EE/Ma/CS 126a: Information Theory

Aman Bhargava

October-December 2022

Contents

0.1 Introduction and Course Information

This document offers an overview of EE/Ma/CS 126a at Caltech. They comprise my condensed course notes for the course. No promises are made relating to the correctness or completeness of the course notes. These notes are meant to highlight difficult concepts and explain them simply, not to comprehensively review the entire course.

Course Information

- Professor: Michelle Effros
- Term: 2022 Fall

Chapter 1

Math Review

1.1 Combinatorics & Probability

Binomial Distribution & Coefficient

- **Bernoulli Process**: Repeated trials, each with one binary outcome. The probability of a positive outcome is $p \in [0, 1]$. Each trial is independent.

- **Binomial Distribution**: Let x represent the number of successful trials in a Bernoulli process repeated n times with success probability p. The binomial distribution gives the probability distribution on x:

$$b(x; n, p) = \binom{n}{k} p^x (1 - p)^{n-x} \tag{1.1}$$

  Which has $\mu = np$, $\sigma^2 = npq$.

- **Intuition for Binomial Distribution**: The probability of observing a sequence with x positive outcomes and $n - x$ negative outcomes is $p^x(1-p)^{n-x}$. There are $\binom{n}{k}$ different sequences (i.e., permutations) that have x positive cases and n negative cases. Thus the total probability of observing x positive cases is given by Eq 1.1.

- **Binomial Coefficient**:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{1.2}$$

1.2 Logarithm Identities

Entropy calculations and manipulations involve a lot of logarithms. They're not so bad once you get to know them, though:

- **Definition**:
$$a = b^{\log_b a}$$

- **Sum-Product**:
$$\log_c(ab) = \log_c a + \log_c b$$

- **Difference-Quotient**:
$$\log(a/c) = \log a - \log c$$
$$\log \frac{1}{a} = -\log a$$

- **Product-Exponent**:
$$\log_c(a^n) = n \log_c(a)$$

- **Swapping Base**:
$$\log_b(a) = \log_a(b)$$

- **Swapping Exponential**:
$$a^{\log n} = n^{\log a}$$

- **Change of Base**;
$$\log_b(a) = \frac{\log_x(a)}{\log_x(b)}$$

Chapter 2

Entropy Definitions

Chapter 2 of Elements of Information Theory.

2.1 Entropy, Conditional Entropy, Joint Entropy

Entropy Definition (Discrete):

$$H(X) = \sum_{x \in \mathcal{X}} p(x) \log(\frac{1}{p(x)}) \tag{2.1}$$

$$= -\sum_{x \in \mathcal{X}} p(x) \log p(x) \tag{2.2}$$

$$= \mathbb{E}[\log \frac{1}{p(x)}] \tag{2.3}$$

Theorem 1. *Properties of Entropy*

1. ***Non-negativity:*** $H(X) \geq 0$ *– Reasoning: Entropy is the sum-product of non-negative terms.*

2. ***Change of base:*** $H_b(X) = (\log_b a) H_a(X)$

3. ***Bernoulli entropy:*** $H(X) = -p \log p - q \log q \equiv H(p)$.

   - $H(p)$ *is a concave function of* p*, peaks at* $p = q = 0.5$.

Joint Entropy: Literally just entropy of vector $[X, Y]^\top$.

$$\begin{aligned} H(X,Y) &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(x,y) && (2.4) \\ &= \mathbb{E}[\log p(x,y)] && (2.5) \\ & && (2.6) \end{aligned}$$

Conditional Entropy: $H(Y|X)$ is the expected entropy of $p(y|x)$ averaged across all x.

$$\begin{aligned} H(Y|X) &= \sum_{x\in\mathcal{X}} p(x) \underbrace{\sum_{y\in\mathcal{Y}} p(y|x)\log p(y|x)}_{H(Y|X=x)} && (2.7) \\ &= -\mathbb{E}[\log p(Y|X)] && (2.8) \end{aligned}$$

Entropy can be thought of as the **uncertainty** in the value of a random variable. High entropy corresponds to a high degree of uncertainty. Conditional entropy $H(Y|X)$ can be thought of as the average **remaining uncertainty** in the value of Y after learning the value of X.

Theorem 2. *Chain Rule for Entropy*

$$\begin{aligned} H(X,Y) &= H(X) + H(Y|X) && (2.9) \\ &= H(Y) + H(X|Y) && (2.10) \end{aligned}$$

It also follows that

$$H(X,Y|Z) = H(X|Z) + H(Y|X,Z) \tag{2.11}$$

Proof sketch:

- *Recall that* $H(X) = -\mathbb{E}[\log p(x)]$ *and* $H(Y|X) = -\mathbb{E}[\log p(y|x)]$.
- $\log p(x) + \log p(y|x) = \log(p(x)\cdot p(y|x)) = \log p(x,y)$.
- *The proof follows from there. You can also write out the full sum form of* $H(X,Y)$ *and recover* $H(X), H(Y|X)$ *from there if you're feeling rigorous.*

2.2 Relative Entropy & Mutual Information

Relative Entropy: $D(p\|q)$ gives a *distance* between distributions $p(x)$ and $q(x)$. Also known as KL divergence.

$$D(p\|q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \tag{2.12}$$

$$= \mathbb{E}_{p(x)}[\log \frac{p(x)}{q(x)}] \tag{2.13}$$

$$\tag{2.14}$$

This also corresponds to the **inefficiency** of using q as a replacement for p when generating codes for tokens drawn from $p(x)$.

- **Average code length with correct** $p(x)$**:** $H(p)$.
- **Average code length with incorrect** $q(x)$**:** $H(p) + D(p\|q)$.

Theorem 3. *Properties of Relative Entropy*

1. ***Asymmetric:*** *In general,* $D(p\|q) \neq D(q\|p)$*.*
2. ***Non-negative:*** $D(p\|q) \geq 0$*.*
3. ***Identity:*** *If* $D(p\|q) = 0$ *then* $p \equiv q$*.*

Conditional Relative Entropy/KL Divergence: Distance between two distributions when conditioned on the same variable. Similar idea of averaging across all values of the conditioning variable.

$$D(p(y|x)\|q(y|x)) = \sum_{x \in \mathcal{X}} p(x) \big[\sum_{y \in \mathcal{Y}} p(y|x) \log \frac{p(y|x)}{q(y|x)} \big] \tag{2.15}$$

$$= \mathbb{E}_{p(x,y)} \log \big[\frac{p(y|x)}{q(y|x)} \big] \tag{2.16}$$

We now move onto **mutual information** – a measure of the dependence of two variables. As we will see, it is the **reduction in uncertainty** of X due to knowing Y, on average.

Mutual Information:

$$I(X;Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log(\frac{p(x,y)}{p(x)p(y)}) \tag{2.17}$$

$$= D\big(p(x,y)\|p(x)p(y)\big) \tag{2.18}$$

$$= \mathbb{E}[\log\frac{p(X,Y)}{p(X)p(Y)}] \tag{2.19}$$

Theorem 4. *Properties of Mutual Information*

- *It is the **divergence** between $p(x,y)$ and $p(x)p(y)$.*
- ***Symmetry:*** $I(X;Y) = I(Y;X)$.
- ***Relation to Entropy:*** *Mutual information is the reduction in uncertainty of each RV expected after discovering the other variable's value.*

$$I(X;Y) = H(X) - H(X|Y) \tag{2.20}$$

$$= H(Y) - H(Y|X) \tag{2.21}$$

$$\tag{2.22}$$

- ***Alternative Entropy Relation:***

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{2.23}$$

$$I(X;X) = H(X) - H(X|X) \tag{2.24}$$

$$= H(X) \tag{2.25}$$

Proof Sketch for (3):

- *Within the definition of $I(X;Y)$ there is a term $\log\frac{p(x,y)}{p(x)p(y)}$.*
- *Once you convert the argument of the log into $p(x|y)/p(x)$, you can separate out $H(X) - H(X|Y)$ using the quotient-difference logarithm rule.*

Conditional Mutual Information: $I(X,Y|Z)$ is the average reduction in uncertainty on the value of X due to knowing Y when Z is given.

$$I(X;Y|Z) = H(X|Z) - H(X|Y,Z) \tag{2.26}$$

$$= \mathbb{E}_{p(x,y,z)}\log\big[\frac{p(X,Y|Z)}{p(X|Z)p(Y|Z)}big] \tag{2.27}$$

$$\tag{2.28}$$

2.3 Chain Rules: $H(\cdot), I(\cdot;\cdot)$

These chain rules end up being very useful in a lot of proofs. Deeply understanding them is a good idea.

Theorem 5. *Entropy chain rule Let $X_1, X_2, \ldots, X_n \sim p(x_1, x_2, \ldots, x_n)$. Then*

$$H(X_1, \ldots, X_n) = \sum_{i=1}^{n} H(X_i | X_{i-1}, X_{i-2}, \ldots, X_1) \tag{2.29}$$

Proof: *Repeatedly apply Equation 2.9.*

Intuition of Chain Rule: It's important to note that the term in the sum is conditioned on elements X_j with $j < i$. Conditioning always reduces entropy, so it's as though the "additional entropy" from the term must be reduced to account for the previous terms already having been added to the total. Also note that any order can suffice – there is no absolute order in the sum.

