Information Theory, Part II

Manan Shah
manan.shah.777@gmail.com
The Harker School

March 21, 2017

This document contains lecture notes from Harker's Advanced Topics in Mathematics class in Information Theory II, taught by Dr. Anuradha Aiyer. This course is the second part of a two part offering that explores the basic concepts of Information Theory, as initially described by Claude Elwood Shannon at Bell Labs in 1948. These notes were taken using TeXShop and LaTeX2ϵ and will be updated for each class. The reader is advised to note any errata at the source control repository https://github.com/mananshah99/infotheory.

Contents

1 Unit 1: Gambling

We'll discuss the duality between the growth rate of investment (i.e. a horse race) and the entropy rate of the horse race and how the side information's financial value is tied to mutual information.

Definition 1 (Horse Race)**.** We have m horses in a race in which the ith horse wins with probability p_i. If horse i wins, the payoff is o_i for 1[1]. We'll assume that the gambler invests his wealth across all horses and doesn't hold on to any of his money. Specifically, b_i is the fraction of wealth invested in horse i where $b_i \geq 0$ and $\sum b_i = 1$. If horse i wins, the gambler wins $o_i b_i$; this case occurs with probability p_i. The wealth at the end of the race is a random variable which we will attempt to maximize.

1.1 Repeated Gambling

Define S_n as the total growth in the gambler's wealth after n races. We have

$$S_n = \prod_{j=1}^{n} S(X_j) \qquad \text{and} \qquad S(X_j) = b(X_j)o(X_j)$$

with X representing the horse that wins (this changes between races). Here, $S(X_i)$ represents the factor by which the gambler's wealth grows. We can define the doubling rate of a race W as

$$E(\log S(X)) = \sum_{k=1}^{m} p_k \log(b_k o_k) = W(b,p)$$

Theorem 1. Let race outcomes $X_1, X_2, X_3, \ldots, X_n$ be identically and independently distributed $\sim p(x)$. The wealth of a gambler using betting strategy b grows exponentially at the rate $W(b,p)$ such that $S_n = 2^{nW(b,p)}$.

Proof. Functions of independent random variables are also independent, so $\log S(X_1), \ldots \log S(X_n)$ are i.i.d. From our earlier definition of S_n we have

$$\frac{1}{n}\log S_n = \frac{1}{n}\sum \log S(X_i)$$

By the weak law of large numbers[2], this equates to $E(\log S(X)) = W(b,p)$. So we can conclude that $S_n = 2^{nW(b,p)}$ and the proof is complete. So if to maximize S_n, we'll need to maximize W. □

Definition 2. The optimum doubling rate over all choices of b_i is

$$W^*(p) = \max_b W(b,p) = \max_{b\colon b_i \geq 0, \sum b_i = 1} \sum_i p_i \log b_i o_i$$

We must formally maximize $W(b,p)$ such that $\sum b_i = 1$. To do this, we'll apply Lagrange optimization. We have

$$J(b) = \sum p_i \log b_i o_i + \lambda \sum b_i$$

[1] There are two ways to describe a bet: either a for 1 or b to 1. The first notation indicates an exchange that happens prior to the race, and the latter indicates and exchange that happens post-race (although in both cases the horses are picked before the race). More concretely, a for 1 indicates that if one places \$1 on a particular horse before the race, the payoff is \$a iff the horse wins and \$0 if the horse loses. b to 1 indicates that one would pay \$1 after the race if a particular horse loses and win \$b if the horse wins. The equivalency between these scenarios is $b = a - 1$.

[2] See `http://mathworld.wolfram.com/WeakLawofLargeNumbers.html` for more information.

Taking the partial with respect to b_i and setting it equal to 0,

$$\frac{\partial J}{\partial b_i} = \frac{p_i}{b_i} + \lambda$$

where $i \in \{1 \dots m\}$. Solving for b_i as a function of p_i and λ, substituting the resulting value into the constraint $\sum b_i = 1$, and evaluating the differential expression with $\lambda = -1$ results in $b_i = p_i$. Technically, we'd have to take the second derivative to prove that this is a maximum; this verification is left to the reader.

Theorem 2. $W^* = \sum p_i \log o_i - H(p)$

Proof.

$$\begin{aligned} W(b,p) &= \sum p_i \log b_i o_i \\ &= p_i \log \left(\frac{b_i}{p_i} \times p_i o_i \right) \\ &= \sum p_i \log o_i - H(p) - D(p||b) \end{aligned}$$

The last term, D, is known as relative entropy. It has some of the same properties of entropy, one of them being that $D \geq 0$. So, $W(b,p) \leq \sum p_i \log o_i - H(p)$ with equality when $p = b$. □

1.2 Kullback–Liebler Divergence

The function D, known as the relative entropy or Kullback-Liebler Divergence, is a measure of distance[3] between two distributions. If p and q are the two distributions, then $D(p||q)$ is a measure of inefficiency of assuming q when the true distribution is p. The average code length for distribution p is $H(p)$, but if we were to use the code for q to encode p, then $H(p) + D(p||q)$ bits.

Definition 3 (Kullback-Liebler Divergence)**.** The KL divergence $D(p||q)$ is expressed as

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)} = E_x \left[\log \frac{p(x)}{q(x)} \right]$$

where $0 \log 0/q = 0$ and $p \log p/0 = \infty$. We can then write $I(X;Y) = \sum \sum p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$ which is simplified to $D(p(x,y)||p(x)p(Y))$

Example 1. $p(0) = 1 - r, q(0) = 1 - s, p(1) = r, q(1) = s$

$$D(p||q) = (1-r) \log \frac{1-r}{1-s} + r \log \frac{r}{s}$$

$$D(q||p) = (1-s) \log \frac{1-s}{1-r} + s \log \frac{s}{r}$$

Example 2. Consider a case with two horses where horse 1 wins with probability p_1 and horse 2 wins with p_2. Assume even odds (2-for-1). (a) What is the optimal bet? (b) Doubling rate? (c) Resulting wealth?

