Notes for EE 229A: Information and Coding Theory
UC Berkeley Fall 2020

Aditya Sengupta

January 17, 2021

Contents

Lecture 1: Introduction 4
1.1 Entropy 5
Lecture 2: Entropy and mutual information, relative entropy 7
2.1 Entropy 7
2.2 Joint Entropy 8
2.3 Conditional entropy 8
2.4 Mutual information 9
2.5 Jensen's inequality 9
Lecture 3: Mutual information and relative entropy 12
3.1 Mutual information 12
3.2 Chain rule for mutual information (M.I.) 12
3.3 Relative entropy 13
Lecture 4: Mutual information and binary channels 15
4.1 Binary erasure channel example 15
4.2 The uniform distribution maximizes entropy 16
4.3 Properties from C-T 2.7 16
Lecture 5: Data processing inequality, asymptotic equipartition 18
5.1 Data Processing Inequality 18
5.2 AEP: Entropy and Data Compression 19
5.3 Data Compression 21
Lecture 6: AEP, coding 22

6.1 Markov sequences . . . 22
6.2 Codes . . . 23
6.3 Uniquely decodable and prefix codes . . . 24
Lecture 7: Designing Optimal Codes, Kraft's Inequality . . . 25
7.1 Prefix codes . . . 25
7.2 Optimal Compression . . . 27
7.3 Kraft's Inequality . . . 27
7.4 Coding Schemes and Relative Entropy . . . 29
7.5 The price for assuming the wrong distribution . . . 32
Lecture 8: Prefix code generality, Huffman codes . . . 33
8.1 Huffman codes . . . 34
8.2 Limitations of Huffman codes . . . 35
Lecture 9: Arithmetic coding . . . 36
9.1 Overview and Setup, Infinite Precision . . . 36
9.2 Finite Precision . . . 37
9.3 Invertibility . . . 37
Lecture 10: Arithmetic coding wrapup, channel coding . . . 38
10.1 Communication over noisy channels . . . 38
Lecture 11: Channel coding efficiency . . . 40
11.1 Noisy typewriter channel . . . 40
Lecture 12: Rate Achievability . . . 42
Lecture 13: Joint Typicality . . . 44
13.1 BSC Joint Typicality Analysis . . . 44
Lecture 14: Fano's Inequality for Channels . . . 47
Lecture 15: Polar Codes, Fano's Inequality . . . 48
15.1 Recap of Rate Achievability . . . 48
15.2 Source Channel Theorem . . . 48
15.3 Polar Codes . . . 49
Lecture 16: Polar Codes . . . 51
Lecture 17: Information Measures for Continuous RVs . . . 53
17.1 Differential Entropy . . . 54
17.2 Differential entropy of popular distributions . . . 55

17.3 Properties of differential entropy 57
Lecture 18: Distributed Source Coding, Continuous RVs 58
18.1 Distributed Source Coding 58
18.2 Differential Entropy Properties 59
18.3 Entropy Maximization 60
18.4 Gaussian Channel Capacity Formulation 62
Lecture 20: Maximum Entropy Principle, Supervised Learning 64
20.1 The principle of maximum entropy 64
20.2 Supervised Learning 65

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 1: Introduction

Lecturer: Kannan Ramchandran *27 August* *Aditya Sengupta*

Note: *LaTeX format adapted from template for lecture notes from CS 267, Applications of Parallel Computing, UC Berkeley EECS department.*

Information theory answers two fundamental questions.

1. What are the fundamental limits of data compression? The answer is the entropy of the source distribution.
2. What are the fundamental limits of reliable communication? The answer is the channel capacity.

Information theory has its roots in communications, but now has influence in statistical physics, theoretical computer science, statistical inference and theoretical statistics, portfolio theory, and measure theory.

Historically, 18th and 19th century communication systems were not seen as a unified field of study/engineering. Claude Shannon saw that they were all connected, and said: Every communication system has the form $f_1(t) \to [T] \to F(t) \to [R] \to f_2(t)$. He made some assumptions that don't hold today; for one thing, he assumed a noiseless channel, and analog signals.

There's a movie about Shannon! The Bit Player.

Shannon called his work "A Mathematical Theory of Communication" - others then called it Shannon's theory of communication. The key insight: no matter what you're using to send and receive messages, you can use the common currency of bits to encode messages.

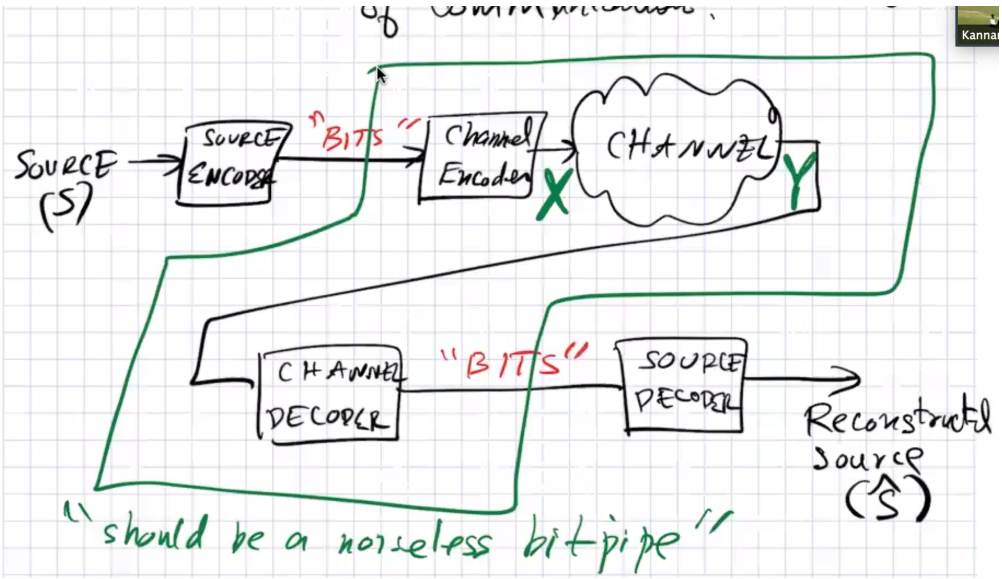


Figure 1.1: Shannon's channel diagram

Shannon introduced three new concepts:

1. The entropy, as above
2. not sure
3. not sure

The separation theorem states that source and channel coders do their jobs optimally and separately for optimal end-to-end performance.

After 70+ years, all communication systems are built on the principles of information theory, and it provides theoretical benchmarks for engineering systems.

1.1 Entropy

In information theory, entropy is a measure of the information content contained in any message or flow of information.

Example 1.1. Take a source $S = \{A, B, C, D\}$ with $\mathbb{P}(x) = \frac{1}{4}$ for each $x \in S$. A uniform source might produce a sequence like $ADCABCBA\ldots$, assuming these are iid. How many binary questions would you expect to ask to figure out what each symbol is?

We expect to ask 2 questions, and because we've done 126 we know this is because

$$H(x) = -\sum_X \mathbb{P}(X) \log_2 \mathbb{P}(X) = -4 \cdot \frac{1}{4} \log_2 \frac{1}{4} = -1 \cdot (-2) = 2 \tag{1.1}$$

and more concretely, you can ask the Huffman tree of questions:

- Is $X \in \{AB\}$?
- If yes, is $X = A$?
  - If yes, $X = A$.
  - If no, $X = B$.
- If no, is $X = C$?
  - If yes, $X = C$.
  - If no, $X = D$.

□

Example 1.2. Consider the same setup as above, but now the pmf is

$$p_A = \frac{1}{2}, p_B = \frac{1}{4}, p_C = \frac{1}{8}, p_D = \frac{1}{8}. \tag{1.2}$$

The same question? Now, there's fewer questions in expectation. We could set up the Huffman tree to algorithmically get this answer. Intuitively, you know you should ask if $X = A$ first, so you can rule out half the sample space. Next, if it's not, you should ask if $X = B$, because B now has half the mass in the marginalized sample space. Finally, if neither of them are true, you should ask if it's C or D.

The expected number of questions in this case is

$$\mathbb{E}[\#Q] = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{4} \cdot 3 = 1.75 \tag{1.3}$$

□

The more probable an outcome is, the less information it's providing. A message is the most informative to you when it's rare. As we saw in the example above, we can quantify this in the entropy:

$$H(S) = \sum_{i \in S} p_i \left[\log_2 \frac{1}{p_i} \right] \tag{1.4}$$

Shannon called this the "self-information" of i. For a random variable, we say

$$H(X) = \mathbb{E}[\log \frac{1}{p_X(x)}] = \sum_x \log_2 \frac{1}{p_X(x)} \tag{1.5}$$

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 2: Entropy and mutual information, relative entropy

Lecturer: Kannan Ramchandran *1 September* *Aditya Sengupta*

2.1 Entropy

Previously, we saw that the entropy of a discrete RV X was given by

$$H(X) = \mathbb{E}[\log \frac{1}{p_X(x)}] = \sum_x \log_2 \frac{1}{p_X(x)} \tag{2.1}$$

We say that the entropy is label-invariant, as it depends only on the distribution and not on the specific values that the variable could take on. This contrasts properties like expectation and variance, which do depend on the values the variable could take on.

Example 2.1. Consider a coin flip, where $X \sim Bern(p) = \begin{cases} 0 & \text{w.p. } 1-p \\ 1 & \text{w.p. } p \end{cases}$. The entropy is

$$H(X) = p \log \frac{1}{p} + (1-p) \log \frac{1}{1-p}. \tag{2.2}$$

This is called the *binary entropy function*, $H(p)$. As a function of p, it appears to be about parabolic, as we see in Figure 2.2.

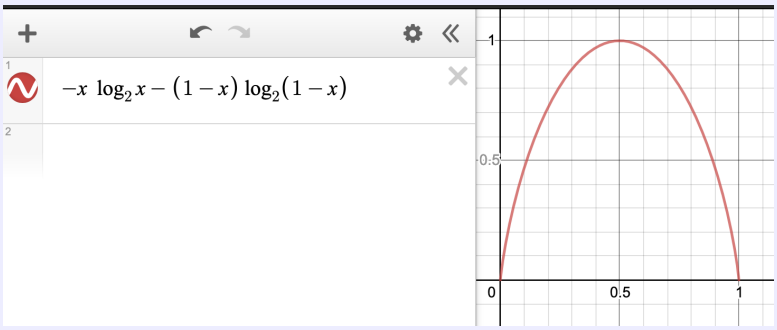


Figure 2.2: The binary entropy function, $H(p)$

Note that if $p = \frac{1}{2}$, we get the maximum amount of information, which makes intuitive sense, as we previously had no reason to favour one outcome over the other. If p is very close to 0 or 1, however, a result of heads/tails respectively is not

very surprising, and so its entropy is less.

□

2.2 Joint Entropy

If we have two RVs, say X_1 and X_2, then their joint entropy follows directly from their joint distribution:

$$H(X_1, X_2) = \mathbb{E}\left[\log_2 \frac{1}{p(X_1, X_2)}\right]. \tag{2.3}$$

If X_1 and X_2 are independent, then $p(X_1, X_2) = p(X_1)p(X_2)$; the distribution splits, and therefore so does the entropy:

$$H(X_1, X_2) = \mathbb{E}\left[\log_2 \frac{1}{p(X_1, X_2)}\right] = \mathbb{E}\left[\log_2 \frac{1}{p(X_1)}\right] + \mathbb{E}\left[\log_2 \frac{1}{p(X_2)}\right] \tag{2.4}$$

Therefore, $X \perp\!\!\!\perp Y \implies H(X, Y) = H(X) + H(Y)$.

We see that the log in the definition of entropy ensures that entropy is additive: the entropy of independent RVs is the sum of the individual entropies.

2.3 Conditional entropy

The more general law based on conditioning is

$$H(X, Y) = H(X) + H(Y|X), \tag{2.5}$$

where

$$H(Y|X) = \mathbb{E}\left[\log \frac{1}{p(y|x)}\right] = \sum_x \sum_y p(x, y) \log \frac{1}{p(y|x)}. \tag{2.6}$$

$H(Y|X)$ is referred to as the *conditional entropy of Y given X*.

We can extend this to more than two variables, if we just look at the definition above with two variables at a time:

$$H(X, Y, Z) = H(X) + H(Y, Z|X) = H(X) + H(Y|X) + H(Z|Y, X). \tag{2.7}$$

This is the *chain rule of entropy*.

Let's break down the expression $H(Y|X)$ a bit more.

$$\begin{aligned} H(Y|X) &= \sum_x \sum_y p(x,y) \log \frac{1}{p(x,y)} \\ &= \sum_x p(x) \sum_y p(y|x) \log \frac{1}{p(y|x)} \\ &= \sum_x p(x) H(Y \mid X = x). \end{aligned} \tag{2.8}$$

And the extension to the three-variable setting:

$$\begin{aligned} H(Y,Z|X) &= \sum_x p(x) H(Y,Z|X=x) \\ &= \sum_x p(x) H(Y|X=x) + \sum_x p(x) H(Z|Y,X=x) \\ &= H(Y|X) + H(Z|Y,X). \end{aligned} \tag{2.9}$$

This is just like $H(Y,Z) = H(Y) + H(Z|Y)$, but conditioning everything on X.

2.4 Mutual information

The mutual information of two RVs, $I(X;Y)$, is given by

$$I(X;Y) = H(X) - H(X|Y). \tag{2.10}$$

We interpret this as "how much information does Y convey about X?"

The mutual information can be shown to be symmetric, i.e. X gives as much information about Y as Y gives about X.

$$I(X;Y) = I(Y;X) \tag{2.11}$$

$$H(X) - H(X|Y) = H(Y) - H(Y|X). \tag{2.12}$$

2.5 Jensen's inequality

Definition 2.1. *A real-valued function f is convex on an interval $[a,b]$ if for any $x_1, x_2 \in [a,b]$ and any λ such that $0 < \lambda < 1$,*

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

A couple useful properties:

- If f is doubly differentiable, then $f''(x) \geq 0 \iff f$ is convex.
- If $-f$ is convex then f is concave.

Theorem 2.1 (Jensen's Inequality for Probabilities)**.** *For any random variable X and any convex function f,*

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)].$$

Proof. Let $t(x)$ be the tangent line of $f(x)$ at some point c. We can say

$$f(x) \geq f(c) + f'(c)(x - c), \tag{2.13}$$

i.e. $f(x)$ is above the tangent line.

Take $c = \mathbb{E}[X]$; then we have

$$f(x) \geq f(\mathbb{E}[X]) + f'(\mathbb{E}[X])(X - \mathbb{E}[X]) \tag{2.14}$$
$$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X]) + f'(\mathbb{E}[X]) \cdot 0 \tag{2.15}$$
$$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X]). \tag{2.16}$$

□

A quick example of this is: $f(x) = x^2$. Jensen's tells us that $\mathbb{E}[X^2] \geq \mathbb{E}[X]^2$, i.e. $\mathbb{E}[X^2] - \mathbb{E}[X]^2 \geq 0$. This makes sense if we think of this expression as $\text{var}(X)$.

If we consider $f(x) = -\log x$, we get

$$\log \mathbb{E}[X] \geq \mathbb{E}[\log X]. \tag{2.17}$$

From this, we can get a couple of useful properties of entropy: by definition, we know that $H(X) \geq 0$, and we can show using Equation 2.17 that entropy is upper-bounded by the size of the alphabet:

$$H(X) = \mathbb{E}[\log p(x)] \leq \log \mathbb{E}[p(x)] = \log |\mathcal{X}| \tag{2.18}$$

More properties of mutual information:

1. $I(X;Y) = I(Y;X)$
2. $I(X;Y) \geq 0 \ \forall X, Y$
3. $I(X;Y) = 0 \iff X \perp\!\!\!\perp Y$.