Theorem 6. *Chain Rule for Mutual Information*

$$I(X_1, \ldots, X_n; Y) = \sum_{i=1}^{n} I(X_i; Y | X_{i-1}, X_{i-2}, \ldots, X_1) \tag{2.30}$$

$$\tag{2.31}$$

Proof: *Start with the entropy definition of mutual information. Then apply the chain rule for entropy (Theorem 5).*

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

Theorem 7. *Chain Rule for Relative Entropy*

$$D(p(x,y)\|q(x,y)) = D(p(x)\|q(x)) + D(p(y|x)\|q(y|x)) \tag{2.32}$$

Proof sketch: *Expand the LHS in log-sum form. Separate the term in the* log *into a sum of two* log *terms corresponding to the two divergences on the RHS.*

2.4 Jensen's Inequality & Consequences

Theorem 8. *Jensen's Inequality (and Convexity) For any convex function f and random variable X,*

$$\mathbb{E}\big[f(x)\big] \geq f(\mathbb{E}[X]) \tag{2.33}$$

Where "convex f in (a, b)" $\iff$

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2) \tag{2.34}$$

for all $x_1, x_2 \in (a, b)$ and $\lambda \in [0, 1]$. ***Strict*** *convexity has equality iff $\lambda \in \{0, 1\}$.* ***Concave*** *$f \iff$ convex $-f$.*

Proof sketch of Jensen's Inequality:

- *Start with $|\mathcal{X}| = 2$. Then we can let a vector $\mathbf{p} \in \mathbb{R}^2$ represent f(x).*
- *$\mathbb{E}[f(x)] = p_1 f(x_1) + p_2 f(x_2)$.*
- *$f(\mathbb{E}[X]) = f(p_1 x_1 + p_2 x_2)$.*
- *Show equivalence of Jensen's inequality $\iff$ $\mathbf{p}$ is a valid PMF.*
- *Expand to $|\mathcal{X}| = k - 1$ (induction proof).*
- *Use continuity arguments for continuous case.*

Theorem 9. *Implications of Jensen's Inequality*

1. ***Information Inequality:*** $D(p\|q) \geq 0$.
2. ***MI Inequality:*** $I(X; Y) \geq 0$.
3. ***Maximum Entropy:*** $H(X) \leq \log|\mathcal{X}|$. *Maximum is achieved by $X \sim uniform(\mathcal{X})$.*
4. ***Information can't Hurt:*** $H(X|Y) \leq H(X)$.
5. ***Entropy Sum Bound:*** $H(X_{1:n}) \leq \sum_{i=1}^{n} H(X_i)$ *with equality for independent X_i.*

2.5 Log-Sum Inequality & Consequences

Theorem 10. *Log-Sum Inequality Let $\{a_i, b_i\}_{i=1}^n$ be **non-negative** numbers. Then*

$$\sum_{i=1}^n a_i \log \frac{a_i}{b_i} \geq \big(\sum_{i=1}^n a_i\big) \log \frac{\sum a_i}{\sum b_i} \tag{2.35}$$

With equality iff $\frac{a_i}{b_i}$ is constant.

Proof sketch:

1. *Introduce $\alpha_i = a_i / \sum a_j$ and $t_i = a_i / b_i$. α_i values comprise a probability distribution (sum to 1).*
2. *Apply **Jensen's** to $\sum \alpha_i f(t_i) \geq f(\sum \alpha_i t_i)$ where $f() = \log()$.*

Theorem 11. *Applications of Log-Sum Inequality*

- ***Convexity of Relative Entropy:*** *$D(p\|q)$ is convex in pairs of p, q.*

$$D(\lambda p_1 + (1-\lambda)p_1 \| \lambda q_1 + (1-\lambda)q_2) \leq \lambda D(p_1\|q_1) + (1-\lambda) D(p_2\|q_2) \tag{2.36}$$

- ***Concavity of Mutual Information:***
  1. ***For fixed** $p(y|x)$**:** $I(X;Y)$ is concave in $p(x)$.*
  2. ***For fixed** $p(x)$**:** $I(X;Y)$ if concave in $p(y|x)$.*
  3. *This property is really handy when doing channel capacity calculations/bounds!*

2.6 Data Processing Inequality

Theorem 12. *Data Processing Inequality Let $X \to Y \to Z$ form a markov chain. That is, $p(x,y,z) = p(x)p(y|x)p(z|y)$. Then*

$$I(X;Y) \geq I(X;Z) \tag{2.37}$$

That is, the mutual information between X, Z is bounded by the mutual information between X, Y.

Proof sketch:

- *Expand $I(X;Y,Z)$ with the chain rule.*
- *($\star$) Observe that $I(X;Z|Y) = 0$ since X, Z are **conditionally independent** given Y (recall Markov theory – head-to-tail connection).*
- *Since $I(X;Y|Z) \geq 0$, we can conclude that $I(X;Y) \geq I(X;Z)$.*

Sufficient Statistics Connection: Assume that $X \to Y$ is a Markov chain. E.g., X are some parameters of an underlying distribution and Y are some data collected from the distribution. If $Z = f(Y)$, then we have $X \to Y \to Z$. Therefore any function Z on the data Y cannot have greater information on the underlying distribution X than Y did in the first place.

A **sufficient statistic** is one that has all the information that the data had about the underlying distribution. That is, $Z = f(Y)$ is a sufficient statistic on Y if $I(Z; X) = I(Y; X)$.

Minimal Sufficient Statistic: Let $\theta \to T(X) \to U(X) \to X$. $T(X)$ is the *minimal sufficient statistic* iff $U(X)$ can be any sufficient statistic and still have $\theta \to T(X) \to U(X) \to X$ be a valid Markov chain.

2.7 Fano's Inequality

Fano's inequality concerns estimators for random variables. Given some correlated random variables X, Y, we want to understand the probability of error when using an estimator $\hat{X}(Y)$ to approximate X. The big reveal is that $\Pr(\text{error}) = \text{function}(H(X|Y))$.

Theorem 13. *Fano's Inequality Let X, Y be dependent random variables. $\hat{X}(Y)$ is an estimator on X, so $X \to Y \to \hat{X}$ is a valid Markov chain for the joint distribution. Then*

$$H(P_{err}) + P_{err} \log |\mathcal{X}| \geq H(X|\hat{X}) \geq H(X|Y) \tag{2.38}$$

A weaker form of the same being:

$$1 + P_{err} \log |\mathcal{X}| \geq H(X|Y) \tag{2.39}$$

$$P_{err} \geq \frac{H(X|Y) - 1}{\log |\mathcal{X}|} \tag{2.40}$$

$$\tag{2.41}$$

Proof Sketch:

- *Represent the "error" event as random variable E. It takes value 1 if $\hat{X} \neq X$ and zero if $\hat{X} = X$.*
- $P_{err} = \mathbb{E}[E]$.
- *Expand $H(E, X|\hat{X}) = H(X|\hat{X}) + H(E|X, \hat{X})$, noting that the last term must be zero.*

- *Also note that $H(E|\hat{X}) \leq H(E)$ (conditioning reduces entropy).*
- *$H(X|E,\hat{X})$ can be expanded and shown to be bounded by $P_e \log|\mathcal{X}|$.*
- *Finally use Markov's inequality for $H(X|\hat{X}) \geq H(X|Y)$.*
- *Combine everything and you should be able to get the strongest version in Equation 2.38.*

You can also strengthen it to

$$H(P_{err}) + P_{err}\log(\underbrace{|\mathcal{X}|-1}_{(\star)}) \geq H(X|Y)$$

since one element of $\mathcal{X}$ is eliminated from the error entropy calculation by random guessing.

"Sharpness" of Fano's Inequality: If you have no information Y to inform $\hat{X}$, your best guess is $\hat{X} = \arg\max_x p(x)$. Equality in Fano's inequality (i.e., $H(P_{err}) + P_{err}\log(|\mathcal{X}|-1) \geq H(X)$) is achieved by $p(\hat{x}) = 1 - P_{err}$ and $p(x \neq \hat{x}) = P_{err}/(|\mathcal{X}|-1)$.

Bound on Collision: Let $X, X' \sim p(x)$. Then $\Pr\{X = X'\} = \sum_{x\in\mathcal{X}}(p(x))^2$. Then

$$\Pr(X = X') \geq 2^{-H(X)} \tag{2.42}$$

with equality if $X \sim unif(\mathcal{X})$.

Proof sketch: Expand RHS with entropy in $\mathbb{E}[]$ form. Apply Jensen's inequality, and you'll recover $\sum p(x)^2$ as the upper bound.

Collisions with Different Distributions: Let $X \sim p(x)$ and $\hat{X} \sim r(x)$. Then

$$\Pr(X \neq \hat{X}) \geq 2^{-H(p)-D(p\|r)} \tag{2.43}$$

$$\Pr(X \neq \hat{X}) \geq 2^{-H(r)-D(r\|p)} \tag{2.44}$$

$$\tag{2.45}$$

Proof sketch: We know the LHS is $\sum_x p(x)r(x)$. Expanding the RHS, we can apply Jensen's inequality if H and D are in $\mathbb{E}$ form. Then we will have a bound equal to LHS.

Chapter 3

Asymptotic Equipartition Theorem (AEP)

3.1 Typicality and AEP Theorem

AEP is the application of the **weak law of large numbers** (WLLN) to entropy. Recall WLLN:

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} X_i = \mathbb{E}[X] \tag{3.1}$$

Applying to entropy,

Theorem 14. *Asymptotic Equipartition Theorem Let $X_1 \dots X_n$ be iid with PMF $p(x)$. Then*

$$\frac{1}{n}\log\frac{1}{p(X_1,\dots,X_n)} \to H(x) \tag{3.2}$$

$$p(X_1,\dots,X_n) \to 2^{-nH(x)} \tag{3.3}$$

in probability. That is, for all $\epsilon > 0$, $\exists$ n such that the difference between the sample entropy (LHS) and the true entropy (RHS) is less than ϵ.

Proof sketch: *Apply WLLN to the definition of entropy. LHS = RHS in limit $n \to \infty$..*

Typical sequences: Sequences of x_i with **sample entropy** $\approx nH(X)$.

Typical set $A_\epsilon^{(n)}$ w.r.t. $p(x)$: A set of sequences $x^n \in \mathcal{X}^n$ that satisfy

$$2^{-n(H(X)+\epsilon)} \leq p(x_1,\dots,x_n) \leq 2^{-n(H(X)-\epsilon)} \tag{3.4}$$

Theorem 15. *Properties of* $A_\epsilon^{(n)}$

1. $x^n \in A_\epsilon^{(n)} \rightarrow$ *sample entropy* $-\frac{1}{n}\log p(x^n) \in [H(X)-\epsilon, H(X)+\epsilon]$.

2. $\Pr\{X^n \in A_\epsilon^{(n)}\} = \Pr\{A_\epsilon^{(n)}\} \geq 1-\epsilon$ *for large* n.

3. $|A_\epsilon^{(n)}| \leq 2^{n(H(x)+\epsilon)}$ *for all* n.

4. $A_\epsilon^{(n)} \geq (1-\epsilon)2^{n(H(x)-\epsilon)}$ *for sufficiently large* n.

