CS5275 Lecture 8: Information Theory

Jonathan Scarlett

March 17, 2025

Useful references:

- Cover/Thomas book "Elements of Information Theory"
- MacKay book: "Information Theory, Inference, and Learning Algorithms"

Categorization of material: Sections 1–4 are "core material"; the rest is optional.

Note: In the initial sections of these notes, we introduce various information measures and some axiomatic motivation, intuition, and properties. In Sections 5 and 6, we overview compression and communication problems where these information measures naturally arise in their fundamental performance limits.

1 Information of an Event

Getting started.

- If we are told that random event A occurred (e.g., coin came up tails, two dice added up to 7, it rained today), how much "information" have we learned?
- Approach: Quantify information without any regard to *significance* or *importance*. It is only $\Pr[A]$ that matters.
  - Things like "importance" are usually too subjective to quantify.
- Generically speaking, if A occurs with probability p, then

$$\text{Information}(A) = \psi(p)$$

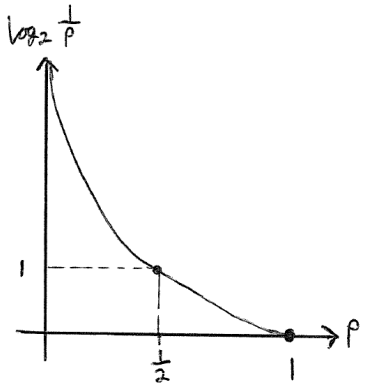
  for some function $\psi(\cdot)$. Perhaps a more intuitive interpretation of $\psi(p)$ is that it quantifies *how surprised we are that event A occurred*. What properties should this function satisfy?

Axiomatic view.

- Here are some very natural properties that we should expect $\psi(p)$ to satisfy:
  1. (Non-negativity) $\psi(p) \geq 0$, i.e., we cannot learn a "negative amount" of information.
  2. (Zero for definite events) $\psi(1) = 0$, i.e., if something was certain to happen, nothing is learned by the fact that it occurred.

3. (Monotonicity) If $p \le p'$, then $\psi(p) \ge \psi(p')$, i.e., the less likely the event was, the more information is learned by the fact that it occurred.
4. (Continuity) $\psi(p)$ is continuous in p, i.e., small changes in probability don't cause drastic changes in information.
5. (Additivity under independence) $\psi(p_1 p_2) = \psi(p_1) + \psi(p_2)$. If A and B are independent events with probabilities p_1 and p_2, then $A \cap B$ has probability $p_1 p_2$, and the information learned from both A and B occurring is the sum of the two individual amounts of information (because they are independent!)

- It can be shown that only $\psi(p) = \log_b \frac{1}{p}$ (for some base $b > 0$) satisfies all three
  - We focus on $b = 2$, which means information is measured in "bits". Another common choice is $b = e$, which means information is measured in "nats".
  - All choices of b are equivalent up to scaling by a universal constant (e.g., number of nats $= (\log_e 2) \times$ number of bits). This is much the same as how we can measure distance in meters, kilometers, inches, or miles, but converting from one to another just amounts to scaling.
  - So being told that a probability-p event occurred gives us $\log_2 \frac{1}{p}$ "bits" of information.
  - An illustration:



2 Information of a Random Variable – Entropy

Definition.

- Let X be a discrete random variable with probability mass function (PMF) P_X
- According to the previous section, if we observe $X = x$ then we have learned $\log_2 \frac{1}{P_X(x)}$ bits of information. The **(Shannon) entropy** is simply the average of this value with respect to P_X:

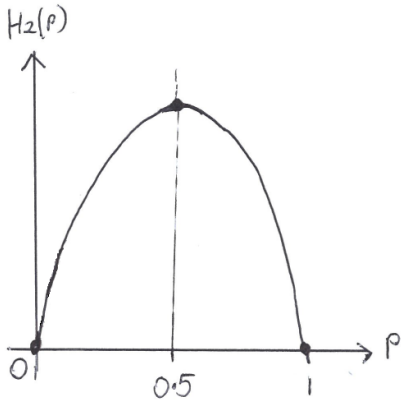
$$\begin{aligned} H(X) &= \mathbb{E}_{X \sim P_X}\left[\log_2 \frac{1}{P_X(X)}\right] \\ &= \sum_x P_X(x) \log_2 \frac{1}{P_X(x)}. \end{aligned}$$

  - Note the convention $0 \log \frac{1}{0} = 0$, which is reasonable since $\lim_{p \to 0} p \log_2 \frac{1}{p} = 0$.
- Can be viewed as a measure of *information in* X or *uncertainty in* X (these are not contradictory)

- **Note.** Here and throughout the vast majority of the course, we only consider *discrete-valued* random variables that can only take on a finite number of values. We will cover *continuous-valued* random variables much later.

Examples.

- Binary source:
  - Suppose $X \sim \text{Bernoulli}(p)$ for some $p \in (0,1)$ (i.e., $P_X(1) = 1 - P_X(0) = p$)
  - Then we get
$$H(X) = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}. \tag{1}$$
The right hand side, as a function of p, is known as the *binary entropy function*. Since this quantity will be used frequently throughout the course, we give it a formal definition: $H_2(p) = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}$ for $p \in [0,1]$. An illustration:



- Uniform source:
  - Suppose X is uniform on a finite set $\mathcal{X}$ (i.e., $P_X(x) = \frac{1}{|\mathcal{X}|}$ for each $x \in \mathcal{X}$, where $|\mathcal{X}|$ is the cardinality of $\mathcal{X}$)
  - Then we get
$$H(X) = \mathbb{E}\left[\log_2 \frac{1}{1/|\mathcal{X}|}\right] = \log_2 |\mathcal{X}|.$$
This is intuitive, e.g., with 10 bits we can produce $|\mathcal{X}| = 2^{10}$ combinations of bits.

(Optional**) Axiomatic view [Shannon].**

- Suppose that X is a discrete random variable taking N values, with probabilities $\mathbf{p} = (p_1, \ldots, p_N)$. If we consider a general information measure of the form
$$\Psi(\mathbf{p}) = \Psi(p_1, \ldots, p_N),$$
then what properties should it satisfy?

