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 $[X, Y]^\top$.

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

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

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

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*

$$H(X, Y) = H(X) + H(Y|X) \tag{2.9}$$
$$= H(Y) + H(X|Y) \tag{2.10}$$

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.

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

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

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. $x^n \in A_\epsilon^{(n)} \to$ *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 $p(x)/p(x)$ $\square$

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*

$$= (\sum_{x \in \mathcal{X}} D^{-\ell(x)})^k \tag{4.18}$$

$$\sum_{x^k \in \S^k} D^{-\ell(x^k)} = \sum_{m=1}^{k\ell_{max}} a(m) D^{-m} \tag{4.19}$$

$$\leq \sum_{m=1}^{k\ell_{max}} D^m D^{-m} \tag{4.20}$$

$$= \ell_{max} k \tag{4.21}$$

- *So now we know that*

$$(\sum_{x \in \mathcal{X}} D^{-\ell(x)})^k \leq \ell_{max} k \tag{4.22}$$

$$\implies \sum_{x \in \mathcal{X}} D^{\ell(x)} \leq (\ell_{max} k)^{1/k} \tag{4.23}$$

$$\implies \sum_{x \in \mathcal{X}} D^{-\ell(x)} \leq 1 \tag{4.24}$$

  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* $\square$.

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* $\mathbb{E}[\ell(x)]$*!* $\square$

- *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)* $\square$

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* $\mathbf{p}' \to \mathbf{p}$ *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 Sardinas-Patterson Test for Unique Decodability