Proof Sketches:

1. *Rearrangement of AEP/definition of* $A_\epsilon^{(n)}$.

2. *Apply the lower bound on* $\Pr x^n : x^n \in A_\epsilon^{(n)}$.

3. $\Pr\{A_\epsilon^{(n)}\} \leq 1$. *But we also know that* $\Pr(x^n) \geq 2^{-n(H(x)+\epsilon)}$ *if* $x^n \in A_\epsilon^{(n)}$. *Do the math.*

4. $\Pr\{A_\epsilon^{(n)}\} \geq 1-\epsilon$ *(from 2). Since typical sequences* x^n *have maximum probability* $2^{-n(H(X)-\epsilon)}$, *we can derive this bound.*

Chapter 4

Data Compression

Tail end of EIT chapter 3 and the entirety of chapter 5.

Data Compression – Problem Statement: We want to find the *shortest* codes for transmitting sequences x^n where each token is iid with PMF $p(x)$.

- We can leverage notions of **typicality** since we know that $X^n \in A_\epsilon^{(n)}$ with arbitrarily high probability for large n.
- **Simple implementation:** Introduce some ordering to elements in $A_\epsilon^{(n)}$. Since the size of $A_\epsilon^{(n)} \approx 2^{nH(X)}$, a binary index would need $n(H+\epsilon)+1$ bits. This is a pretty good code already!
- For items not in $A_\epsilon^{(n)}$, we can prepend a flag bit and use codes of length $n \log|\mathcal{X}|+1$.

4.1 Definitions & Source Coding Theorem

Compression/Source Coding Preliminaries:

- **Source:** Random variable $X : x \in \mathcal{X}$.
- **Codeword Alphabet:** $\mathcal{D}$ is a D-ary alphabet.
- **Source Code** $C : \mathcal{X} \to \mathcal{D}^*$ maps from the source alphabet to strings of arbitrary length from the code alphabet.
- **Expected Length** $L(C) = \sum_{x\in\mathcal{X}} p(x)\ell(x)$ where $\ell(x) = |C(x)|$.

- **Assumption:** We tend to represent every D-ary alphabet with natural numbers $\{0, 1, \dots, 1 - D\}$.
- **Non-singular:** Code C is non-singular iff

$$x_1 \neq x_t \implies C(x_1) \neq C(x_2) \tag{4.1}$$

- **Extension Code:** C^* is the extension of C –

$$C^*(x_1, x_2, \dots) = C(x_1)C(x_2)\dots \tag{4.2}$$

  I.e., the concatenation of $C(x_i)$.

- **Uniquely Decodable:** C is UD if C^* is non-singular.
- **Prefix Code:** No codeword $c(x_1)$ is a prefix of another codeword $c(x_2)$. This also means one can decode without any future information – it is an **instantaneous code**.

Theorem 16. *Source Coding Theorem Let X^n be iid with each $X_i \sim p(x)$. Then **there exists** a code with lengths satisfying:*

$$\mathbb{E}\big[\frac{1}{n}\ell(X^n)\big] \leq H(X) + \epsilon \tag{4.3}$$

for n sufficiently large. In other words, we can represent sequences of length n using $nH(X)$ bits on average!

Proof sketch:

- *For large n, $\Pr\{A_\epsilon^{(n)}\} \to 1$.*
- *Then $\mathbb{E}[\ell(X^n)] = \sum_{x^n \in \mathbb{X}^n} p(x^n)\ell(x^n)$.*
- *Split the sum into $x^n \in A_\epsilon^{(n)}$ and the compliment.*
- *Apply codes of length $n(H+\epsilon)+2$ for the first sum and lengths $n \log |\mathcal{X}| + 2$ to the second (1 flag bit, 1 rounding bit).*
- *Simplify using $\Pr\{A_\epsilon^{(n)}\}$ and upper bounds on $A_\epsilon^{(n)}$ size.*

4.2 Kraft Inequality

The Kraft inequality answers the question, “how short can codes get?”

Theorem 17. *Kraft Inequality Let $C : \mathcal{X}^n \to \mathcal{D}^*$ be a* ***prefix code****. Let $\{\ell_i\}_{i=1}^m$ be the codeword lengths for each $x^n \in \mathcal{X}^n$ (i.e., $m = |\mathcal{X}^n|$. Then*

$$\sum_{i=1}^{m} D^{-\ell_i} \leq 1 \tag{4.4}$$

And conversely: For any $\{\ell_i\}$ satisfying the inequality, ***there exists a prefix code with those lengths****!*

Proof sketch:

- *Consider a tree structure for all possible $\mathcal{D}^*$. Each layer adds one character, etc.*
- *At the deepest layer ℓ, there are D^ℓ leaf nodes.*
- *Each non-leaf node on layer ℓ_i has $D^{\ell-\ell_i}$ descendants.* ***To maintain prefix quality****, all descendants are eliminated!*
- *The number of descendants on the final layer must sum to $D^{\ell_{max}}$.*
- *Manipulate these equations and the inequality will pop out :)*

Theorem 18. *Extended Kraft Inequality Even for an infinite prefix code (i.e., countably infinite codewords), the lengths $\{\ell_i\}_{i=1}^\infty$ will satisfy*

$$\sum_{i=1}^{\infty} D^{-\ell_i} \leq 1 \tag{4.5}$$

Proof sketch: *Apply analogy to floating point numbers/place value. ”Descendants” represent intervals, they must be disjoint, etc.*

4.3 Finding Optimal Codes

We learned from the Kraft Inequality (Equation 4.4) that the codeword lengths are constrained to follow $\sum D^{-\ell_i} \leq 1$. Optimal codes will therefore solve the following optimization problem:

$$\min_{\{\ell_i\}} \sum_{i=1}^{m} p_i \ell_i \tag{4.6}$$

$$s.t. \sum_{i=1}^{m} D^{-\ell_i} \leq 1 \tag{4.7}$$

Lagrange Multiplier Solution: $\ell_i^* = -\log_D p_i$ satisfies Kraft inequality and minimizes $\mathbb{E}[\ell_i]$ – expected code length converges to $H_D(x)$. However, we need to round off code lengths so they're integers!

Theorem 19. *Expected code length inequality For a D-ary code and random variable $X : x \in \mathcal{X}$,*

$$L \geq H_D(X) \tag{4.8}$$

with equality iff $D^{-\ell_i} = p_i$ where $p_i = p(x_i)$.

Proof sketch: *Start by expanding $L - H_D(X)$. Combine the sums using product-sum log rule and recover $L - H_D(X) = D(\mathbf{p}\|\mathbf{r})$ where $r_i = D^{-\ell_i}/\sum_j D^{-\ell_j}$. By non-negativity of $D(\|)$, the proof is done.*

D-adic Distributions: $p(x)$ is D-adic if $\exists\{n_i\}$ such that

$$p(x_i) = D^{-n_i} \tag{4.9}$$

where each n_i is a natural number.

Generally, we want a D-adic distribution $p(x)$ so that our codes achieve expected length $L = \sum p_i \ell_i = H_D(X)$. Failing that, we want to **minimize** $D(d - adic\|p(x))$. We are approximating $p(x)$ with the closet D-adic distribution. That's really all that coding methods boil down to!

- Shannon-Fano: Offers good, easy, suboptimal codes.
- Huffman: Truly optimal codes based on $p(x)$ – we actually find the nearest D-adic distribution!

To summaryze: we have just established some bounds on **code lengths** $\{\ell_i\}_{i=1}^m$ for n-ary transmissions from the set $\mathcal{X}^n$, we can go ahead and take a look at the bounds on **expected code lengths** $L = \mathbb{E}[\ell_i] = \sum_{x^n \in \mathcal{X}^n}$! This will help us understand the true practical information transmission limitations – after all, our concern is primarily in aggregate behavior for communication systems.

Theorem 20. *1-Bit Bound on Expected Code Length L Let L represent the expected code length $L = \mathbb{E}_{p(x)}[\ell(x)]$. Then the optimal code will produce the minimum expected code length L^* where L^* is bounded as*

$$H(X) \leq L^* \leq H(X) + 1 \tag{4.10}$$

Proof sketch:

- *Recall that the optimal code will have lengths $\{\ell_i\}$ that give rise to the nearest diadic distribution $p(x_i) \approx D^{\ell_i}/\sum D^{\ell_i}$.*
- *Let us introduce* $\mathbf{r}$ *as follows:*

$$r_i = \{\frac{D^{-\ell_i}}{\sum_j D^{\ell_j}}\}$$

- *Then we can reframe the search for optimal code lengths $\{\ell_i^*\}$ as the minimization of the following function:*

$$\min_{\{\ell_i\}} \left[D(\mathbf{p}\|\mathbf{r})\right] - \log(\sum D^{-\ell_i})$$

Instead of doing anything clever, let's just use the ***rounded version of the non-integer solution****. That is,*

$$\ell_i = \lceil \log_D \frac{1}{p_i} \rceil \tag{4.11}$$

Arbitrary Closeness to Optimal $L = H(X)$: Observe that we are able to make the per-character average length $\frac{1}{n}L \to \frac{1}{n}H(X^n) = nH(X)$ arbitrarily close by increasing n since $H(X^n) + 1$ is an upper bound on optimal L^*.

Wrong Code Cost: If we create optimal codes according to $q(x)$, then $\ell_i = \lceil \log \frac{1}{q(x_i)} \rceil$. The resulting code length under real conditions $p(x)$ will be $\mathbb{E}_p[\ell(X)]$. It is given by

$$H(p) + D(p\|q) \leq \mathbb{E}_p[\ell(X)] < H(p) + D(p\|q) + 1 \tag{4.12}$$

Proof sketch: Start by expanding $\mathbb{E}_{p(x)}[\ell(X)]$ into sum form, with $\ell(x) = \lceil \log \frac{1}{q(x)} \rceil$. You'll get a sum-log term that you can expand into $H(p)$ and $D(p\|q)$ by multiplying the term in the log by $p(x)/p(x)$ □

4.3.1 Generalizing Kraft Inequality to all UD Codes

So far, we have been proving bounds for expected code length in **prefix codes**. We now generalize these findings to all **uniquely decodable** codes. The big takeaway is that UD codes cannot out-perform prefix codes with respect to code length.

