Professor Emmanuel Abbe Scribe: Amy Hua

*** denotes a chapter that is currently being edited.

Contents

1 Preliminaries

Our digital world relies heavily on our ability to extract, store, and transfer information. Consequently, much effort has been devoted to perform efficiently such tasks. This class covers the fundamental limits and algorithms for data compression and transmission, providing an introduction to information theory and coding theory. It also discusses connections with unsupervised machine learning problems, in particular with data clustering.

We will cover three broad areas:

- Compression of information
- Transmission of information
- Extraction of information (unsupervised learning).

1.1 Compression

Our general problem is to take some data D and put it through some compressor to get some data D'. We want the following:

1. D' needs to preserve the information in D.

2. D' needs to have less "volume" than D.

We have a trivial solution to each: to (1), the identity map, and to (2), the empty solution. But how do we find an appropriate tradeoff between the two? To get both (1) and (2), we want to exploit the redundancy or structure of the original data.

Definition 1.1. (Volume). The number of bits used to encode the data.

Example 1.2. Some words occur more often or in pairs in text documents.

Example 1.3. (Weather data). A toy example follows. We assume four possible states—sunny (with probability 1/2), cloudy (pr. 1/4), rainy (pr. 1/8), and snowy (pr. 1/8). The most naive approach would take 2 bits to encode each of these likelihoods (e.g., 00 for sunny, 01 for cloudy, 10 for rainy, and 11 for snowy). The expectation of the number of bits needed to encode the weather is 2 bits.

What does it mean to "preserve" the information? We want to be able to go from D' uniquely back to D. For instance, the following encoder is unique: 0 for sunny, 10 for cloudy, 110 for rainy, 1111 for snowy (this is called a **prefix-free code**, which we will go into detail soon). We can encode sunny—rainy—cloudy with 011010. The expected number of bits is $\frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 = \frac{7}{4} < 2$ bits.

Question 1.4. What is the optimal tradeoff?

Definition 1.5. (Entropy). Take x, denoting an outcome state, and $P(x)$, denoting the number of bits for the encoder corresponding to a given state x. Entropy is defined as

$$H(P) = \sum_{x \in \mathcal{X}} P(x) \log_2 \frac{1}{P(x)}.$$

Achieving the entropy of P means that the code uses the optimal expected number of bits.

Example 1.6. (YES/NO question game). We have a probability distribution on some finite set. The goal is to minimize the expected number of questions to guess the correct element in the finite set. A greedy approach is to group the elements into two roughly equally-sized groups. However, Huffman encoding is the optimal solution to this problem. Interestingly, the answer to this question is the same as the answer to our previous question.

In this class and most of basic information theory, redundancy is modelled with probability distributions. Note that the most random, in some sense, distribution is a completely uniform one, where we have no bias in predicting one outcome over another. On the other hand, the least random, a completely deterministic, distribution involves an outcome with pr. 1 and all other outcomes with pr. 0. The non-uniformity of distributions will be exploited to find an optimal code.

Remark 1.7. (Applications). Lossless compression examples include ZIP, PDF, PNG, and GIF. Lossy compression examples include JPEG and MPEG.

1.2 Transmission

The general problem is to take some data D and take it through a noisy channel to get data D'. This time, we want the following:

1. We need to be able to recover D from D'.
2. We need to have the least delay.

The goal, now, is to *add* redundancy to the data to protect it from the noise.

Question 1.8. What is the optimal tradeoff?

Example 1.9. We revisit the weather example from our compression discussion. The naive approach is to take 2 bits to encode each likelihood: 00 for sunny, 01 for cloudy, 10 for rainy, and 11 for snowy. What if the channel erases 1 bit? We wouldn't be able to recover the data. Now, say we encode via 000000 for sunny, 111000 for cloudy, 000111 for rainy, and 111111 for snowy. If the channel erases 2 bits, i.e., we see $0xx000$, and we can still recover the data. What if the channel erases 3 bits? We can take 000000, 111100, 001111, and 110011.

Example 1.10. (Applications). Transmission examples include SMS, email, telephone, Skype, the Internet, and storage.

Remark 1.11. Compression and transmission often both need to be performed.