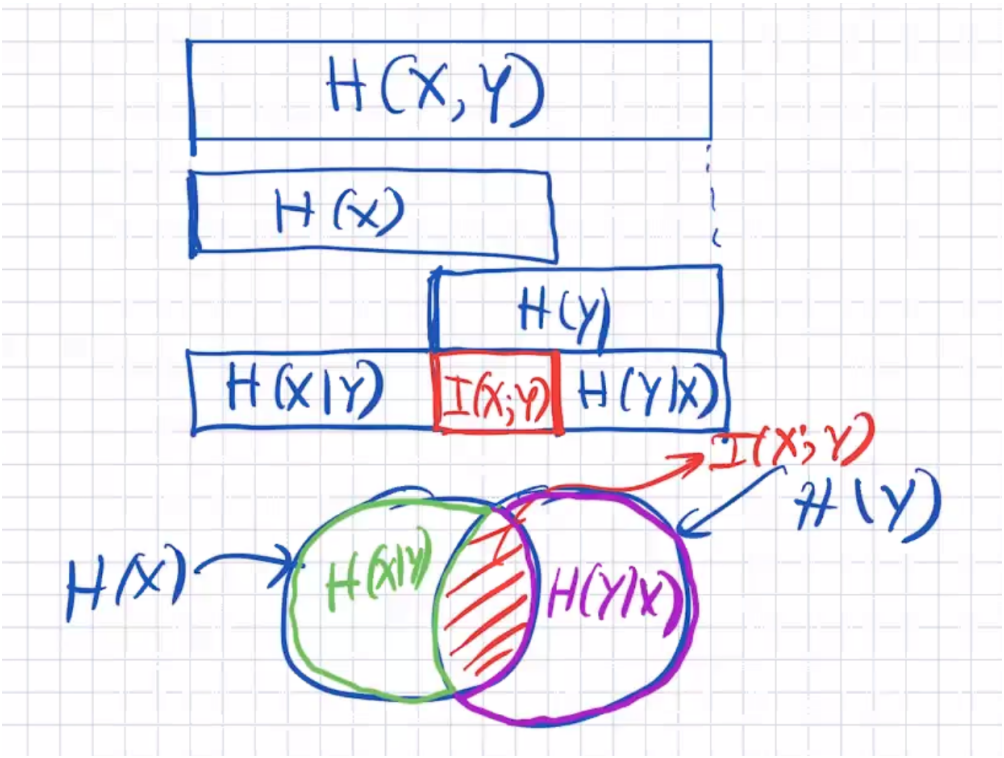


Figure 2.3: Mutual information diagrams

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 3: Mutual information and relative entropy

Lecturer: Kannan Ramchandran *3 September* *Aditya Sengupta*

3.1 Mutual information

We'll start with a closed form for mutual information in terms of a probability distribution.

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) && (3.1)\\ &= \sum_x p(x) \log \frac{1}{p(x)} - \sum_x \sum_y p(x,y) \log \frac{1}{p(x|y)} && (3.2)\\ &= \sum_x \sum_y p(x,y) \log \frac{1}{p(x)} - \sum_x \sum_y p(x,y) \log \frac{1}{p(x|y)} && (3.3)\\ &= \sum_x \sum_y p(x,y) \log \frac{p(x|y)}{p(x)}. && (3.4) \end{aligned}$$

We can use this to show that mutual information is symmetric:

$$\begin{aligned} \frac{p(x|y)}{p(x)} &= \frac{p(x,y)}{p(x)p(y)} = \frac{p(y|x)}{p(y)} && (3.5)\\ I(X;Y) &= \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} && (3.6)\\ &= \sum_x \sum_y p(x,y) \log \frac{1}{p(x)p(y)} - \sum_x \sum_y p(x,y) \log \frac{1}{p(x,y)} && (3.7)\\ &= H(X) + H(Y) - H(X,Y). && (3.8) \end{aligned}$$

Mutual information must be nonnegative, and $I(X;Y) = 0 \iff X \perp\!\!\!\perp Y$. From this, we get that $H(X,Y) = H(X) + H(Y)$ if and only if $X \perp\!\!\!\perp Y$.

3.2 Chain rule for mutual information (M.I.)

Suppose we have three RVs, X, Y_1, Y_2. Consider $I(X;Y_1,Y_2)$, which we can interpret as the amount of information that (Y_1,Y_2) give us about X. We can split this up similarly to how we would for entropy:

$$I(X;Y_1,Y_2) = I(X;Y_1) + I(X;Y_2|Y_1) \tag{3.9}$$

3.3 Relative entropy

Relative entropy is also known as Kullback-Leibler (K-L) divergence.

Definition 3.1. *The relative entropy between two distributions p, q is*

$$D(p \parallel q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}, \tag{3.10}$$

or alternatively,

$$D(p \parallel q) = \mathbb{E}_{x \sim p} \left[\log \frac{p(x)}{q(x)} \right]. \tag{3.11}$$

Relative entropy has the following properties:

1. In general, $D(p \parallel q) \neq D(q \parallel p)$:(
2. $D(p \parallel p) = 0$
3. $D(p \parallel q) \geq 0$ for all distributions p, q, with equality iff $p = q$.

Example 3.1. Let $X_1 \sim Bern(1/2)(p), X_2 \sim Bern(1/4)(q)$. We can verify that the relative entropy is not symmetric.

$$\begin{aligned} D(p \parallel q) &= \mathbb{E}_{x \sim p} \left[\log \frac{p(x)}{q(x)} \right] = \frac{1}{2} \log \frac{1/2}{3/4} + \frac{1}{2} \log \frac{1/2}{1/4} && (3.12) \\ &= \frac{1}{2} \log \frac{2}{3} + \frac{1}{2} \log 2 && (3.13) \\ &= 0.20752 && (3.14) \end{aligned}$$

$$\begin{aligned} D(q \parallel p) &= \mathbb{E}_{x \sim q} \left[\log \frac{q(x)}{p(x)} \right] = \frac{3}{4} \log \frac{3/4}{1/2} + \frac{1}{4} \log \frac{1/4}{1/2} && (3.15) \\ &= \frac{3}{4} \log \frac{3}{2} + \frac{1}{4} \log 2 && (3.16) \\ &= 0.68872. && (3.17) \end{aligned}$$

□

$I(X;Y)$ can be expressed in terms of the relative entropy between their joint distribution and the product of their marginals.

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{3.18}$$

$$= D(p(x,y) \parallel p(x) \cdot p(y)). \tag{3.19}$$

Theorem 3.1. *Let $p(x), q(x), x \in \mathcal{X}$ be two PMFs. Then $D(p \parallel q) \geq 0$, with equality iff $p(x) = q(x)\ \forall x \in \mathcal{X}$.*

Proof. Let $A = \{x \mid p(x) > 0\}$ be the support set of $p(x)$.

From the definition,

$$-D(p \parallel q) = -\sum_{x \in A} p(x) \log \frac{p(x)}{q(x)} \tag{3.20}$$

$$= \sum_{x \in A} p(x) \log \frac{q(x)}{p(x)} = \mathbb{E}_{x \sim p}\left[\log \frac{q(x)}{p(x)}\right] \tag{3.21}$$

Using Jensen's inequality,

$$-D(p \parallel q) \leq \log \sum_{x \in A} p(x) \frac{q(x)}{p(x)} = \log \sum_{x \in A} q(x). \tag{3.22}$$

Since q and p may not have exactly the same support, $\sum_{x \in A} q(x) \leq 1$ and so

$$-D(p \parallel q) \leq \log 1 = 0 \tag{3.23}$$

$$D(p \parallel q) \geq 0. \tag{3.24}$$

□

Remark 3.2. *Since $\log t$ is concave in t, we have equality in 3.22 iff $\frac{q(x)}{p(x)} = c$ everywhere, i.e. $q(x) = cp(x)\ \forall x \in A$. Due to normalization, this can only occur when $c = 1$, i.e. they are identical inside the support. Further, we have equality in 3.23 iff the support of both PMFs is the same. Putting those together, the distributions must be exactly the same. Therefore, $D(p \parallel q) = 0 \iff p(x) = q(x)\ \forall x \in \mathcal{X}$.*

Corollary 3.3.

1. *$I(X;Y) = D(p(x,y) \parallel p(x) \cdot p(y)) \geq 0$*
2. *$H(X|Y) \leq H(X)$. Conditioning reduces entropy.*

We can interpret this as saying that on average, the uncertainty of X after we observe Y is no more than the uncertainty of X unconditionally. "More knowledge cannot hurt".

Among all possible distributions over a finite alphabet, the uniform distribution achieves the maximum entropy.

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 4: Mutual information and binary channels

Lecturer: Kannan Ramchandran *8 September* *Aditya Sengupta*

4.1 Binary erasure channel example

Example 4.1. Consider the binary erasure channel where $X \in \{0,1\}$ and $Y \in \{0,1,*\}$, Let $Y = X$ with probability $\frac{1}{2}$ and $Y = *$ with probability $\frac{1}{2}$, individually for either case of X. (todo if I care, BEC diagram). Suppose $X \sim Bern(1/2)$, so that

$$Y = \begin{cases} 0 & \text{w.p.} \frac{1}{4} \\ * & \text{w.p.} \frac{1}{2} \\ 1 & \text{w.p.} \frac{1}{4} \end{cases} \tag{4.1}$$

$$H(X) = H_2\left(\frac{1}{2}, \frac{1}{2}\right) \tag{4.2}$$

$$H(Y) = H_3\left(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right) = 1.5 \tag{4.3}$$

We can find the conditional entropies, using the fact that X is symmetric:

$$H(Y|X) = H(Y|X=0) = H_2\left(\frac{1}{2}, \frac{1}{2}\right) = 1 \tag{4.4}$$

$$H(X|Y=y) = \begin{cases} 0 & y \in \{0,1\} \\ 1 & y = * \end{cases} \tag{4.5}$$

$$H(X|Y) = \mathbb{E}[H(X|Y=y)] = 0.5 \tag{4.6}$$

Therefore, the mutual information either way is

$$H(X) - H(X|Y) = 1 - 0.5 = 0.5 \tag{4.7}$$

$$H(Y) - H(Y|X) = 1.5 - 1 = 0.5 \tag{4.8}$$

□

4.2 The uniform distribution maximizes entropy

Theorem 4.1. *Let an RV X be defined on $\mathcal{X}$ with $|\mathcal{X}| = n$. Let U be the uniform distribution on $\mathcal{X}$. Then $H(X) \leq H(U)$.*

Proof.

$$H(U) - H(X) = \sum_{\mathcal{X}} \frac{1}{n} \log n + \sum_x p(x) \log p(x) \tag{4.9}$$

$$= \sum_x p(x)(\log n) + \sum_x p(x) \log p(x) \tag{4.10}$$

$$= \sum_x p(x) \log \left(\frac{p(x)}{1/n} \right) \tag{4.11}$$

$$= D(p \parallel U) \geq 0. \tag{4.12}$$

□

4.3 Properties from C-T 2.7

Theorem 4.2 (Log-Sum Inequality)**.** *For nonnegative numbers $(a_i)_{i=1}^n$ and $(b_i)_{i=1}^n$,*

$$\sum_{i=1}^n a_i \log \frac{a_i}{b_i} \geq \left(\sum_{i=1}^n a_i \right) \log \frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i}, \tag{4.13}$$

with equality if and only if $\frac{a_i}{b_i} = const.$

Proof. (Almost directly from C&T 2.7.1)

$f(t) = t \log t$ is convex on positive inputs, and therefore Jensen's inequality applies; let $\{\alpha_i\}$ be a partition of unity, then

$$\sum_i \alpha_i f(t_i) \geq f \left(\sum_i \alpha_i t_i \right), \tag{4.14}$$

for any $\{t_i\}$ such that all elements are positive. In particular, this works for $\alpha_i = \frac{b_i}{\sum_j b_j}$ and $t_i = \frac{a_i}{b_i}$; substituting this into Jensen's gives us

$$\sum \frac{a_i}{\sum b_j} \log \frac{a_i}{b_i} \geq \sum \frac{a_i}{\sum b_j} \log \sum \frac{a_i}{\sum b_j}. \tag{4.15}$$

□

Theorem 4.3 (Convexity of Relative Entropy)**.** *$D(p \parallel q)$ is convex in (p, q), i.e. if $(p_1, q_1), (p_2, q_2)$ are two pairs of distribution, then*

$$D(\lambda p_1 + (1-\lambda)p_2 \parallel \lambda q_1 + (1-\lambda)q_2) \leq \lambda D(p_1 \parallel q_1) + (1-\lambda)D(p_2 \parallel q_2) \tag{4.16}$$

Theorem 4.4 (Concavity of entropy)**.** *$H(p)$ is concave in p.*

Proof.

$$H(p) = \log |\mathcal{X}| - D(p \parallel U) \tag{4.17}$$

Since relative entropy is convex in p, its negative must be concave (and this is not affected by the constant offset.) □

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 5: Data processing inequality, asymptotic equipartition

Lecturer: Kannan Ramchandran *10 September* *Aditya Sengupta*

5.1 Data Processing Inequality

The data processing inequality states that if $X \to Y \to Z$ is a Markov chain, then $p(Z|Y,X) = p(Z|Y)$ and so

$$I(X;Y) \geq I(X;Z) \tag{5.1}$$
$$I(Y;Z) \geq I(X;Z). \tag{5.2}$$

More rigorously,

$$\begin{aligned} I(X;Y,Z) &= I(X;Y) + I(X;Z|Y) = I(X;Y) \\ &= I(X;Z) + I(X;Y|Z) \end{aligned} \tag{5.3}$$

$$\begin{aligned} I(X,Y;Z) &= I(X;Z) + I(Y;Z|X) = I(X;Z) \\ &= I(Y;Z) + I(X;Z|Y) \end{aligned} \tag{5.4}$$

Corollary 5.1. *If $Z = g(Y)$, we have $I(X;Y) \geq I(X;g(Y))$.*

Proof. $X \to Y \to g(Y)$ forms a Markov chain. □

Corollary 5.2. *If $X \to Y \to Z$ then $I(X;YZ) \leq I(X;Y|Z)$.*

Proof.

$$\begin{aligned} I(X;YZ) &= I(X;Y) + I(X;Z|Y) \\ & I(X;Z) + I(X;Y|Z) \end{aligned} \tag{5.5}$$

□

If $X - Y - Z$ forms a Markov chain, then the dependency between X and Y is decreased by observation of a "downstream" RV Z.

How crucial is the Markov chain assumption? If $X - Y - Z$ do not form a chain, then it's possible for $I(X;Y|Z) > I(X,Y)$.

Example 5.1. Let $X, Y \sim Bern(1/2)$ and $I(X;Y) = 0$. Let $Z = X \oplus Y$. Therefore $Z = \mathbb{1}\{X \neq Y\}$. Consider

$$I(X;Y|Z) = H(X|Z) - H(X|Y,Z) = H(X) = 1 - 0 = 1 \tag{5.6}$$

and $I(X;Y) = 0$. These do not form a chain but still satisfy the property. □

5.2 AEP: Entropy and Data Compression

Entropy is directly related to the fundamental limits of data compression.

1. For a sequence of n i.i.d. RVs, $X_i \sim Bern(1/2)$, we need $nH(X_1) = n$ bits.
2. If $X_i \overset{i.i.d.}{\sim} Bern(0.11)$, we need $nH(X_1) = nH_2(0.11) = \frac{n}{2}$ bits.

For a rough analysis of the entropy of a sequence, consider $(X_i)_{i=1}^n \sim Bern(p)$. The probability of a particular sequence drawn from this distribution having k ones and $n-k$ zeros is

$$\begin{aligned} p(X_1, \ldots, X_n) &= p^k (1-p)^{n-k} && (5.7)\\ &= 2^{k \log p + (n-k)\log(1-p)} && (5.8)\\ &= 2^{-n\left(\frac{k}{n}\log p + \frac{n-k}{n}\log(1-p)\right)} && (5.9) \end{aligned}$$

By the law of large numbers, $k \simeq np$, so we can more simply write

$$\begin{aligned} p(X_1, \ldots, X_n) &= 2^{-n\left(p \log \frac{1}{p} - (1-p)\log \frac{1}{1-p}\right)} && (5.10)\\ &= 2^{-nH_2(p)}. && (5.11) \end{aligned}$$

These are typical sequences with the same probability of occurring. Although there are 2^n possible sequences, the "typical" ones will have probability $2^{-nH(X)}$. The number of typical sequences is

$$\binom{n}{np} = \frac{n!}{(np)!(n-np)!} \tag{5.12}$$

and by Stirling's approximation we get (setting $\bar{p} = 1 - p$):

$$\binom{n}{np} \approx \frac{(n/e)^n}{(np/e)^{np}(n\bar{p}/e)^{n\bar{p}}} \tag{5.13}$$

$$= p^{-np}\bar{p}^{-n\bar{p}} \tag{5.14}$$

$$= 2^{n\left(p\log\frac{1}{p}+\bar{p}\log\frac{1}{\bar{p}}\right)} \qquad = 2^{nH_2(p)} \tag{5.15}$$

If $n = 1000, p = 0.11$, then $H(X) = \frac{1}{2}$. The number of possible sequences is 2^{1000}, but there are only about 2^{500} typical sequences.