(a) The optimal bet is according to the probabilities of the horses, (b) The doubling rate is $1 - H(p)$, and the resulting wealth (c) is $2^{n(1-H(p))}$

[3] This isn't technically a measure of distance as it doesn't satisfy the triangle inequality

We further have that $W(b,p) = \sum p \log \frac{p_i}{r_i} - \sum p \log \frac{p}{b} = D(p||r) - D(p||b)$ where $r_i = 1/o_i$. The doubling rate is the difference between the distance of the bookie's estimates from the truth. The gambler only makes money when b is closer than r. When the odds are m-for-1, we have

$$W^*(p) = D(p||1/m) = \log m - H(p)$$

and $W^*(p) + H(p) = \log m$.

Example 3. Three horses run a race. A gambler offers 3-for-1 odds on each horse. Fair odds under the assumption that all horses are equally likely to win. $p = (1/2, 1/4, 1/4)$. (a) Expected wealth, (b) b^*, (c) W^*

(a) We have that $W(b) = \sum p_i \log b_i o_i = \sum p_i \log 3b$ since $o_i = 1/3$ due to fair odds. Therefore, $W(b) = \sum p_i \log 3 + \sum p_i \log b_i = \log 3 + \sum p_i \log b_i$. (b) $b^* = p = (1/2, 1/4/, 1/4)$ and (c) $W^* = W(b^*) - \log 3 - 3/2$. Note that we can solve (c) with the identity discussed above.

1.3 The Value of Side Information

One measure is the increase in the doubling rate based on the information. We'll connect this increase with mutual information (as we connected W^* with KL divergence and entropy before). Define $X \in \{1, 2, \dots m\}$ as the horse betting space, $p(x)$ as the probabilities associated with $1 \to m$, $o(x)$ for 1 odds, and y as the side information. Furthermore, we have $\sum_x b(x|y)$ as the conditional betting depending on side information y and $b(x|y)$ as the proportion of wealth bet on horse x when y is observed. Based on these definitions, we have

$$W^*(X) = \max_{b(x)} \sum_x p(x) \log b(x) o(x)$$

and given our side information,

$$W^*(X|Y) = \max_{b(x|y)} \sum_{x,y} p(x,y) \log b(x|y) o(x)$$

so we have

$$\Delta W = W^*(X|Y) - W(X)$$

Theorem 3. The increase doubling rate ΔW due to side information Y for a horse race X is $\Delta W = I(X;Y)$.

Proof. We have that $b^*(x|y) = p(x|y)$. Since $W^*(X|Y) = \max_{b(x|y)} E(\log S)$[4]. This equates to $\max_{b(x|y)} \sum p(x,y) \log[o(x) b(x|y)]$. So, we have that

$$W^*(X|Y) = \sum p(x,y) \log[p(x) p(x|y)] = \sum p(x) \log o(x) - H(X|Y)$$

Without side information $W^* = \sum p(x) \log o(x) - H(X)$, so we have $\Delta W = \sum p(x) \log o(x) - H(X|Y) - [\sum p(x) \log o(x) - H(X)]$. Finally, we have $\Delta W = H(X) - H(X|Y) = I(X;Y)$. □

Example 4. Given a three horse race $p = (1/2, 1/4, 1/4)$ with odds with respect to the false distribution $r_1, r_2, r_3 = (1/4, 1/4, 1/2)$ and $o_1, o_2, o_3 = (4, 4, 2)$[5]. Find (a) the entropy of the race and (b) (b_1, b_2, b_3) such that compounded wealth $\to \infty$.

The entropy of the race is easily calculated as 3/2. It's intuitive that $b_i = o_i p_i$ so we have $(2, 1, 1/2)$, which we re-normalize to $(4/7, 2/7, 1/7)$. Our final $W = \sum p_i \log b_i o_i$.

[4] S was defined earlier as the aggregate wealth

[5] This is because $o_i = 1/r_i$ when determining the odds given the false distribution. "Fair odds" are defined such that $\sum 1/o_i = 1$

Example 5. Let the distribution be (p_1, p_2, p_3) with odds $o = (1, 1, 1)$ and wealth proportions $b = (b_1, b_2, b_3)$. $S_n \to 0$ exponentially. (a) Find the exponent, (b) b^*, and (c) What p causes $S_n \to 0$ at the fastest rate.

We always have that $b_i = \frac{p_i o_i}{\sum_i b_i}$, so we can write $b^* = p$. Furthermore, the exponent is simply the doubling rate $W = \sum p_i \log b_i o_i$, and the P that causes $S_n \to 0$ most quickly is the one that maximizes $H(p)$ or $p = (1/3, 1/3, 1/3)$.

Example 6. Now assume you have the most common form of side information, the past performance of the horses. If the results of each successive horse race are independent, then this additional information will not reveal anything new about the race. Suppose instead that the races are dependent. How might we calculate $W^*(X_k | X_{k-1}, X_{k-2}, ..., X_1)$?

At the end of of n races,

$$S_n = \prod_i S(X_i)$$

Therefore,

$$\begin{aligned}\frac{1}{n}\mathrm{E}\left[\log S_n\right] &= \frac{1}{n}\sum_{i=1}^{n}\mathrm{E}\left[\log S(X_i)\right] \\ &= \frac{1}{n}\sum_{i=1}^{n}\left[\log m - H(X_i|...)\right] \\ W^* &= \log m - \frac{H(X_1, X_2, ..., X_n)}{n}\end{aligned}$$

The term $\frac{H(X_1, X_2, ..., X_n)}{n}$ can be thought of as the average entropy across all the races.

