{x \in \mathcal{X}} P(X=x) \ell(x)$ over ℓ
 - ▶ with the constraints: (i) $\sum_{x \in \mathcal{X}} 2^{-\ell(x)} \leq 1$
 - ▶ (ii) $\ell(x) \in \mathbb{N} \quad \forall x \in \mathcal{X}$
- ▶ **Idea:** relax constraint: (ii') $\ell(x) \in \mathbb{R}_{>0} \quad \forall x \in \mathcal{X}$
 - ⇒ Minimization runs over more functions ℓ .
 - ⇒ *lower bound*: $\inf_{(i),(ii')} \mathbb{E}_P[\ell(X)] \leq \inf_{(i),(ii)} \mathbb{E}_P[\ell(X)]$
- ▶ **Observation:** solution satisfies: (i') $\sum_{x \in \mathcal{X}} 2^{-\ell(x)} = 1$
 - ▶ Enforce via Lagrange multiplier $\lambda \in \mathbb{R}$:
 - find stationary point (w.r.t. both ℓ and λ) of $\mathcal{L}(\ell, \lambda) := \sum_{x \in \mathcal{X}} P(X=x) \ell(x) + \lambda \left(\sum_{x \in \mathcal{X}} 2^{-\ell(x)} - 1 \right)$

Proof of Source Coding Theorem

- ▶ Solution of the relaxed optimization problem: $\ell(x) = \underbrace{-\log_2 P(X=x)}_{\text{"information content"}} \in \mathbb{R}_{\geq 0}$.
- ▶ Let's now constrain $\ell(x)$ again to integer values $\forall x \in \mathcal{X}$.
 - ⇒ **lower bound** on expected bit rate ("the bad news"):

$$\mathbb{E}_P[|C(X)|] \geq \underbrace{\mathbb{E}_P[-\log_2 P(X=x)]}_{H_P(X)} \quad \forall \text{ uniquely decodable } C.$$

- ▶ **Upper bound** on the *optimal* expected bit rate ("the good news"):

- ▶ *Shannon Code*: set $\ell(x) := \lceil -\log_2 P(X=x) \rceil \in \mathbb{N}$.
- ▶ Satisfies Kraft inequality: $\sum_{x \in \mathcal{X}} 2^{-\lceil -\log_2 P(X=x) \rceil} \leq \sum_{x \in \mathcal{X}} 2^{\log_2 P(X=x)} = 1$.
- ⇒ \exists uniquely decodable code C_ℓ with:

$$|C_\ell(x)| = \ell(x) < -\log_2 P(X=x) + 1 \quad \forall x \in \mathcal{X}.$$

algorithms and discrete math

Ende

Quantifying Uncertainty in Bits (for Discrete Data)



- ▶ **Information content:** $-\log_2 P(X=x)$: The (amortized) bit rate for encoding the given message x with a code that is optimal (in expectation) for the data source P .
- ▶ **Entropy:** $H_P(X) = \mathbb{E}_P[-\log_2 P(X)] \equiv H[P(X)] \equiv H[P]$: The expected bit rate for encoding a (random) message from data source P with a code that is optimal for P .
= How many bits does receiver need (in expectation) to reconstruct X ?
= How many bits does receiver need (in expectation) to resolve any *uncertainty* about X ?
- ▶ **Cross entropy:** $H[P, Q] = \mathbb{E}_P[-\log_2 Q(X)] \geq H[P]$:
The expected bit rate when encoding a message from data source P with a code that is optimal for a model Q of the data source (\Rightarrow practically achievable expected bit rate).
 \rightarrow We'd want to minimize this over the model Q . \rightarrow Maximum likelihood estimation.
- ▶ **Kullback-Leibler divergence:** $D_{KL}(P \parallel Q) = H[P, Q] - H[P] = \mathbb{E}_P\left[-\log_2 \frac{Q(X)}{P(X)}\right] \geq 0$:
Overhead (in expected bit rate) due to a mismatch between the true data source P and its model Q (also called “*relative entropy*”).