The ratio of typical sequences to the total number of sequences is

$$\frac{2^{nH(X)}}{2^n} = 2^{n(H(X)-1)} \overset{n\to\infty}{\longrightarrow} 0, \tag{5.16}$$

for $H(X) \neq 1$.

Lemma 5.3 (WLLN for entropy)**.** *Let $(X_i)_{i=1}^n \overset{i.i.d.}{\sim} p$. Then by the weak law of large numbers,*

$$-\frac{1}{n}\log p(X_1,\ldots,X_n) \xrightarrow{p} H(X) \tag{5.17}$$

Proof. RVs $p(X_i)$ are i.i.d., and so by

$$-\frac{1}{n}\log p(X_1,\ldots,X_n) = -\frac{1}{n}\sum_{i=1}^{n}\log p(X_i) \xrightarrow{p} -\mathbb{E}[\log p(X)] = H(X). \tag{5.18}$$

□

The AEP enables us to partition the sequence $(X_i)_{i=1}^n \sim p$ into a typical set, containing sequences with probability approximately $2^{-nH(X)}$ (with leeway of ϵ), and an atypical set, containing the rest. This will allow us to compress any sequence in the typical set to $n(H(X)+\epsilon)$ bits.

Definition 5.1. *A typical set $A_\epsilon^{(n)}$ with respect to a distribution p is the set of sequences $(x_1,\ldots,x_n) \in \mathcal{X}^n$ with probability*

$$2^{-n(H(X)+\epsilon)} \leq p(x_1,\ldots,x_n) \leq 2^{-n(H(X)-\epsilon)}. \tag{5.19}$$

Example 5.2. Suppose in our $Bern(0.11)$ sequence, with $n = 1000$, we let $\epsilon = 0.01$. Then the typical set is

$$\{(X_1,\ldots,X_n) \mid 2^{-510} \leq p(X_1,\ldots,X_n) \leq 2^{-490}\} \tag{5.20}$$

□

Some properties of the typical set are as follows:

1. For a sequence of RVs $(X_i)_{i=1}^n \overset{i.i.d.}{\sim} p$ and sufficiently large n,

$$\mathbb{P}((X_i) \in A_\epsilon^{(n)}) \geq 1 - \epsilon \tag{5.21}$$

2. $(1-\epsilon)2^{n(H(X_1)-\epsilon} \leq |A_\epsilon^{(n)}| \leq 2^{n(H(X_1)+\epsilon}$

5.3 Data Compression

Theorem 5.4. *Let $\{X_i\}_{i=1}^n \overset{i.i.d.}{\sim} Bern(p)$ and $\epsilon > 0$. Then there exists an invertible mapping of $(X_1, \ldots, X_n)$ into binary codewords of length $l(X_1, \ldots, X_n)$ where*

$$\mathbb{E}[l(X_1, \ldots, X_n)] \leq n[H(X) + 2\epsilon]. \tag{5.22}$$

Proof. Consider our scheme which encodes $A_\epsilon^{(n)}$ using codewords of length $n(H(X)+\epsilon)$, and the atypical set using codewords of length n. Averaging over those two and using property 1 above, the expected value of a sequence length does not exceed the typical set values by more than ϵ. □

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 6: AEP, coding

Lecturer: Kannan Ramchandran *15 September* *Aditya Sengupta*

Previously, we saw that exploiting the AEP for compression gave us an expected value of compressed length slightly greater than $nH(X)$:

$$\mathbb{E}[l(X_i)] = n(H(X) + \epsilon)\mathbb{P}(A_\epsilon^{(n)}) + n(1 - \mathbb{P}(A_\epsilon^{(n)})) \approx n(H(X) + 2\epsilon) \tag{6.1}$$

With more careful counting, the code length of a typical sequence is $n(H(X) + \epsilon) + 1$.

6.1 Markov sequences

Shannon considered the problem of modeling and compressing English text. To do this, we'll need to introduce the concept of entropy rate.

Definition 6.1. *For a sequence of RVs $X_1, \dots, X_n$, where the sequence is not necessarily i.i.d., the entropy rate of the sequence is*

$$H(X) = \lim_{n\to\infty} \frac{1}{n} H(X_1, \dots, X_n), \tag{6.2}$$

if the limit exists.

For instance, if the sequence is i.i.d., $H(X) = H(X_1)$. In some (contrived) cases, $H(X)$ is not well-defined; see C&T p75.

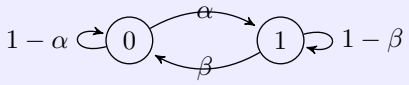
We can find the entropy rate of a stationary Markov chain:

$$H(X_1, \dots, X_n) = H(X_1) + \sum_{i=1}^{n-1} H(X_{i+1}|X_i) \tag{6.3}$$

$$= H(X_1) + (n-1)H(X_2|X_1) \tag{6.4}$$

As $n \to \infty$, the entropy rate just becomes $H(X_2|X_1)$.

Example 6.1. Consider the following Markov chain.



For example, we can take $\alpha = \beta = 0.11$. Then, our entropy rate tends to

$$H(X_2|X_1) = H(X_2|X_1 = 0)\mathbb{P}(X_1 = 0) + H(X_2|X_1 = 1)\mathbb{P}(X_1 = 1). \tag{6.5}$$

We can find that the stationary distribution in general is $\left[\frac{\beta}{\alpha+\beta}, \frac{\alpha}{\alpha+\beta}\right]$, so that lets us plug into the conditional entropy:

$$H(X_2|X_1) = H_2(0.11) \cdot \frac{1}{2} + H_2(0.89) \cdot \frac{1}{2} = 0.5 \tag{6.6}$$

Therefore, we can compress the Markov chain of length 1000 to about 500 bits. □

6.2 Codes

Previously, we used the AEP to obtain coding schemes that asymptotically need $H(X_1)$ bits per symbol to compress the source $X_1, \ldots, X_n \overset{i.i.d.}{\sim} p$.

Now, we turn our attention to "optimal" coding schemes that compress $X_1, \ldots, X_n \overset{i.i.d.}{\sim} p$ for finite values of n.

Definition 6.2. *A code $C : \mathcal{X} \to \{0,1\}^l$ is an injective function that maps letters of $\mathcal{X}$ to binary codewords.*

We denote by $C(x)$ the code for x, and we denote by $l(x)$ the length of the code, for $x \in \mathcal{X}$.

Definition 6.3. *The expected length L of a code C is defined as*

$$L \triangleq \sum_{x \in \mathcal{X}} l(x)p(x). \tag{6.7}$$

We want our codes to be uniquely decodable, i.e. any x can only have one representation in code. Specifically, we will focus on a class of codes called *prefix codes*, which have nice mathematical properties.

Our task is to devise a coding scheme C from a class of uniquely decodable prefix codes such that L is minimized.

Example 6.2. Let $\mathcal{X} = \{a, b, c, d\}$ and let $p_X = \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\}$. We want a binary code, $\mathcal{D} = \{0, 1\}$. If we Huffman encode, this turns out to be

$$c(a) = 0, c(b) = 10, c(c) = 110, c(d) = 111 \tag{6.8}$$
$$l(a) = 1, l(b) = 2, l(c) = 3, l(d) = 3. \tag{6.9}$$

We see that in this case, $L = H(p)$: we've achieved the optimal average length (although we haven't shown it's optimal yet).

□

The optimal length is not always the entropy of the distribution, due to "discretization" effects.

Example 6.3. Let $\mathcal{X} = \{a, b, c\}$ and $p_X = \{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$. The Huffman code might be $a \to 0, b \to 10, c \to 11$ or equivalent. $L = \frac{1+2+2}{3} = 1.66$, and $H(X) = \log_2 3 = 1.58$. Here the entropy cannot be reached.

□

6.3 Uniquely decodable and prefix codes

Suppose we have a code that has $a \to 0, d \to 0$. This is a singular code at the symbol level, because if we get a 0 we do not know whether to decode it to a or d.

Suppose we have a code $a \to 0, b \to 010, c \to 01, d \to 10$. If we get the code 010, we do not know whether to decode it to b, ca, or ad. This is a *singular code in extension space*.

A code is called *uniquely decodable* if its extension is non-singular. A code is called a *prefix code* if no codeword is a prefix of any other.

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 7: Designing Optimal Codes, Kraft's Inequality

Lecturer: Kannan Ramchandran *17 September* *Aditya Sengupta*

Our goal with coding theory is to find the function C to minimize the average message length L. Codes must also have certain constraints placed on them: they must be uniquely decodable, and we will focus on prefix codes.

Definition 7.1. *A code is called uniquely decodable (UD) if its extension C^* is nonsingular.*

A code is UD if no two source symbol sequences correspond to the same encoded bitstream.

Definition 7.2. *A code is called a prefix code or instantaneous code if no codeword is the prefix of any other.*

$$\text{prefix codes} \subset \text{uniquely decodable codes} \subset \text{codes}$$

Cover and Thomas table 5.1 shows some examples of codes that are various combinations of singular, UD, and prefix.

TABLE 5.1 Classes of Codes

X	Singular	Nonsingular, But Not Uniquely Decodable	Uniquely Decodable, But Not Instantaneous	Instantaneous
1	0	0	10	0
2	0	010	00	10
3	0	01	11	110
4	0	10	110	111

Figure 7.4: C&T Table 5.1: classes of codes

Prefix codes automatically do the job of separating messages, which we might do with a comma or a space.

7.1 Prefix codes

We can describe all prefix codes in terms of a binary tree. For example, let $a = 0, b = 10, c = 110, d = 111$; Figure 7.5 shows the corresponding tree. The red nodes are leaves, which correspond to full codewords, and the other nodes are interior nodes, which correspond to partial codewords.

Here, we can verify that $H(X) = L(C) = 1.75$ bits.

More generally, we can show that for any dyadic distribution, i.e. a distribution in which all of the probabilities are 2^{-i} for $i \in \mathbb{Z}^+$, there exists a code for which the codeword length for symbol j is exactly equal to $\frac{1}{\log_2 p_j(x)}$ and as a result, $L(C) = H(X)$ because

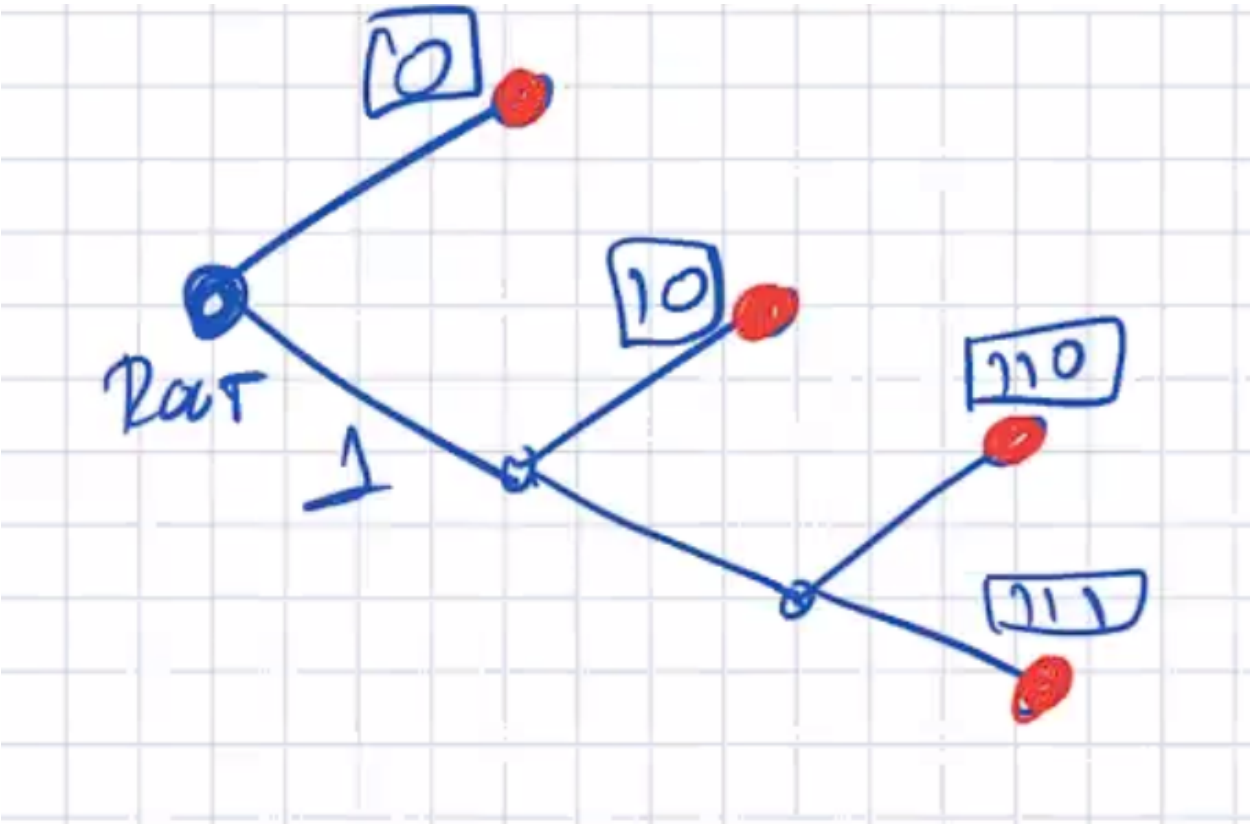


Figure 7.5: A tree corresponding to a prefix code

$$L(C) = \mathbb{E}[l_j(x)] = \sum_j p_j(x) \log_2 \frac{1}{\log_2 p_j(x)} = H(X). \tag{7.1}$$

For distributions that are not dyadic, there is a gap; for example, the optimal code for the distribution $\left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right]$ has $L(C) = 1.66$ bits whereas $H(X) = 1.58$ bits.

7.2 Optimal Compression

Seeing that $L(C) = H(X)$ in nice cases and $L(C) > H(X)$ in some other cases, we are motivated to ask the following questions:

1. Are there codes that can achieve compression rates lower than H?
2. If not, what is the best achievable compression rate compared to H?

We can set this up as an optimization problem: let $X \sim p$ be defined over an alphabet $\mathcal{X} = \{1, 2, \ldots, m\}$. Without loss of generality, assume $p_i \geq p_j$ for $i < j$. We want to find the prefix code with the minimum expected length:

$$\min_{\{l_i\}} \sum_i p_i l_i, \tag{7.2}$$

subject to l_i all being positive integers and l_i all being the codeword lengths of a prefix code. This is not a tractable optimization problem; one reason is that it deals with integer optimization, and continuous methods will not work on discrete problems. Another reason is that the space of prefix codes is too big and too difficult to enumerate for us to optimize over.

7.3 Kraft's Inequality

To make the optimization problem easier to deal with, we introduce Kraft's inequality.

Lemma 7.1. *The codeword lengths l_i of a binary prefix code must satisfy*

$$\sum_i 2^{-l_i} \leq 1. \tag{7.3}$$

Conversely, given a set of lengths satisfying Equation 7.3, there exists a prefix code with those lengths.

Proof. (in the forward direction) Consider a binary-tree representation of the prefix code. Because the code is a prefix code, no codeword can be the prefix of any other. Therefore, we can prune all the branches below a codeword node.