2 Unit 2: Statistics

2.1 Introduction

Statistics considers two types of studies: observational and experimental. In an experimental study, treatments are assigned to subjects; this is not the case in observational studies. The first part of this topic was covered with traditional statistics worksheets involving the definition of p-values and statistical tests (t, z, etc.) A diagram that connects these concepts is as follows:

Information Theory	Statistics	Machine Learning
Source	Observational Studies	Unsupervised Learning
Channel	Experimental Studies	Supervised Learning

We'll start by modeling the source, which has a distribution Q and outputs the vector X. We will define a notion of X as too ridiculous to have come from Q. It is incorrect to define the ridiculousness criterion as "$\text{Prob}_Q(X)$ is small" as looking at one event whose probability will almost always be small is insufficient.

Definition 4. X is "too ridiculous" to have originated from source Q if and only if probability $\text{Prob}_Q(X$ and its entire subsequent tail) is sufficiently small[6]. That is to say, given X we can define

$$S_X = \{X' \mid \text{Prob}_Q(X') \leq \text{Prob}_Q(X)\}$$

[6] Define ϵ as in traditional proofs to quantify this

We then have that X is too ridiculous to have come from Q if the probability $\text{Prob}_Q(S_X)$ is sufficiently small.

Definition 5 (Confidence Interval)**.** Given X, identify all Qs it may have originated from. The confidence interval[7] is defined as

$$\{Q \mid X \text{ is not too ridiculous to have originated from } Q\}$$

Definition 6 (Hypothesis Test)**.** Given hypothesis Q, find all X that will allow for disproving Q. The rejection interval (or region) is defined as

$$\{X \mid X \text{ is too ridiculous to have come from } Q\}$$

2.2 Two Significant Theorems

We'll discuss two important theorems that define the notions of $\text{Prob}_Q(X)$ and $\text{Prob}_Q(S)$.

2.2.1 Prob$_Q(X)$

Assume that we are given a source Q that produces observed data outputs $X_1 \dots X_N$ (all abbreviated as the vector X) and that the values are identically and independently distributed. We will attempt to define the value $\text{Prob}_Q(X)$. If we were to histogram the vector X (with the y axis representing the frequency of occurrences of ξ in X and the x axis representing the discrete values $\xi_1 \dots \xi_k)$[8], each value ξ_i would have an associated frequency N_i. Call this histogram P_X with $N_1 + N_2 + \cdots + N_k = N$. We then have

$$\begin{aligned}\text{Prob}_Q(X) &= Q(X_1) \times Q(X_2) \times \cdots \times Q(X_N) \\ &= Q(\xi_1)^{N_1} \times Q(\xi_2)^{N_2} \times \cdots \times Q(\xi_k)^{N_k} \\ &= 2^{-[N_1 \log \frac{1}{Q(\xi_1)} + N_2 \log \frac{1}{Q(\xi_2)} + \cdots + N_k \log \frac{1}{Q(\xi_k)}]} \\ &= 2^{-N[P_X(\xi_1) \log \frac{1}{Q(\xi_1)} + \cdots + P_X(\xi_k) \log \frac{1}{Q(\xi_k)}]}\end{aligned}$$

where $Q(X_i)$ is the probability of seeing X_i in the distribution of Q. We can multiply each log term by $P_X(\xi_i)$ on the numerator and denominator to express the value as a function of the KL divergence and entropy. We therefore have the following result[9].

Theorem 4. $\text{Prob}_Q(X) = 2^{-N[D(P_X || Q) + H(P_X)]}$

2.2.2 Prob$_Q(S)$

We can begin by writing

$$\begin{aligned}\text{Prob}_Q(S) &= \sum_{X \in S} \text{Prob}_Q(X) \\ &= \sum_{X \in S} 2^{-N[D(P_X || Q) + H(P_X)]}\end{aligned}$$

[7] It's better to write this as a confidence region as opposed to a confidence interval if we're working in spaces of higher dimensionality than $\mathbb{R}^1$

[8] X comprises of discrete values that are represented by ξ

[9] We've only proved the result for a discrete i.i.d distribution, but it can be shown to be applicable to continuous distributions (with differential entropy). This result cannot, however, be extended to non-i.i.d distributions because of the first step

With Q representing the true distribution, we have a space of histograms $\{P_X \, \forall \, X \in S\}$. We want to identify the distribution P^* that is "closest" to Q. By the Pythagorean inequality (which we'll prove later), we can write the above expression as

$$\text{Prob}_Q(S) \leq \sum_{X \in S} 2^{-N[D(P_X||P^*)+D(P^*||Q)+H(P_X)]} = 2^{-ND(P^*||Q)} \sum_{X \in S} 2^{-N[D(P_X||P^*)+H(P_X)]}$$

which can be written as

$$2^{-ND(P^*||Q)} \sum_{X \in S} \text{Prob}_{P^*}(X) = 2^{-ND(P^*||Q)} \text{Prob}_{P^*}(S)$$

Since the probability term is less than or equal to one, we've therefore bounded $\text{Prob}_Q(S)$.

Theorem 5 (Sanov's Theorem)**.** $\text{Prob}_Q(S) \leq 2^{-ND(P^*||Q)}$

2.2.3 Proof of the Pythagorean Inequality

Theorem 6 (Pythagorean Inequality)**.** $D(P||Q) \geq D(P||P^*) + D(P^*||Q)$ where Q is the true distribution and P is the estimated distribution. P^* is the closest distribution in terms of KL divergence.