Theorem 1.12. *(Separation, or Shannon, theorem). *** See original diagram.*

1.3 Information extraction

We pick a subset of topics related to data clustering (e.g., image segmentation, customer segmentation, medical diagnosis). Note that we can build a graph of similarity, which is closely related to compression. We will cover the following clustering topics:

- Hierarchical
- Center-based
- Spectral
- Axiomal
- Probabilistic.

2 Fix-to-variable codes for lossless compression

2.1 Notation

In these notes, we use the following notation.

- $\mathcal{X}$ denotes a finite set called the **source alphabet**, where the elements of $\mathcal{X}$ are called the **source symbols**. For example, $\mathcal{X}$ could be the 6 outcomes of a dice roll or the 26 letters of the English alphabet.
- $\{0,1\}^*$ denotes the set of all binary sequences of arbitrary length (not necessarily finite). $\{0,1\}^* = \{\emptyset, 0, 1, 00, 01, 10, 11, 000, \ldots\}$.
- $[n] := \{1, 2, \ldots, n\}$ denotes the set of the first n positive integers.
- For some integer $n \geq 1$, $\mathcal{X}^n = \mathcal{X} \times \cdots \times \mathcal{X}$ denotes the n-th Cartesian product of $\mathcal{X}$, i.e., the set of vectors of length n valued in $\mathcal{X}$. ($\times$ is the tensor product.)

2.2 Uniquely decodable and prefix-free codes

Definition 2.1. (Source encoder and codewords). A source encoder on a source alphabet $\mathcal{X}$ is a map $C : \mathcal{X} \to \{0,1\}^*$. $\{C(x)\}_{x\in\mathcal{X}}$ are called the codewords. Assume that we deal with codes for finite sequences.

Remark 2.2. Note that, for now, we will often use "encoder" and "code" interchangeably.

Definition 2.3. A source encoder for a source alphabet $\mathcal{X}$ is extended to $\mathcal{X}^n$ with the map $C^n : \mathcal{X}^n \to \{0,1\}^*$ defined by $(x_1, \ldots, x_n) \longmapsto C(x_1)C(x_2)\cdots C(x_n)$, which we call **concatenation** of codewords. With a small abuse of notation, we will often use C directly in place of C^n to denote a sequence of source symbols.

Example 2.4. $\mathcal{X} = \{A, B, C\}$. $C(A) = 0$, $C(B) = 10$, $C(C) = 11$. $C^3(A, B, A) = 0100$.

Definition 2.5. (Uniquely decodable (UD)). A source encoder C is uniquely decodable (UD) if its extension C^n is injective for any $n \geq 1$, i.e., $\forall\ n \geq 1$ and $\forall\ x_1, \ldots, x_n, y_1, \ldots, y_n \in \mathcal{X}$ such that $C^n(x_1, \ldots, x_n) = C^n(y_1, \ldots, y_n)$, then $x_1 = y_1, \ldots, x_n = y_n$.

Example 2.6. (Prefix). 0110 is a prefix of 01101110.

Definition 2.7. (Prefix-free (PF)). A source encoder C is prefix-free (PF) if for any $x, y \in \mathcal{X}$, $x \neq y$, $C(x)$ is not a prefix of $C(y)$.

Example 2.8. The codewords $\{0, 10, 110, 1110\}$ are prefix-free, but the codewords $\{01, 011\}$ are not prefix-free since 011 starts with 01.

Remark 2.9. PF codes are UD. In fact, they are instantaneously decodable symbol by symbol. However, there are UD codes that are not PF. (Ex: suffix-free codes. Other examples also exist.)

Remark 2.10. A PF code can be represented using a binary tree, where the codewords must terminate at the leaves of the tree.

2.3 Length of a code

Definition 2.11. (Length function). For a source code C on $\mathcal{X}$, we define the length function of C to be

$$\ell_C : \mathcal{X} \to \mathbb{Z}_+,$$
$$x \longmapsto \text{length}(C(x)).$$

(Note that $0 \in \mathbb{Z}_+$ by convention.)

Example 2.12. length(01101) = 5. (The length function simply counts how many bits are in a codeword.)

Remark 2.13. The length function can be applied to a sequence of codewords—it adds up the length of each codeword.