- Three natural properties:
  1. (Continuity) $\Psi(\mathbf{p})$ is continuous as a function of $\mathbf{p}$. Again, small changes in the distribution don't give large changes in information/uncertainty.

2. (Uniform case) If $p_i = \frac{1}{N}$ for $i = 1, \ldots, N$, then $\Psi(\mathbf{p})$ is increasing in N. That is, being uniform over a larger set of outcomes always means more information/uncertainty.
3. (Successive decisions) The following always holds:

$$\Psi(p_1, \ldots, p_N) = \Psi(p_1 + p_2, p_3, \ldots, p_N) + (p_1 + p_2)\Psi\left(\frac{p_1}{p_1 + p_2}, \frac{p_2}{p_1 + p_2}\right).$$

This can be viewed as drawing from the distribution on X by first drawing from the corresponding distribution that doesn't distinguish two symbols (the ones with probabilities p_1 and p_2), and then drawing another random variable to resolve those two symbols if needed (which only happens a fraction $p_1 + p_2$ of the time). The total information/uncertainty is the sum of the information/uncertainty from each of the two stages.

- It can be shown that only $\Psi(X) = \text{constant} \times H(X)$ satisfies all three.

Variations.

- Joint entropy of two random variables (X, Y):

$$\begin{aligned} H(X, Y) &= \mathbb{E}_{(X,Y)\sim P_{XY}}\left[\log_2 \frac{1}{P_{XY}(X, Y)}\right] \\ &= \sum_{x,y} P_{XY}(x, y) \log_2 \frac{1}{P_{XY}(x, y)}. \end{aligned}$$

We can similarly define $H(X, Y, Z)$ or larger collections such as $H(X_1, \ldots, X_n)$.

- Conditional entropy of Y given X:

$$\begin{aligned} H(Y|X) &= \mathbb{E}_{(X,Y)\sim P_{XY}}\left[\log_2 \frac{1}{P_{Y|X}(Y|X)}\right] \\ &= \sum_{x,y} P_{XY}(x, y) \log_2 \frac{1}{P_{Y|X}(y|x)} \\ &= \sum_{x} P_X(x) H(Y|X = x), \end{aligned} \tag{2}$$

where in the last line, $H(Y|X = x) = \sum_y P_{Y|X}(y|x) \log_2 \frac{1}{P_{Y|X}(y|x)}$ is simply the entropy of the distribution $P_{Y|X}(\cdot|x)$ on Y. We can similarly define quantities like $H(Y_1, Y_2|X_1, X_2)$.

  - Intuition: $H(Y|X = x)$ is the uncertainty in Y after having observed that $X = x$. The conditional entropy $H(Y|X)$ simply averages such a quantity over X, so it represents the average remaining uncertainty in Y after observing X.

Example:

- Consider the joint distribution described as follows:

$$P_X(0) = 1 - p, \quad P_X(1) = p,$$

$$P_{Y|X}(y|x) = \begin{cases} 1 - \delta & y = x \\ \delta & y \neq x, \end{cases}$$

where X and Y are both $\{0,1\}$-valued.

- That is, $X \sim \text{Bernoulli}(p)$ and then Y is generated by flipping X with probability δ.
- We have the following:
  - $H(X) = H_2(p)$ as already done above.
  - Similarly, $H(Y) = H_2(q)$ where $q = P_Y(1) = p(1-\delta) + (1-p)\delta$.
  - $H(Y|X) = \sum_x H(Y|X=x)$, but $H(Y|X=x)$ is $H_2(\delta)$ for both x values, so $H(Y|X) = H_2(\delta)$.
  - $H(X;Y)$ could be computed directly based on the 4 joint probabilities $(p\delta, p(1-\delta), (1-p)\delta, (1-p)(1-\delta))$, but an easier way it to use the property $H(X,Y) = H(X) + H(Y|X)$ (see *chain rule* below), giving $H(X,Y) = H_2(p) + H_2(\delta)$.
  - Computing $H(X|Y)$ directly would require using Bayes' rule to get an expression for $P_{X|Y}$ and substituting into $H(X|Y) = \sum_y P_Y(y) H(X|Y=y)$. This gets a bit messy, so is skipped here. (Again, the chain rule gives an easier approach via $H(X,Y) = H(Y) + H(X|Y)$.)

2.1 Properties of Entropy

- **Non-negativity:**

$$H(X) \geq 0$$

with equality if and only if X is deterministic.

  - Intuition: Information/uncertainty cannot be negative
  - Proof: The "information of an event" $\log_2 \frac{1}{p}$ is always non-negative for $p \in [0,1]$, so entropy is the average of a quantity that is always non-negative, and so is itself non-negative. Moreover, only $p = 1$ gives $\log_2 \frac{1}{p} = 0$, so $H(X) = 0$ if and only if X is deterministic.

- **Upper bound:** If X takes values on a finite alphabet $\mathcal{X}$, then

$$H(X) \leq \log_2 |\mathcal{X}|$$

with equality if and only if X is uniform on $\mathcal{X}$. This similarly implies $H(X|Y) \leq \log_2 |\mathcal{X}|$.

  - Intuition: The uniform distribution has the most uncertainty.
  - Proof: Let P be the distribution of X, and let Q be the uniform distribution on $\mathcal{X}$, so that $Q(x) = \frac{1}{|\mathcal{X}|}$ for all x. Then note that

$$\begin{aligned}
\sum_x P(x) \log_2 \frac{P(x)}{Q(x)} &= \sum_x P(x) \log_2 \left(|\mathcal{X}| \cdot P(x) \right) \\
&= \log_2 |\mathcal{X}| + \sum_x P(x) \log P(x) \\
&= \log_2 |\mathcal{X}| - H(X).
\end{aligned}$$

    In Section 3 we will show that the left-hand side is non-negative for *any* distributions P and Q, with equality if and only if $P = Q$. Specialized to the above choices of P and Q, we get $\log_2 |\mathcal{X}| - H(X) \geq 0$ with equality if and only if P is uniform, as desired.