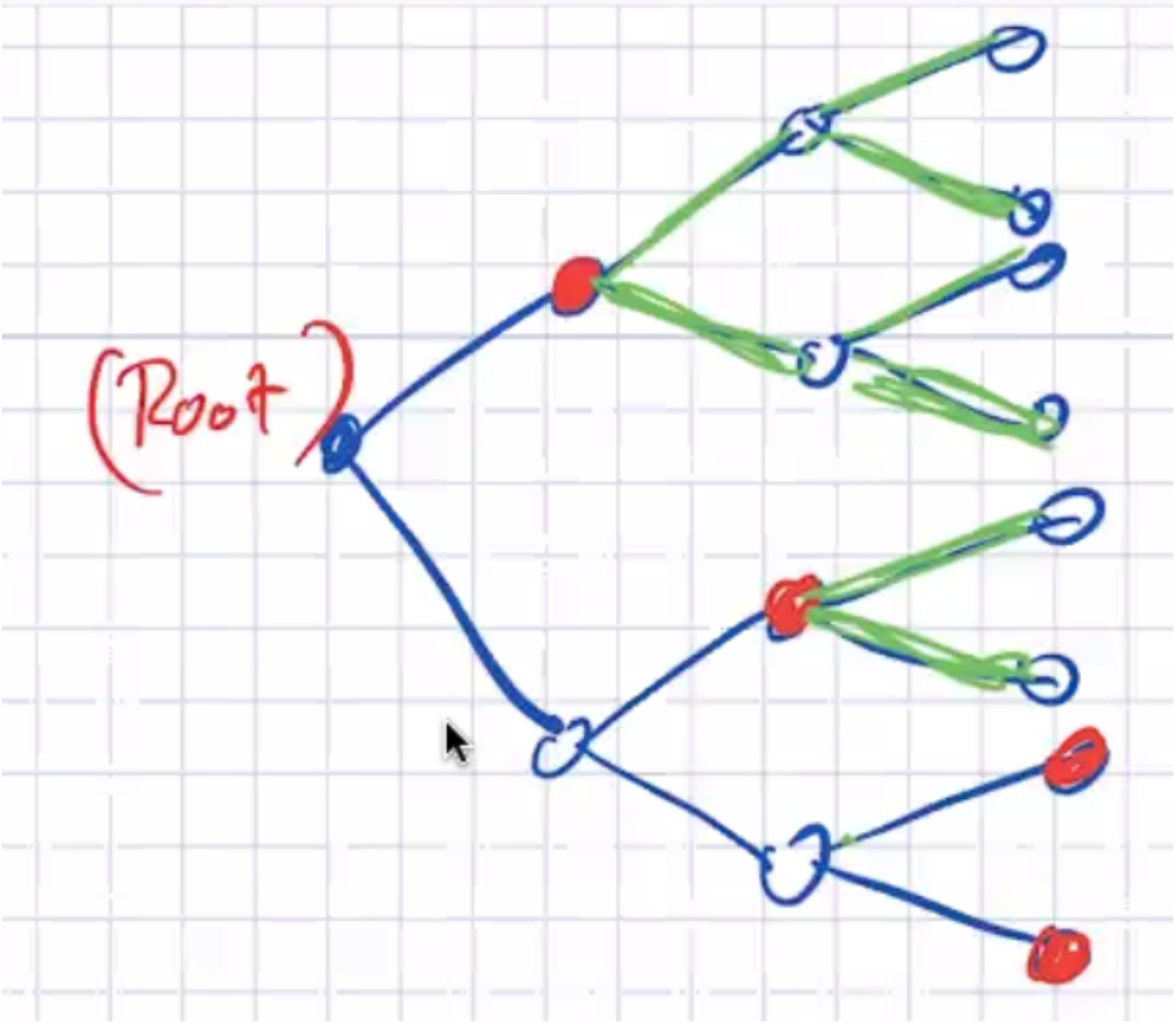


Figure 7.6: A pruned (in green) binary prefix tree.

Let $l_{\max}$ be the maximum level of the tree (the length of the longest codeword.) Each codeword at level l_i "knocks out" $2^{l_{\max}-l_i}$ descendants, so we knock out a total of $\sum_{i=1}^{m} 2^{l_{\max}-l_i}$ nodes. This must be less than the number of nodes that exist, so we get

$$\sum_{i=1}^{m} 2^{l_{\max}-l_i} \leq 2^{l_{\max}}. \tag{7.4}$$

Dividing by $2^{l_{\max}}$ on both sides, we get Kraft's inequality. □

The converse can be shown by construction.

Kraft's inequality can be intuitively understood by partitioning the probability space (the real line from 0 to 1) by halves recursively: codewords at level l_i span a fraction 2^{-l_i} of the probability space.

Now, we can rewrite our optimization problem, since Kraft's inequality has given us a way of translating the condition of being a prefix code into math:

$$\begin{aligned} &\min_{\{l_i\}} \sum_i p_i l_i \\ \text{s.t.} l_i \in \mathbb{Z}^+, &\sum_{i=1}^{m} 2^{-l_i} \leq 1 \end{aligned} \tag{7.5}$$

This is an integer programming problem, and it's still not clear if it can be solved intuitively.

7.4 Coding Schemes and Relative Entropy

$$L = \sum_{i=1}^{m} p_i l_i = \sum_{i=1}^{m} p_i \log \frac{1}{2^{-l_i}} \tag{7.6}$$

Define

$$Z = \sum_i 2^{-l_i} \tag{7.7}$$

$$q_i \triangleq \frac{2^{-l_i}}{Z}, \tag{7.8}$$

where Z is defined analogous to physics-entropy as the partition function. We can rewrite the length:

$$L = \sum_i p_i \log \frac{1}{q_i Z} = \sum_i p_i \log \frac{1}{q_i} + \sum_i p_i \log \frac{1}{Z} \tag{7.9}$$

$$= \log \frac{1}{Z} + \sum_i p_i \log \frac{1}{q_i} \tag{7.10}$$

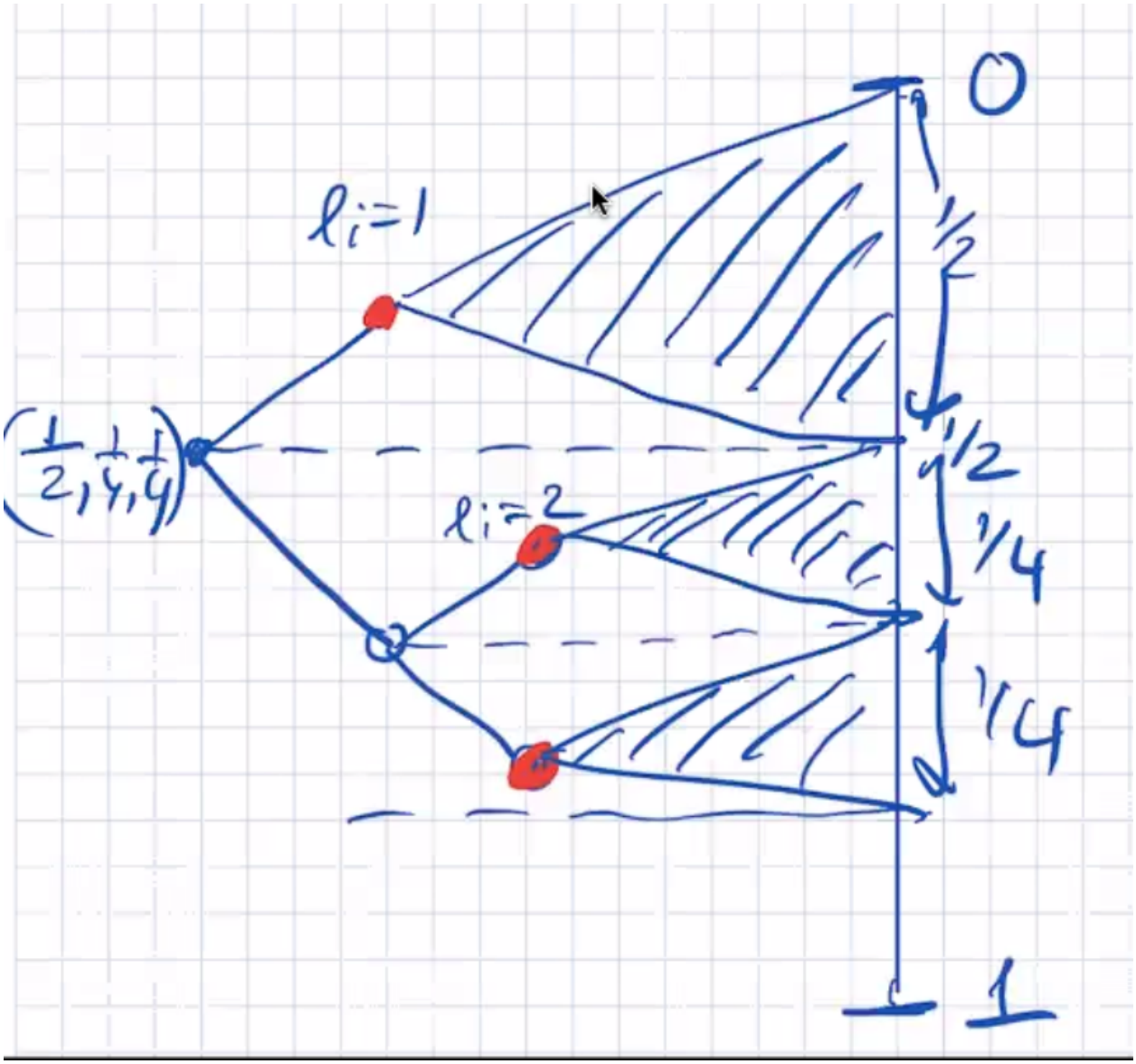


Figure 7.7: Kraft's inequality intuitively

The $\log \frac{1}{Z}$ is referred to as the slack in Kraft's inequality; if it is 0, then Kraft's inequality is an exact equality. We can split up the other term into absolute and relative entropies:

$$L = \sum_i p_i \log \frac{p_i}{q_i} + \sum_i p_i \log \frac{1}{p_i} + \log \frac{1}{Z} \tag{7.11}$$

$$= D(p \parallel q) + H(X) + \log \frac{1}{Z}. \tag{7.12}$$

The slack and the relative entropy must both be nonnegative, and therefore $L \geq H(X)$; it's impossible to beat the entropy. For equality, we need $Z = 1$, which roughly states that the code should span the whole probability space ("don't be dumb"), and we need $p = q$ (otherwise the relative entropy will be nonzero), i.e. $p_i = 2^{-l_i}$, which gives us the condition that the probabilities must be dyadic.

Generally, to get the best code length while satisfying Kraft's inequality, we set

$$\tilde{l_i} = \lceil \log_2 \frac{1}{p_i} \rceil. \tag{7.13}$$

We can show that these lengths still satisfy Kraft's inequality:

$$\sum_{i=1}^{m} 2^{-\tilde{l_i}} = \sum_{i=1}^{m} 2^{-\lceil \log \frac{1}{p_i} \rceil} \tag{7.14}$$

$$\leq \sum_{i=1}^{m} 2^{\log \frac{1}{p_i}} = \sum_{i=1}^{m} p_i = 1. \tag{7.15}$$

And we can show that this average length deviates from the entropy by at most one bit:

$$\tilde{L} = \mathbb{E}[\tilde{l}(X)] \leq \mathbb{E}\left[1 + \log \frac{1}{p(X)}\right] = 1 + H(X). \tag{7.16}$$

This is not always good news if $H(X) \ll 1$; for example, if $X \sim Bern(0.0001)$ and $H(X) \approx 2.1 \times 10^{-5}$ bits. How do we reduce the one bit of redundancy? We could encode multiple symbols at once, and in general amortize the one-bit code over n symbols to get an upper bound of $\frac{1}{n}$.

$$H(X_1, \ldots, X_n) \leq \mathbb{E}[l^*(X_1, \ldots, X_n)] \leq H(X_1, \ldots, X_n) + 1 \tag{7.17}$$

$$\frac{H(X_1, \ldots, X_n)}{n} \leq \frac{\mathbb{E}[l^*(X_1, \ldots, X_n)]}{n} \leq \frac{1}{n} + \frac{H(X_1, \ldots, X_n)}{n}. \tag{7.18}$$

For the limit as $n \to \infty$,

$$\lim_{n \to \infty} \frac{\mathbb{E}[l^*(X_1, \ldots, X_n)]}{n} = H \tag{7.19}$$

$$H = \lim_{n \to \infty} \frac{H(X_1, \ldots, X_n)}{n}. \tag{7.20}$$

Therefore, we achieve the entropy rate!

Note that this is a very general result: $X_1, \ldots, X_n$ are not assumed to be i.i.d. or Markov.

7.5 The price for assuming the wrong distribution

Consider expected length under $p(x)$ of the code lengths $l(x) = \log \frac{1}{q(x)}$ (dropping the ceiling for simplicity).

$$L = \sum_i p_i \log \frac{1}{q_i} = \sum_i p_i \log \frac{1}{p_i} + \sum_i p_i \log \frac{p_i}{q_i} \tag{7.21}$$

$$= D(p \parallel q) + H(p). \tag{7.22}$$

The cost of being wrong about the distribution is the distance (relative entropy) between the distribution you think it is, and the distribution it actually is.

$$L = H + \text{“price of being dumb”} + \text{“price of being wrong”} \tag{7.23}$$

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 8: Prefix code generality, Huffman codes

Lecturer: Kannan Ramchandran *22 September* *Aditya Sengupta*

In case a prefix code satisfies Kraft's inequality exactly, i.e. $\sum_i 2^{-l_i} = 1$, we call it a *complete* prefix code.

Since we've decided to restrict our attention to prefix codes, we might expect that we lose some optimality or efficiency by making this choice. However, prefix codes are actually fully optimal and efficient, i.e. the general class of uniquely decodable codes offers no extra gain over prefix codes. We formalize this:

Theorem 8.1 (McMillan)**.** *The codeword lengths of any UD binary code must satisfy Kraft's inequality. Conversely, given a set of lengths $\{l_i\}$ satisfying Kraft's inequality, it is possible to construct a uniquely decodable code where the codewords have the given code lengths.*

The backward direction is done already, because we know this is true of prefix codes, and prefix codes are a subset of uniquely decodable codes.

Proof. Consider C^k to be the kth extension of the base (U.D.) code C formed by the concatenation of k repeats of the base code C. By definition, if C is U.D., so is C^k.

Let $l(x)$ be the codelength of C. For C^k, the length of the codeword is

$$l(x_1, x_2, \ldots, x_k) = \sum_{i=1}^{k} l(x_i). \tag{8.1}$$

We want to show that $\sum_i 2^{-l_i} \leq 1$. The trick is to consider the kth power of Kraft's inequality:

$$\left(\sum_i 2^{-l_i}\right)^k = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_k} 2^{-l(x_1)-l(x_2)-\cdots-l(x_k)} \tag{8.2}$$

$$= \sum_{x_1, x_2, \ldots, x_k \in \mathcal{X}^k} 2^{-l(x_1,\ldots,x_k)} \tag{8.3}$$

Suppose $l_{\max}$ is the longest codelength in C. Then

$$\left(\sum_i 2^{-l_i}\right)^k = \sum_{m=1}^{kl_{\max}} a(m) 2^{-m}, \tag{8.4}$$

where $a(m)$ represents the number of code sequences of length m. $a(m) \leq 2^m$, because a code sequence of length m has 2^m possible choices and in general not all of these sequences are used. Therefore

$$\left(\sum_i 2^{-l_i}\right)^k \leq \sum_{m=1}^{kl_{\max}} 1 = kl_{\max}. \tag{8.5}$$

Therefore

$$\left(\sum_i 2^{-l_i}\right)^k \leq k l_{\max}. \tag{8.6}$$

Since this has to work for any k, and in general exponential terms are faster than linear terms, the only way for this inequality to be satisfiable is if the base for the exponentiation is less than or equal to 1. More concretely,

$$\sum_i 2^{-l_i} \leq (k l_{\max})^{1/k}. \tag{8.7}$$

This should be satisfiable in the limit $k \to \infty$, and we can show that $\lim_{k\to\infty} (k l_{\max})^{1/k} = 1$. □

8.1 Huffman codes

We have seen that $L = \mathbb{E}[l^*] \geq H(X)$, but can we find l^* efficiently?

Recall that our general optimization problem requires that we minimize $\sum_x p(x)l(x)$ such that $l(x) \in \mathbb{Z}^+$ and such that Kraft's inequality is satisfied. To design the code, we make the following observations:

- We want the highest-probability codewords to correspond to the shortest codelengths; i.e. if $i < j \implies p_i \geq p_j$ then $i < j \implies l_i \leq l_j$. This is provable by contradiction.
- If we use all the leaves in the prefix code tree (and we should), then the deepest two leaves will be on the same level, and therefore the two longest codewords will differ only in the last bit.