Question 2.14. Does the length function of a PF or UD code need to satisfy some properties?

Theorem 2.15. *(Kraft's inequality). (a) If a source code C on $\mathcal{X}$ is PF, then*

$$\sum_{x \in \mathcal{X}} 2^{-\ell_C(x)} \leq 1.$$

(b) If a set of positive integers $\{\ell(x)\}_{x \in \mathcal{X}}$ satisfies $\sum_{x \in \mathcal{X}} 2^{-\ell(x)} \leq 1$, then there exists a PF code matching those lengths.

Proof. (a) Define a binary tree corresponding to the prefix-free code C. Extend the tree so that each leaf also has a neighboring leaf. Now, consider a random walk on this tree, with probability 1/2 of going up and probability 1/2 of going down at each intersection, independently of its past. The probability that the random walk halts is 1 as the tree is finite, and the probability that the random walk halts in a codeword is less than or equal to 1 because of the extension of the tree. The probability that the random walk halts in a codeword $C(x)$ is $2^{-\ell_C(x)}$, and since the events of the random walk halting in a particular codeword are mutually exclusive, we get

$$\sum_{x \in \mathcal{X}} 2^{-\ell_C(x)} \leq 1.$$

(b) *** See diagram. A codeword $(c_1, \ldots, c_k)$ of a PF code is a leaf of the binary tree, and it can be equivalently characterized by a dyadic interval of length 2^{-k} in $[0, 1]$, which contains all values $\sum_{i=1}^{k} c_i 2^{-i} + \sum_{i=k+1}^{\infty} b_i 2^{-i}$ for all possible binary sequences $\{b_i\}_{i \geq k+1}$, i.e., the suffix values.

Let $\{\ell(x)\}_{x \in \mathcal{X}}$ be a set of positive integers satisfying Kraft's inequality. Define $P(x) = 2^{-\ell(x)}$. Partition the interval $[0, 1]$ into dyadic intervals of length $P(x)$, which can be packed without overlap since $\sum_{x \in \mathcal{X}} P(x) \leq 1$. The code is then given by assigning the PF codeword to each dyadic interval as previously explained. □

Thus, we have shown that PF codes are UD and must satisfy Kraft's inequality.

2.4 Probabilistic source models

We now put a probabilistic model on the source.

Definition 2.16. (Discrete memoryless source (DMS)). A discrete memoryless source (DMS) is a discrete-time random process producing a sequence of random variables $\{X_i\}_{i\geq 1}$ such that all X_i's are mutually independent and each X_i has the same distribution P on $\mathcal{X}$, which we call the **source distribution**.

Example 2.17. $\mathcal{X} = \{A, B, C, D\}$. $P(A) = \frac{1}{2}, P(B) = \frac{1}{4}, P(C) = \frac{1}{8}, P(D) = \frac{1}{8}$.

Remark 2.18. $\ell_C(X)$ is a random variable if X is a random source symbol variable distributed from P on $\mathcal{X}$ for a code C on $\mathcal{X}$. $\ell_C(X)$ takes value $\ell_C(x)$ with probability $P(x)$.

Definition 2.19. The expected length of a code C under a source distribution P is defined by

$$\overline{L}(C, P) = \sum_{x\in\mathcal{X}} \ell_C(x)P(x).$$

Example 2.20. $\mathcal{X} = \{A, B, C, D\}$. $P(A) = \frac{1}{2}, P(B) = \frac{1}{4}, P(C) = \frac{1}{8}, P(D) = \frac{1}{8}$. $C(A) = 0,\ C(B) = 10,\ C(C) = 110,\ C(D) = 111$. The expected length of C is $\overline{L}(C, P) = 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{8} + 3 \cdot \frac{1}{8} = \frac{7}{4}$.

Remark 2.21. Take a sequence $\ell_C(X_1), \ldots, \ell_C(X_n)$.

$$\mathbb{E}[\ell_C(X_1) + \cdots + \ell_C(X_n)] = \mathbb{E}(\ell_C(X_1)) + \cdots + \mathbb{E}(\ell_C(X_n)) = n\mathbb{E}(\ell_C(X_1)) = n\overline{L}(C, P).$$

So studying the one-shot problem captures studying the block DMS sequence.