- **Chain rule (two variables):**

$$H(X,Y) = H(X) + H(Y|X)$$

  - Intuition: The overall information in (X,Y) is the information in X plus the remaining information in Y after observing X.
  - Proof: For $(X,Y) \sim P_{XY}$, we have

$$\begin{aligned} H(X,Y) &= \mathbb{E}\left[\log \frac{1}{P_{XY}(X,Y)}\right] \\ &= \mathbb{E}\left[\log \frac{1}{P_X(X)P_{Y|X}(Y|X)}\right] \\ &= \mathbb{E}\left[\log \frac{1}{P_X(X)} + \log \frac{1}{P_{Y|X}(Y|X)}\right] \\ &= H(X) + H(Y|X). \end{aligned}$$

  - Note: We can swap the roles of X and Y, giving $H(X,Y) = H(Y) + H(X|Y)$.

- Chain rule (general):

$$H(X_1, \ldots, X_n) = \sum_{i=1}^{n} H(X_i|X_1, \ldots, X_{i-1}).$$

  - Intuition: Similar to the two-variable case.
  - Proof: Similar to the two-variable case, but instead use the expansion $P_{X_1 \ldots X_n} = P_{X_1} \times P_{X_2|X_1} \times P_{X_3|X_1X_2} \times \ldots \times P_{X_n|X_1,\ldots,X_{n-1}}$.

- Conditioning reduces[1] entropy:

$$H(X|Y) \leq H(X)$$

  with equality if and only if X and Y are independent.

  - Intuition: Having additional information cannot increase uncertainty *on average*.[2]
  - Proof: Equivalent to the property $I(X;Y) \geq 0$ to be proved in Section 4.1.

- Sub-additivity:

$$H(X_1, \ldots, X_n) \leq \sum_{i=1}^{n} H(X_i)$$

  with equality if and only if $X_1, \ldots, X_n$ are independent.

  - Intuition: The uncertainty in several random variables is no more than the sum of individual uncertainty in each one.
  - Proof: Apply "conditioning reduces entropy" to each summand in the general chain rule formula above.

[1] More precisely, does not increase

[2] In contrast, $H(X|Y=y)$ for a particular y could exceed $H(X)$. For example, suppose that $P_X(0) = 1 - P_X(1) = 0.99$, so that $X = 1$ is very rare. But if Y is sufficiently correlated with X, we could have $P_{X|Y}(0|0) = \frac{1}{2}$, meaning that being told $Y = 0$ made us much less certain about X.

3 A Useful Measure Between Distributions – KL Divergence

- For two PMFs P and Q on a finite alphabet $\mathcal{X}$, the *Kullback-Leibler (KL) divergence* (also known as *relative entropy*) is given by

$$\begin{aligned} D(P\|Q) &= \sum_x P(x) \log_2 \frac{P(x)}{Q(x)} \\ &= \mathbb{E}_{X\sim P}\left[\log_2 \frac{P(X)}{Q(X)}\right]. \end{aligned}$$

- Can be viewed as a kind of "distance" between P and Q, but it is not a distance function in the mathematical sense (in general it is not symmetric and doesn't satisfy the triangle inequality).

- **Claim.** For any distributions P and Q, we have

$$D(P\|Q) \geq 0$$

with equality if and only if $P = Q$.

  - Proof:

$$\begin{aligned} -D(P\|Q) &= \sum_x P(x) \log \frac{Q(x)}{P(x)} \\ &\overset{(a)}{\leq} \sum_x P(x)\left(\frac{Q(x)}{P(x)} - 1\right) \\ &= \sum_x Q(x) - \sum_x P(x) \\ &= 0, \end{aligned}$$

    where (a) uses the inequality $\log \alpha \leq \alpha - 1$, which is easily verified graphically. Equality holds in $\log \alpha \leq \alpha - 1$ if and only if $\alpha = 1$, which means that equality holds in (a) if and only if $\frac{Q(x)}{P(x)} = 1$ for all x (i.e., $P = Q$).

- **(**Optional**)** While we will focus more on entropy and mutual information, we note that KL divergence has other useful properties beyond the one above, e.g.:

  - A chain rule holds; for two variables, it is

$$D(P_{XY}\|Q_{XY}) = D(P_X\|Q_X) + D(P_{Y|X}\|Q_{Y|X}|P_X),$$

    where $D(P_{Y|X}\|Q_{Y|X}|P_X) = \sum_x P_X(x) D(P_{Y|X}(\cdot|x)\|Q_{Y|X}(\cdot|x))$.

  - A data processing inequality holds: Given P_X, Q_X, if we form P_Y, Q_Y by applying the same transformation to X (i.e., $P_Y(y) = \sum_x P_X(x)V(y|x)$ and $Q_Y(y) = \sum_x Q_X(x)V(y|x)$ for some randomized transformation V), then

$$D(P_Y\|Q_Y) \leq D(P_X\|Q_X).$$

In other words, "processing" X to produce Y can only make the distributions become closer together, not further apart.

- The KL divergence (and in fact, also entropy and mutual information) is used extensively in other fields like statistics and machine learning. Some example uses (stated only very roughly here) are:
  - In data compression, if the true source is distribution is P but we use an algorithm that wrongly assumes it is Q, then we pay a penalty of $D(P\|Q)$ in the average number of bits per symbol;
  - In statistics, if $\mathbf{X} = (X_1, \ldots, X_n)$ is i.i.d. with $X_i \sim Q$, then the probability that the observed proportions of symbols match P is roughly $2^{-nD(P\|Q)}$ when n is large. You can look up *Sanov's theorem* for a more precise and more general statement.

4 Information Between Random Variables – Mutual Information

Definition.

- Mutual information:

$$I(X;Y) = H(Y) - H(Y|X).$$

- Intuition:
  - $H(Y)$ is the *a priori uncertainty* in Y
  - $H(Y|X)$ is the *remaining uncertainty* in Y after observing X (on average)
  - Hence, $I(X;Y)$ is the *amount of information about Y we resolve by observing X* (on average).

Variations.

- Joint version:

$$I(X_1, X_2; Y_1, Y_2) = H(Y_1, Y_2) - H(Y_1, Y_2|X_1, X_2).$$

- Conditional version:

$$I(X;Y|Z) = H(Y|Z) - H(Y|X, Z).$$

Examples.