Proof. Consider any P in the convex vector space of histograms S. By the definition of a convex vector space,

$$P_\lambda = \lambda P + (1 - \lambda) P^*$$

If we let $\lambda = 0$, $P_\lambda = P^*$, but since we know that P^* is defined to be the minimum of all P in set S (according to the condition $D(P^*||Q) = \min_{P \in S} D(P||Q)$), we know that it is also the minimum of $D(P_\lambda||Q)$ along the path $P^* \to P$. So, $D_\lambda = D(P_\lambda||Q) = \sum P_\lambda \log P_\lambda / Q$. Therefore, $dD_\lambda / d\lambda$ as a function of λ is nonnegative at $\lambda = 0$ because as $\lambda \to 0$, the divergence decreases. Specifically,

$$\frac{dD_\lambda}{d\lambda} = \sum_{\text{bins}} \left[(P - P^*) + (P - P^*) \log \frac{P_\lambda}{Q} \right]$$

At $\lambda = 0$, $P_\lambda = P^*$. We also know that $\sum P(X) = \sum P^*(X) = 1$ by the property of a pdf. We can then write

$$\left. \frac{dD_\lambda}{d\lambda} \right|_{\lambda=0} = \sum [P(X) - P^*(X)] \log \frac{P^*}{Q}$$

This simplifies to $\sum P \log \frac{P^*}{Q} - \sum P^* \log \frac{P^*}{Q} \geq 0$. The first term can be written as $\sum P \log \frac{P^*}{P} \frac{P}{Q} - \sum P^* \log \frac{P^*}{Q}$ which completes the proof (as the initial point $\lambda = 0 \geq 0$ so the remainder of the function is). □

2.3 Connections to Statistics

From the definition of KL divergence, we can write $D[\text{Bernoulli}(p_1)||\text{Bernoulli}(p_2)]$ as

$$p_1 \log \frac{p_1}{p_2} + (1 - p_1) \log \frac{1 - p_1}{1 - p_2}$$

which is equivalent to[10]

$$\frac{1}{\ln 2} \left[p_1 \ln \frac{p_1}{p_2} + (1 - p_1) \ln \frac{1 - p_1}{1 - p_2} \right]$$

[10] Converting log to ln

A second order approximation of $\ln(1/1-\epsilon) \approx \epsilon + \epsilon^2/2$ allows us to write

$$\frac{1}{\ln 2}\left[p_1\left(\left(1-\frac{p_2}{p_1}\right)+\frac{1}{2}\left(1-\frac{p_1}{p_2}\right)^2\right)+(1-p_1)\left(\left(1-\frac{1-p_2}{1-p_1}\right)-\frac{1}{2}\left(1-\frac{1-p_2}{1-p_1}\right)^2\right)\right]$$

which simplifies to

$$\left(-\frac{1}{2\ln 2}\right)\frac{(p_1-p_2)^2}{p_1(1-p_1)}$$

What we've just obtained is an approximation to the KL divergence between two binomials. Furthermore, the KL divergence between Gaussian distributions $D[N(\mu_1,\sigma_1^2)||N(\mu_2,\sigma_2^2)]$ is

$$\frac{1}{\ln 2}\left[\ln\frac{\sigma_2}{\sigma_1}+\frac{1}{2}\left(\frac{\sigma_1^2}{\sigma_2^2}-1\right)+\frac{(\mu_1-\mu_2)^2}{2\sigma_2^2}\right]$$

In the special case that $\sigma_1=\sigma_2=\sigma$ we have

$$\frac{1}{2\ln 2}\frac{(\mu_1-\mu_2)^2}{\sigma^2}$$

Note for future reference that

$$D(f||q)=\int_x f\log\frac{f}{q}dx$$

2.3.1 Connection 1: Observation Studies for Proportion

Consider a binomial distribution with probability p originating from source Q (that outputs values $x_1, x_2, \ldots, x_N$). We've defined X is too ridiculous to have come from Q as $2^{-ND(P^*||Q)} < \epsilon$ (by extension of Sanov's theorem). But since P^* (our best guess of the distribution) is equivalent to Binomial($\hat{p}$)[11], we can write

$$2^{-N\left[\frac{1}{2\ln 2}\frac{(\hat{p}-p)^2}{\hat{p}(1-\hat{p})}\right]} < \epsilon$$

Which is equivalent to

$$\frac{1}{2\ln 2}\frac{(\hat{p}-p)^2}{(\hat{p}(1-\hat{p}))/N} > \log\frac{1}{\epsilon}$$

We can rewrite this as

$$\frac{(\hat{p}-p)^2}{(\hat{p}(1-\hat{p}))/N} > \ln\frac{1}{\epsilon^2}$$

Taking the square root of both sides, we have

$$\frac{|\hat{p}-p|}{\sqrt{\frac{\hat{p}(1-\hat{p})}{N}}} > \underbrace{\sqrt{\ln\frac{1}{\epsilon^2}}}_{\text{call this constant } C}$$

We can now express this in a more familiar form. In particular, given $X = \{P \mid P$ lies in the interval $\hat{p} \pm C\sqrt{\frac{\hat{p}(1-\hat{p})}{N}}\ \}$, then we have the hypothesis test that $\hat{p}$ is "too ridiculous" to have come from X if it lies in the rejection region $\{\ \frac{|\hat{p}-p|}{\sqrt{\frac{\hat{p}(1-\hat{p})}{N}}} > C\ \}$.