This gives rise to Huffman's algorithm:

1. Take the two least probable symbols. They correspond to the two longest codewords, and differ only in the last bit.
2. Combine these into a "super-symbol" with probability equal to the sum of the two individual probabilities, and repeat.

Example 8.1. Let $\mathcal{X} = \{a, b, c, d, e\}$ and $P_X = \{0.25, 0.25, 0.2, 0.15, 0.15\}$.

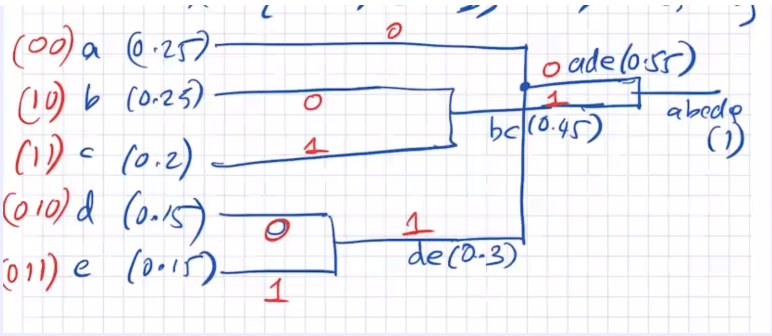


Figure 8.8: Example Huffman encoding

The average length of this encoding is $L = 2.3$, and the entropy is around $H(X) = 2.28$.

□

8.2 Limitations of Huffman codes

"you know how they say 'I come to bury Caesar, not to praise him'? I've come here to kill Huffman codes." - Prof. Ramchandran

There are several reasons not to like Huffman codes.

1. Computational cost: the time complexity of making the Huffman tree is $O(m \log m)$ where $|\mathcal{X}| = m$. For a block of length n, the alphabet size is m^n, and so this gets painful quickly.
2. They are designed for a fixed block length, which is not a widely applicable assumption
3. The source is assumed to be i.i.d., and non-stationarity is not handled.

To address this, we'll introduce concepts like arithmetic coding.

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 9: Arithmetic coding

Lecturer: Kannan Ramchandran *24 September* *Aditya Sengupta*

9.1 Overview and Setup, Infinite Precision

Arithmetic coding is a (beautiful!) entropy coding scheme for compression that operates on the entire source sequence rather than one symbol at a time. It takes in a message all at once, and returns a binary bitstream corresponding to it.

Consider a source sequence, such as "the quick brown fox jumps over the lazy dog". Suppose this has some encoding to a binary string, such as 011011110110000. We can interpret this as a binary fraction by prepending a 0., so that we get 0.011011110110000, a number in $[0, 1)$. Note that these are binary fractions, so 0.01 represents $\frac{1}{4}$.

The key concept behind arithmetic coding is that we can separate the encoding/decoding aspects from the probabilistic modeling aspects without losing optimality.

Example 9.1. Consider a dyadic distribution, $X \sim p$, $\mathcal{X} = \{1, 2, 3, 4\}, p_X = \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\}$. We write the CDF in terms of binary fractions; in general if we want to represent the interval $[a, b)$ we'd like to choose a binary fraction that corresponds to a, and such that the number of bits used corresponds to the length $b - a$. For example, the length spanned by $x = 2$ is $\frac{1}{4}$ and so we want $-\log \frac{1}{4} = 2$ bits.

$$F_X(x) = \begin{cases} 0.0 & x = 1 \\ 0.10 & x = 2 \\ 0.110 & x = 3 \\ 0.111 & x = 4 \end{cases} \tag{9.1}$$

Further, we can continue subdividing each interval to assign longer messages! For example, if we subdivide $[0, 1/2)$, we can assign codewords to messages starting with 1, i.e. "$11'' \to [0, 1/4)$, "$12'' \to [1/4, 1/8)$ and so on.

□

In the limit, the subinterval representing $(X_1, X_2, \ldots, X_n)$ shrinks to a single point.

Definition 9.1. *Given a sequence of RVs $X_1, X_2, \cdots \sim p$ over $\mathcal{X} = \{0, 1\}$ where $X_i \sim Bern(p)$, the arithmetic code for $x = 0.X_1X_2\ldots$ is*

$$F_X(x) = U = 0.U_1U_2U_3\ldots, \tag{9.2}$$

where $U_1, U_2, \ldots$ are the encoded symbols for $X_1, X_2, \ldots$ and $F_X(x) = \mathbb{P}(X < x)$.

Theorem 9.1. *The arithmetic code F achieves the optimal compression rate. Equivalently, the encoded symbols $U_1, U_2, \ldots$ are i.i.d. $Bern(1/2)$ RVs and cannot be further compressed.*

Lemma 9.2. *If X is a continuous RV with cdf F, then $F_X(x) \sim Unif[0,1]$.*

Proof. Let $U \sim Unif[0,1]$:

$$\mathbb{P}(U \leq u) = \mathbb{P}(F_X(x) \leq u) = \mathbb{P}(X \leq F_X^{-1}(u)) = F_X(F_X^{-1}(u)). \tag{9.3}$$

□

The proof of the theorem is in C&T probably.

9.2 Finite Precision

Suppose we have $\mathcal{X} = \{A, B, C\}$ and $p_X = \{0.4, 0.4, 0.2\}$. Suppose we want to encode ACAA.

We assign $A \to [0, 0.4), B \to [0.4, 0.8), C \to [0.8, 1)$. The encoding of A gives us the interval $[0, 0.4)$, and we further subdivide this interval. The next symbol is a C so we get the interval $[0.32, 0.4)$ (this encodes the distribution $X_2|X_1$, in general the second symbol's distribution doesn't have to be the same distribution as the first). The next A gives us $[0.32, 0.352)$, and the final A gives us $[0.32, 0.3328)$. Finally, we reserve a code symbol for the end of a string, so that we know exactly when a string ends.

The subinterval's size represents $\mathbb{P}(ACAA) = \mathbb{P}(X_1 = A)\mathbb{P}(X_2 = C|X_1 = A)\mathbb{P}(X_3 = A|X_1X_2 = AC)\mathbb{P}(X_4 = A|X_1X_2X_3 = ACA)$. Note that we can capture arbitrary conditioning in the encoding!

9.3 Invertibility

We can show that $l(X^n) \leq \log \frac{1}{p(X^n)} + 2$.

We can use a Shannon-Fano-Elias code argument to show this: for any message, the binary interval corresponding to it must fully engulf the probability interval of the sequence. The +2 comes from ensuring that the interval represented by the code is fully inside the probability interval of the sequence, for which we may have to subdivide twice.

If we have a source interval with a marginally smaller probability than the optimal coding interval size, there's no way to identify whether the interval belongs to the source interval or those above/below it given just the coding interval. When we divide it in half, the top and bottom halves still have ambiguity: in order to ensure there's an interval that is unambiguously within the optimal interval, we need to divide twice and use the two "inner" midpoints.

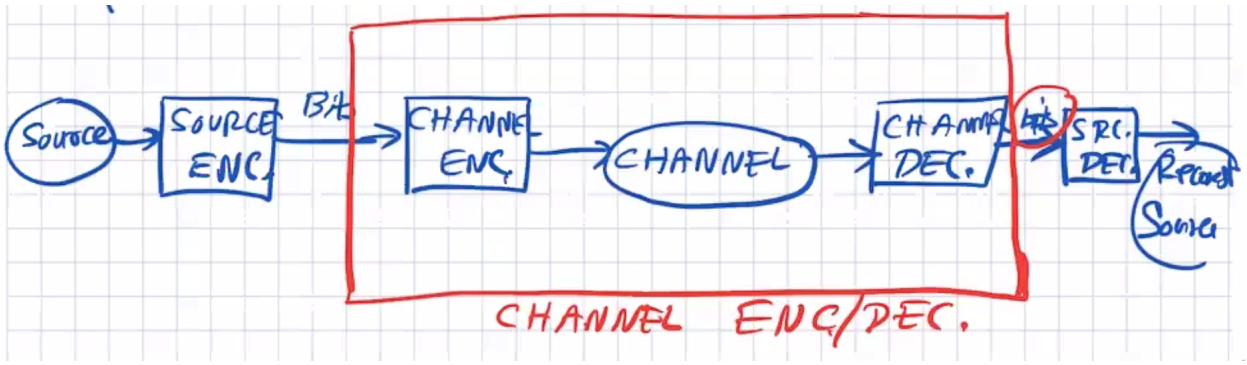


Figure 10.9: Channel diagram

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 10: Arithmetic coding wrapup, channel coding

Lecturer: Kannan Ramchandran *29 September* *Aditya Sengupta*

Arithmetic coding has several advantages over, e.g., Huffman codes:

1. Complexity: to communicate a string of N symbols from an alphabet of size $|\mathcal{X}| = m$, both the encoder and the decoder need to compute only Nm conditional probabilities Contrast this cost with the alternative of using a large block-length Huffman code, where all the possible block sequences have to have their probabilities evaluated.
2. Flexibility: can be used with any source alphabet and any encoded alphabet.
3. Arithmetic coding can be used with any probability distribution that can change from context to context.
4. Can be used in a "universal" way. Arithmetic coding does not require any specific method of generating the prediction probabilities, and so these may be naturally produced using a Bayesian approach. For example, if $\mathcal{X} = \{a, b\}$, Laplace's rule gives us that $\mathbb{P}(a|x_1, x_2, \ldots, x_{n-1}) = \frac{F_a+1}{F_a+F_b+2}$ (where F_a, F_b are the number of occurrences so far of a and b.) Thus the code adapts to the true underlying probability distribution.

10.1 Communication over noisy channels

Consider a block diagram source-to-encoder-to-noisy-channel-to-decoder-to-reconstructed-source.

Even though the channel is noisy, we would like to ensure that there is no noise induced overall between the source encoder and source decoder, while maximizing the channel's capacity, i.e. the maximum number of bits per use of the channel such that communication remains reliable.

Say we give the channel an input W and that the output is $\hat{W}$. We can say the channel takes in and spits out bits, which means we need to add an encoder before the channel and a decoder after.

1. The encoder converts $W \sim U[0, 2^m - 1]$ (an optimally compressed source with m bits) to a string of n bits $X_1, \ldots, X_n$.
2. The channel takes in this sequence of bits and returns corrupted bits $Y_1, \ldots, Y_n$ following the probability $\mathbb{P}(Y_1, \ldots, Y_n | X_1, \ldots, X_n)$.
3. The decoder takes in the corrupted bits and outputs the optimal decoding $\hat{W}$.

We define two metrics to assess channel performance:

1. Probability of error, $P_e = \mathbb{P}(W \neq \hat{W})$
2. Rate $R = \frac{\log_2 M}{n} = \frac{m}{n}$.

Communication is an optimization problem between these two factors. We can find an optiml probability of error for a fixed data rate R and length $|W|$: $P(err)^* = \min_{f,g} P(err)$, where f and g are the encoder and decoder.

The simplest possible strategy is a repetition code, where each encoded symbol is sent n times. Suppose the channel is a $BEC(p)$. The probability of error is $P_e = \frac{1}{2}p^n$ (every single bit needs to be erased), and the rate is $R = \frac{1}{n}$. The rate is low while the probability of error is also low; we'd like to bring the rate up without increasing P_e much.

Shannon showed that at some value of the rate, P_e goes to zero and reliable communication is possible.

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 11: Channel coding efficiency

Lecturer: Kannan Ramchandran *1 October* *Aditya Sengupta*

We're going to focus on the *discrete memoryless channel*, with the property that $\mathbb{P}(Y^n|X^n) = \prod_{i=1}^n \mathbb{P}(y_i|x_i)$.

The capacity of a channel is $C = \max_{p(x)} I(X;Y)$. This is intuitively something we want to maximize: we want to make it so that the output tells us as much as possible about the input. The channel is specified by the joint distribution $p(x,y)$, in which $p(x)$ is to be designed (based on how the encoder works) and $p(y|x)$ is fixed (it characterizes the channel).

Example 11.1. Consider a noiseless channel, where $H(Y|X) = 0$. If $X \sim Bern(1/2)$ then the maximum is achieved: $C = \max_{p(x)} H(X) = 1$.

□

11.1 Noisy typewriter channel

Suppose $\mathcal{X} = \{A, B, C, D\}$, and the channel takes each letter to itself with probability 1/2 and to its next neighbor (D going to A) with probability 1/2.

Due to this symmetry, $H(Y|X) = 1$, and so to maximize the capacity we want to maximize $H(X)$. In the uniform case, this comes to $C = 2 - 1 = 1$.

For a BSC with flip probability p, the channel capacity is $1 - H(p)$, because the maximum initial entropy is 1 and the minimum conditioned flip entropy is $H(p)$.

For a BEC, $H(X) = H_2(\alpha)$ (where α is a starting Bernoulli parameter), and for the conditional entropy:

$$H(X|Y=0) = 0 \tag{11.1}$$
$$H(X|Y=1) = 0 \tag{11.2}$$

$$H(X|Y=e) = \sum_{x\in\{0,1\}} p(x|Y=e)\frac{1}{\log p(x|Y=e)} \tag{11.3}$$
$$= (1-\alpha)\log\frac{1}{1-\alpha} + \alpha\log\frac{1}{\alpha} \tag{11.4}$$

Therefore the capacity is

$$C = \max_{p(x)} H_2(\alpha) - pH_2(\alpha) = 1 - p. \tag{11.5}$$

Theorem 11.1 (Shannon: Channel Coding)**.** *Any rate $R < C$ is achievable, meaning that there exists a scheme for communication at rate R such that $P_e \to 0$. Conversely, if $R > C$, P_e is strictly bounded away from 0.*

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 12: Rate Achievability

Lecturer: Kannan Ramchandran *6 October* *Aditya Sengupta*

Shannon's capacity achievability argument will end up concluding that the capacity of a $BEC(p)$ cannot exceed $1-p$, but that the capacity $1-p$ is achievable. Even with the aid of a "feedback oracle", the capacity of a $BEC(p)$ is at most $1-p$.

To prove the achievability, make a codebook. Let $R = \frac{\log M}{n}$ and $M = 2^{nR}$. Write a codebook with $n-$length codewords for each of the M possible messages, and use the encoding scheme that we saw in 126 (just copy that over). The probability of error with this encoding scheme is

$$P_e = \mathbb{E}_c[P_e(c)] = \sum_{c \in C} \mathbb{P}(C = c)\mathbb{P}(\hat{W} \neq W | c) \tag{12.1}$$

$$= \sum_{c} \mathbb{P}(\hat{W} \neq W, C = c) \tag{12.2}$$

$$= \mathbb{P}(\hat{W} \neq W). \tag{12.3}$$

The channel action is as follows:

1. The encoder and decoder generate a random codebook C based on iid fair coin flips $\sim Bern(1/2)$.
2. A message W is chosen at random from $[M]$.
3. The wth codeword $\vec{c}_w = (c_w^i)$ is transmitted over the $BEC(p)$ channel.
4. The receiver gets Y^n according to $\mathbb{P}(Y^n | X^n)$.
5. Decoding rule: if there is exactly one codeword consistent with the received Y^n, then declare the message associated with that codeword as the decoded message; else, declare error.

Example 12.1. Suppose $M = 4, R = \frac{1}{2}$, and the codebook is

	1	2	3	4
1	0	0	0	0
2	0	1	1	0
3	1	0	0	1
4	1	1	1	1

Here, if we send 0000, that gets uniquely decoded as 1.

If we send 0ee0, it is uncertain whether the message was 1 or 2, so an error is declared.