1. If X and Y are independent, then it is straightforward to compute $H(Y|X) = H(Y)$, giving $I(X;Y) = 0$ (i.e., independent random variables do not reveal any information about each other).
2. If $Y = X$, then it is straightforward to compute $H(Y|X) = H(X|X) = 0$, and hence $I(X;X) = H(X)$ (i.e., the amount of information a random variable reveals about itself is the entropy).
3. In the example given shortly after Eq. (2), we computed $H(Y|X) = H_2(\delta)$ and $H(Y) = H_2(p(1-\delta) + (1-p)\delta)$, which we can substitute into $I(X;Y) = H(Y) - H(Y|X)$ to get the mutual information. In particular, when $p = \frac{1}{2}$ we simply get $I(X;Y) = 1 - H_2(\delta)$.

4.1 Properties of Mutual Information

- **Alternative forms:**

$$\begin{aligned} I(X;Y) &= D(P_{XY} \| P_X \times P_Y) \\ &= \mathbb{E}\left[\log_2 \frac{P_{XY}(X,Y)}{P_X(X)P_Y(Y)} \right] = \sum_{x,y} P_{XY}(x,y) \log_2 \frac{P_{XY}(x,y)}{P_X(x)P_Y(y)} \\ &= \mathbb{E}\left[\log_2 \frac{P_{Y|X}(Y|X)}{P_Y(Y)} \right] = \sum_{x,y} P_{XY}(x,y) \log_2 \frac{P_{Y|X}(y|x)}{P_Y(y)}. \end{aligned}$$

  - Proof: Substituting $H(Y) = \mathbb{E}\big[\log_2 \frac{1}{P_Y(Y)}\big]$ and $H(Y|X) = \mathbb{E}\big[\log_2 \frac{1}{P_{Y|X}(Y|X)}\big]$ into the definition of mutual information gives $I(X;Y) = \mathbb{E}\big[\log_2 \frac{P_{Y|X}(Y|X)}{P_Y(Y)}\big]$. Multiplying the numerator & denominator by $P_X(X)$ gives $\mathbb{E}\big[\log_2 \frac{P_{XY}(X,Y)}{P_X(X)P_Y(Y)}\big]$, from which the remaining equalities follow easily.
  - The expression $D(P_{XY} \| P_X \times P_Y)$ has the interpretation of measuring "how far X and Y are from being independent".

- **Symmetry:** We have

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

  and in particular

$$I(X;Y) = I(Y;X)$$

  which also implies

$$I(X;Y) = H(X) - H(X|Y).$$

  - Intuition: X and Y reveal an equal amount of information about each other (or maybe this is not that intuitive!)
  - Proof: We have from the above alternative form that

$$\begin{aligned} I(X;Y) &= \mathbb{E}\left[\log_2 \frac{P_{XY}(X,Y)}{P_X(X)P_Y(Y)} \right] \\ &= \mathbb{E}\left[\log_2 \frac{1}{P_X(X)} + \log_2 \frac{1}{P_Y(Y)} + \log_2 P_{XY}(X,Y) \right] \\ &= H(X) + H(Y) - H(X,Y), \end{aligned}$$

    where we first expanded the logarithm, and then applied the definition of (joint) entropy.

- **Non-negativity:** $I(X;Y) \geq 0$ with equality if and only if X and Y are independent.

  - Intuition: One random variable cannot tell us a "negative amount" of information about the other.
  - Proof: Using the above-established identity $I(X;Y) = D(P_{XY} \| P_X \times P_Y)$, this is just a special case of $D(P\|Q) \geq 0$ with equality if and only if $P = Q$.

- **Upper bounds:** We have

$$\begin{aligned} I(X;Y) &\leq H(X) \leq \log_2 |\mathcal{X}| \\ I(X;Y) &\leq H(Y) \leq \log_2 |\mathcal{Y}|. \end{aligned}$$

- Intuition: The information X reveals about Y (mutual information) is at most the prior information in X (entropy).
- Proof: To show that $I(X;Y) \leq H(X)$, combine $I(X;Y) = H(X) - H(X|Y)$ (see above) and $H(X|Y) \geq 0$ (conditional or unconditional entropy is never negative). We already showed $H(X) \leq \log_2 |\mathcal{X}|$ earlier, and the remaining claims follow by symmetry, reversing the roles of X and Y.

- **Chain rule:**

$$I(X_1, \ldots, X_n; Y) = \sum_{i=1}^{n} I(X_i; Y | X_1, \ldots, X_{i-1}).$$

  - Intuition: Similar to the chain rule for entropy.
  - Proof: Write $I(X_1, \ldots, X_n; Y) = H(X_1, \ldots, X_n) - H(X_1, \ldots, X_n|Y)$ and apply the chain rule for entropy to both terms.

- **Data processing inequality:** If Z depends on (X, Y) only through Y (often stated via the terminology "$X \to Y \to Z$ forms a Markov chain", and equivalent to the statement "X and Z are conditionally independent given Y"), then

$$I(X; Z) \leq I(X; Y).$$

  - Intuition: Processing Y (to produce Z) cannot increase the information available regarding X.
  - Proof: As stated above, the statement "Z depends on (X, Y) only through Y" is equivalent to "Z and X are conditionally independent given Y". This means that the property $P_{Z|XY} = P_{Z|Y}$ (as assumed in the result) is equivalent to $P_{X|YZ} = P_{X|Y}$. To deduce the result, we write

$$\begin{aligned} I(X; Z) &\overset{(a)}{=} H(X) - H(X|Z) && (3)\\ &\overset{(b)}{\leq} H(X) - H(X|Y, Z) && (4)\\ &\overset{(c)}{=} H(X) - H(X|Y) && (5)\\ &\overset{(d)}{=} I(X; Y), && (6) \end{aligned}$$

    where (a) and (d) use the definition of mutual information, (b) follows since conditioning reduces entropy, and (c) holds because $H(X|Y,Z) = \mathbb{E}\big[\log \frac{1}{P_{X|YZ}(X|Y,Z)}\big] = \mathbb{E}\big[\log \frac{1}{P_{X|Y}(X|Y)}\big] = H(X|Y)$ by the above-established fact $P_{X|YZ} = P_{X|Y}$.
  - Variations:
    - If $X \to Y \to Z$ then $I(X; Z) \leq I(Y; Z)$.
    - If $W \to X \to Y \to Z$ then $I(W; Z) \leq I(X; Y)$.