Agenda



1

Theory of
Communication

2

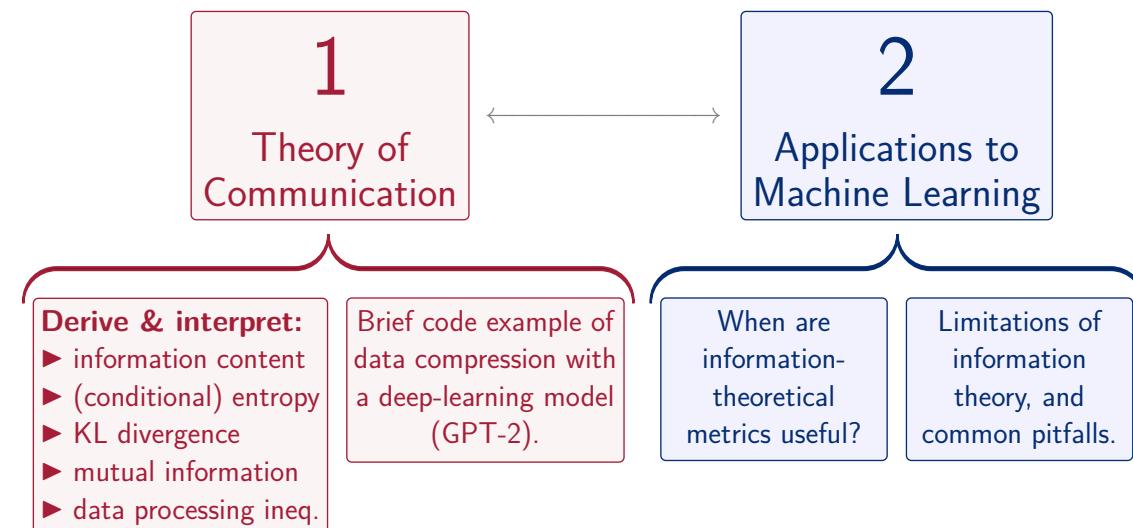
Applications to
Machine Learning

- Derive & interpret:**
- ▶ information content
 - ▶ (conditional) entropy
 - ▶ KL divergence
 - ▶ mutual information
 - ▶ data processing ineq.

Brief code example of
data compression with
a deep-learning model
(GPT-2).

When are
information-
theoretical
metrics useful?

Limitations of
information
theory, and
common pitfalls.



Joint, Marginal, and Conditional Entropy

Consider a data source $P(X, Y)$ that generates pairs $(x, y) \sim P$:

$$P(X, Y) = P(X) P(Y | X) = P(Y) P(X | Y).$$

► **Joint information content**, i.e., information content of the entire message (x, y) :

$$-\log_2 P(X=x, Y=y) = -\log_2 P(X=x) - \log_2 P(Y=y | X=x).$$

► **Joint entropy**:

$$H_P((X, Y)) = \mathbb{E}_{P(X, Y)}[-\log_2 P(X, Y)] = \mathbb{E}_{P(X) P(Y|X)}[-\log_2 P(X) - \log_2 P(Y|X)]$$

$$\begin{aligned} &= \underbrace{\mathbb{E}_{P(X)}[-\log_2 P(X)]}_{(\text{marginal}) \text{ entropy } H_P(X)} + \underbrace{\mathbb{E}_{x \sim P(X)}[\underbrace{\mathbb{E}_{P(Y|X=x)}[-\log_2 P(Y | X=x)]}_{=: H_P(Y | X=x) = \text{entropy of the}}]}_{=: \text{conditional entropy } H_P(Y | X)} \\ &\quad \text{conditional distribution } P(Y | X=x) \end{aligned}$$

► $H_P((X, Y)) = H_P(X) + H_P(Y | X) = H_P(Y) + H_P(X | Y)$

Mutual Information

Reminder: $H_P(Y | X) := \mathbb{E}_P[-\log_2 P(Y | X)] = \mathbb{E}_{x \sim P(X)}[\underbrace{\mathbb{E}_{P(Y|X=x)}[-\log_2 P(Y | X=x)]}_{=: H_P(Y | X=x) = \text{entropy of the}}]$

$$H_P((X, Y)) = H_P(X) + H_P(Y | X).$$

$$= H_P(Y | X=x) = \text{entropy of the conditional distribution } P(Y | X=x)$$

Let's encode a given message (x, y) :

- (a) encode x with optimal code for $P(X)$; then encode y with optimal code for $P(Y | X=x)$;
- (b) encode (x, y) using an optimal code for the data source $P(X, Y)$;
- (c) encode x with optimal code for $P(X | Y=y)$; then encode y with optimal code for $P(Y)$.
- (d) encode x with optimal code for $P(X)$; then encode y with optimal code for $P(Y)$;

Expected bit rate:

$H_P(X)$	$H_P(Y X)$
----------	--------------

$H_P((X, Y))$	
---------------	--

$H_P(X Y)$	$H_P(Y)$
--------------	----------

$H_P(X)$	$H_P(Y)$
----------	----------

Interpretations of the Mutual Information $I_P(X; Y)$



The following expressions for $I_P(X; Y)$ are equivalent:

$$\begin{aligned} \textcircled{1} \quad I_P(X; Y) &= H_P(X) + H_P(Y) - H_P((X, Y)) \\ &= D_{\text{KL}}(P(X, Y) \parallel P(X)P(Y)) \geq 0 \end{aligned}$$

Interpretation: how much would ignoring correlations between X, Y hurt expected compression performance?

$$\textcircled{2} \quad I_P(X; Y) = H_P(Y) - H_P(Y|X)$$

Interpretation: how many bits of information does knowledge of X tell us about Y (in expectation)?
(reduction of uncertainty, “expected information gain”)

$$\textcircled{3} \quad I_P(X; Y) = H_P(X) - H_P(X|Y)$$

Interpretation: how many bits of information does knowledge of Y tell us about X (in expectation)?