[11] That is, we make our best guess given the observed probability distribution

2.3.2 Connection 2: Observation Studies for Mean

Consider a normal distribution with mean μ and standard deviation σ originating from source Q (that outputs values $x_1, x_2, \ldots, x_N$). We've defined X is too ridiculous to have come from Q as $2^{-ND(P^*||Q)} < \epsilon$ (by extension of Sanov's theorem). But since P^* (our best guess of the distribution) is equivalent to $N(\bar{x}, \sigma^2)$, we can write

$$2^{-N\left[\frac{1}{2\ln 2}\frac{(\bar{x}-\mu)^2}{\sigma^2}\right]} < \epsilon$$

Which is equivalent to

$$\frac{(x-\mu)^2}{\sigma^2/N} > 2\ln 2 \log\frac{1}{\epsilon}$$

Taking the square root of both sides, we have

$$\frac{|\bar{x}-\mu|}{\sigma/\sqrt{N}} > \underbrace{\sqrt{\ln\frac{1}{\epsilon^2}}}_{\text{call this constant } C}$$

We can now express this in a more familiar form. In particular, the confidence interval given X from distribution Q is $\{\mu | \frac{|\bar{x}-\mu|}{\sigma/\sqrt{N}} \leq C\} = \{\mu \mid \mu \text{ lies in the interval } \bar{x} \pm C\sigma/\sqrt{N}\}$

2.3.3 Connection 3: Experimental Studies (General)

In an experimental study, we have inputs $x_1, x_2, \ldots, x_N$ which pass through a channel $Q(y|x)$ to produce outputs $y_1, y_2, \ldots, y_N$. Our job is to guess $Q(y|x)$ or refute $\hat{P}(y|x)$ that has been guessed by someone else. Given that X went into Q, Y is too ridiculous to have come out of Q by a certain criterion. To define this criterion, assume that $X = \langle x_1, x_2, \ldots, x_N\rangle$ can assume possible values $\xi_1, \xi_2, \ldots, \xi_k$. Now we segregate Y according to $\xi_1, \xi_2, \ldots, \xi_k$. The vector Y is split into $y_{\xi_1}, y_{\xi_2}, \ldots, y_{\xi_k}$ where each y_{ξ_i} represents the components of Y where the corresponding x have values equal to ξ_i. We can next segregate the true conditional distribution $Q(y|x)$ according to $\xi_1, \xi_2, \ldots, \xi_k$ (that is, $Q_{\xi_i}(y) = Q(y|x = \xi_i)$). We define multiple tails, one for each input value, where

$$\text{tail of } y_{\xi_i} = \{y'_{\xi_i} \mid \text{Prob}_{Q_{\xi_i}}(\text{empirical histogram of } y'_{\xi_i}) \leq \text{Prob}_{Q_{\xi_i}}(\text{empirical histogram of } y_{\xi_i})\}$$

Given that X went into Q and Y is too ridiculous to have come out of Q, we can write

$$\text{Prob}_{Q_{\xi_1}}(\text{tail of } y_{\xi_1}) \times \text{Prob}_{Q_{\xi_2}}(\text{tail of } y_{\xi_2}) \times \cdots \times \text{Prob}_{Q_{\xi_k}}(\text{tail of } y_{\xi_k}) < \epsilon$$

Recalling our earlier definition of $\text{Prob}_Q(\text{tail of } X) \leq 2^{-ND(P^*||Q)}$, we have that P^* is the closest histogram to Q among all the histograms you can draw of things in the tail of X. We can thus replace each of the probability terms with $2^{-N_i D(P^*_{\xi_i}||Q_{\xi_i})} \mid i \in 1\ldots k$.

2.3.4 Connection 3a: Experimental Studies for Proportion

In this case, we have $Q(y|x) \to Q_{\xi_1}(y) \sim \text{Binomial}(p_1)$ and $Q(y|x) \to Q_{\xi_2}(y) \sim \text{Binomial}(p_2)$. We can follow the same derivation as the previous steps (but keeping in mind the N_i terms) to obtain

$$\frac{(\hat{p_1}-p_1)^2}{(\hat{p_1}(1-\hat{p_1}))/N_1} + \frac{(\hat{p_2}-p_2)^2}{(\hat{p_2}(1-\hat{p_2}))/N_2} > \ln\frac{1}{\epsilon^2}$$

Moving the $\ln 1/\epsilon^2$ to the other side, we have an equation for an ellipse (if we graph with respect to the variables p_1 and p_2). All non rejected proportions are within the area of this ellipse, which is centered at $(\hat{p_1}, \hat{p_2})$ and has a horizontal axis radius of $\sqrt{\ln \frac{1}{\epsilon^2} \frac{\hat{p_1}(1-\hat{p_1})}{N_1}}$. We can obtain a confidence interval by setting $p_1 = p_2$, which allows us to draw two tangents to the ellipse. We can find the intersection points of the tangents with the ellipse, and the distance between these points defines the confidence interval.

To identify the confidence interval, consider two tangents $y - x = C_1$ and $y - x = C_2$. We need to find $C_2 - C_1$. Consider the equation of an ellipse

$$\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} = 1$$

Implicitly differentiating both sides yields

$$\frac{dy}{dx} = -\frac{b^2}{a^2}\left(\frac{x-h}{y-k}\right)$$

which we can equate to 1 to obtain

$$x - h = \pm\sqrt{\frac{a^4}{b^2 + a^2}}$$

Plugging this into the ellipse equation yields

$$y - k = \mp\frac{b^2}{a^2}(x-h)$$

We therefore have two pairs of x and y (one where x is $+$ and y is $-$, and vice-versa). We can then subtract these two to obtain $C_1 - C_2 = 2\sqrt{a^2 + b^2}$. Expressing this int terms of proportions yields the confidence interval

$$(\hat{p_1} - \hat{p_2}) \pm C\sqrt{\frac{\hat{p_1}(1-\hat{p_1})}{N_1} + \frac{\hat{p_2}(1-\hat{p_2})}{N_2}}$$

3 Unit 3: Kolmogorov Complexity

So far we have talked about descriptive complexity and related information theory to pdfs. Kolmogorov Complexity is a measure of algorithmic complexity, which consists of a set of subjective statements that a human or a compiler interprets. Kolmogorov Complexity is independent of compiler choice.