- **Partial sub-additivity:** If $(Y_1, \ldots, Y_n)$ are conditionally independent given $(X_1, \ldots, X_n)$, and in addition Y_i depends on $(X_1, \ldots, X_n)$ only through X_i, then

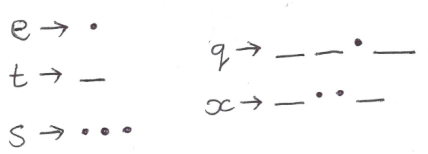
$$I(X_1, \ldots, X_n; Y_1, \ldots, Y_n) \leq \sum_{i=1}^{n} I(X_i; Y_i).$$

  (You can try proving this as an exercise, or see Notes #1 at `https://www.comp.nus.edu.sg/~scarlett/CS3236_notes/` for the proof.) However, without the conditional independence assumptions, this property may fail to hold.

5 (**Optional**) A Brief Overview of Data Compression

Familiar examples of compression.

- When we compress a file to .zip or .rar it gets smaller, and yet we can still recover the contents. How/why is this possible? This is the problem of **lossless compression**.
- When we convert a file from .bmp to .jpeg, we lose some quality, but hopefully not too much. However, we cannot convert back to the higher-quality image. This is the problem of **lossy compression**.
- Concepts in compression go further back than computers – recall Morse code:



Example 1: Sparse binary string.

- Suppose that we want to efficiently store

$$0000000\mathbf{1}000000000000000000000\mathbf{1}000000000000000000000000\mathbf{1}000000000,$$

  i.e., a string of length 64 with only three 1s and the rest 0s.
- Storing this "as is" requires 64 bits (a bit being a 1 or 0).
- Alternative scheme:
  - Index the string positions from 0 to 63, and consider their binary format (e.g., $0 \to 000000$, $7 \to 000111$, $63 \to 111111$)
  - Store the 3 positions where the long string has value 1, using 6 bits per position.
- This permits only 18 bits of storage instead of 64.

Example 2: Equal number of 1s and 0s.

- Suppose that we need a system that can compress sequences of the form

$$1011010101100111010011001100100101101011101010100010010010100 11,$$

  i.e., still length 64, but now half ones and half zeros.
- Again, storing "as is" requires 64 bits.
- The number of strings with half zeros and half ones is $\binom{64}{32}$. Let's aim to compress them down to some number $L < 64$ of bits. How small can L be?
- With L bits (each 0 or 1), we can make 2^L combinations. Since each of the $\binom{64}{32}$ strings have to be stored as a different combination, we clearly need $2^L \geq \binom{64}{32}$, or equivalently

$$L \geq \log_2 \binom{64}{32} \approx 60.7.$$

(More generally, compressing N different strings down to L bits without loss requires $L \geq \log_2 N$.)

- So we can't hope to do much better than direct storage!

Example 3: English text.

- English text clearly has a fair bit of redundancy, since we can "throw away" several letters but still (usually) recover the original text:

C_N Y__ F_LL _N TH_ V_W_LS _N TH_S S_NT_NC_?

- If we (somewhat naively) store English text in some binary format in a letter-by-letter fashion, we can exploit the fact that some letters are more common than others, e.g., map 'e' to a short binary sequence, and 'x' to a long binary sequence.
  - Morse code is an early example of this idea (but not quite "binary"!)
  - Can we construct an "optimal" mapping?
- The savings are much greater if we exploit the fact that different *groups* of letters are more likely to appear together (e.g., if we have already seen "Fill in the blan", then there is clearly a much more likely letter than 'e' coming next!)
- Spoiler: While it requires 5 bits (or at least $\log_2 27 \approx 4.75$) to uniquely identify one of 27 characters ('a' to 'z' and also spaces), the actual "information content" of each letter in English text is only about 1.34 bits. This will mean that we can compress down by a factor of at least $3\times$ without losing anything.
  - See the appendix of this document for further details

Information-theoretic viewpoint.

- Information theory adopts probabilistic models, e.g., a string of 1000 English characters is modeled by some joint distribution $P_{X_1 X_2 \ldots X_{999} X_{1000}}$, where each X_i takes some value in $a \ldots z$ (or space).
  - Probabilistic modeling can provide a very good trade-off between accuracy in modeling the real world vs. tractability of the mathematical analysis.
- Two distinct approaches to compression:
  - (Variable-length) Map more probable sequences to shorter binary strings, at the expense of mapping less probable sequences to longer strings. **How low can the average length be?**
  - (Fixed-length) Map the most probable sequences to binary strings of a given length, at the expense of not having enough such strings for the low-probability sequences. **How low can the length be while having a very low probability of failure?**
- **Source coding theorem (informal).** In both of these settings, the fundamental compression limit is given by the Shannon entropy H. The (average) storage length can be arbitrarily close to H, but can never be any lower than H.

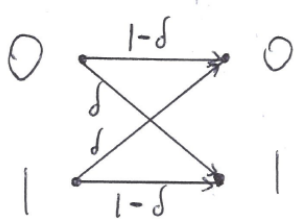
6 (**Optional**) A Brief Overview of Data Communication

Familiar examples of communication.

- When military pilots want to read a sequence of letters over an intercom, they use "alpha", "bravo", "Charlie", etc.
- If someone on the other end of the phone is having trouble hearing us, we might repeat the same thing 2–3 times to make sure they hear it.
- If we're talking to someone in their non-native language, we might talk slower.
- Our WiFi slows down as we move further away from the router.
- Common theme: Send information *more slowly* but also *more reliably.*
  - For a given reliability, how slow do we need to go?

Simple communication setting.