Theorem 21. *McMillan Theorem – Any UD Code Satisfies Kraft Inequality For any uniquely decodable code $c : \mathcal{X}^n \to \mathcal{D}^*$, the code lengths $\ell_i = |c(x_i)|$ for each $x_i \in \mathcal{X}$ must satisfy*

$$\sum D^{-\ell_i} \leq 1 \tag{4.13}$$

And conversely: Given any $\{\ell_i\}$ satisfying the Kraft inequality, we can construct a UD code that has lengths $\{\ell_i\}$.

Proof sketch: *Our goal is to show that the code lengths must satisfy $\sum_{x \in \mathcal{X}} D^{-\ell(x)} \leq 1$.*

- *We start by considering the number of different D-ary codewords of length n the code $c(x)$ can produce.*
- *Since there only exist D^n unique D-ary sequences of length n, $c^k()$ (the kth extension of c) can only produce D^n sequences of length n.*
- ***Trick**: We do some cursed manipulations of the following term –*

$$\begin{align} &= \big(\sum_{x \in \mathcal{X}} D^{-\ell(x)}\big)^k \tag{4.14}\\ &= \big(\sum_{x_1 \in \mathcal{X}} D^{-\ell(x_1)}\big)\big(\sum_{x_2 \in \mathcal{X}} D^{-\ell(x_2)}\big) \dots \big(\sum_{x_k \in \mathcal{X}} D^{-\ell(x_k)}\big) \tag{4.15}\\ &= \sum_{x_{1:k} \in \mathcal{X}^k} D^{-\ell(x_1)} D^{-\ell(x_2)} D^{-\ell(x_k)} \tag{4.16}\\ & \tag{4.17} \end{align}$$

  The reason this is a valid manipulation is that you can push around $\sum_{x_i}$ around in the product as long as x_i remains within its argument field.

- *We now apply a change of variables. Specifically, for each sum $\sum_{x^k} D^{-\ell(x_k)}$, we replace it with*

$$\sum_{m=1}^{k\ell_{max}} a(m) D^{-m}$$

  where $a(m)$ is the number of m-long codes. This is equivalent to $\sum_x D^{-\ell(x)}$ – we are just grouping and calculating by code lengths m.

- *$a(m)$ is the number of m-long codes, and there cannot be more than D^m of those. So all of a sudden, we have*

$$
\begin{aligned}
&= (\sum_{x \in \mathcal{X}} D^{-\ell(x)})^k && (4.18)\\
\sum_{x^k \in \S^k} D^{-\ell(x^k)} &= \sum_{m=1}^{k\ell_{max}} a(m) D^{-m} && (4.19)\\
&\leq \sum_{m=1}^{k\ell_{max}} D^m D^{-m} && (4.20)\\
&= \ell_{max} k && (4.21)
\end{aligned}
$$

- *So now we know that*

$$
\begin{aligned}
(\sum_{x \in \mathcal{X}} D^{-\ell(x)})^k &\leq \ell_{max} k && (4.22)\\
\implies \sum_{x \in \mathcal{X}} D^{\ell(x)} &\leq (\ell_{max} k)^{1/k} && (4.23)\\
\implies \sum_{x \in \mathcal{X}} D^{-\ell(x)} &\leq 1 && (4.24)
\end{aligned}
$$

  With the last transformation justified by the fact that $\lim_{k \to \infty} \ell_{max} k = 1$

- *Therefore Kraft's inequality holds not only for prefix codes, but for all UD codes □.*

4.4 Huffman Codes

Huffman coding is provably optimal – that is, it yields the minimum possible expected code length $L = \mathbb{E}_{p(x)}[\ell(x)]$. The general idea is to continually generate a **shared prefix** for the D least likely remaining symbols. Once grouped, the codes for each of the least likely symbols will only differ in the last code character. The weird **recursive** part of the algorithm is how the new **shared prefix** is treated like a single source symbol in the next iteration – hence why the algorithm is so popular as a dynamic programming/recursion exercise.

Overall, the intuition boils down to this: compression is aided by **degeneracy** – when very similar codes can represent many different objects. This is an exploitation in the structure of the data: every time we group together the least likely terms, we are **adding 1** to their code length. The other symbols that were untouched essentially get to survive another iteration without having a character added to their code. In many ways, this is like the opposite of PCA, matching pursuit, and other compression techniques that first generate representations of information contained in the **most common** elements. One can also connect the ideas behind Huffman coding to concepts in **search** – unlike most search algorithms where the performance is averaged evenly across search time to all elements, Huffman codes incorporate **weights** (i.e., probability of a source token) into the search procedure. You can even have the weights violate the "sum to 1" rule of probability distributions. Best of all, it's still a provably optimal scheme with respect to code length/number of search queries!

Overall, Huffman codes are pretty neat :)

Algorithm Sketch:

1. Construct a table of source symbols $x \in \mathcal{X}$ and their probabilities $p(x)$. Order this table from high to low probability.

2. For $D \geq 3$, add dummy symbols with 0 probability such that the number of symbols is $1 + k(D-1)$ for integer k.

3. **For each iteration:** Combine the least likely D symbols into one symbol for the next iteration.

   (a) Draw arrows from each of the combined symbols to a new entry in a new probability column. Assign each arrow a number from $\{0, \ldots, D\}$.

   (b) Repeat this process with the new probability column.

4. The result of this iterative procedure will be a tree-like structure with each path ending at a single symbol in the final column with probability 1.

5. To construct the codes for each source symbol: Follow the arrows back from the $\Pr = 1$ top-level node.

4.4.1 Optimality of Huffman Codes

Like with most recursive algorithms, the proof for correctness/optimality is inductive. The central idea to keep in mind is our **optimality condition**: we with to solve $\min \sum p_i \ell_i$.

Big Picture: To prove the optimality of Huffman codes, we will do the following:

1. **Canonical codes**: Optimal codes can be hard to reason about since there are lots of different optimal codes. Some will be prefix codes, others won't. Therefore, the first thing we do is **narrow the search** for optimal codes to a certain class. To do this, we need to prove the existence of optimal codes of this class. We call these codes "**canonical codes**", and their properties will enable the rest of the proof.

2. **Induction**: Now that we have established what a "canonical code" is, we get to the fun part! We start by naming the "combine the D least likely symbols" operation from the Huffman code generation process a "**Huffman reduction**". We then show that, assuming that the code of the **reduced** set of symbols is optimal/canonical, then so will the **unreduced** code.

So, let's get started! First, we define a **canonical code**.

Theorem 22. *Existence of Canonical Codes For all $p(x)$, there exists an **optimal source code** with the following properties:*

1. *$p(x_1) > p(x_2) \implies \ell(x_1) \leq \ell(x_2)$.*

2. *The two longest $\ell(x_i), \ell(x_j)$ are equal.*

3. *The two longest $c(x_i), c(x_j)$ differ **only in the final bit**. They also correspond to the **least likely source symbols**.*

*We call optimal codes with these properties **canonical codes** (it sounds awfully clever).*

Proof sketch:

- *How to show $p(x_1) > p(x_2) \implies \ell(x_1) \leq \ell(x_2)$:*

  1. *Imagine swapping the codewords for x_1, x_2.*
  2. *Then the sum $\sum p_i \ell_i$ will be strictly greater than if they were unswapped* $\square$

- *How to show that the two longest $\ell(x_i), \ell(x_j)$ will have the same lengths:*

  1. `Assume toward a contradiction` *that they have different lengths.*
  2. *Recall the prefix property: the second longest cannot be a prefix of the longest.*

3. *Therefore deleting the last character(s) of the longest would yield a shorter code that is still unique.*
4. *This would strictly reduce $\mathbb{E}[\ell(x)]$! □*

- *How to show that the two longest codes only differ in the final bit AND correspond to the lowest probability symbols:*

  1. *For an optimal prefix code, imagine that the longest two codes are NOT siblings. That is, they differ in more than just the final bit.*
  2. *Then we should be able to delete the last bit of each of them and still have a UD/prefix code (since none of the other codes will be prefixes of them).*
  3. *We can also just make them siblings with no cost to code lengths.*
  4. *Therefore, the longest two codes should only differ in the last bit. They should also correspond to the lowest probability $x \in \mathcal{X}$ given property (1) □*

Now that we've established that canonical codes exist and are optimal, we can move on and prove that Huffman codes are canonical at each iteration and are therefore optimal!

Huffman Reduction: We define a Huffman reduction on a probability distribution $p(x)$ where $x \in \mathcal{X} = \{x_1, \ldots, x_m\}$. Let $\mathbf{p} = [p(x_1), \ldots, p(x_m)]^\top$ be ordered from most likely $p(x_1)$ to least likely $p(x_m)$. Then the Huffman reduction of $\mathbf{p}$ is

$$\mathbf{p}' = \big[p_1, p_2, \ldots, p_{m-2}, \underbrace{p_{m-1} + p_m}_{\text{reduction!}}\big]^\top \tag{4.25}$$

Now all we have to do is show that the **optimal code** for the reduced $\mathbf{p}'$ can be expanded to the optimal code for unreduced $\mathbf{p}$.

Theorem 23. *Optimality of Huffman Codes Let C^* be a Huffman code and C' be any uniquely decodable code. Then*

$$L(C^*) \leq L(C') \tag{4.26}$$

Proof sketch:

1. *Huffman code generation is essentially a series of* ***expansion operations*** *from a reduced $\mathbf{p}'$ to an unreduced $\mathbf{p}$*

2. *Use **proof by contradiction** to show that codes extended from $\mathbf{p}' \rightarrow \mathbf{p}$ maintain optimality.*

3. *Therefore, if the first code is optimal (must be since it's 1 element), then the rest of the expansions must also be optimal. Therefore, Huffman coding produces optimal codes* $\square$

4.5 Shannon Codes

Shannon codes give a procedure to construct a prefix code with lengths $\ell(x) = \lceil \log \frac{1}{p(x)} \rceil$. These aren't optimal like Huffman codes, but they're pretty good. In fact, there's a whole section in the textbook dedicated to its "competitive optimality" with Huffman codes. Perhaps people feel the need to defend Shannon coding because Shannon is so central to information theory. Regardless of whether it is deserved, you'll probably be asked to do a Shannon code example at some point in your life, and it's a somewhat elegant mathematical construction.