3.1 Turing Machine

A Turing machine is a device that converts input to output using a set of predefined rules. An example of a basic Turing machine is an FSM which receives input one bit at a time, and given its current state and input, writes something to the output or write something in its ”work tape,” or internal memory. The only stipulation is that the input has to be processed sequentially (right to left).

3.2 Definitions

$$x : \text{finite length binary string}$$
$$U : \text{universal computer}$$
$$l(x) : \text{length of string } x$$
$$U(p) : \text{output of computer} U \text{when presented with program } p$$

Definition 7. Kolmogorov Complexity Kolmogorov complexity $K_u(x)$ of a string x with respect to U is $K_U(x) = \min_{p:U(p)=x} l(p)$, that is, the length of the shortest program which produces output x when fed into universal computer U.

3.3 Examples

Example 7. The description of string 010101010101...01 is simply an alternating pattern of 0s and 1s.

Example 8. The description of string 0110101000001001111001101100...00001000 is the binary expansion of $\sqrt{2} - 1$

Example 9. Describing pseudo-random string 1101111001110101111101101111110...100111011 is slightly more complex. Let k be the number of 1s in the sequence. A table with all possible sequences with k 1s can be precomputed and stored in the universal computer. The index of the particular sequence can be precomputed and passed as the input of the program. Using this strategy, the minimum program length is $logn + nH\frac{k}{n}$.

3.4 A Theorem

Theorem 7. Universality of K.C. If U is a universal computer, then for any other computer, A,

$$K_u(x) = K_A(x) + C_A$$

for all strings $x \in 0, 1$ where C_A does not depend on x.

3.4.1 Proof

Assume we have P_A for computer A to print x.

- $A(P_A) = x$
- precede this program by a simulation program S_A which tells U to simulate A.
- computer U will then interpret instructions in the program for A, perform calc., print x.
- program for $U : p = S_A p_A$ with length $l(p) = l(S_A) + l(p_A) = C_A + l(p_A)$
- $K_U(x) = \min l(p) = \min_{A(p)=x} l(p_A) + C_A = K_A(x) + C_A$
- When the program size is large, the constant is deemed irrelevant, and from now on, we'll assume the Turing machine is universal.

3.5 Another Theorem

Theorem 8. $K(x|l(x)) \leq l(x) + c$

The reasoning is that if we know $l(x)$, the end of the program is clearly defined. A program for printing x is "Print the following l-bit sequence: $x_1x_2x_3x_{l(x)}$.

If $l(x)$ is not known, we need to find a way to let the Turing machine know that we have reached the end of the program. A simple way to do this is to add an additional symbol so that $K(x) \leq K(x|l(x)) + 2\log l(x) + c$.

3.5.1 Proof

Assume we have a Turing machine that accepts 01 as a comma. Let $l(x)$ be n. To describe $l(x)$ repeat every bit of n twice, then end with 01. If the length is 5, we pass 11001101. The length of the description of $l(x)$ using this method is $2\log n + 2$ bits. Another way to do this would be using recursion.

3.6 Yet Another Theorem

Theorem 9. Number of strings x with complexity $K(x) < k$ satisfies $|x \in 0, 1^* : K(x) < k| < 2^k$.

3.6.1 Yet Another Proof

There are $2^k - 1 < 2^k$ sequences with length $< k$.

3.7 Some Entropy Notation

Let's start by introducing some "new" definitions for entropy. We know that $H(p)$ is given by

$$H_0(p) = -p\log p - (1-p)\log(1-p).$$

Along the same lines, let's write the entropy of a mean of values x_i as follows:

$$H_0\left(\frac{1}{n}\sum x_i\right) = -\overline{x}_n \log \overline{x}_n - (1-\overline{x}_n)\log(1-\overline{x}_n).$$

This shouldn't be confused with the entropy of $\overline{x}$ across many trials; it's the entropy of Bern$(p = \overline{x}_n)$.

3.8 Lemma

As an intermediate step in another proof, we'd like to show that

$$\binom{n}{k} \leq 2^{nH_0(\frac{k}{n})}.$$

Stirling's approximation formula will help here:

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n.$$

Proof. We'll write $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ and use Stirling's approximation:

$$\begin{aligned}
\ln\binom{n}{k} &= \frac{1}{2}\ln(2\pi) + \frac{1}{2}\ln n + n\ln n - n \\
&\quad - \left[\frac{1}{2}\ln(2\pi) + \frac{1}{2}\ln k + k\ln k - k\right] \\
&\quad - \left[\frac{1}{2}\ln(2\pi) + \frac{1}{2}\ln(n-k) + (n-k)\ln(n-k) - (n-k)\right] \\
\frac{1}{n}\ln\binom{n}{k} &= \frac{1}{2n}\left(\ln\left[\frac{n}{k(n-k)}\right]\right) + \ln n - \left(\frac{k}{n}\right)\ln k - \left(\frac{n-k}{n}\right)\ln(n-k) - \frac{1}{2n}\ln(2\pi) \\
&= -\frac{k}{n}\ln\left(\frac{k}{n}\right) - \left(1-\frac{k}{n}\right)\ln\left(1-\frac{k}{n}\right) \\
&= H_0\left(\frac{k}{n}\right).
\end{aligned}$$

This above derivation isn't perfect, but the idea is that we're using Stirling's approximation to show this Lemma – nbd if the algebra above doesn't *entirely* work out.

3.9 Basic Kolmogorov Complexity Examples

Let's look at some concrete examples of $K_U(x)$. For a string of repeating zeros, the number of bits in the program is constant. Same for some other repeating pattern.