- Let's suppose that we are communicating in binary:
  - A "transmitter" sends a sequence of 0s and 1s
  - A "receiver" receives the sequence *with some corruptions*: Each bit is flipped (from 0 to 1, or from 1 to 0) independently with probability $\delta \in \left(0, \frac{1}{2}\right)$.
  - This is depicted in the following "channel transition diagram":



  - e.g., The sequence 01101000 might be received as 0**0**101**1**00 (two corruptions)

Approach 1: Uncoded communication.

- Suppose that the transmitter wants to send one of 16 messages (e.g., it has done a weather reading and wants to send one of 16 possibilities among "sunny", "rainy", "partly cloudy", etc.)
- Naively, it can do this by mapping each outcome to a unique sequence of 4 bits (e.g., sunny $\to$ 0000, rainy $\to$ 1010, etc.)
- Since each bit is flipped with probability δ, the probability of all 4 bits coming out correct is $(1-\delta)^4$. For instance, if $\delta = 0.1$, we have $\mathbb{P}[\text{correct}] = 0.9^4 = 0.6561$.
- Things get worse as we send more messages, e.g., if we encode one of $2^8 = 256$ messages to a length-8 binary string and transmit it, we get $\mathbb{P}[\text{correct}] = (1-\delta)^8$, which is roughly 0.43 when $\delta = 0.1$.

Approach 2: Repetition code.

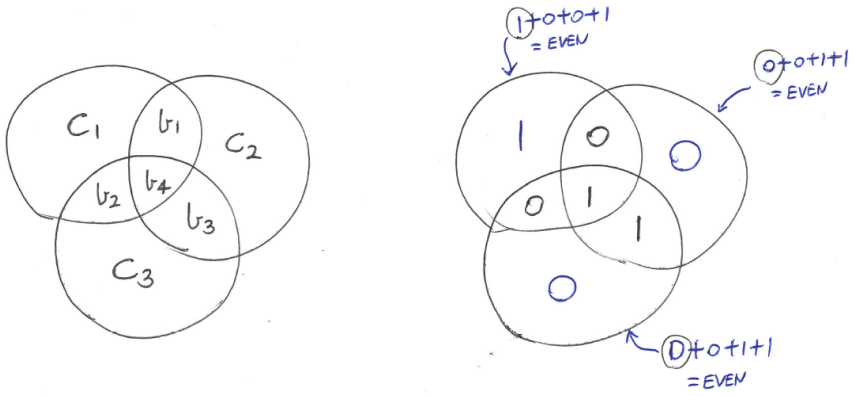
- As mentioned above, let's try transmitting slower but more reliably!
- Let's start with just sending one of two messages, which we will label as 0 and 1.
- Repetition code R_3 of length 3:
  - To send "0", transmit the sequence "000"
  - To send "1", transmit the sequence "111"
  - At the receiver, take the majority vote (e.g., 000 or 010 get decoded as "0", whereas 111 or 110 get decoded as "1")
- Clearly, we get correct decoding if there are no flips or one flip, so $\mathbb{P}[\text{correct}] = (1-\delta)^3 + 3\delta(1-\delta)^2$, which equals 0.972 when $\delta = 0.1$.
- We can then transmit, say, one of 16 messages by mapping (e.g.) 0101 to 000111000111. The probability of getting back the correct message is $0.972^4 \approx 0.893$
  - A fair bit more reliable than uncoded – but three times slower!
- We can do the same with more repetitions:
  - e.g., map 0101 to 000000011111110000000011111111 (repetition code R_7)
  - Any given bit (out of the 4 sent) is decoded correctly with probability

$$(1-\delta)^7 + 7\delta(1-\delta)^6 + \binom{7}{2}\delta^2(1-\delta)^5 + \binom{7}{3}\delta^3(1-\delta)^4 \approx 0.9973$$

    (This is the probability that a Binomial$(7, 0.1)$ random variable is at most 3)
  - The overall message is decoded correctly with probability $\approx 0.9973^4 \approx 0.989$.
  - Now the communication is very reliable, but we are 7 times slower than uncoded! Do we have to keep getting slower and slower?

Approach 3: Hamming code.

- Here we give a famous example of how to map a binary string of length 4 (so 16 messages) to a binary string of length 7 while still being able to correct one bit flip.
- The technique: In the following figure, fill in $b_1b_2b_3b_4$ ('b' for 'bit') with the original four bits, and assign $c_1c_2c_3$ ('c' for 'check') the values that make the three bits in their circle add up to an even number. (An example is shown on the right)



- Observe that any single bit flip (whether it be one of the b_i or one of the c_i) changes a unique combination of circles from "even" to "odd"! Therefore, if a single bit flip occurs, we can uniquely identify which bit caused it, and therefore correct it.
  - We can also distinguish the case that no bit flips occurred, and hence all 3 circles remain "even".
- Therefore

$$\begin{aligned}\mathbb{P}[\text{correct}] &\geq \mathbb{P}[\text{zero or one bit flip(s)}] \\ &= (1-\delta)^7 + 7\delta(1-\delta)^6 \\ &\approx 0.85,\end{aligned}$$

  where the last line holds when $\delta = 0.1$.
- Nearly as reliable as the repetition code, despite mapping to only 7 bits instead of 12! (i.e., we are transmitting a fair bit "less slowly")

Information-theoretic results.

- **Definition.** If we map k bits to n bits in the encoding procedure, then the *rate* is $\frac{k}{n}$ (e.g., $\frac{4}{7}$ for the above Hamming code, $\frac{1}{3}$ for the repetition code, 1 for uncoded)
- Clearly, there is an inherent trade-off between rate and error probability.
  - Higher rate = Send faster
  - Lower error probability = Send more reliably
- **Channel coding theorem (informal).** There exists a channel-dependent quantity called the *(Shannon) capacity* C such that arbitrarily small error probability can be achieved for all rates less than C, but for no rates higher than C. Specifically, we have $C = \max_{P_X} I(X;Y)$, where P_X is optimized and $P_{Y|X}$ is the channel transition law.
  - In the above example with $\delta = 0.1$, we get $C \approx 0.531$ (more generally $C = 1 - H_2(\delta)$). So for arbitrarily small error probability (e.g., $\mathbb{P}[\text{error}] \leq 10^{-10}$), not only is it unnecessary to multiply the number of bits by a higher and higher number, but we can get away with fewer than double the original number (!)