□

The probability of error is then

$$\overline{P_e} = \mathbb{P}(W \neq \hat{W}) = \mathbb{P}(\hat{W} \neq 1 | W = 1) \tag{12.4}$$

$$= \mathbb{P}(\cup_{w=2}^{M} x^n(w) = x^n(1) \text{ in all the unerased bit locations}) \quad \leq \sum_{w=2}^{M} \mathbb{P}(x^n(w) = x^n(1) \text{ in all the unerased bit locations}) \tag{12.5}$$

By LLN, the number of erasures tend to np with probability 1, and so the number of unerased bit locations tends to $n(1-p)$ with probability 1. Since each bit is independently generated with probability $\frac{1}{2}$, we have

$$\mathbb{P}(\hat{W} = 2 | W = 1) = \frac{1}{2^{n(1-p)}} \tag{12.6}$$

And therefore

$$\overline{P_e} < 2^{nR} 2^{-n(1-p)} = 2^{-n((1-p)-R)} \tag{12.7}$$

Therefore, as $n \to \infty, \overline{P_e} \to 0$ if $1 - p > R$. Therefore, for any $\epsilon > 0$, $R = 1 - p - \epsilon$ is achievable.

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 13: Joint Typicality

Lecturer: Kannan Ramchandran *8 October* *Aditya Sengupta*

We know what typicality with respect to one sequence means; now we look at typicality with a set of multiple sequences. We've already seen what a typical sequence of RVs is, and in particular we saw the definition of a width-ϵ typical set:

$$A_\epsilon^{(n)} = \{x^n \mid |\log p(x^n) + nH(X)| \leq n\epsilon\}. \tag{13.1}$$

For what follows, denote typicality by saying $p(x^n) \sim 2^{-nH(X)}$.

Now, we extend this:

Definition 13.1. *For iid RVs X^n and iid RVs Y^n, (x^n, y^n) is called a jointly typical sequence if*

$$p(x^n) \sim 2^{-nH(X)} \tag{13.2}$$
$$p(x^n) \sim 2^{-nH(Y)} \tag{13.3}$$
$$p(x^n, y^n) \sim 2^{-nH(X,Y)}. \tag{13.4}$$

13.1 BSC Joint Typicality Analysis

Consider a $BSC(p)$ where $X \sim Bern(1/2)$, i.e. the output Y is related to the input X by

$$Y = X \oplus Z \tag{13.5}$$

where $Z \sim Bern(p)$.

The joint entropy is

$$H(X,Y) = H(X) + H(Y|X) = 1 + H_2(p) \tag{13.6}$$

Let (x^n, y^n) be jointly typical. Then, $p(x^n) \sim 2^{-n}, p(y^n) \sim 2^{-n}$, and $p(x^n, y^n) \sim 2^{-n(1+H_2(p))}$.

From the asymptotic equipartition property (almost all the probability space is taken up by typical sequences), we get that

$$|\{x^n \mid p(x^n) \sim 2^{-n}\} \sim 2^n \tag{13.7}$$
$$|\{y^n \mid p(y^n) \sim 2^{-n}\} \sim 2^n \tag{13.8}$$
$$|\{(x^n, y^n) \mid p(x^n, y^n) = 2^{-n(1+H_2(p))}\}| \sim 2^{n(1+H_2(p))} \ll 2^{2n}. \tag{13.9}$$

Next, consider the conditional distribution: for jointly typical (x^n, y^n),

$$p(y^n|x^n) = \frac{p(x^n, y^n)}{p(x^n)} \sim \frac{2^{-nH(X,Y)}}{2^{-nH(X)}} \tag{13.10}$$

and so, for jointly typical (x^n, y^n),

$$p(y^n|x^n) \sim 2^{-nH(Y|X)} \tag{13.11}$$

If you imagine a space of all the possible sequences x^n, you can draw "noise spheres" of typical sequences that are arrived at by going through the BSC. The size of these noise spheres is $\binom{n}{np} \approx 2^{nH_2(p)}$ by Stirling's approximation.

The number of disjoint noise spheres (each corresponding to a single transmitted codeword $x^n(\omega)$) is less than or equal to $2^{nH(Y)}/2^{nH(Y|X)} = 2^{nI(X;Y)}$. Therefore, the total number of codewords M is

$$M = 2^{nR} \leq 2^{nI(X;Y)} \tag{13.12}$$

and so

$$\frac{\log_2 M}{n} = R \leq I(X;Y) = C. \tag{13.13}$$

We have therefore shown that rates of transmission over the BSC necessarily satisfy $R \leq C$. Now, we show that any $R \leq C$ is achievable. This can be done by the same codebook argument as for the BSC. The receiver gets $Y^n = X^n(\omega) \oplus Z^n$. Without loss of generality, we'll assume $W = 1$ was sent.

From there, the decoder does typical-sequence decoding: it checks whether $(X^n(1), Y^n)$ is jointly typical. Let Z_i^n be the noise candidate associated with message i.

We can see that

$$Z_i^n = X^i(1) \oplus Y^i \ \forall i \in \{1, \ldots, M\} \tag{13.14}$$

This gives us a sequence of noise values, and we want this sequence to look like $Bern(1/2)$.

Therefore, our decoding rule is as follows: the decoder computes $y^n \oplus x^n(w)$ for all $w \in \{1, 2, \ldots, M\}$. Eliminate all w that $Y^n \oplus X^n(w) \notin A_\epsilon^{(n)}$. If only one message survives, declare it the winner. Else, declare error.

$$\overline{P_e} = \mathbb{E}_c[P_e(c)] = P(\hat{W} \neq W) \tag{13.15}$$

$$\leq \mathbb{P}(\bigcup \{Z^n(w) \in A_\epsilon^{(n)}\}) \cup (somethingImissed) \tag{13.16}$$

$$< M\mathbb{P}(Z^n(2) \in A_\epsilon^{(n)} | W = 1) \tag{13.17}$$

$$= 2^{nR} \frac{|A_\epsilon^{(n)}}{\text{total number of possible sequences}} \tag{13.18}$$

$$= 2^{nR} 2^{n(H_2(p)+1)} / 2^n \tag{13.19}$$

$$= 2^{-n(1-H_2(p)-\epsilon-R)}. \tag{13.20}$$

Therefore, if $R < 1 - H_2(p) - \epsilon$, the probability of error goes to 0. Therefore, a rate of $1 - H_2(p)$ is achievable by making ϵ arbitrarily small.

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 14: Fano's Inequality for Channels

Lecturer: Kannan Ramchandran *12 October* *Aditya Sengupta*

I missed this lecture and later referred to the official scribe notes.

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 15: Polar Codes, Fano's Inequality

Lecturer: Kannan Ramchandran *15 October* *Aditya Sengupta*

15.1 Recap of Rate Achievability

Fano's inequality tells us that for any estimator $\hat{X}$ such that $X \to Y \to \hat{X}$ is a Markov chain with $P_{err} = \mathbb{P}(\hat{X} \neq X)$,

$$H_2(P_e) + P_e \log |\mathcal{X}| \geq H(X|Y). \tag{15.1}$$

To prove the converse for random coding, that $R \leq C$ is achievable, we set up the Markov chain $W \to X^n \to Y^n \to \hat{W}$:

$$nR = H(W) = I(W; \hat{W}) + H(W|\hat{W}) \tag{15.2}$$
$$\leq I(X^n; Y^n) + 1 + P_e^{(n)} nR \tag{15.3}$$

Therefore $R \leq C + \frac{1}{n} + RP_e^{(n)}$; in the limit $n \to \infty, P_e^{(n)} \to 0$.

15.2 Source Channel Theorem

Theorem 15.1. *If $V_1, \dots V_n$ is a finite-alphabet stochastic process for which $H(\nu) < C$, there exists a source-channel code with $P_{err} \to 0$.*

Conversely, for any stationary process, if $H(\nu) > C$ then P_{err} is bounded away from zero, and it is impossible to transmit the process over the channel reliably (i.e. with arbitrarily low P_{err}).

Proof. For achievability, we can use the two-stage separation scheme: Shannon compression and communication using random codes. If $H(\nu) < C$, reliable transmission is possible.

For the converse, we want to show that $\mathbb{P}(V^n \neq \hat{V}^n) \to 0$ implies that $H(\nu) \leq C$ for any sequence of source-channel codes:

$$X^n(\nu^n) : V^n \to X^n \tag{15.4}$$
$$g_n(Y^n) : \mathcal{Y}^n \to V^n \tag{15.5}$$

Consider the Markov chain $V^n \to X^n \to Y^n \to \hat{V}^n$. Fano's inequality tells us

$$H(V^n|\hat{V}^n) \leq 1 + P(V^n \neq \hat{V}^n) \log |\nu^n| = 1 + P(V^n \neq \hat{V}^n) n \log |\nu|. \tag{15.6}$$

For a source channel code:

$$\begin{aligned} H(\nu) &\leq \frac{H(V_1, \ldots, V_n)}{n} = \frac{H(V^n)}{n} && (15.7)\\ &= \frac{1}{n}\left[H(V^n|\hat{V}^n) + I(V^n; \hat{V}^n)\right] && (15.8)\\ &\leq \frac{1}{n}\left[1 + \mathbb{P}(V^n \neq \hat{V}^n) n \log |\nu| + I(X^n; Y^n)\right] && (15.9)\\ &\leq \frac{1}{n} + \mathbb{P}(V^n \neq \hat{V}^n) \log |\nu| + C && (15.10) \end{aligned}$$

If we let $n \to \infty$, then $\mathbb{P}(\hat{V}^n \leq V^n) = 0$ and therefore $H(\nu) \leq C$. □

15.3 Polar Codes

Polar codes were invented by Erdal Arikan in 2008. They achieve capacity with encoding and decoding complexities of $O(n \log n)$, where n is the block length.

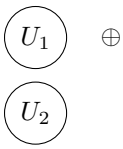
Among all channels, there are two types of channels for which it's easy to communicate optimally:

1. The noiseless channel, $X - Y$; Y determines X.
2. The useless channel, X Y; $Y \perp\!\!\!\perp X$.

In a perfect world, all channels would be one of these extremal types. Arikan's polarization is a technique to convert binary-input channels to a mixture of binary-input extremal channels. This technique is information-lossless and is of low complexity!

Consider two copies of a binary input channel, $W : \{0, 1\} \to Y$. The first one takes in X_1 and returns Y_1 and the second takes in X_2 and returns Y_2.

Set $X_1 = U_1 \oplus U_2, X_2 = U_2$, where $U_1, U_2 \sim Bern(1/2)$ independently.



X_1, X_2 are also independent $Bern(1/2)$ RVs.

$$I(X_1; Y_1) + I(X_2; Y_2) = I(X_1 X_2; Y_1 Y_2) = I(U_1 U_2; Y_1 Y_2) \tag{15.11}$$

Use the shorthand $I(W) = I(\text{input}; \text{output})$ where the binary input is uniform on $\{0, 1\}$. Then the above equation combined with the chain rule gives us

$$2I(W) = I(U_1; Y_1Y_2) + I(U_2; Y_1, Y_2|U_1) \tag{15.12}$$

We can rewrite the second term as

$$2I(W) = I(U_1; Y, Y_2) + I(U_2; Y_1, Y_2, U_1) \tag{15.13}$$

Denote the first term by $I(W^-)$ and the second by $I(W^+)$:

$$2I(W) = I(W^-) + I(W^+). \tag{15.14}$$

We'll now show two things:

1. W^- and W^+ are associated with particular channels that satisfy the extremal property we saw before.
2. $I(W^-) \leq I(W) \leq I(W^+)$.

Given the mutual information expressions, the channel W^- takes in as input U_1 and outputs (Y_1, Y_2). U_2 is also an input but we're not looking at it for now, I think.

Example 15.1. For a $BEC(p)$,

$$(Y_1, Y_2) = \begin{cases} (U_1 \oplus U_2, U_2) & \text{w.p.} (1-p)^2 \\ (e, U_2) & \text{w.p.} p(1-p) \\ (U_1 \oplus U_2, e) & \text{w.p.} p(1-p) \\ (e, e) & \text{w.p.} p^2 \end{cases} \tag{15.15}$$

What can we say about the mutual information?

W^- is a $BEC(p^-) = BEC(1-(1-p)^2) = BEC(2p-p^2)$. If $p = \frac{1}{2}, p^- = \frac{3}{4}$ and so $I(W^-) = \frac{1}{4}$. Therefore, by mutual information balance we get that $I(W^+) = \frac{3}{4}$. □

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 16: Polar Codes

Lecturer: Kannan Ramchandran *27 October* *Aditya Sengupta*

Recall that the polar transform changes assumed $Bern(1/2)$ random variables U_1, U_2 to X_1, X_2 by the relationship $X_1 = U_1 \oplus U_2, X_2 = U_2$.

We showed last time that

$$2I(X_1; Y_1) = I(U_1, U_2; Y_1, Y_2) = I(U_1; Y_1, Y_2) + I(U_2; Y_1, Y_2|U_2), \tag{16.1}$$

and we can write this as the combination of two channels:

$$2I(W) = I(W^-) + I(W^+). \tag{16.2}$$

W^- is a worse channel than W, and W^+ is a better channel than W.

Suppose W is a $BEC(p)$; we can then show that W^- is a $BEC(2p - p^2)$. Similarly, we can look at W^+:

$$I(W^+) : U_2 \to Y_1 Y_2 U_2 \tag{16.3}$$

$$(Y_1, Y_2, U_1) = \begin{cases} (U_1 \oplus U_2, U_2, U_1) & \text{w.p.} (1-p)^2 \\ (e, U_2, U_1) & \text{w.p.} p(1-p) \\ (U_1 \oplus U_2, e, U_1) & \text{w.p.} (1-p)p \\ (e, e, U_1) & \text{w.p.} p^2 \end{cases} \tag{16.4}$$

Therefore W^+ is a $BEC(p^2)$, as U_2 can be recovered with probability $1 - p^2$.

We can now verify that the polar transformation preserves mutual information:

$$\frac{I(W^-) + I(W^+)}{2} = \frac{1 - (2p - p^2) + 1 - p^2}{2} = 1 - p = I(W) \tag{16.5}$$

The synthetic channels are not real, and this is fine for W^-, but not for W^+, since we only have $\hat{U_1}$ and not the real U_1: U_1 is not actually observed by the receiver. To get around this, we impose a decoding order.

Consider a genie-aided receiver that processes the channel outputs:

$$\tilde{U_1} = \phi_1(Y_1, Y_2) \tag{16.6}$$

$$\tilde{U_2} = \phi_2(Y_1, Y_2, U_1) \tag{16.7}$$

where U_1 is perfectly known by the genie.Consider also the implementable receiver

$$\hat{U_1} = \phi_1(Y_1, Y_2) \tag{16.8}$$

$$\hat{U_2} = \phi_2(Y_1, Y_2, \hat{U_1}) \tag{16.9}$$

We claim that if the genie-aided receiver makes no errors, then neither does the implementable receiver. That is, the block error events $\{(\tilde{U_1}, \tilde{U_2}) \neq (U_1, U_2)\}$ and $\{(\hat{U_1}, \hat{U_2}) \neq (U_1, U_2)\}$ are identical.

If the genie is wrong, then $U_1 \neq \tilde{U_1}$. Therefore we have an overall error: $(\tilde{U_1}, \tilde{U_2}) \neq (U_1, U_2)$, and the implementable receiver would have an error too. It is possible for an error in $\hat{U_1}$ to propagate, but we don't care about that, because in that case the genie also failed to decode U_1 and so we declare an error in both anyway.

The main idea behind polar codes is to transform message bits such that a fraction $C(W)$ of those bits "see" noiseless channels, whereas a fraction $1 - C(W)$ of bits "see" useless channels. We can do this recursively: take two W^- copies and two W^+ and repeat the whole process.

Some diagrams and stuff tell us that the 4×4 polar transform has a matrix

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}, \tag{16.10}$$

which is the tensor operation $\otimes$: $P_4 = P_2 \otimes P_2$.

$$P_4 = \begin{bmatrix} P_2 & P_2 \\ 0 & P_2 \end{bmatrix}. \tag{16.11}$$

In general, $P_{2^{k+1}} = P_{2^k} \otimes P_{2^k}$.

For concreteness, with a $BEC(1/2)$ with three splits, we can translate it into eight channels: W^{+++} has a probability of success of 0.9961.

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 17: Information Measures for Continuous RVs

Lecturer: Kannan Ramchandran *3 November* *Aditya Sengupta*

We saw last time that the capacity of the Gaussian channel, $Y_i = X_i + Z_i$ where $Z_i \sim \mathcal{N}(0, \sigma^2)$ is

$$C = \frac{1}{2}\log_2(1 + SNR), \tag{17.1}$$

where $SNR = \frac{P}{\sigma^2}$. Similarly to the discrete case, we can show any rate up to the capacity is achievable, via a similar argument (codebook and joint typicality decoding).