$H_P((X, Y))$		$I_P^{\textcircled{1}}$
$H_P(X)$	$H_P(Y)$	
$H_P(X)$	$H_P(Y X)$	$I_P^{\textcircled{2}}$
$I_P^{\textcircled{3}}$	$H_P(X Y)$	$H_P(Y)$

Note: “in expectation” is an important qualifier here.
Conditioning on a specific x can increase the entropy of Y :
 $H_P(Y|X) \leq H_P(Y)$ (always),
but:
 $H_P(Y|X=x) > H_P(Y)$ is possible for some (atypical) x .

Continuous Data (Pedestrian Approach)



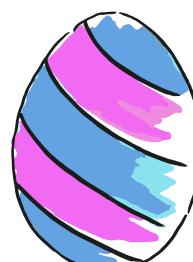
Recall: optimal lossless code C_{opt} for a data source P : $H_P(X) \leq \mathbb{E}_P[|C_{\text{opt}}(X)|] < H_P(X) + 1$

- ▶ Lossless compression is only possible on a *discrete* (i.e., countable) message space \mathcal{X} .
(Because $\mathcal{X} \xrightarrow{\text{lossless code } C \text{ (injective)}} \{0, 1\}^* \xrightarrow{\text{injective}} \mathbb{N}$.)

Simple *lossy* compression of a message $X \in \mathbb{R}^n$: (an “act of desperation” — M.P.)

- ▶ Require that reconstruction X' satisfies $|X'_i - X_i| < \frac{\delta}{2} \forall i \in \{1, \dots, n\}$ for some $\delta > 0$.
- ▶ Let $\hat{X} := \delta \times \text{round}\left(\frac{1}{\delta}X\right)$. $\implies |\hat{X}_i - X_i| \leq \frac{\delta}{2} \forall i$.
- ▶ Compress $\hat{X} \in \delta\mathbb{Z}^n$ losslessly using induced model $P(\hat{X})$. \implies Reconstruction $X' = \hat{X}$.
- ▶ $P(\hat{X} = \hat{x}) = P\left(X \in \bigtimes_{i=1}^n [\hat{x}_i - \frac{\delta}{2}, \hat{x}_i + \frac{\delta}{2}]\right) = \int_{X \in \bigtimes_{i=1}^n [\hat{x}_i - \frac{\delta}{2}, \hat{x}_i + \frac{\delta}{2}]} p(x) d^n x \approx \delta^n p(\hat{x}) + o(\delta^n)$
- ▶ $H_P(\hat{X}) \approx - \sum_{\hat{x} \in \delta\mathbb{Z}^n} \delta^n p(\hat{x}) \log_2(\delta^n p(\hat{x}))$
 $\approx - \int p(x) (\underbrace{\log_2 p(x)}_{\text{“differential entropy” } h_P(X)} + n \log_2 \delta) d^n x = \underbrace{\mathbb{E}_P[-\log_2 p(X)]}_{\delta \rightarrow 0 \rightarrow \infty} + n \log_2(1/\delta)$

How Does Discretization Relate to IMPRS-IS?



Easter egg for attendees of the tutorial.

(Not on handouts, sorry. You should have been there 😊.)

Examples of Differential Entropies

- ▶ **Uniform distribution:** $P(X) = \mathcal{U}(\mathcal{X})$
- ▶ **Density:** $p(x) = \frac{1}{\text{Vol}(\mathcal{X})} \quad \forall x \in \mathcal{X}$
- ▶ **Differential entropy:** $h_P(X) = \mathbb{E}_P[-\log_2 p(X)] = \mathbb{E}_P\left[-\log_2 \frac{1}{\text{Vol}(\mathcal{X})}\right] = \log_2(\text{Vol}(\mathcal{X}))$
- ▶ **Note:** if $\text{Vol}(\mathcal{X}) < 1$ then $h_P(X) < 0$.
→ Nothing to see here, h_P is only meaningful up to an infinite additive constant.
- ▶ **Normal distribution:** $P(X) = \mathcal{N}(\mu, \Sigma)$ (with $X, \mu \in \mathbb{R}^n$ and $\Sigma \in \mathbb{R}^{n \times n}$)
- ▶ **Density:** $p(x) = \mathcal{N}(x; \mu, \Sigma) = \frac{1}{\sqrt{\det(2\pi\Sigma)}} \exp\left[-\frac{1}{2}(x - \mu)^\top \Sigma^{-1} (x - \mu)\right]$
- ▶ **Differential entropy:** $h_P(X) = \mathbb{E}_P[-\log_2 p(X)] = \frac{1}{2} \log_2(\det \Sigma) + \underbrace{\frac{n}{2} \log_2(2\pi e)}_{\text{const.}}$