Example 10. The Mandelbrot set is a particular set of complex numbers that produces nice looking fractal boundaries. Turns out that the Kolmogorov complexity of this is also constant, very close to zero, showing how theoretical or visual complexity does not necessarily correlate with $K_U(x)$.

Example 11. If we have an integer n, $K(n) \leq \log n + c$, since it requires $\log n$ bits to represent the integer.

Example 12. Sequence of n bits where k of the bits are ones. Here, k can go from 0 to n, and another counter index i can go from 0 to $\binom{n}{k}$. So the length of the program is $\ell(p) = 2\log k + \log\binom{n}{k} + c$. Then, we can apply our previous Lemma for a new theorem!

Theorem 10. $K(x_1, x_2, \ldots, x_n \mid n) \leq nH_0\left(\frac{1}{n}\sum x_i\right) + 2\log k + c$.

Note: we can discuss the Kolmogorov complexity of integers in the same way of binary strings, because of example 11. For some integer n,

$$K(n) = \min_{p:U(p)>n} \ell(p).$$

3.10 Kolmogorov Complexity and Entropy

The expected value of a random sequence is close to Shannon entropy. Specifically, we'll prove that it satisfies a similar bound to Huffman. Just as we used the Kraft inequality to prove Huffman, we'll use the Kraft inequality to prove this.

Theorem 11. For any computer U,

$$\sum_{p:U(p) \text{ halts}} 2^{-\ell(p)} \leq 1.$$

This can be trivially proved once we realize that the set of all halting program forms a prefix free set.

Theorem 12. Let x_i be drawn i.i.d. according to a probability mass function $f(x)$ where $x \in X$ and X is a finite alphabet.

$$f(x^n) = \prod_{i=1}^{n} f(x_i)$$

Then there exists a constant c such that

$$H(X) \leq \frac{1}{n} \sum_{x^n} f(x^n) K(x^n|n) \leq H(X) + \frac{|x| \log n}{n} + \frac{c}{n}$$

In other words, for an n-length string drawn from an alphabet X, the average complexity of the string times the probability of seeing it is bounded by the entropy of the source and the entropy of a source plus some constant.

Proof. For the lower bound: All allowed programs satisfy the prefix-free property and thus their lengths satisfy the Kraft inequality. For each x^n, the shortest program p is assigned such that $U(p, n) = x^n$. By the source coding theorem:

$$\sum_{x^n} f(x^n) K(x^n|n) \geq H(x_1, x_2, \ldots, x_n) = nH(X)$$

Since $x_1, x_2, \ldots, x_n$ are independent.
For the upper bound: Look at a binary RV, X. Thus, $x_1, x_2, \ldots, x_n$ are i.i.d. $\sim$ Bernoulli(Q)

$$K(x_1, x_2, \ldots, x_n|n) \leq nH_0(\frac{1}{n} \sum x_i) + 2 \log n + c$$

$$\begin{aligned} E[K(x_1, x_2, \ldots, x_n|n)] &\leq nE[H_0(\frac{1}{n} \sum x_i)] + 2 \log n + c \\ &\leq nH_0(\frac{1}{n} \sum E[x_i]) + 2 \log n + c \\ &\leq nH_0(\theta) + 2 \log n + c \end{aligned}$$

Note that this is not a sufficient proof for the general case of any random variable X; however, the general proof is beyond the scope of this course.

Example 13. Let $x, y \in \{0, 1\}$. Argue that $K(x, y) \leq K(x) + K(y) + c$.
This is trivially true as the worst case of a program that outputs x and y is simply a program that outputs x followed by a program that outputs y. In the best case, if x and y are the same, then the program will simply output x twice.

Example 14. Argue that $K(n) \leq \log n + 2 \log \log n + c$.
To represent an integer, n, one needs $\log n$ bits to describe it. However, just describing the integer is not enough to uniquely decode the number. Instead, we need to transmit the number and the length of the number twice in order to to make it uniquely decodable. Take 30. In binary, this is 11110. We need to first append the length of the string "doubled up" along with some terminating sequence that delineates the length of the integer from the integer itself. The length of 30 is $\lceil \log 30 \rceil = 5$. The length of 5 is $\lceil \log 5 \rceil = 3$ Instead of representing this 5 using these three bits as 011, we will use six bits and instead write it as 001111. To show that the length part has stopped and the description of the number itself has begun, we will add a "01" between the length and number to mark the transition. Therefore, 30 would be described as *001111***01**11110. The italic portion represents the length of the string, the bolded is the divider, and the normal font text is the number itself.

Example 15. Argue that $K(n_1 + n_2) \leq K(n_1) + K(n_2) + c$
Given that the longer of the two strings n_1 and n_2 is k bits long, the addition operation cannot produce a string more that $k + 1$ bits long. The worst case of sending n_1 and then n_2 is send a $2k$ bit string while sending the result of the addition is $k + 1$ bits thus the sending the addition is necessarily smaller than sending the concatenation.

Example 16. Consider an n x n array x of 0's and 1's such that x has n^2 bits. Find $K(x|n)$ to the first order if you wish to transmit a horizontal line of x.
All you must do is transmit the row number, which at first order takes $K(x|n) \leq \log n + c$.

Example 17. Now we wish to transmit a a small square subregion of the array. What is the complexity? You must transmit the row number and column number of the top-left corner of the square along with the square length so the complexity will be: $K \leq 3(\log n + 2 \log \log n) + c$.

Example 18. Transmitting the union of two lines either horizontal or vertical: We must transmit the row/column number of each line so $K \leq 2 \log n + c$.

3.11 Incompressible Sequences

We now come to the problem of how easy or difficult it is to describe a number. Long sequences and large integers (such as the first n digits of π or the result of $2^{2^{2^2}}$), while perhaps long or unwieldy are relatively simple to calculate; therefore, they are easy to compress as well. From this, we conclude that the probability that a sequence can be compressed by more than k bits is no greater than 2^{-k}.