Similarly as the discrete-alphabet setting, we expect $\frac{1}{2}\log(1 + SNR)$ to be the solution of a mutual information maximization problem with a power constraint:

$$\max_{p(x)} I(X;Y) \text{ s.t.} \mathbb{E}[X^2] \leq P, \tag{17.2}$$

where $I(X;Y)$ is some notion of mutual information between continuous RVs X, Y. This motivates the fact that we need to introduce the notion of information measures, like we saw in the discrete case (mutual information, KL divergence/relative entropy, and entropy) for continuous random variables before we can optimize something like Equation 17.2.

Chapter 8 of C&T has more detailed exposition on this topic.

As our first order of business, let's look at mutual information. We can try and make a definition that parallels the case of DRVs (discrete RVs). Recall that for DRVs X, Y,

$$I(X;Y) = \mathbb{E}\left[\log \frac{p(x,y)}{p(x)p(y)}\right] \tag{17.3}$$

The continuous case is similar: all we have to do is replace the PMFs $p(x), p(y), p(x,y)$ with their PDFs $f(x), f(y), f(x,y)$.

Definition 17.1. *The mutual information for CRVs $X, Y \sim f$ is*

$$I(X;Y) \triangleq \mathbb{E}\left[\log \frac{f(x,y)}{f(x)f(y)}\right] \tag{17.4}$$

To show this is a sensible definition, we show it is an approximation to the discretized form of X, Y. Divide up the real line into width-Δ subintervals. For $\Delta > 0$, define $X^\Delta = i\Delta$ if $i\Delta \leq X \leq (i+1)\Delta$. Then, for small Δ,

$$p(X^\Delta) \approx \Delta f(x) \tag{17.5}$$
$$p(Y^\Delta) \approx \Delta f(y) \tag{17.6}$$
$$p(X^\Delta, Y^\Delta) \approx \Delta^2 f(x) f(y) \tag{17.7}$$

Then,

$$I(X^\Delta; Y^\Delta) = \mathbb{E}\left[\log \frac{p(X^\Delta, Y^\Delta)}{p(X^\Delta)p(Y^\Delta)}\right] \tag{17.8}$$
$$= \mathbb{E}\left[\log \frac{f(x,y)\Delta^2}{f(x)\Delta f(y)\Delta}\right] \tag{17.9}$$
$$= I(X;Y) \tag{17.10}$$

Therefore, we see that

$$\lim_{\Delta \to 0} I(X^\Delta; Y^\Delta) = I(X;Y) \tag{17.11}$$

17.1 Differential Entropy

Once again proceeding analogous to the discrete case, we see that

$$I(X;Y) = \mathbb{E}\left[\log \frac{f(X,Y)}{f(X)f(Y)}\right] = \mathbb{E}\left[\log \frac{f(Y|X)}{f(Y)}\right] \tag{17.12}$$
$$= \mathbb{E}\left[\log \frac{1}{f(Y)}\right] - \mathbb{E}\left[\frac{1}{f(Y|X)}\right] \tag{17.13}$$

We would like to define the entropy of Y to be $\mathbb{E}\left[\log \frac{1}{f(Y)}\right]$, and the conditional entropy of Y given X to be $\mathbb{E}\left[\log \frac{1}{f(Y|X)}\right]$. But we have to deal with the fact that $f(Y), f(Y|X)$ are not probabilities. The intuition that entropy is the number of bits required to specify an RV breaks down, because we require infinite bits to specify a continuous RV. But it is still convenient to have a measure for CRVs. This motivates the definitions we'd like to have

$$h(Y) \triangleq \mathbb{E}\left[\log \frac{1}{f(Y)}\right] \tag{17.14}$$
$$h(Y|X) \triangleq \mathbb{E}\left[\log \frac{1}{f(Y|X)}\right] \tag{17.15}$$

Remark 17.1. *There are similarities and dissimilarities between $H(X)$ and $h(X)$.*

1. *$H(X) \geq 0$ for any DRV X, but $h(X)$ need not be a nonnegative quantity for any CRV, as probability density could be any positive number.*

2. *$H(X)$ is label-invariant, but $h(X)$ depends on the label. For example, let $Y = aX$ for a scalar a. For discrete RVs, $H(Y) = H(X)$. However, for continuous RVs, we know that $f_Y(y) = \frac{1}{|a|} f_X\left(\frac{y}{a}\right)$. Therefore*

$$h(Y) = \mathbb{E}\left[\log \frac{1}{f_Y(y)}\right] = \mathbb{E}\left[\log \frac{|a|}{f_X\left(\frac{Y}{a}\right)}\right] \tag{17.16}$$

$$= \log|a| + \mathbb{E}\left[\log \frac{1}{f_X\left(\frac{Y}{a}\right)}\right] \tag{17.17}$$

$$= \log|a| + h(X) \tag{17.18}$$

Therefore $h(aX) = h(X) + \log a$.

In the vector case, if $Y = Ax$, $h(Ax) = \log|\,|A|\,| + h(x)$.

17.2 Differential entropy of popular distributions

17.2.1 Uniform

Let $X \sim Unif([0, a])$.

$$h(X) = \mathbb{E}\left[\log \frac{1}{f(X)}\right] = \mathbb{E}\left[\log \frac{1}{a}\right] = \log a. \tag{17.19}$$

Note that for $a < 1, \log a < 0$ and $h(X) < 0$.

For $a = 1, h(X) = 0$. We reconcile this in terms of physical intuition by thinking of $2^{h(X)} = 2^{\log a} = a$ as the meaningful quantity: this is the volume of the support set of the RV.

We can relate differential entropy to its discrete version as well; define X^Δ as above, then

$$H(X^\Delta) = -\sum_i p_i \log p_i \tag{17.20}$$

$$= -\sum_i f(x_i)\Delta \log(f(x_i)\Delta) \tag{17.21}$$

$$= -\sum_i \Delta f(x_i) \log f(x_i) - \sum_i \Delta \log \Delta \tag{17.22}$$

and as $\Delta \to 0$, this goes to the Riemann sum, i.e.

$$H(X^\Delta) \to -\int f(x) \log f(x)\, \mathrm{d}x - \Delta \log \Delta \tag{17.23}$$

and so

Theorem 17.2.

$$\lim_{\Delta \to 0} H(X^{\Delta}) + \Delta \log \Delta = h(f) = h(X). \tag{17.24}$$

The entropy of an $n-$bit quantization of a CRV $X \approx h(X) + n$. If $X \sim Unif[0,1], h = 0, H^{\Delta} = n$.

17.2.2 Normal

Let $X \sim \mathcal{N}(0,1)$.

$$\log_2 \frac{1}{f_X(x)} = \frac{1}{2}\log_2(2\pi) + \frac{x^2}{2}\log_2 e \tag{17.25}$$

Therefore

$$h(X) = \frac{1}{2}\log(2\pi) + \frac{1}{2}\log_2 e\mathbb{E}[X^2] \tag{17.26}$$

$$= \frac{1}{2}\log_2(2\pi e). \tag{17.27}$$

Transformation rules tell us that if $Z \sim \mathcal{N}(\mu, \sigma^2)$, then

$$h(Z) = \frac{1}{2}\log 2\pi e\sigma^2 \tag{17.28}$$

Theorem 17.3. *For a constant a, $I(aX;Y) = I(X;Y)$.*

Proof.

$$I(aX;Y) = h(aX) - h(aX|Y). \tag{17.29}$$

We know $h(aX) = h(X) + \log|a|$. Similarly,

$$h(aX|Y) = \int_y h(aX|Y=y) f_Y(y)\,\mathrm{d}y \tag{17.30}$$

$$= \int_y \left(h(X|Y=y) + \log|a|\right) f_Y(y)\,\mathrm{d}y \tag{17.31}$$

$$= h(X|Y) + \log|a|, \tag{17.32}$$

and so

$$I(aX;Y) = h(X) + \log|a| - h(X|Y) - \log|a| = I(X;Y) \tag{17.33}$$

□

17.3 Properties of differential entropy

1. The chain rule works the same way as for DRVs (exercise!)

2. For relative entropy, we naturally want to choose

$$D(f \parallel g) \triangleq \mathbb{E}_{X\sim f}\left[\log \frac{1}{g(x)}\right] \tag{17.34}$$

   One of the most important properties of discrete relative entropy is its non-negativity. We'd like to know if it holds in the continuous world. According to C&T p252, it does!

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 18: Distributed Source Coding, Continuous RVs

Lecturer: Kannan Ramchandran *10 November* *Aditya Sengupta*

18.1 Distributed Source Coding

To motivate the idea of distributed source coding, consider the problem of compressing encrypted data. We start with some message X, which we compress down to $H(X)$ bits. Then, we encrypt it using a cryptographic key K. The bits are still $H(X)$ after this.

What if we did this in reverse? Suppose we first encrypt X using the key K, and then compress the resulting message Y. We claim this can still be compressed down to $H(X)$ bits. How does this work?

Slepian and Wolf considered this problem in 1973, as the problem of *source coding with side information* (SCSI). X, K are correlated sources, and K is available only to the decoder. They showed that if the statistical correlation between X and K are known, there is no loss of performance over the case in which K is known at the encoder. More concretely, if maximal rates R_K, R_X are achievable for compressing K and X separately, then the joint rate (R_K, R_X) is achievable simultaneously.

Example 18.1. Let X, K be length-3 binary data where all strings are equally likely, with the condition that the Hamming distance between X and K is at most 1 (i.e. X and K differ by at most one bit).

Given that K is known at both ends, we can compress X down to two bits. We can do this by considering $X \oplus K$ and noting that this can only be $000, 100, 010, 001$. Indexing these four possibilities gives us X compressed in two bits.

From the Slepian-Wolf theorem, we know that we can compress X to two bits even if we don't know K at the encoder. We can show this is achievable as follows: suppose $X \in \{000, 111\}$.

- If $X = 000$, then $K \in \{000, 001, 010, 100\}$.
- If $X = 111$, then $K \in \{111, 110, 101, 011\}$.

These two sets are mutually exclusive and exhaustive. Therefore, the encoder does not have to send any information that would help the decoder distinguish between 000 and 111, as the key will help it do this anyway. We can use the same codeword for $X = 000$ and $X = 111$.

Partition $\mathbb{F}_2^3$ into four cosets like this:

$$\{000, 111\} \to 00 \tag{18.1}$$
$$\{001, 110\} \to 01 \tag{18.2}$$
$$\{010, 101\} \to 10 \tag{18.3}$$
$$\{100, 011\} \to 11 \tag{18.4}$$

The index of the coset is called the *syndrome* of the source symbol. The encoder sends the index of the coset containing X, and the decoder finds a codeword in the given coset that is closest to K.

□

More generally, we partition the space of possible source symbols into smaller circles. The encoder sends information specifying which circle we look in, and the decoder looks within that circle and uses the key to decode the correct source symbol. The main idea here is that the key can be used to make a joint decoder and decrypter.

18.2 Differential Entropy Properties

One of the most important properties of relative entropy in the discrete case is its non-negativity. This still holds in the continuous case:

Theorem 18.1. *(C&T p252): $D(f \parallel g) \geq 0$.*

Proof. Similar to the discrete setting, let f, g be the two PDFs and let S be the support set of f.

$$D(f \parallel g) = \mathbb{E}_{X\sim f}\left[\log \frac{f(x)}{g(x)}\right] \tag{18.5}$$
$$= \mathbb{E}_{X\sim f}\left[-\log \frac{g(x)}{f(x)}\right] \tag{18.6}$$
$$\geq -\log \mathbb{E}_{X\sim f}\left[\frac{g(x)}{f(x)}\right] \text{ (Jensen's)} \tag{18.7}$$
$$= -\log\left[\int_{x\in S} f(x)\frac{g(x)}{f(x)}\,\mathrm{d}x\right] \tag{18.8}$$
$$= -\log\left[\int_{x\in S} g(x)\,\mathrm{d}x\right] \tag{18.9}$$
$$\geq 0 \tag{18.10}$$

The argument of the integral is less than or equal to 1, so its negative log must be nonnegative. □

Corollary 18.2. *For CRVs X and Y,*

$$I(X;Y) \geq 0. \tag{18.11}$$

Proof.

$$I(X;Y) = D(f_{XY}(x,y) \parallel f_X(x)f_Y(y)) \geq 0 \tag{18.12}$$

□

Corollary 18.3. *For CRVs X and Y,*

$$h(X|Y) \leq h(X), \tag{18.13}$$

and $h(X|Y) = h(X)$ iff $X \perp\!\!\!\perp Y$.

Theorem 18.4. *$h(X)$ is a concave function in X.*

Proof. Exercise. □

18.3 Entropy Maximization

We have seen that if an RV X has support on $[K] = \{1, \ldots, K\}$, and $X \sim p$, then the discrete distribution p maximizing $H(X)$ is the uniform distribution, i.e. $\mathbb{P}(X = i) = \frac{1}{K}$ for all $1 \leq i \leq K$. We can prove this as a theorem:

Theorem 18.5. *For a given alphabet, the uniform distribution achieves maximum entropy.*

We have seen this proof before, but we repeat it so that we have a proof technique we can extend to the continuous setting.

Proof. Let U be the uniform distribution on $[K]$. Let p be an arbitrary distribution on X.

$$D(p \parallel U) \geq 0, X \sim p \tag{18.14}$$

$$D(p \parallel U) = \sum_x p(x) \log \frac{p(x)}{U(x)} \tag{18.15}$$

$$= \sum_x p(x) \log p(x) + \sum_x p(x) \log \frac{1}{U(x)} \tag{18.16}$$

$$= -H(X) - \sum_x \log[U(x)] \cdot p(x) \tag{18.17}$$

$$= -H(X) - \log[U(x)] \underbrace{\sum_x p(x)}_{1} \tag{18.18}$$

$$= -H(X) - \underbrace{\sum_x U(x) \log U(x)}_{-H(U)} \tag{18.19}$$

$$= -H(X) + H(U). \tag{18.20}$$

Therefore

$$D(p \parallel U) = H(U) - H(X) \geq 0, \tag{18.21}$$

and so

$$H(U) \geq H(X) \tag{18.22}$$

□

Now, we can do the analogous maximization in the continuous case. However, we need to add a constraint: over all distributions on the reals, $\max_f h(X) \to \infty$ (for example, if we take $X \sim U[0, a]$, then $h(X) = \log a$ which blows up as $a \to \infty$.)

Therefore, the analogous problem is

$$\max_f h(X) \text{ s.t.} \mathbb{E}[X^2] \leq \alpha. \tag{18.23}$$

Theorem 18.6. *The Gaussian pdf achieves maximal differential entropy in 18.23 subject to the second moment constraint.*

Proof. This follows similarly to the discrete case and the uniform distribution. Note that this is essentially "guess-and-check": to optimize without knowing up front that the Gaussian is the answer, we could use Lagrange multipliers, but these become unwieldy and difficult.