Remark 2.22. $\frac{1}{n}\left(\ell_C(X_1) + \cdots + \ell_C(X_n)\right) = \frac{1}{n}\sum_{i=1}^{n} \ell_C(X_i) \xrightarrow{\text{in } P} \mathbb{E}(\ell_C(X_1))$, as $X_1, \ldots, X_n$ are i.i.d. By the law of large numbers,

$$\forall\ \varepsilon > 0,\ \mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} \ell_C(X_i) - \mathbb{E}(\ell_C(X_1))\right|\right) \xrightarrow{n\to\infty} 0.$$

In other words, the average length of a long sequence of codewords from a code C is, with high probability, close to $\overline{L}(C, P)$. Again, the one-shot problem is enough to capture the block DMS sequence.

2.5 Efficient prefix-free codes

Question 2.23. We ask a few questions:

- For a given source distribution P, what is the minimum value that $\overline{L}(C, P)$ can take for a UD source code C?
- For a given source distribution P, what is the minimum value that $\overline{L}(C, P)$ can take for a PF source code C?
- For a given source distribution P, how can we construct a source code C that is UD and achieves the minimum value that $\overline{L}(C, P)$ can take for a UD code?

Remark 2.24. Note that, generally, in this class, $\log = \log_2$. There will only be one lecture (on Gaussian channels) in which $\log = \log_e$.

Remark 2.25. Note that since $\overline{L}(C, P)$ depends only on the code C through its length function, and since the length function of PF codes is characterized by Kraft's inequality, we can pose the questions above as an optimization problem. We first address these questions with PF codes and will discuss the general case of UD codes in the future.

$$\begin{aligned} \min_{\{\ell_C(x)\}_{x\in\mathcal{X}}} \quad & \sum_{x\in\mathcal{X}} P(x)\ell_C(x) \\ \text{s.t.} \quad & \sum_{x\in\mathcal{X}} 2^{-\ell_C(x)} \le 1 \\ & \ell_C(x) \in \mathbb{Z}_+ \ \forall\ x \in \mathcal{X}. \end{aligned}$$

Defining $q(x) := 2^{-\ell_C(x)}$, we can write the problem equivalently as

$$\begin{aligned} \min_{\{q(x)\}_{x\in\mathcal{X}}} \quad & \sum_{x\in\mathcal{X}} P(x) \log \frac{1}{q(x)} \\ \text{s.t.} \quad & \sum_{x\in\mathcal{X}} q(x) \le 1 \\ & q(x) \in 2^{-\mathbb{Z}_+} \ \forall\ x \in \mathcal{X}, \end{aligned}$$

which is lower bounded by

$$\begin{aligned} \min_{\{q(x)\}_{x\in\mathcal{X}}} \quad & \sum_{x\in\mathcal{X}} P(x) \log \frac{1}{q(x)} \\ \text{s.t.} \quad & \sum_{x\in\mathcal{X}} q(x) \le 1 \\ & q(x) \ge 0 \ \forall\ x \in \mathcal{X}. \end{aligned}$$

Claim 2.26. *For all q such that $q(x) \ge 0\ \forall\ x \in \mathcal{X}$, with $\sum_{x\in\mathcal{X}} q(x) \le 1$, and for all probability distributions P on $\mathcal{X}$, i.e., for any source distribution P on $\mathcal{X}$ and any PF code C on $\mathcal{X}$, we have*

$$\overline{L}(C, P) := \sum_{x\in\mathcal{X}} P(x) \log \frac{1}{q(x)} \ge \sum_{x\in\mathcal{X}} P(x) \log \frac{1}{P(x)} =: H(P).$$

Proof.