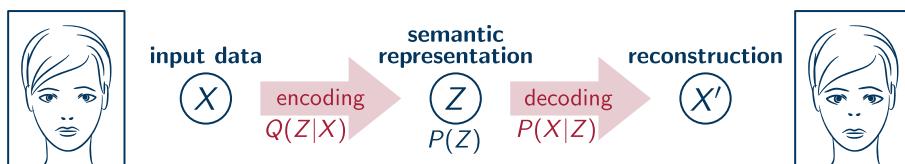
KL-Divergence Between Continuous Distributions

- ▶ **Differential entropy** (reminder): $h_P(X) = \mathbb{E}_P[-\log_2 p(X)]$
→ Relation to entropy of discretization \hat{X} : $H_P(\hat{X}) \approx h_P(X) + n \log_2(1/\delta) \xrightarrow{\delta \rightarrow 0} \infty$
- ▶ **Differential cross entropy** (less common): $h[P(X), Q(X)] = \mathbb{E}_P[-\log_2 q(X)]$
→ Relation to discretization: $H[P(\hat{X}), Q(\hat{X})] \approx h[P(X), Q(X)] + n \log_2(1/\delta) \xrightarrow{\delta \rightarrow 0} \infty$
- ▶ **Kullback-Leibler divergence** between discretized distributions $P(\hat{X})$ and $Q(\hat{X})$:

$$\begin{aligned} D_{\text{KL}}(P(\hat{X}) \| Q(\hat{X})) &= H[P(\hat{X}), Q(\hat{X})] - H_P(\hat{X}) \\ &\approx h[P(X), Q(X)] + n \log_2(1/\delta) - (h_P(X) + n \log_2(1/\delta)) \\ &= \mathbb{E}_P\left[-\log_2 \frac{q(X)}{p(X)}\right] =: D_{\text{KL}}(P(X) \| Q(X)) \xrightarrow{\text{(possibly)}} < \infty \end{aligned}$$
- ⇒ **Interpretation:** $D_{\text{KL}}(P \| Q)$ = modeling overhead, in the limit of infinitely fine quantization.
- ▶ **Generalization (density-free):** $D_{\text{KL}}(P \| Q) = - \int \log_2 \left(\frac{dQ}{dP} \right) dP$

(Variational) Information Bottleneck

- ▶ **Example:** β -variational autoencoder (similar for supervised models (Alemi et al., ICLR 2017))



- ▶ **Loss function:** $\mathbb{E}_{x \sim \text{data}} \left[\mathbb{E}_{Q(Z|X=x)} [-\log P(X=x | Z)] + \beta D_{\text{KL}}(Q(Z | X=x) \| P(Z)) \right]$

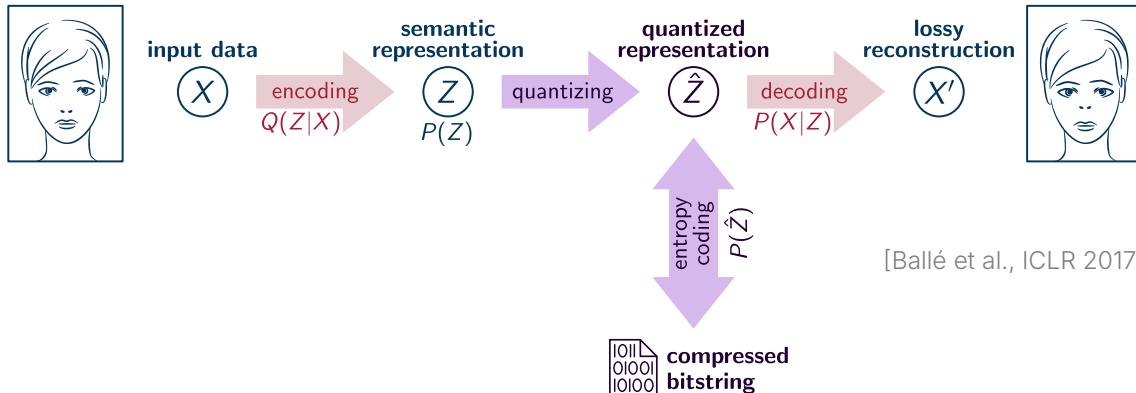
$$D_{\text{KL}}(\dots \| \dots) = \begin{cases} \text{information in } z \sim Q(Z | X=x) \\ \text{for someone who doesn't know } x \\ \text{(i.e., they only know } P(Z)) \end{cases} - \begin{cases} \text{information in } z \sim Q(Z | X=x) \\ \text{for someone who knows } x \\ \text{(i.e., they know } Q(Z | X=x)) \end{cases}$$