Big Picture: I'm not sure if this is how they were developed, but this is the story I tell myself.

1. We know that $\ell(x) = \lceil \log \frac{1}{p(x)} \rceil$ is achievable. But if someone asks me to actually code some x, where do I even get the values for $c(x)$? Do I make them randomly?

2. **CDF encoding idea:** The CDF of a random variable has a really nice property: let $F(x)$ be the CDF of random variable X. Then $F : \mathcal{X} \rightarrow [0, 1]$ in a one-to-one manner! So let's try using the bits of $F(x)$ to create our code $c(x)$!

3. That's great, but $F(x)$ could take an infinitey number of bits to encode! We need to **truncate** them somehow...

4. **Truncation technique:** Just take the first $\ell(x) = \lceil \log \frac{1}{p(x)} \rceil + 1$ bits from $F(x)$ to get truncated $\lfloor F(x) \rfloor_{\ell(x)}$.

5. We can show that this will still produce a prefix code via **extreme cleverness**.

I'll elaborate more on that last point (why the truncation technique ends up working) momentarily, but that's pretty much how Shannon coding works. The only important caveat is that we don't use $F(x)$ directly – we use bits from a modified CDF $\tilde{F}(x) = F(x) + \frac{1}{2}p(x)$. This corresponds to the midpoint between $F(x-1)$ and $F(x)$.

Big result: The scheme gives us a code with expected length $L \leq H(X) + 2$. Aren't we so lucky for that.

Why does $\lceil \tilde{F}(x) \rceil_{\ell(x)}$ work? What a great question! Let's start by considering what "prefix-free" corresponds to in the context of using bits from the binary representation of $\tilde{F}(x)$.

- Let's assume you truncate to digit ℓ after the decimal point. Under the prefix condition, no codes that share those first ℓ digits.
- This corresponds to **claiming** the range $[0.z_1 z_2 \dots z_\ell, 0.z_1 z_2 \dots z_\ell + \frac{1}{2^\ell}]$. That last $+\frac{1}{2^\ell}$ term corresponds to the extra value if all the digits past z_ℓ were ones.
- Now you just have to show that there's no **overlap** in the **claimed regions** of each x!
- At that point, it's just a bunch of symbol shunting. So I really can't be bothered to write it all out, and I suggest you refrain as well – I suspect that symbol shunting is bad for one's health.

That's pretty much Shannon coding. You get some decent preformance out of it, it uses some clever ideas, and sometimes thats enough. I'm not going to get into the "competitive optimality" too much since it's terribly boring, but these are the main takeaways:

1. If $ell(x)$ are Shannon code lengths and $\ell'(x)$ are any other UD code, then $\Pr(\ell(x) \geq \ell'(x) + c) < \frac{1}{2^{c-1}}$.
2. Given a dyadic $p(x)$ (expressible in base 2), let $\ell(x) = -\log p(x)$ and $\ell'(x)$ be any other Shannon code. Then $\Pr(\ell(X) < \ell'(x)) \geq \Pr(\ell(X) > \ell'(x))$.

See? I told you it was boring.

An important note on "competitive optimality": Most of this stuff is framed in probabilities that one is shorter/longer than the other. The reason Shannon coding secretly sucks is that it's pretty suboptimal in practical **expected length**. If you have a distribution $p(x_1) = 0.001, p(x_2) = 0.999$ then Shannon coding would tell you to use insane large code lengths for x_1 for no reason. The point is, don't believe all the propaganda about Shannon coding.

Chapter 5

Entropy of Markov Processes

We all know that Markov processes are cool. But what are their entropies?

This gets down to the question, “but what if the elements of the sequences aren’t iid?”. Markov processes offer a nice way to understand that.

Big Result: The entropy of random variables $X_{1:n}$ generated by a Markov process grows linearly w.r.t. n at rate $H(\mathcal{X})$. This is called the **entropy rate** of the system, and represents the **best achievable data compression** for the process.

5.1 Markov Definitions

Stochastic Process: A sequence $\{X_i\}_{i=1}^{n}$ of random variables. They are allowed to have arbitrary dependence with PMF $p(x_{1:n})$.

$$\Pr\Big\{(X_1, \dots, X_n) = (x_1, \dots, x_n)\Big\} = p(x_1, \dots, x_n) \tag{5.1}$$

Stationary Stochastic Process: When $p(x^n)$ is invarient to any shifts in time for all n.

Markov Chain/Process:

$$\Pr(x_{n+1}|x_{1:n}) \equiv \Pr(x_{n+1}|n) \tag{5.2}$$

This also implies that the following factorization of the PMF is valid:

$$p(x^n) = p(x_1)p(x_2|x_1)\dots p(x_n|x_{n-1}) \tag{5.3}$$

Time Invariant Markov Chain: When $p(x_{n+1}|x_n)$ depends only on the **values** x_{n+1}, x_n – not n.

Irreducibility: When there exists a non-zero probability path between all states $i \to j$ for $i, j \in \mathcal{X}$.

Aperiodic: The **largest common factor** for all closed path lengths is 1. This ensures that the steady state behaviour is not periodic.

Stationary distribution: Let $\mathbf{P}$ be the transition matrix. Then the probability distribution $\mathbf{p} = [p(x_1) \dots p(x_n)]$ is the steady state distribution if

$$\mathbf{p} = \mathbf{P}\mathbf{p} \tag{5.4}$$

5.2 Entropy Rate

Theorem 24. *Entropy Rate Definition For any stochastic process $\{X_i\}_{i=1}^n$, the entropy rate is defined as*

$$H(\mathcal{X}) = \lim_{n\to\infty} \frac{1}{n} H(X_1, X_2, \dots, X_n) \tag{5.5}$$

when the limit exists.

Related Quantity: We define $H'(\mathcal{X})$ as

$$H(\mathcal{X}) = \lim_{n\to\infty} H(X_n | X_{n-1}, \dots, X_1) \tag{5.6}$$

When it exists.

We will also make use of the **Cesaro mean** to prove equivalence between $H(\mathcal{X})$ and $H'(\mathcal{X})$.

Theorem 25. *Cesaro Mean If $a_n \to a$ and $b_n = \frac{1}{n}\sum_{i=1}^n a_i$, then*

$$b_n \to a \tag{5.7}$$

Proof sketch: *Just do a $\delta - \epsilon$ proof for this.*

Cesaro Mean → Existence of $H'(\mathcal{X})$: We know that conditioning reduces entropy. Therefore, $H(X_n|X_{1:n-1})$ is **non-increasing** with n (assuming *stationary* process) and has limit $H'(\mathcal{X})$. Therefore, the limit exists! □

Theorem 26. *Entropy Rate $H(\mathcal{X}) \equiv H'(\mathcal{X})$ Let's start by expanding the definition of $H(\mathcal{X})$:*

$$H(\mathcal{X}) = \lim_{n\to\infty} \frac{1}{n} H(X_1, \ldots, X_n) \tag{5.8}$$

$$= \lim_{n\to\infty} \frac{1}{n} \left[\sum_{i=1}^{n} \underbrace{H(X_i|X_{1:i-1})}_{\to H'(\mathcal{X})} \right] \tag{5.9}$$

*The transformation from line 1-2 is justified by the **chain rule for entropy**. By the Cesaro mean theorem, the last line implies that $H(\mathcal{X}) = H'(\mathcal{X})$.* □

Entropy of 1st Order Stationary Markov Chain: The big takeaway is that for transition matrix $\mathbf{P}$ and steady state distribution μ,

$$H(\mathcal{X}) = -\sum_{ij} \mu_i \mathbf{P}_{ij} \log \mathbf{P}_{ij} \tag{5.10}$$

Proof sketch: We know that $H(\mathcal{X}) = H'(\mathcal{X}) = \lim_{n\to\infty} H(X_n|X_{1:n-1})$. However, since it's a 1st order markov chain, $\lim_{n\to\infty} H(X_n|X_{1:n-1}) \equiv H(X_n|X_{n-1})$. Now we just need to calculate $H(X_2|X_1)$ where $X_1 \sim \mu$ and X_2 is... also $\sim \mu$:)

Chapter 6

Channel Capacity

We've spent a lot of time on what essentially amounts to data compression. If you ever make a friend, though, you may actually need to *send* data. So let's study "channels" – stuff you would send that information through.

Discrete Channel Definition:

1. Input alphabet $\mathcal{X}$.
2. Output alphabet $\mathcal{Y}$.
3. Transition matrix $p(y|x)$ – this is the actual channel.

It should be noted that we mainly care about **memoryless** channels where the input at a given time depends only on the current input symbol.

6.1 Channel Capacity

Channel Capacity: The amount of information you can transmit on average via a single use of the channel with vanishingly small error probability.

$$C = \max_{p(x)} \{I(X;Y)\} \tag{6.1}$$

Where $p(y|x)$ is dictated by the channel itself. We basically get to "pick" out $p(x)$ to maximize C. We generally want to get as close to the theoretical channel capacity as possible. But before we do anything so practical, we need to faff around with channel capacity calculations and properties of channel capacity.

Top 10 Tips for Calculating Channel Capacity: Probably not actually 10, but you get the point.

- **Start with the definition.** If no inspiration strikes, just write down the definition. It's particularly important to write the equivalent forms of $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$.
- **Worst case: leverage convexity and solve for 0 gradient.** Recall how $I(X;Y)$ is convex w.r.t. $p(x)$ given fixed $p(y|x)$. Representing $p(x)$ as a vector of values, you can solve for the point with zero gradient of $I(X;Y)$ with respect to each of the values in $p(x)$.
- **One-to-many but disjoint** Y yields capacity equal to the number of output clusters.
- **(Maximum) number of bits transmitted** per use of the channel is identical to capacity. Sometimes this is enough for a hand-wavey calculation of capacity if you already know what it is.
- **Noisy typewriter** is a big example to keep in mind. If the output is basically a "diffused" version of the input, a uniform distribution will suffice to maximize. This is basically the same as a symmetric channel.
- **Symmetry is your friend.** Check if the channel is symmetric first – it gives you great guarantees (e.g., uniform distribution maximizes capacity).
- **Binary erasure channel** with erasure probability α has capacity $1-\alpha$ if there are two elements in the input space and 3 elements in the output space (i.e., two correspond to the input and 1 is the "erased" outcome).