– Caveat: We may need to code over a much longer block length (e.g., map $k = 5000$ bits to $n = 10000$ bits, rather than mapping $k = 3$ bits to $n = 6$ bits)

Principles of information theory:

- First fundamental limits, then practical methods
- First asymptotic results, then finite-length refinements
- Mathematically tractable yet powerful probabilistic models

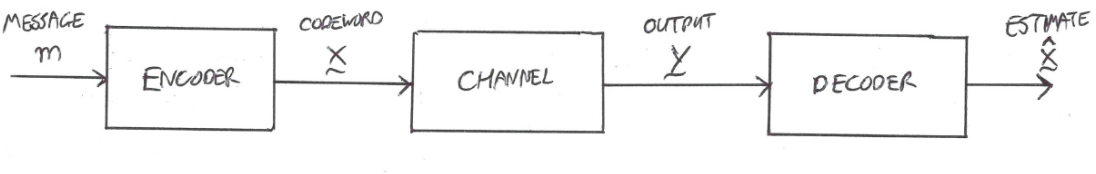
7 (**Optional**) Proof Outline for the Channel Coding Theorem

- The proofs of the source and channel coding theorems use similar ideas; we will focus on the latter.
- The proof is split into two statements:
  - (Achievability) For any transmission rate $R < C$, there exists a sequence of codes (indexed by the block length n) such that $P_e \to 0$.
  - (Covnerse) For any transmssion rate $R > C$, it is impossible to obtain $P_e \to 0$, regardless of the choice of code (in fact a stronger statement $P_e \to 1$ can be shown).

  Note that 'Achievability' means 'mathematical existence property' and 'Converse' means 'mathematical impossibility result'.

Outline of achievability proof:

- The setup is depicted as follows:



- We need to specify the behavior of both the encoder and decoder:
  - (Encoder) Given a message m, produce a length-n codeword $\mathbf{x}^{(m)}$. The codewords can be collected into a codebook $\{\mathbf{x}^{(i)}\}$.
  - (Decoder) Given knowledge of the coedbook but not the specific message m, and also given the received sequence $\mathbf{y}$, produce an estimate $\hat{m}$.
- Codebook design:
  - Designing a good codebook explicitly is very difficult (it was the focus of decades of research after information theory was introduced!)
  - Instead, the original proofs show the *existence* of good codebooks in a non-constructive way.

- This was done using *random coding* (a case of *the probabilistic method*) – analyze the average performance of a codebook whose codewords have entries drawn independently from P_X. If the average performance is "good", then the best specific code's performance is certainly no worse.

- Decoder design and analysis:
  - There are multiple decoding rules that suffice to prove the channel coding theorem, with the most powerful (optimal) rule being maximum-likelihood decoding. The rule usually found in textbooks is *joint typicality decoding*.
  - Roughly, joint typicality decoding is based on searching for a codeword that "looks like" it was drawn i.i.d. from $P_X \times P_{Y|X}$ (with P_X from random coding, and $P_{Y|X}$ being the channel).
  - Under random coding, it can be shown that the correct codeword passes this joint typicality condition with probability approaching one, whereas any incorrect codeword only passes it with probability roughly $2^{-nI(X;Y)}$. Hence, if the number of messages is 2^{nR} with $R < I(X;Y)$, then the decoder succeeds with high probability.
  - Optimizing P_X gives the capacity formula $C = \max_{P_X} I(X;Y)$.

Outline of converse proof:

- The proof is roughly outlined as follows:
  - Relate the error probability to $I(m;\hat{m})$, where m a uniformly random message and $\hat{m}$ is the final estimate. Intuitively, if $\hat{m} = m$ with high probability then $I(m;\hat{m})$ should be large, and the contrapositive version of this leads to a statement on the error probability.
  - Use a property called *data processing inequality* to bound $I(m;\hat{m}) \leq I(\mathbf{X};\mathbf{Y})$, where $\mathbf{X}$ is the transmitted codeword and $\mathbf{Y}$ is the channel output. Intuitively, the end-to-end information that $\hat{m}$ reveals about m cannot exceed the bottleneck imposed by $(\mathbf{X},\mathbf{Y})$ in between.
  - Use mutual information properties (e.g., chain rule) and the memoryless channel assumption to show that $I(\mathbf{X};\mathbf{Y}) \leq \sum_{i=1}^{n} I(X_i;Y_i)$. This is upper bounded by nC since $C = \max_{P_X} I(X;Y)$, and putting everything together completes the proof.

- The first of these steps uses *Fano's inequality*, which is very widespread for proving converse results in information theory and statistics. In generic notation (renaming $(m,\hat{m})$ to $(X,\hat{X})$), the inequality is

$$H(X|\hat{X}) \leq H_2(P_\mathrm{e}) + P_\mathrm{e} \log_2\big(|\mathcal{X}| - 1\big),$$

  where $P_\mathrm{e} = \mathbb{P}[\hat{X} \neq X]$, and $\mathcal{X}$ is the set of values X can take.
  - Intuition. To resolve the uncertainty in X given $\hat{X}$, we can first ask whether the two are equal, which bears uncertainty $H_2(P_\mathrm{e})$. In the case that they differ, which only occurs a fraction P_e of the time, the remaining uncertainty is at most $\log_2\big(|\mathcal{X}| - 1\big)$, since the uniform distribution maximizes entropy.
  - Notice that mutual information doesn't appear above, but when X is uniform we have $I(X;\hat{X}) = H(X) - H(X|\hat{X})$, and substituting Fano's inequality (with $\log_2\big(|\mathcal{X}| - 1\big)$ weakened to $\log_2|\mathcal{X}|$ and $H_2(P_\mathrm{e})$ weakened to 1) and $H(X) = \log_2|\mathcal{X}|$ gives the following:

$$I(X;\hat{X}) \geq (1 - P_\mathrm{e}) \log_2|\mathcal{X}| - 1,$$

or equivalently $P_e \geq 1 - \frac{I(m;\hat{m})+1}{\log_2 |\mathcal{X}|}$. (Intuitively, to have small P_e we need the "learned information" $I(X;\hat{X})$ to match the prior uncertainty $\log_2 |\mathcal{X}|$.)