⇒ { Capture as much (x -independent) information about z in the prior $P(Z)$ as possible.
Encode as little (unnecessary) information in $Q(Z | X=x)$ as possible.

Remark: Data Compression With VAEs



- So far, no compression: Z still takes up lots of memory (even if its inf. content is low).
- Real compression has to actually reduce Z to its information content: *entropy coding*



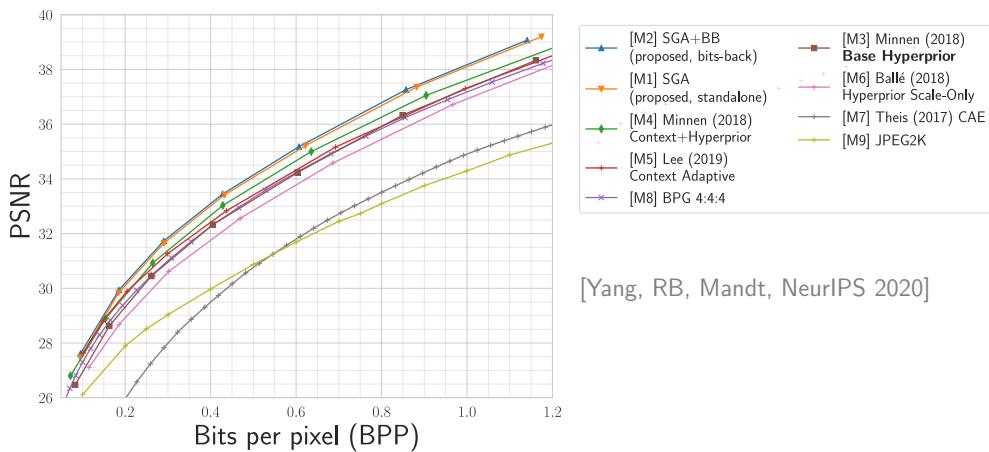
Robert Bamler - Universität Tübingen - Tutorial on Information Theory With Applications to Data Compression - IMPRS-IS Boot Camp 2024 - slides and code available at <https://bamler-lab.github.io/bootcamp24>

| 33

Rate/Distortion Trade-off



- Tuning β allows us to trade off *bit rate* against *distortion*.



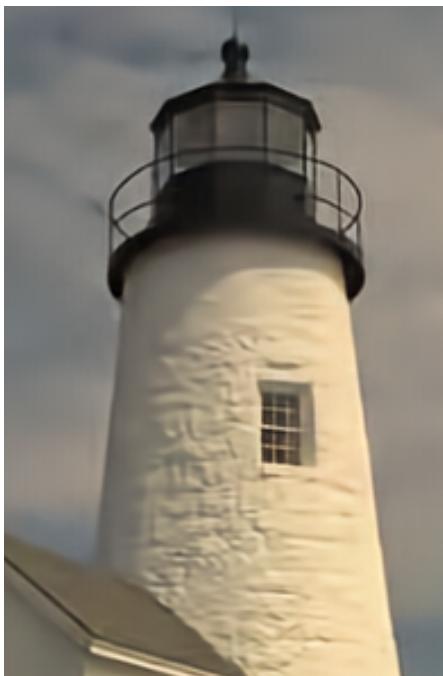
Robert Bamler - Universität Tübingen - Tutorial on Information Theory With Applications to Data Compression - IMPRS-IS Boot Camp 2024 - slides and code available at <https://bamler-lab.github.io/bootcamp24>