Top 10 Properties of Channel Capacity:

1. $C \geq 0$
2. $C \leq \min\{\log \mathcal{X}, \log \mathcal{Y}\}$.
3. $I(X;Y)$ is a continuous function of $p(x)$ and is a **concave** function w.r.t. $p(x)$. Therefore local optima are global optima.
4. There's no closed form solution for capacity in the general case.

6.2 Symmetric Channels

Symmetric channels are great. Can't get enough of them. Nobody loves symmetric channels more than I do.

Symmetric Channel Definition: When the rows of the channel are all permutations of eachother, and so are all the columns. Note that the xth row of the matrix corresponds to the input x and the yth column corresponds to the probability of that output. Essentially, the **rows sum to 1**.

Properties of Symmetric Channels:

- The maximizing $p(x)$ for $I(X;Y)$ is the uniform distribution.
- $I(X;Y) \leq \log|\mathcal{Y}| - H(\mathbf{r})$ where $\mathbf{r}$ is a row of the transition matrix $(p(:|x))$.

Weakly Symmetric Channels: Rows are all permutations of eachother and all columns sum to the same value. Capacity is still $C = \log|\mathcal{Y}| - H(\mathbf{r})$ and is still achieved by a uniform distribution on the input.

6.3 Channel Coding Theorem

Theorem 27. *Channel Coding Theorem For a discrete memoryless channel, all rates $R < C$ are achievable.*

Yup, that's the big takeaway. We need to go through a lot to prove it, though. We start with defining some error probability metrics and what exactly "achievable" means. To keep things classy and WLLN-ish, we do most of this in the $n \to \infty$ regime. That lets us invoke AEP theorem and assume that pretty much every input/output is **jointly typical** (which we will also need to define – it's not that bad though). After all that, we're ready to prove the channel coding theorem. We do this by randomly generating our codewords, then focusing on just the first one $c(x_1)$. Since everything's random, $c(x_1)$ can stand in for every codeword when we calculate the probability of error. The receiver just uses **joint typicality decoding** to determine which output input sequence x^n matches the received y^n. Thanks to AEP stuff, error goes to zero and everyone's happy.

So let's get cracking with all the definitions and theorems! Lots to do.

6.3.1 Definitions

Discrete memoryless channel: Defined above, represented as $\mathcal{X}, p(y|x), \mathcal{Y})$.

nth extension of DMC: Since there's no memory, $(\mathcal{X}^n, p(y^n|x^n), \mathcal{Y}^n$ is well defined. Just let $p(y^n|x^n) = \prod_{i=1}^n p(y_i|x_i)$.

(M, n) code: Given a set of M messages $\{1, \dots, M\}$, the (M, n) code gives an encoding function $X^n : \{1, \dots, M\} \to \mathcal{X}^n$. It essentially uses n-long sequences from alphabet $\mathcal{X}$ to encode one of M messages. It also gives a decoding function $g : \mathcal{Y}^n \to \{1, \dots, M\}$. It's a more general scheme than it may seem since the M messages can apply to any collection of M objects – maybe even source codes.

Conditional probability of error: Given that our input message is $i \in \{1, \dots, M\}$, the conditional probability of error λ_i is

$$\begin{aligned} \lambda_i &= \Pr\{g(Y^n) \neq i | X^n = x^n(i)\} && (6.2) \\ &= \sum_{y^n \in \mathcal{Y}^n} p(y^n|x^n(i)) \mathbb{I}[g(y^n) \neq i] && (6.3) \\ & && (6.4) \end{aligned}$$

Maximum probability of error: $\lambda^{(n)}$ is exactly what it sounds like:

$$\lambda^{(n)} = \max_{i \in [M]} \lambda_i \tag{6.5}$$

Arithmetic average error probability: Also exactly what it sounds like.

$$P_e^{(n)} = \frac{1}{M} \sum_{i=1}^{M} \lambda_i \tag{6.6}$$

It's the general error probability if input messages are chosen uniformly from $[M]$.

Rate of (M, n) code: Number of bits per transmission

$$R = \frac{\log M}{n} \tag{6.7}$$

Though not all rates are *achievable*...

Achievability: A rate R is achievable if $\exists$ a **sequence** of $(\lceil 2^{nR} \rceil, n)$ codes such that

$$\lim_{n\to\infty} \lambda^{(n)} = 0 \tag{6.8}$$

Capacity: The **supremum** of all **achievable** rates. Importantly, this implies that a rate less than the capacity implies that an error probability approaching zero with large block sizes is possible!

6.3.2 Joint Typicality

Theorem 28. *Jointly Typical Sets Definition $A_\epsilon^{(n)} = \{(x^n, y^n)\}$ are the set of $\{(x^n, y^n)\}$ that have **sample entropies** that are ϵ-close to the true entropies (calculated by $p(x^n, y^n)$).*

$$\begin{aligned}
A_\epsilon^{(n)} = \{(x^n, y^n) \in \mathcal{X}^n \times \mathcal{Y}^n : & \quad (6.9)\\
\Big|\frac{-1}{n} \log p(x^n) - H(X)\Big| \le \epsilon, & \quad (6.10)\\
\Big|\frac{-1}{n} \log p(y^n) - H(Y)\Big| \le \epsilon, & \quad (6.11)\\
\Big|\frac{-1}{n} \log p(x^n, y^n) - H(X, Y)\Big| \le \epsilon, & \quad (6.12)\\
\} & \quad (6.13)
\end{aligned}$$

Theorem 29. *Joint AEP Let (X^n, Y^n) be sequences generated with with distribution $p(x^n, y^n) = \prod_{i=1}^n p(x_i, y_i)$ (iid). Then*

$$\begin{aligned}
\Pr\{(X^n, Y^n) \in A_\epsilon^{(n)}\} &\to 1 \text{ as } n \to \infty & (6.14)\\
|A_\epsilon^{(n)}| &\le 2^{n(H(X,Y)+\epsilon)} & (6.15)\\
\Pr\{(\tilde{X}^n, \tilde{Y}^n) \in A_\epsilon^{(n)}\} &\le 2^{-n(I(X;Y)-3\epsilon)} & (6.16)\\
\Pr\{(\tilde{X}^n, \tilde{Y}^n) \in A_\epsilon^{(n)}\} &\ge (1-\epsilon) 2^{-n(I(X;Y)+3\epsilon)} & (6.17)
\end{aligned}$$

*Where $(\tilde{X}^n, \tilde{Y}^n) \sim p(x^n)p(y^n)$ are **independently sampled** sequences.*

The big upside are those last two statements – as long as you increase n, the probability of incorrectly associating two $\tilde{X}^n, \tilde{Y}^n$ that aren't actually sampled from $p(x, y)$ is vanishingly small as long as you use jointly typical decoding.

Proof sketch: Probability of $A_\epsilon^{(n)} \to 1$ Literally just use the law of large numbers. Sample entropy convergest to expected value which is the actual entropy for all 3 of the conditions in the definition of $A_\epsilon^{(n)}$.

Proof sketch: $|A_\epsilon^{(n)}| \leq 2^{n(H(X,Y)+\epsilon)}$ The definition states the max/min probability of the sequence (x^n, y^n) is it's in $A_\epsilon^{(n)}$ – you just need to rephrase the entropy constraint. Using this, and the fact that the probability of the typical set in total is less than 1, you get this bound.

Proof sketch: Independently sampled sequences have low probability in typical set. This proof is actually a little cursed. I'm just going to have to write the 3 cursed lines:

$$\Pr\left((\tilde{X}, \tilde{Y}) \in A_\epsilon^{(n)}\right) = \sum_{(x^n, y^n) \in A_\epsilon^{(n)}} p(x^n) p(y^n) \tag{6.18}$$

$$\leq \underbrace{2^{n(H(X,Y)+\epsilon)}}_{|A_\epsilon^{(n)}|} \cdot \underbrace{2^{-n(H(X)-\epsilon)} \cdot 2^{-n(H(Y)-\epsilon)}}_{\text{Max } p(\tilde{x}^n), p(\tilde{y}^n) : (\tilde{x}^n, \tilde{y}^n) \in A_\epsilon^{(n)}} \tag{6.19}$$

$$= 2^{-n(I(X;Y)-3\epsilon)} \tag{6.20}$$

6.3.3 Proof of Channel Coding Theorem

Finally we can get to the channel coding theorem proof! After all the work we have put in, it's not so bad – the starting point is easy to forget, though.

Messaging Procedure for Channel Coding Theorem: Just to make it completely clear, this is the procedure we use to transmit and receive messages under the channel coding theorem:

1. Sender selects some message $W \in [M]$ from the index set to send. We generally assume that this is uniformly sampled.
2. Sender encodes i as $x^n(W) \in \mathcal{X}^n$ – the codeword for index set message W.
   - The "codebook" C consists of all $x^n(w): \ w \in [M]$.
3. Sender chucks $x^n(i)$ into the channel $p(y^n|x^n)$ and the receiver samples y^n from that channel's probabilistic output.
4. The receiver **guesses** the original message W using the following estimation procedure: We find $\hat{W}$ such that

- Find $\hat{W} : (x^n(\hat{W}), y^n) \in A_\epsilon^{(n)}$ – i.e., jointly typical decoding.
- **Non-Unique Case** If $\nexists W' \neq \hat{W}$ such that $(x^n(W'), y^n) \in A_\epsilon^{(n)}$ AND $(x^n(\hat{W}), y^n) \in A_\epsilon^{(n)}$, **RETURN ERROR**.

Our goal is essentially to understand the bounds on error probabilities associated with this procedure. Errors occur when we return **ERROR** and also when we incorrectly decode $\hat{W}$ – i.e., $\hat{W} \neq W$.

Starting the Channel Coding Theorem Proof: Our goal is to show that all rates of transmission $R < C$ are **achievable** – that is, we can construct a sequence of (M, n) codes with that rate R such that the **maximum error probability** $\lambda^{(n)}$ goes to zero as n goes to infinity.