8 (**Optional**) Broader Uses of Information Theory

Information theory is used extensively in computer science, statistics, machine learning, and beyond, e.g.:

- Lower bounds on sample/query complexity for statistical problems (e.g., estimation, optimization, randomized algorithms)
- Information-theoretic analysis of algorithms, both computationally efficient and inefficient (e.g., an interesting recent example is watermarking of Large Language Models)
- Information measures used in machine learning and other areas (e.g., in sequential decision-making, make decisions that maximize information gain)

See Notes #7 at `https://www.comp.nus.edu.sg/~scarlett/CS3236_notes` for a more detailed overview. Below we give one example in the first category.

Example: Binary search

- Suppose that there are n integers sorted in non-decreasing order, and we want to find the first one to exceed a threshold γ. We can only interact with the list by asking questions of the form "Does the i-th element exceed γ?".
  - We will focus on the case that the allowed number of queries Q is pre-specified (with a value that may depend on n), but this can easily be relaxed.
- Noiseless case: (Every query answer is correct)
  - We can solve this using binary search: Query the middle element, then recurse left or right depending on the answer, and after roughly $Q \approx \log_2 n$ iterations we will have the answer.
  - A simple counting argument shows that no algorithm can do better: If the list is of the form $0 \ldots 01 \ldots 1$ and $\gamma = \frac{1}{2}$, then there are n possible locations of the transition from 0 to 1. But the algorithm must produce a different output for each such case, so the list of query answers must also be different. With Q queries there are 2^Q possible sequences of outcomes, so we need $2^Q \geq n$, or $Q \geq \log_2 n$.
  - Fano's inequality (see below) can also used to deduce a similar conclusion, with the mutual information simply bounded upper by the number of queries (because each query outcome is binary so can only contribute at most 1 bit).
  - Similar ideas can be applied to other problems, e.g., for comparison-based sorting we get that $\log_2(n!) = \Theta(n \log n)$ comparisons are needed.
- Noisy case: (Each query answer is only correct with probability $1 - \delta$)
  - Having each query answer be flipped w.p. δ amounts to having the answers passed through a binary symmetric channel (BSC), so we can view this through the lens of communication.

– In particular, if $\mathbf{X}$ is the set of queries made and $\mathbf{Y}$ is the sequence of responses, then an argument based on Fano's inequality gives the following lower bound on error probability:

$$P_{\mathrm{e}} \geq 1 - \frac{I(\mathbf{X};\mathbf{Y}) + 1}{\log_2 n}$$

when we consider sequences of the form $0 \ldots 01 \ldots 1$ with the transition from 0 to 1 being uniformly random over n possibilities.

– But by the same analysis as in channel coding, we can derive $I(\mathbf{X};\mathbf{Y}) \leq QC$ where C is the BSC capacity and Q is the number of queries.

– It follows that Q needs to be roughly $\frac{\log_2 n}{C}$ to have high reliability (i.e., $P_{\mathrm{e}} \approx 0$). Using the channel capacity formula $C = 1 - H_2(\delta)$, this can further be shown to behave as $\Theta\big(\frac{\log_2 n}{(1/2-\delta)^2}\big)$.

– Algorithms matching this query complexity to within constant factors is given in the paper "Computing with noisy information" (1994), and the query complexity with precise constants is given in the paper "Optimal bounds for noisy sorting" (2023).

(**Optional**) Appendix: Entropy of English Text

- Shannon's famous 1948 paper discussed several (intentionally over-simplified) probabilistic models for generating English text; see Figure 1 below.
- Stated differently, #3 generates each letter conditioned on the previous one, #4 conditions on the previous two, #5 lets the "alphabet" $\mathcal{X}$ be the set of all words rather than the set of all characters and generates each word independently, and #6 generates each word conditioned on the previous one.
- **Fundamental question:** How much information (entropy) does each letter of English text tell us?
  – The entropy $H(X)$ of a single character doesn't capture the fact that previous characters help in predicting the next one.
  – As detailed in Chapter 4 of Cover/Thomas, a more meaningful measure in such scenarios is

$$H(X_n | X_1, \ldots, X_{n-1}),$$

representing the uncertainty of a given character given all of the previous ones.

  – Fitting a model to English text and then calculating the entropy of that model is prone to be inaccurate (too complex to fit a very accurate model!) – is there a simpler approach?
- **Key idea:** The entropy is closely related to *how many guesses are needed (on average) before the correct character is guessed*, by an optimal guessing algorithm.
  – Intuitively, entropy is a measure of uncertainty, and higher uncertainty means more guesses will be needed on average.
  – Writing an optimal guessing algorithm is hard (though an interesting machine learning problem!), so experiments were done under the assumption that *humans are near-optimal guessers.*

1. Zero-order approximation (symbols independent and equiprobable).

   XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZL-HJQD.

2. First-order approximation (symbols independent but with frequencies of English text).

   OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL.

3. Second-order approximation (digram structure as in English).

   ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TU-COOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE.

4. Third-order approximation (trigram structure as in English).

   IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONS-TURES OF THE REPTAGIN IS REGOACTIONA OF CRE.

5. First-order word approximation. Rather than continue with tetragram, ... , n-gram structure it is easier and better to jump at this point to word units. Here words are chosen independently but with their appropriate frequencies.

   REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NAT-URAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.

6. Second-order word approximation. The word transition probabilities are correct but no further struc-ture is included.

   THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHAR-ACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

Figure 1: Excerpt from Shannon's paper.

  - Using some theory behind the "optimal guessing" viewpoint, and observing the average number of guesses that several humans required, it was estimated that the entropy of English text is at most **1.34 bits per character**
  - This is much less than the highest possible value of $\log_2 27 \approx 4.75$ with 27 characters (a-z and "space")
  - Note: Recent developments in Language Models suggest that the "correct" value may even be significantly smaller than 1.34 (details omitted).

- See Chapter 6 of Cover/Thomas for further details.