| 34





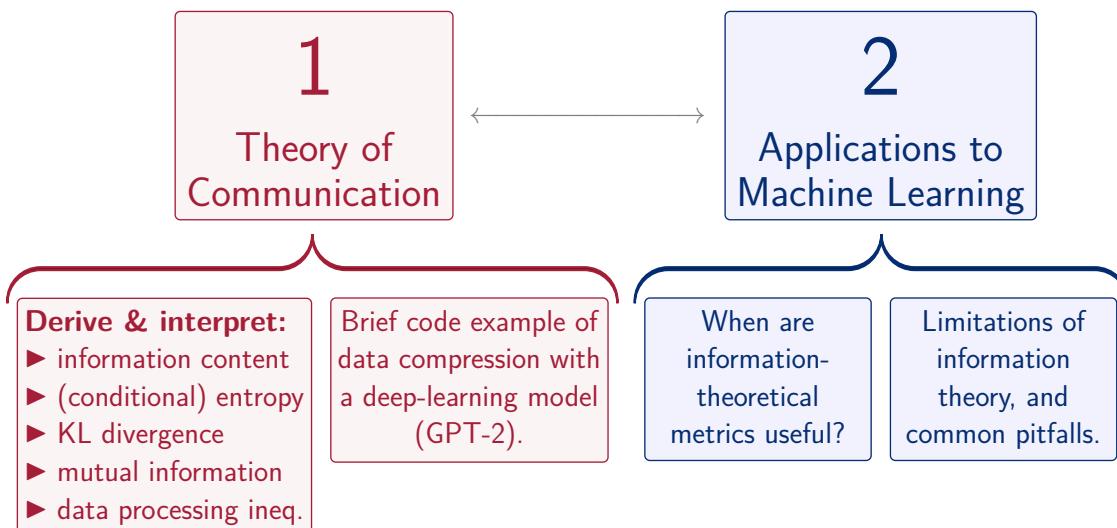
BPG 4:4:4
left: 0.143 bit/pixel
right: 0.14 bit/pixel



VAE-based
left: 0.142 bit/pixel
right: 0.13 bit/pixel
[Yang, RB, Mandt,
NeurIPS 2020]



JPEG
left: 0.142 bit/pixel
right: 0.14 bit/pixel



Mutual Information for Continuous Random Vars

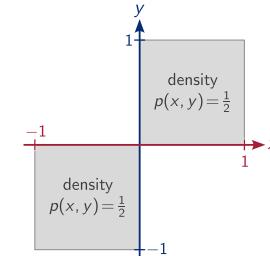


$I_P(X; Y) = D_{\text{KL}}(P(X, Y) \parallel P(X) P(Y)) = \mathbb{E}_P \left[-\log_2 \frac{p(X) p(Y)}{p(X, Y)} \right]$ (if densities p exist)

- ▶ **Exercise:** let $X' = f(X)$, $Y' = g(Y)$, where f and g are differentiable *injective* functions.
Convince yourself that I_P is independent of representation, i.e., $I_P(X'; Y') = I_P(X; Y)$.

Example 1a:

- ▶ $P(X) = P(Y) = \mathcal{U}([-1, 1]) \implies h_P(X) = h_P(Y) = \log_2(2) = 1.$
- ▶ $P(Y | X) = \begin{cases} \mathcal{U}([-1, 0)) & \text{if } X < 0; \\ \mathcal{U}(0, 1)) & \text{if } X \geq 0. \end{cases}$
- ▶ $\implies h_P(Y | X) = \mathbb{E}_{x \sim P(X)} [h_P(Y | X=x)] = \log_2(1) = 0.$
- ▶ **Mutual information:** $I_P(X; Y) = h_P(Y) - h_P(Y | X) = 1 - 0 = 1$ bit.
- ▶ **Interpretation:** observing X tells us (only) the sign of Y . $\implies 1$ bit of information.

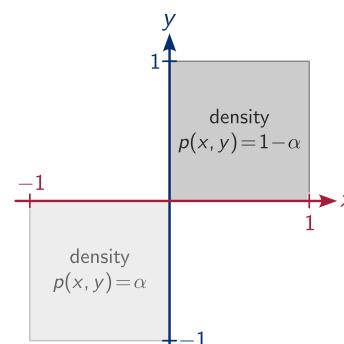


Mutual Information for Continuous Random Vars



Example 1b: non-uniform $P(X)$.

- ▶ $p(y) = \begin{cases} \alpha & \text{if } y \in [-1, 0); \\ 1 - \alpha & \text{if } y \in [0, 1]. \end{cases}$ (for $\alpha \in [0, 1]$)
- ▶ $\implies h_P(Y) = - \int_{-1}^1 p(y) \log_2 p(y) dy$
 $= -\alpha \log_2(\alpha) - (1 - \alpha) \log_2(1 - \alpha) =: H_2(\alpha)$
- ▶ $P(Y | X) = \begin{cases} \mathcal{U}([-1, 0)) & \text{if } X < 0; \\ \mathcal{U}(0, 1)) & \text{if } X \geq 0. \end{cases}$ (as before)
- ▶ $\implies h_P(Y | X) = 0$ (as before)
- ▶ **Mutual information:** $I_P(X; Y) = h_P(Y) - h_P(Y | X) = H_2(\alpha) - 0 = H_2(\alpha) \leq 1$ bit.
- ▶ **Interpretation:** observing X still tells us sign(Y) with certainty, but sign(Y) now carries less than one bit of information (in expectation) if $\alpha \neq \frac{1}{2}$.