That's all a bit of a mouthful, but you should read it until it's seared into your memory. We start the channel coding theorem by replacing M with 2^{nR} in the $(2^{nR}, n)$ code statement. Recall that rate $R = \log(M)/n$. So for any given n, R, setting $M := 2^{nR}$ will guarantee that rate. From there, you just have to prove **achievability** – that $\lambda^{(n)} \to 0$.

So, to summarize our roadmap:

1. We want to show that rates $R < C$ are achievable.
2. Achievability of $R \equiv \exists$ a sequence of (M, n) codes with rate R such that max error probability $\lambda^{(n)} \to 0$ as $n \to \infty$.
3. We set $M := 2^{nR}$. This yields a code with rate $R = \log(M)/n = \log(2^n R)/n = R$.
4. To show **achievability**, we now just need to show that $\lambda^{(n)} \to 0$ for some sequence of codes!

Random Codes: Big Idea 1 As everyone likes to harp on about, Shannon was a very clever guy. One of the clever things he did was use **random codes** to prove the channel coding theorem. For each of the $M = 2^{nR}$ input messages, he generated a random $x^n(i) : \ i \in [M]$ where each index $x_j \sim p(x)$ for some fixed $p(x)$. We then show that $\lambda^{(n)} \to 0$ under the condition $R < C$ when we use **joint typicality decoding**. Hopefully this makes you glad we did all that stuff with joint AEP, because we basically just get to yeet that in at the end to prove that the error rates go to zero.

Bound Error Rate for $x^n(1)$: Big Idea 2 Another life hack we get to use in the proof is to use $x^n(1)$ (i.e., the codeword for input number 1 from index set $[M] = 2^{nR}$) as a stand-in for all $x^n(i)$ when bounding error probabilities. This is allowable since all the $x^n(i)$ were generated randomly using the same process. This really improves our quality of life over the course of the proof.

Channel Coding Theorem Proof Sketch:

- Message is sent: $W \to x^n(W) \to y^n$ where each element of $x^n(W)$ is iid with $p(x)$.
- Sender and receiver both have access to codebook $C = \{x^n(w)\}_{w=1}^{M}$ as well as $p(y|x), p(y^n|x^n)$. They are therefore both able to construct and search $A_\epsilon^{(n)}$
- **Errors** occur when $\hat{W} \neq W$ and when $\exists W' = \hat{W}$ that is also yields jointly typical $x^n(W'), y^n$.
- Let $x^n(1)$ stand in for all $x^n(w): \ w \in [M]$ as above.
- Let $\mathcal{E}$ be the "error" event $\{\hat{W}(y^n) \neq W\}$ OR the decoding procedure produces an error. Then

$$\Pr(\mathcal{E}) = \Pr(\mathcal{E}|W = 1) \tag{6.21}$$

- Let E_i be the event of decoding $\hat{W} = i$ when $W = 1$. That is,

$$E_i = \{(X^n(i), Y^n) \in A_\epsilon^{(n)}\} \tag{6.22}$$

- $\mathcal{E}$ can be written as a union of individual outcomes decoding E_i as follows:

$$\Pr(\mathcal{E}|W = 1) = \Pr\left(\neg E_1 \cup E_2 \cup E_3 \cup \cdots \cup E_{2^{nr}}|W = 1\right) \tag{6.23}$$

$$\leq \Pr(\neg E_1|W = 1) + \sum_{i=2}^{2^{nR}} P(E_i|W = 1) \tag{6.24}$$

- Now we get to use the AEP outcomes on each of those terms!
  - $Pr(\neg E_1|W_1) \to 0$ as $n \to \infty$ by joint AEP. All sequences (x^n, y^n) sampled iid from $p(x, y) = p(y|x)p(x)$ will end up in $A_\epsilon^{(n)}$ if n is increased enough.

  - We use the **independence statement** in the joint AEP to bound the rest of the terms $\Pr(E_{i\neq 1}|W_1)$. For $i \neq 1$, $x^n(i)$ is independent from $x^n(1)$ since the code generation process uses iid sampling of $p(x)$ to generate each code. Therefore, we can view the pair of "mismatch" sequences $x^n(i \neq 1), y^n$ as **independently sampled** from $p(x), p(y)$ respectively.
  - Joint AEP states that the probability of two **independently sampled** $(\tilde{x}^n, \tilde{y}^n) \in A_\epsilon^{(n)}$ is

$$\Pr\{(\tilde{X}^n, \tilde{Y}^n) \in A_\epsilon^{(n)}\} \leq 2^{-n(I(X;Y)-3\epsilon)} \tag{6.25}$$

- **Home stretch!**

$$\begin{aligned}\Pr(\mathcal{E}) &= \Pr(\mathcal{E}|W=1) &(6.26)\\ &\leq \Pr(\neg E_1|W=1) + \sum_{i=2}^{2^{nR}} P(E_i|W=1) &(6.27)\\ &\leq \epsilon + (2^{nR}-1)(2^{-n(I(X;Y)-3\epsilon)}) &(6.28)\\ &\leq \epsilon + (2^{3n\epsilon})(2^{-n(I(X;Y)-R)}) &(6.29)\end{aligned}$$

  And so, as long as the second exponential argument $I(X;Y) - R > 0$, the whole thing will vanish to zero as $n \to \infty$. Thus we have proven that all rates $R < I(X;Y)$ are **achievable** (i.e., vanishingly small error probability as block length is increased)! □

That's the meat of the proof! All that's left is some fairly elementary strengthening of a few of the statements.

Strengthening the Channel Coding Theorem:

- Use the capacity-maximizing $p^*(x)$ so that the condition $R < I(X;Y) \to R < C$.
- Search for the **optimal codebook** to get average error probability less than 2ϵ.

$$C^* = \arg\min_C \Pr(\mathcal{E}|C) \tag{6.30}$$

- **Markov Inequality Trick:** Let's throw away the worst half of C^*. This gives us a bound on the **maximum value** of $\lambda_i(C^*)$ rather than just guarantees on average/expected value. Half of the λ_i values **must** have values all less than 4ϵ if the average overall is 2ϵ. This results in 2^{nR-1} remaining codewords with new rate $R' = R - \frac{1}{n}$, but we get in return $\max \lambda_i = 4\epsilon$!

That's pretty much it!

6.4 Converse to Channel Coding Theorem

Todo

6.5 Hamming Codes

Todo

6.6 Feedback Codes

Todo

Chapter 7

Continuous Stuff

If you've been outside or touched grass at any point in your life, you may have noticed that many things are actually continuous and not discrete. Unfortunately for you, we've spend most of the course looking at discrete distributions and codes, so now we need to make great haste to cover our ass and make sure this stuff works for continuous random variables.

7.1 Differential Entropy

Differential entropy is defined for **continuous random variables** with CDF $F(x)$ and PDF $f(x)$. A term that often comes up is the "support set" of $f(x)$ – it's just $S = \{x \in \mathcal{X} : \; f(x) > 0\}$. Because god forbid you just say "regions where $f(x) > 0$...

It's also good to keep in mind that the existence of integrals and PDF's can't be taken for granted. Everything is qualified with "if it exists".

Differential Entropy Definition:

$$h(X) = -\int_S f(x) \log f(x) dx \tag{7.1}$$

$$= -\mathbb{E}_{f(x)}[\log f(x)] \tag{7.2}$$

Concerningly, you can have negative differential entropies. You can interpret $n + h(X)$ as the **number of bits** required to get n-bit accuracy on X. The reason you can have negative $h(X)$ is that sometimes you can get n-bit accuracy with less than n bits – e.g., if you know the RV X will **always** have 3 leading zeros, you get 3 bits of accuracy for "free".

Theorem 30. *Chain Rule for Differential Entropy*

$$h(X_1, \dots, X_n) = \sum_{i=1}^{n} h(X_i|X_{1:i-1} \tag{7.3}$$

Proof sketch: *Same as for discrete case.*

Theorem 31. *Hadamard's Inequality Let* $\mathbf{X} \sim \mathcal{N}(0, \mathbf{K})$*. Then*

$$|\mathbf{K}| \leq \prod_{i=1}^{n} \mathbf{K}_{ii} \tag{7.4}$$

Proof sketch: *Based on the chain rule for differential entropy plus the fact that conditioning always reduces entropy:*

$$h(X_{1:n}) = \sum_{i=1}^{n} h(X_i|X_{1:i-1}) \leq \sum_{i=1}^{n} h(X_i) \tag{7.5}$$

And we know that the diagonals contain the variances of each independnet X_i*.*

Theorem 32. *Translation Doesn't Affect Differential Entropy*

$$h(X + c) \equiv h(X) \tag{7.6}$$

Theorem 33. *Scaling Affects Differential Entropy*

$$h(aX) = h(X) + \log a \tag{7.7}$$

Proof sketch: *Let* $Y = aX$*. Then* Y*'s PDF is*

$$f_Y(Y) = \frac{1}{|a|} f_X(\frac{y}{a}) \tag{7.8}$$

Now solving for $h(aX) = h(Y)$ *via the integral definition of differential entropy:*

$$\begin{aligned} h(aX) &= -\int f_Y(y) \log f_Y(y) dy && (7.9) \\ &= -\int \frac{1}{|a|} f_X(y/a) \log \left(\frac{1}{|a|} f_X(\frac{y}{a}) \right) && (7.10) \\ &= -\int f_X(x) \log f_x(x) dx + \log |a| && (7.11) \\ &= h(X) + \log |a| && (7.12) \end{aligned}$$

Theorem 34. *Differential Entropy of Linear Transformation on Multivariable*

$$h(\mathbf{AX}) = h(\mathbf{X}) + \log|\det(\mathbf{A})| \tag{7.13}$$

Proof sketch: *This is a corollary of "Scaling Affects Differential Entropy".*

Theorem 35. *Multivariate Normal has Maximum Entropy over All Distributions with the same Covariance Matrix*

Let $\mathbf{X} \in \mathbb{R}^n$ be a random variable with $\mu = 0$, $\mathbf{K} = \mathbb{E}\left[\mathbf{X}\mathbf{X}^\top\right]$. Then

$$h(\mathbf{X}) \leq \frac{1}{2}\log(2\pi e)^n |\mathbf{K}| \tag{7.14}$$

With equality ***iff*** $\mathbf{X} \sim \mathcal{N}(0, \mathbf{K})$.