Theorem 13. Given that $X_1, X_2, \ldots, X_n \sim \text{Bernoulli}(\frac{1}{2})$, $P(K(X_1, X_2, \ldots, X_k|n) < n - k) < 2^{-k}$

Proof.

$$\begin{aligned}
P(K(X_1, X_2, \ldots, X_k|n) < n - k) \\
&= \sum_{x_1,x_2,\ldots,x_n : K(x_1,x_2,\ldots,x_n|n)<n-k} P(x_1, x_2, \ldots, x_n) \\
&= \sum_{x_1,x_2,\ldots,x_n : K(x_1,x_2,\ldots,x_n|n)<n-k} 2^{-n} \\
&= |\{x_1, x_2, \ldots, x_n : K(x_1, x_2, \ldots, x_n|n) < n - k\}| 2^{-n} \\
&< 2^{n-k} \times 2^{-n} \\
&= 2^{-k}
\end{aligned}$$

From this theorem we gather that most sequences have a complexity close to their length.

Example 19. Given a sequence of length n, the probability that the complexity of that string is $n-5$ is $\frac{1}{32}$.

Definition 8. A sequence $x_1, x_2, \ldots, x_n$ is said to be random if $K(x_1, \ldots, x_n|n) \geq n$.

Definition 9. An infinite string x is incompressible if $\lim\limits_{n\to\infty} \dfrac{K(x_1, x_2, \ldots, x_n|n)}{n} = 1$

3.12 Occam's Razor

3.12.1 Laplace's Method

Laplace was curious about the question of whether or not the sun will rise the next morning given that it has risen every morning for the all of recorded history. Ignoring the physics of this, we can say that the random variable of "rising", X can be written as

$$X \sim \text{Bernoulli}(\theta)$$

We'll assume that θ is uniformly distributed on the unit interval.

$$P(X_{n+1} = 1|X_1 = 1, X_2 = 1, \ldots, X_n = 1) = \frac{P(X_{n+1} = 1, X_1 = 1, X_2 = 1, \ldots, X_n = 1)}{P(X_1 = 1, X_2 = 1, \ldots, X_n = 1)}$$

We will assume each X_k to be i.i.d. and now sum over all θ yielding:

$$\frac{\displaystyle\int_0^1 \theta^{n+1} \mathrm{d}\theta}{\displaystyle\int_0^1 \theta^{n} \mathrm{d}\theta} = \frac{n+1}{n+2}$$

You'll note that as n approaches infinity, this probability approaches 1.

3.12.2 Kolmogorov's Method

Now let's look at this from a computing standpoint. What's the probability of seeing a 1 after n 1's? As before, this is

$$\frac{P(\text{All sequences with } 1^n \text{ \& next bit } 1)}{P(\text{All sequences with } 1^n)}$$

We'll note that the simplest program that produces the desired output is one that just prints 1's forever.

Now let's look at the situation where we have the sequence $1^n 0 \ldots$. The complexity of this sequence is roughly

$$K(n) \approx \log n + \mathcal{O}(\log\log n)$$

But now let us consider the probability of seeing any sequence $1^n\, 0\, y$ where y is some arbitrary binary string. The probability of seeing any of these string is

$$\sum_y P(1^n\, 0\, y) \approx P(1^n 0) = 2^{-\log n} = \frac{1}{n}$$

Substituting that into our original equation:

$$P(0|1^n) = \frac{P(1^n\,0)}{P(1^n\,0) + P(1^{n+1})} = \frac{\frac{1}{n}}{\frac{1}{n} + c} = \frac{1}{cn+1}$$

Clearly, as n approaches infinity, $P(0|1^n)$ approaches zero. As we'd expect, the more times the sun rises, the less likely it is that it won't!

This is a toy example of Occam's Razor. The fundamental principle of the concept is that of two propositions, one of which is more likely than the other (i.e. the sun rising the next versus the sun not rising), we should accept the more probable explanation and can furthermore prove that in future experiments the more probable explanation will occur.

Appendix A—Quotes

- “I have a problem. It’s called gambling.” (Dr. Aiyer)
- “What’s E? Entropy?” (David Zhu)
- “So is it the strong law of weak numbers?” (Jerry Chen)
- “It’s like a half life... but it’s a double life” (Steven Cao)
- “Isn’t this just Lagrange?”
  10 minutes later..
  “Wait, how do you do Lagrange again?” (Swapnil Garg)
- “I’m just amazed that you manage to learn something” (Dr. Aiyer)
- (Looking at Σ) That’s a backwards ξ! (Steven)
- “What’s a binomial distribution?” (Dr. Aiyer)
  “$a + bx$” (Shaya)
- “An approximation of $\ln(1 - x)$ is $1 + x + x^2 + x^3 + \dots$” (Misha Ivkov)
- “Where does ϵ go to get a haircut?”
  “The epSALON” (Shaya)
- “So the sample mean is $\hat{\mu}$...” (Manan Shah)
- “Yes, $e^{1/2}$ on most days is $\sqrt{e}$...” (Dr. Aiyer)
- “We’re going to try to finish this today (Ha!)” (Dr. Aiyer)
- “Why are we using a × for multiplication in 2017?”
  “I thought we were at the age where we didn’t have to use anything...” (Rajiv Movva)
- “Why do we discuss confidence intervals instead of regions in statistics?” (Dr. Aiyer)
  “We approximate the ellipse to a square” (Shaya)
- “Hey, hey, Rajiv has allrajivs” (Vedaad)
  “I thought about that, but I figured it was too bad to say” (Shaya)
  “You should never Shayaway from a pun” (Misha).
- Tuesday, March 21, 2017 – Vedaad came to class *early*.