Example 2: Gaussian Signal With Gaussian Noise

Consider an *analog* signal $x \sim \mathcal{N}(0, \sigma_s^2)$, sent over a *noisy* channel (e.g., voltage on a wire).

⇒ Receiver receives a somewhat corrupted signal: $y \sim \mathcal{N}(x, \sigma_n^2)$.

Mutual information: $I_P(X; Y) = h_P(Y) - h_P(Y | X)$

$$\blacktriangleright p(y) = \mathbb{E}_{P(X)}[p(y | X)] = \int \mathcal{N}(x; 0, \sigma_s^2) \mathcal{N}(y; x, \sigma_n^2) dx = \mathcal{N}(y; 0, \sigma_s^2 + \sigma_n^2)$$

$$\Rightarrow I_P(X; Y) = h_P(Y) - h_P(Y | X) = \frac{1}{2} \log_2(\sigma_s^2 + \sigma_n^2) - \frac{1}{2} \log_2(\sigma_n^2) = \frac{1}{2} \log_2\left(1 + \frac{\sigma_s^2}{\sigma_n^2}\right).$$

Interpretation: σ_s^2 / σ_n^2 is the *signal-to-noise ratio* (SNR).

- ▶ For $\text{SNR} \rightarrow 0$, we have $I_P(X; Y) \rightarrow 0$; ⇒ receiver receives no meaningful information.
- ▶ But, as long as $\text{SNR} > 0$, one can still extract *some* information from the received signal.
- ▶ In the theory of channel coding (aka error correction), $P(Y | X)$ models a communication channel. Its *channel capacity* $C := \sup_{P(X)} I_P(X; Y)$ is the number of bits that can be transmitted noise-free per invocation of the noisy channel (in the limit of long messages).

Data Processing Inequality I: Intuition

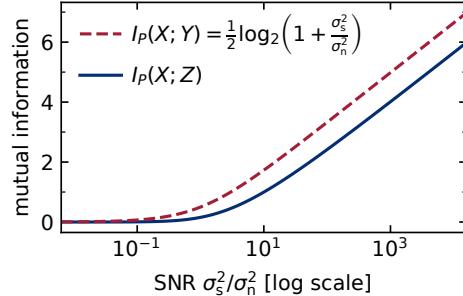
Remember when we were all still young and looking at slide 40?

$$I_P(X; Y) = D_{\text{KL}}(P(X, Y) \| P(X) P(Y)) = \mathbb{E}_P \left[-\log_2 \frac{p(X) p(Y)}{p(X, Y)} \right] \quad (\text{if densities } p \text{ exist})$$

- ▶ **Exercise:** let $X' = f(X)$, $Y' = g(Y)$, where f and g are differentiable *injective* functions. Convince yourself that I_P is independent of representation, i.e., $I_P(X'; Y') = I_P(X; Y)$.

Question: what do *non-injective* transformations do to the mutual information?

- ▶ **Example:** start from last slide:
 $X \sim \mathcal{N}(0, \sigma_s^2)$; $Y|X \sim \mathcal{N}(X, \sigma_n^2)$.
- ▶ Then, consider $Z := Y^2$.
- ▶ Is $I_P(X; Z)$ $\begin{cases} \text{larger than,} \\ \text{smaller than,} \\ \text{or equal to} \end{cases}$ $I_P(X; Y)$?



Data Processing Inequality II: Formalization

Consider a **Markov chain**: $X \rightarrow Y \rightarrow Z$, i.e., $P(X, Y, Z) = P(X) P(Y|X) P(Z|Y)$.

- ⇒ X and Z are conditionally independent given Y (i.e., $P(X, Z | Y) = P(X|Y) P(Z|Y)$).
- ⇒ $Z \rightarrow Y \rightarrow X$ is a Markov chain (i.e., $P(X, Y, Z) = P(Z) P(Y|Z) P(X|Y)$).

Theorem (data processing inequality): “once we’ve removed some information from a random variable, further processing cannot restore the removed information.”

- ▶ $I_P(X; Y) \geq I_P(X; Z) \quad \text{and} \quad I_P(Y; Z) \geq I_P(X; Z)$ $(\forall \text{ Markov chains } X \rightarrow Y \rightarrow Z)$.