Let $\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$, the pdf of the standard normal, and let f be an arbitrary pdf on X.

$$D(f \parallel \phi) = \mathbb{E}_{X \sim f}\left[\log \frac{f(x)}{\phi(x)}\right] \tag{18.24}$$

$$= \mathbb{E}_{X \sim f}[\log f(x)] + \mathbb{E}_{X \sim f} \log \frac{1}{\phi(X)} \tag{18.25}$$

$$\tag{18.26}$$

The first term is just $-h(X)$. Looking at only the second term (and integrating implicitly over all the reals)

$$\mathbb{E}_{X \sim f}\left[\log \frac{1}{\phi(x)}\right] = \int f(x) \log \frac{1}{\phi(x)} \, \mathrm{d}x \tag{18.27}$$

$$= \int f(x) \left[\log \sqrt{2\pi} + (\log_2 e) \frac{x^2}{2}\right] \mathrm{d}x \tag{18.28}$$

$$= \frac{1}{2} \log(2\pi) \int f(x) \, \mathrm{d}x + \int f(x) (\log_2 e) \frac{x^2}{2} \, \mathrm{d}x \tag{18.29}$$

$$= \frac{1}{2} \log(2\pi) + \frac{\log_2 e}{2} \underbrace{\mathbb{E}[X^2]}_{\alpha} \tag{18.30}$$

Since we make the constraint an equality (force $\mathbb{E}X \sim f[X^2] = \alpha$ to avoid any slack), we can say that

$$\mathbb{E}_{X\sim f}[X^2] = \mathbb{E}_{X\sim \phi}[X^2], \tag{18.31}$$

and since $\mathbb{E}[\log \frac{1}{\phi(x)}]$ depends only on $\mathbb{E}[X^2]$, in the original relative entropy statement we can replace $X \sim f$ with $X \sim \phi$, and so

$$D(f \parallel g) = -h(X) + h(X_G) \geq 0. \tag{18.32}$$

Therefore $h(X_G) \geq h(X)$ as desired. □

18.4 Gaussian Channel Capacity Formulation

Recall that the Gaussian channel problem was to maximize capacity subject to a power constraint:

$$C = \max_{p(x)} I(X;Y) \tag{18.33}$$

$$\text{s.t.}\mathbb{E}[X^2] \leq P, \tag{18.34}$$

where $Y = X + Z$ where $Z \sim \text{i.i.d.}\mathcal{N}(0, \sigma^2)$.

Notice that the power constraint is exactly the second moment constraint we just saw. It is equivalent to require that $\mathbb{E}[X^2] = P$. Consider the objective:

$$I(X;Y) = h(Y) - h(Y|X) \tag{18.35}$$

$$= h(Y) - h(X + Z|X) \tag{18.36}$$

$$= h(Y) - h(Z|X) \tag{18.37}$$

$$= h(Y) - h(Z), \tag{18.38}$$

where the last step follows because $Z \perp\!\!\!\perp X$. $h(Z)$ is fixed, so an equivalent problem is

$$\max_f h(Y) \text{s.t.} \mathbb{E}[X^2] = P \tag{18.39}$$

or, writing $Y = X + Z$ and using the independence of X and Z again,

$$\max_f h(X + Z) \text{s.t.} \mathbb{E}[(X + Z)^2] = P + \sigma^2 \tag{18.40}$$

From the derivation of the entropy-maximizing continuous distribution, the entropy of $X + Z$ subject to a second-moment constraint being maximal implies that $X + Z$ is Gaussian. We know that Z is Gaussian as well, so X has to be Gaussian. Using linearity of independent variance, we find that $X \sim \mathcal{N}(0, P)$. Therefore

$$I(X;Y) = h(X_G + Z) - h(Z) \tag{18.41}$$

$$= \frac{1}{2}\log[2\pi e(P + \sigma^2)] - \frac{1}{2}\log[2\pi e\sigma^2] \tag{18.42}$$

$$= \frac{1}{2}\log\left(1 + \frac{P}{\sigma^2}\right), \tag{18.43}$$

which was Shannon's original formula.

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 19: Maximum Entropy Principle, Supervised Learning

Lecturer: Kannan Ramchandran *12 November* *Aditya Sengupta*

Let X be a continuous RV with differential entropy $h(X)$. Let $\hat{X}$ be an estimate of X and let $\mathbb{E}[(X-\hat{X})^2]$ be the expected MSE.

Theorem 19.1. *The MSE is lower-bounded by*

$$\mathbb{E}[(X-\hat{X})^2] \geq \frac{1}{2\pi e} 2^{2h(X)} \tag{19.1}$$

with equality iff X is Gaussian and $\hat{X}$ is the mean of X.

Proof. Let $\hat{X}$ be any estimator of X.

$$\begin{aligned}
\mathbb{E}[(X-\hat{X})^2] &\geq \min_{\hat{X}} \mathbb{E}[(X-\hat{X})^2] && (19.2)\\
&= \mathbb{E}[(X-\mathbb{E}[X])^2] && (19.3)\\
&= \text{var}(X) && (19.4)\\
&\geq \frac{1}{2\pi e} 2^{2h(X)} && (19.5)
\end{aligned}$$

If $\text{var}(X) = \sigma^2$, then for any RV X with that variance we know that $h(X) \leq h(X_G) = \frac{1}{2}\log\big(2\pi e \sigma^2\big)$. Therefore

$$\sigma^2 \geq \frac{2^{2h(X)}}{2\pi e} \tag{19.6}$$

□

19.1 The principle of maximum entropy

Given some constraints on an underlying distribution, we should base our decision on the distribution that has the largest entropy. We can use this principle for both supervised and unsupervised tasks. However, we have access to only a few samples, which are not enough to learn a high-dimensional probability distribution for X.

The idea for how to resolve this is we want to limit our search to a candidate set: we want to find $\max_{p_X \in \Gamma} H(X)$. We look at the exponential family:

$$\Gamma = \{p_X \mid \mathbb{E}[\gamma_i(X)] = \alpha_i; i = 1, 2, \ldots, m\} \tag{19.7}$$

Let p^* be the max-ent distribution over Λ and let q be any distribution. Using the same decomposition as last time,

$$0 \leq D(q \parallel p^*) = -H(X) + \mathbb{E}_{X\sim q}\left[\log \frac{1}{p(X)}\right]. \tag{19.8}$$

If we can show that $\mathbb{E}_{X\sim q}\left[\log \frac{1}{p(X)}\right] = \mathbb{E}_{X\sim p^*}\left[\log \frac{1}{p(X)}\right]$ then we can conclude that $H(X) \leq H(X^*)$

To do this, we can pick a $p^*(x)$ such that $\log \frac{1}{p^*(x)}$ has the exact same functional form as the constraint set Γ.

(some stuff skipped)

Theorem 19.2. *If for coefficients $\lambda_0, \ldots, \lambda_m$, p^* defined as follows satisfies the constraints given for Λ, then $p^* \in \Lambda$ is the max-ent distribution.*

$$p^*(x) = \exp\{\lambda_0\} \tag{19.9}$$

some stuff here

19.2 Supervised Learning

Suppose we have a feature vector $\vec{X} = (X_1, \ldots, X_p)$ and want to predict a target variable Y. If $Y \in \mathbb{R}$ we have a regression problem; if Y is discrete we have a classification problem.

A sensible extension of the max-ent principle to the supervised setting is

$$\max_{p_{\vec{X},Y}\in\Gamma} H(Y|\vec{X}) \tag{19.10}$$

For concreteness, let's look at logistic regression. This part of the lecture was essentially a later homework problem, so I'll just present that. The problem required that we show that maximizing the conditional entropy is equivalent to solving a logistic regression problem (with constraints on the first and second order moments).

We apply the maximum entropy principle. We have constraints on $\mathbb{E}[X_iX_j]$ and on $\mathbb{E}[X_iY]$; we only want to consider the latter, as this is a conditional density and so the former terms can be put into the normalizing constant. The optimal distribution has the form

$$p^*(y|x) = \exp\left(\lambda_0(x) + \sum_i \lambda_1(x_i)\right) \tag{19.11}$$

Further, since we only have first-order dependence in the constraints that involve y, $\sum_i \lambda_1(x_i) = e^{\lambda^\intercal \mathbf{x}}$ (we have linear dependence). Therefore

$$p^*(y|x) = g(\mathbf{x})e^{y\lambda^\intercal \mathbf{x}} \tag{19.12}$$

To compute the normalizing constant, take $y = 0$ and $y = 1$ and require that they sum to 1:

$$g(\mathbf{x})\left(1 + e^{\lambda^\intercal \mathbf{x}}\right) = 1 \tag{19.13}$$

Therefore, the optimal distribution is

$$p^*(y|x) = \frac{e^{y\lambda^\intercal x}}{1 + e^{\lambda^\intercal x}} \tag{19.14}$$

EE 229A: Information and Coding Theory **Fall 2020**

Lecture 20: Fisher Information and the Cramer-Rao Bound

Lecturer: Kannan Ramchandran *17 November* *Aditya Sengupta*

A standard problem in statistical estimation is to determine the parameter of a distribution from a sample of data, e.g. $X_1, \dots, X_n \sim \mathbb{N}(\theta, 1)$. The MMSE of θ is

$$\bar{X_n} = \frac{1}{n} \sum_i X_i. \tag{20.1}$$

Here, we'll give information-theoretic foundations to this.

Definition 20.1. *Let $\{f(x;\theta)\}$ denote an indexed family of densities, such that $f(x;\theta) \geq 0$ and $\int f(x;\theta)\mathrm{d}x = 1$ for all $\theta \in \Theta$, where Θ is the parameter set.*

Definition 20.2. *An estimate for θ given a sample size n is a function $T : X^n \to \Theta$.*

We want to ensure the estimator approximates the value of the parameter, i.e. $T(X^n) \approx \theta$.

Definition 20.3. *The bias of an estimator $T(X_1, \dots, X_n)$ for the parameter θ is defined as $\mathbb{E}_\theta[T(X_1, \dots, X_n) - \theta]$ (taking the expectation with respect to the density corresponding to the indexed θ.)*

The estimate is unbiased if $\mathbb{E}_\theta[T(\cdot) - \theta]$. An unbiased estimator is good, but not good enough. We also want a low estimation error, and this error should go to 0 as $n \to \infty$.

Definition 20.4. *An estimator $T(X_1, \dots, X_n)$ for θ is said to be consistent in probability if $T(X_1, \dots, X_n) \to \theta$ in probability as $n \to \infty$.*

We are also interested in the finite-n case:

Definition 20.5. *An estimator $T_1(X_1, \dots, X_n)$ is said to dominate another estimator $T_2(X_1, \dots, X_n)$ if for all $\theta \in \Theta$,*

$$\mathbb{E}[(T_1(X_1, \dots, X_n) - \theta)^2] \leq \mathbb{E}[(T_2(X_1, \dots, X_n) - \theta)^2] \tag{20.2}$$

This raises a natural question: is there a "best" estimator of θ that dominates every other estimator?

We derive the CRLB (Cramer-Rao lower bound) on the MSE of any unbiased estimator.

First, define the score function of a distribution:

Definition 20.6. *The score V is an RV defined as*

$$V = \frac{\partial}{\partial \theta} \ln f(x;\theta) = \frac{\frac{\partial}{\partial \theta} f(x;\theta)}{f(x;\theta)} \tag{20.3}$$

For brevity, we denote by $l(x;\theta)$ the log-likelihood $\ln f(x;\theta)$.

Example 20.1. Let $f(x;\theta) = \frac{1}{\sqrt{2\pi}} e^{-(x-\theta)^2/2}$. Then

$$V = l'(X;\theta) = \frac{\partial}{\partial\theta}[\ln \frac{1}{\sqrt{2\pi}} - \frac{(x-\theta)^2}{2}] \tag{20.4}$$

$$= \frac{2(x-\theta)}{2} = x - \theta. \tag{20.5}$$

□

We can find the expectation of the score:

$$\mathbb{E}[V] = \mathbb{E}_\theta[l'(x;\theta)] = \mathbb{E}_\theta\left[\frac{f'(x;\theta)}{f(x;\theta)}\right] \tag{20.6}$$

$$= \int \frac{f'(x;\theta)}{f(x;\theta)} f(x;\theta)\mathrm{d}x \tag{20.7}$$

$$= \frac{\partial}{\partial\theta}\int f(x;\theta)\mathrm{d}x \tag{20.8}$$

$$= \frac{\partial}{\partial\theta} 1 = 0. \tag{20.9}$$

(note that we can't always swap the order of integration and differentiation, but just ignore that).

Therefore we see $\mathbb{E}[V^2] = \operatorname{var} V$.

Definition 20.7. *The Fisher information $J(\theta)$ is the variance of the score V.*

$$J(\theta) = \mathbb{E}_\theta\left[\left(\frac{\partial}{\partial\theta}\ln f(x;\theta)\right)^2\right] \tag{20.10}$$

A useful property is $J(\theta) = -\mathbb{E}_\theta[l''(x;\theta)]$, which you can show using the chain rule and quotient rule.

We can interpret l'' as the curvature of the log-likelihood.

If we consider a sample of n RVs $X_1, \ldots, X_n \sim f(x;\theta)$, we can show that

$$V(X_1, \ldots, X_n) = \sum_{i=1}^n V(X_i), \tag{20.11}$$

and

$$J_n(\theta) = \mathbb{E}_\theta\{l'(x^n;\theta)^2\} = \mathbb{E}_\theta\{\left[\sum_{i=1}^n V(X_i)\right]^2\}. \tag{20.12}$$

Therefore $J_n(\theta) = nJ(\theta)$. In other words, the Fisher information in a random sample of size n is simply n times the Fisher information.

Finally, we can show the CRLB:

Theorem 20.1. *For any unbiased estimator $T(X)$ of a parameter θ, then* $\operatorname{var} T \geq \frac{1}{J(\theta)}$.

For example, we know that the sample mean of n Gaussians with mean θ and variance σ^2 is distributed according to $\mathcal{N}\left(\theta, \frac{\sigma^2}{n}\right)$.

We can show this estimator meets the CRLB with equality:

$$J(\theta) = \mathbb{E}[V^2] = \mathbb{E}\left[\left(\frac{x_i - \theta}{\sigma^2}\right)^2\right] = \frac{1}{\sigma^2} \tag{20.13}$$

$$J_n(\theta) = nJ(\theta) = \frac{n}{\sigma^2}. \tag{20.14}$$

Therefore $\operatorname{var} T = \frac{1}{J_n(\theta)}$.

Therefore $\bar{X}_n$ is the MVUE (minimum-variance unbiased estimator), since it satisfies the CRLB with equality. This means it is also an efficient estimator.

Proof. (of CRLB) Let V be the score function and let T be any estimator. By the Cauchy-Schartz inequality,

$$\left[\mathbb{E}_\theta(V - \mathbb{E}_\theta V)(T - \mathbb{E}_\theta T)\right]^2 \leq \mathbb{E}_\theta(V - \mathbb{E}_\theta V)^2 \cdot \mathbb{E}_\theta(T - E_\theta T)^2. \tag{20.15}$$

Since T is unbiased, $E_\theta T = 0$. We also know $E_\theta V = 0$, and $\operatorname{var} V = J(\theta)$ definitionally. Therefore

$$\left[\mathbb{E}_\theta(V \cdot T)\right]^2 \leq J(\theta) \operatorname{var}_\theta(T) \tag{20.16}$$

Further,

$$\mathbb{E}_\theta VT = \int \frac{\frac{\partial}{\partial\theta} f(x;\theta)}{f(x;\theta)} T(x) f(x;\theta) \mathrm{d}x \tag{20.17}$$

$$= \int \frac{\partial}{\partial\theta} f(x;\theta) T(x) dx \tag{20.18}$$

$$= \frac{\partial}{\partial\theta} \int f(x;\theta) T(x) dx \tag{20.19}$$

$$= \frac{\partial}{\partial\theta} \mathbb{E}_\theta T \tag{20.20}$$

$$= \frac{\partial}{\partial\theta} \mathbb{E}_\theta \theta \tag{20.21}$$

$$= 1 \tag{20.22}$$

Therefore

$$1 \leq J(\theta) \operatorname{var}_\theta T \tag{20.23}$$

$$\operatorname{var} T \geq \frac{1}{J(\theta)}. \tag{20.24}$$

□

Using the same arguments, we can show that for any estimator, biased or unbiased,

$$\mathbb{E}[(T-\theta)^2] \geq \frac{1 + b_T'(\theta)}{J(\theta)} + b_T^2(\theta), \tag{20.25}$$

where $b_T(\theta) = \mathbb{E}[T - \theta]$.

We can generalize the concept of Fisher information to the multi-parameter setting, in which we define the Fisher information matrix $J(\theta)$ to be

$$J_{ij}(\theta) = \int f(x;\theta) \left[\frac{\partial}{\partial \theta_i} \ln f(x;\theta_i)\right] \left[\frac{\partial}{\partial \theta_j} \ln f(x;\theta_j)\right]. \tag{20.26}$$

In this case, the CRLB becomes

$$\Sigma \geq J^{-1}(\theta), \tag{20.27}$$

where Σ is the covariance matrix. This means $\Sigma - J^{-1}(\theta)$ is positive semidefinite.

Note: there were two more lectures after this one, both of which were guest lectures: the first by TA Avishek Ghosh and the second by postdoc Amirali Aghazadeh.