Proof sketch: *You start with g as a density with the same covariance as $\phi \sim \mathcal{N}(0, \mathbf{K})$. You then expand the expression $0 \leq D(g\|\phi)$ to recover $h(g), h(\phi)$ – this os only possible because "g, ϕ give the same moments in $\log\phi(\mathbf{x})$", which isn't really elaborated on in the textbook's proof. Regardless, you get equivalence between $D(g\|\phi) = h(\phi) - h(g)$, and since $D \geq 0$, you prove that $h(\phi)$ is the upper bound on $h(g)$ with the same covariance.*

Theorem 36. *Estimator Error Bound from Differential Entropy For any random variable X and estimator $\hat{X}$,*

$$\mathbb{E}\left[X - \hat{X}\right]^2 \geq \frac{1}{2\pi e}e^{2h(X)} \tag{7.15}$$

With equality ***iff*** *X is gaussian and $\hat{X} = \mu_X$.*

Proof sketch: *Let's solve for the minimum expected squared error.*

- *$\hat{X} = \mathbb{E}[X]$ is the best estimator.*
- *$\mathbb{E}[X - \mathbb{E}[X]]^2 \equiv var(X)$.*
- *Leveraging "Gaussian maximizes entropy for constant variance", we can then bound $var(X)$ as*

$$var(X) \geq \frac{1}{2\pi e}e^{2h(X)} \tag{7.16}$$

□

7.1.1 Discrete → Differential Entropy

We can connect discrete entropy to differential entropy much like how we make most continuity arguments: we quantize differential entropy into progressively finer discrete Riemann-like approximations of the PDF $f(x)$, then we congratulate ourselves when it converges to our definition of differential entropy.

Quantization Definition: For any continuous random variable X, we define the quantization X^Δ with "bin length" Δ as

$$\begin{aligned} X^\Delta &= x_i \text{ if } i\Delta \leq X \leq (i+1)\Delta && (7.17) \\ \iff \Pr(X^\Delta &= x_i) = f(x_i)\Delta && (7.18) \end{aligned}$$

Convergence of Quantized Entropy → Differential Entropy: Assuming that $f(x)\log f(x)$ is **Riemann integrable**, we have a pretty straightforward convergence of $H(X^\Delta) + \log\Delta \to h(X)$ as $\Delta \to 0$.

$$H(X^\Delta) + \log\Delta \to h(X) \text{ as } \Delta \to 0 \tag{7.19}$$

In other words, the entropy of an n-bit quantization of continuous variable X (i.e., $-\log\Delta$-bit) has discrete entropy approximately equal to $h(X) + n$. The number of partitions is roughly $1/\Delta$ which would require $\log(\frac{1}{\Delta})$ bits.

7.2 Joint and Conditional Differential Entropy

Conditional Differential Entropy: Let $X, Y \sim f(x,y)$ be continuous random variables. Then

$$\begin{aligned} h(X|Y) &= -\int_{x,y} f(x,y)\log\big(f(x|y)\big)dxdy && (7.20) \\ &= h(X,Y) - h(Y) && (7.21) \end{aligned}$$

Note: some of these values may be infinite!

Joint Differential Entropy:

$$h(X,Y) = -\int_{x,y} f(x,y)\log\big(f(x,y)\big) \tag{7.22}$$

Differential Relative Entropy (KL Divergence): $D(f\|g)$ is given by

$$D(f\|g) = \int f \log \frac{f}{g} \tag{7.23}$$

This is finite iff support$(f) \subseteq$ support(g).

Non-negativity of Relative Entropy: To prove that $D(f\|g)$ is non-negative, start with the integral definition. Use Jensen's inequality to move the log out of the integral. Cancel the f's and come to the conclusion that $-D(f\|g) \leq \log 1 = 0$.

- Implies that $I(X;Y) \geq 0$ with equality for independent X, Y.
- Implies that $h(X|Y) \leq h(X)$ with equality for independent X, Y.

Differential Mutual Information:

$$\begin{aligned} I(X;Y) &= \int f(x,y) \log \frac{f(x,y)}{f(x)f(y)} dxdy && (7.24)\\ &= h(X) - h(X|Y) && (7.25)\\ &= h(Y) - h(Y|X) && (7.26)\\ &= h(X) + h(Y) - h(X,Y) && (7.27)\\ &= D\left(f(x,y) \| f(x)f(y)\right) && (7.28) \end{aligned}$$

Master Definition of Differential Mutual Information: $I(X^\Delta; Y^\Delta) \rightarrow I(X;Y)$ as $\Delta \rightarrow 0$. More generally,

$$I(X;Y) = \sup_{\mathcal{P},\mathcal{Q}} I([X]_{\mathcal{P}}; [Y]_{\mathcal{Q}}) \tag{7.29}$$

Where $\mathcal{P}, \mathcal{Q}$ are **finite partitions** of $\mathcal{X}, \mathcal{Y}$. This means that $\cup_i P_i = \mathcal{X}$ and $\cup_i Q_i = \mathcal{Y}$. This is a pretty robust definition since it works even if the PDF's aren't well defined. As above with $\Delta \rightarrow 0$, "continual refinements" of $\mathcal{P}, \mathcal{Q}$ make a monotonically increasing $I([X]_{\mathcal{P}}; [Y]_{\mathcal{Q}}) \rightarrow I(X;Y)$

Properties for MI, relative entropy are essentially the same as in the discrete case.

7.3 Examples Differential Entropy Calculations

Uniform Distribution Entropy: Let

$$f(x) = \begin{cases} \frac{1}{a} & \text{if } x \in [0, a] \\ 0 & \text{else} \end{cases} \tag{7.30}$$

Then $h(X) = \log(a)$ (just solve the integral).

Normal Distribution Entropy: I strongly dislike solving integrals by hand, so I'm just going to put the result here.

$$X \sim \phi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp[\frac{-x^2}{2\sigma^2}] \tag{7.31}$$

$$h(X) = \frac{1}{2} \log(2\pi e \sigma^2) \tag{7.32}$$

Mutual Informatioon between Correlated Gaussians: Let ρ be the correlation coefficient and $(X, Y) \sim \mathcal{N}(0, \mathbf{K})$ where

$$\mathbf{K} = \begin{bmatrix} \sigma^2 & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 \end{bmatrix} \tag{7.33}$$

Then the following are true:

- $h(X) = h(Y) = \frac{1}{2} \log(2\pi e)\sigma^2$.
- $h(X, Y) = \frac{1}{2} \log(2\pi e)^2 |\mathbf{K}|$.

$$\therefore I(X; Y) = \underbrace{\frac{-1}{2} \log(1 - \rho^2)}_{h(X)+h(Y)-h(X,Y)} \tag{7.34}$$

7.4 AEP for Continuous Variables

All this essentially follows from the laws of large numbers, just like in the discrete case.

Theorem 37. *AEP for Continuous Random Variables Let $X_1, \ldots, X_n$ be iid $\sim f(x)$. Then*

$$\frac{-1}{n} \log f(X_1, \ldots, X_n) \to \mathbb{E}[-\log f(X)] = h(X) \tag{7.35}$$

in probability as $n \to \infty$.

Proof sketch: *Weak law of large numbers. The sample entropy converges to the underlying entropy because $h(X)$ is the expected value of the LHS.*

Just like in the discrete case, typical sets are all sequences x^n with sample entropies ϵ-close to the underlying entropy.

Theorem 38. *Definition: Typical Sets for Continuous Random Variables For any $\epsilon > 0$ and any n, the typical set $A_\epsilon^{(n)}$ wrt $f(x)$ is*

$$A_\epsilon^{(n)} = \left\{x^n \in S^n : \left|\tfrac{-1}{n} \log f(x_1, \ldots, x_n) - h(X)\right| \leq \epsilon\right\} \tag{7.36}$$

Where S is the support set of f, $f(x_1, \ldots, x^n) = \prod_{i=1}^n f(x_i)$.

Most of the properties of the typical set are the same as in the discrete case. The main difference is that cardinality $\left|A_\epsilon^{(n)}\right|$ is replaced by volume $\mathrm{Vol}\left(A_\epsilon^{(n)}\right)$.

Volume Definition: For a set $A \subset \mathbb{R}^n$, the volume is defined as

$$\mathrm{Vol}(A) = \int_A dx_1 dx_2 \ldots dx^n \tag{7.37}$$

Theorem 39. *Properties: Typical Sets for Continuous Random Variables Let $A_\epsilon^{(n)}$ be the typical set of n-long sequences iid $\sim f(x)$. Then*

$$\Pr(A_\epsilon^{(n)}) > 1 - \epsilon \textit{ for } n \textit{ sufficiently large} \tag{7.38}$$

$$\textit{Vol}(A_\epsilon^{(n)}) \leq 2^{n(h(X)+\epsilon)} \textit{ for all } n \tag{7.39}$$

$$\textit{Vol}(A_\epsilon^{(n)}) \geq (1-\epsilon) 2^{n(h(X)-\epsilon)} \textit{ for } n \textit{ sufficiently large} \tag{7.40}$$

Proof sketch:

1. $\Pr(A_\epsilon^{(n)}) > 1 - \epsilon$ *for large n because all sequences have probability that converges to $2^{nh(X)}$ for n sufficiently large. Therefore one can always pick some n given a desired ϵ value such that the condition is satisfied.*

2. *$Vol(A_\epsilon^{(n)}) \leq 2^{n(h(X)+\epsilon)}$ because the probability of $A_\epsilon^{(n)}$ cannot exceed 1. Since the minimum probability if an element in the set is $2^{-n(h(X)+\epsilon)}$, we can work this out just like in the discrete case.*

3. *$Vol(A_\epsilon^{(n)}) \geq (1-\epsilon)2^{n(h(X)-\epsilon)}$ because the probability of $A_\epsilon^{(n)} > 1-\epsilon$ according to property 1. Apply the same reasoning as before to derive this limit (using the max probability of a given sequence the set).*

Chapter 8

Annoying Mechanical Procedures

Have you ever dreamed of being an automaton? Do you wish that electronic computers had never been invented so that you could be a professional human computer? If you answered yes to either of these questions, this chapter is for you!

8.1 Sardinas-Patterson Test for Unique Decodability