Proof:

Inf.-Theoretical Bounds on Model Performance

Consider a classification task: assign label Y to input data X : learn $P(Y|X)$

- Data generative distribution: $P(X, Y_{g.t.}) = P(Y_{g.t.})P(X|Y_{g.t.})$

⇒ Markov chain: $Y_{g.t.} \xrightarrow{\text{data gen.}} X \xrightarrow{\text{classifier}} Y$

- Perfect classification would mean $Y = Y_{g.t.} \Rightarrow I_P(Y_{g.t.}; Y) = H_P(Y_{g.t.}) - \underbrace{H_P(Y_{g.t.}|Y)}_{=0}$

- More generally: high accuracy \Rightarrow high $I_P(Y_{g.t.}; Y) \Rightarrow$ high $I_P(Y_{g.t.}; X) \geq I_P(Y_{g.t.}; Y)$:

Bound: accuracy $\leq f^{-1}(I_P(Y_{g.t.}; X))$ where $f(\alpha) = H_P(Y_{g.t.}) + \alpha \log_2 \alpha + (1 - \alpha) \log_2 \frac{1-\alpha}{\# \text{classes}-1}$
[Meyen, 2016 (MSc thesis advised by U. von Luxburg)]

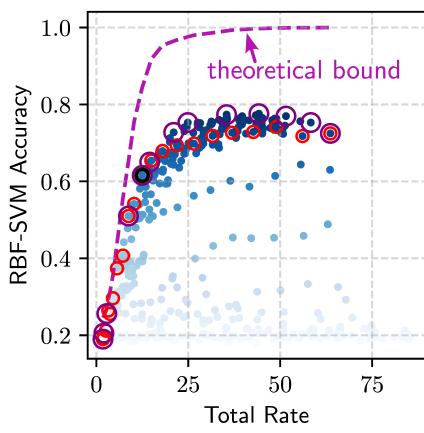
- Now introduce a preprocessing step: $Y_{g.t.} \xrightarrow{\text{data gen.}} X \xrightarrow{\text{preprocessing}} Z \xrightarrow{\text{classifier}} Y$

- Theoretical bound now: accuracy $\leq f^{-1}(I_P(Y_{g.t.}; Z)) \leq f^{-1}(I_P(Y_{g.t.}; X))$
(by information processing inequality and monotonicity of f).

⇒ Information theory suggests: preprocessing can only hurt (bound on) downstream performance.

Limitations of Information Theory

[Tim Xiao, RB, ICLR 2023]



- **Observation:** classification accuracy *decreases* for very large rate (= bound on mutual information).

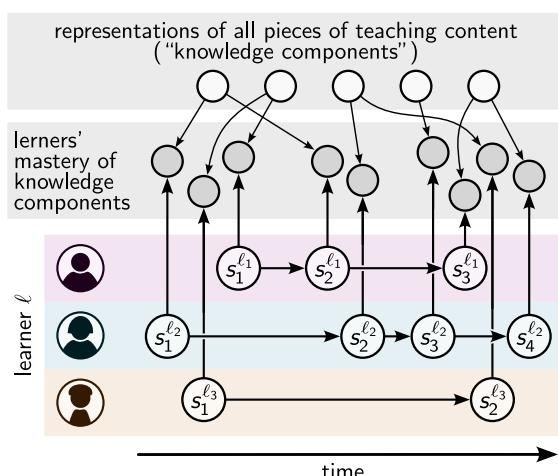
- **Explanation:** information theory doesn't consider (computational/modeling) *complexity*.

► Forcing the encoder to throw away some of the (least relevant) information can make downstream tasks *easier in practice*.

► **Note:** it's the *information* bottleneck that can make downstream processing easier, not any (possible) dimensionality reduction.

(In fact, many downstream tasks become *easier in higher dimensions* → kernel trick.)

Be Creative! You Now Have the Tools for It.



We want to quantify:

- How **specific** are learner representations s for their learner ℓ ?

$$I_P(s; \ell) = H_P(\ell) - H_P(\ell | s)$$

- How **consistent** are representations for a fixed learner if we train on different subsets of time steps?

$$\mathbb{E}_{\ell_{\text{sub}}} [I_P(s^{\ell}; \ell_{\text{sub}})]$$

- **Disentanglement**, i.e., how informative is each *component* of $s \in \mathbb{R}^n$ about learner identity ℓ ?

$$H_P(s) - H_P(s | \ell)_{\text{diag}}$$

[Hanqi Zhou, RB, CM Wu, Á Tejero-Cantero, ICLR 2024]