$$\begin{aligned} \sum_{x\in\mathcal{X}} P(x) \log \frac{1}{P(x)} - \sum_{x\in\mathcal{X}} P(x) \log \frac{1}{q(x)} &= \sum_{x\in\mathcal{X}} P(x) \log \frac{q(x)}{P(x)} \\ &\le \sum_{x\in\mathcal{X}} P(x) \frac{1}{\ln 2} \left(\frac{q(x)}{P(x)} - 1 \right) \\ &= \frac{1}{\ln 2} \left(\sum_{x\in\mathcal{X}} q(x) - \sum_{x\in\mathcal{X}} P(x) \right) \\ &\le \frac{1}{\ln 2} (1 - 1) \\ &= 0, \end{aligned}$$

since $\log(g) \leq \frac{1}{\ln 2}(g-1)$ for any $g \geq 0$. Thus, we have proven that

$$\begin{aligned} \min_{\{q(x)\}_{x\in\mathcal{X}}} \quad & \sum_{x\in\mathcal{X}} P(x) \log \frac{1}{q(x)} \\ \text{s.t.} \quad & \sum_{x\in\mathcal{X}} q(x) \leq 1 \\ & q(x) \in 2^{-\mathbb{Z}_+} \ \forall \ x \in \mathcal{X} \end{aligned}$$

is lower bounded by

$$\begin{aligned} \min_{\{q(x)\}_{x\in\mathcal{X}}} \quad & \sum_{x\in\mathcal{X}} P(x) \log \frac{1}{q(x)} \\ \text{s.t.} \quad & \sum_{x\in\mathcal{X}} q(x) \leq 1 \\ & q(x) \geq 0 \ \forall \ x \in \mathcal{X}, \end{aligned}$$

which is lower bounded by

$$\sum_{x\in\mathcal{X}} P(x) \log \frac{1}{P(x)} =: H(P).$$

□

Remark 2.27. The bound of the previous theorem is tight. For example, consider a probability distribution P that takes values $\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right)$. Then, the PF code $(0, 10, 110, 111)$ achieves $H(P) = \frac{7}{4}$.

Question 2.28. When is the equality met in general?

3 Bounds on the expected length

3.1 Generalizing Kraft's inequality to all uniquely decodable codes

Theorem 3.1. *(McMillan). For a UD code C on $\mathcal{X}$, we have*

$$\sum_{x\in\mathcal{X}} 2^{-\ell_C(x)} \leq 1.$$

Note that we already have the converse from Kraft's inequality: if $\sum_{x\in\mathcal{X}} 2^{-\ell_C(x)} \leq 1$ is satisfied, then there exists a PF (and, hence, a UD) code matching the lengths in C.

Proof. Let C be a UD code, and let $\alpha := \sum_{x\in\mathcal{X}} 2^{-\ell_C(x)}$ (so $\alpha > 0$). We prove the following "astucious" claim: there exists a constant $\beta > 0$ such that $\forall\ k$ large enough, $\alpha^k \leq \beta k$. We have

$$\begin{aligned}\alpha^k &= \sum_{x_1\in\mathcal{X}} 2^{-\ell_C(x_1)} \cdot \ldots \cdot \sum_{x_k\in\mathcal{X}} 2^{-\ell_C(x_k)} \\ &= \sum_{x_1,\ldots,x_k\in\mathcal{X}^k} 2^{-(\ell_C(x_1)+\ldots+\ell_C(x_k))}.\end{aligned}$$

Let

$$L_{\max} := \max_{x\in\mathcal{X}} \ell_C(x).$$

Thus, since $\ell_C(x) \geq 1\ \forall\ x$,

$$\sum_{i=1}^{k} \ell_C(x_i) \in \{k, \ldots, k * L_{\max}\}.$$

We can rewrite α^k as

$$\sum_{l=k}^{kL_{\max}} \sum_{x_1,\ldots,x_k\in\mathcal{X}^k} 2^{-l} \text{ s.t. } \sum_{i=1}^{k} \ell_C(x_i) = \ell.$$

Define

$$\begin{aligned}A(\ell) &= \sum_{x_1,\ldots,x_k\in\mathcal{X}^k} 1 \text{ s.t. } \sum_{i=1}^{k} \ell_C(x_i) = \ell \\ &= \left|\left\{x_1, \ldots, x_k \in \mathcal{X}^k \ : \ \sum_{i=1}^{k} \ell_C(x_i) = \ell\right\}\right|.\end{aligned}$$

Thus,

$$\begin{aligned}\alpha^k &= \sum_{l=k}^{kL_{\max}} \sum_{x_1,\ldots,x_k\in\mathcal{X}^k} 2^{-l} \text{ s.t. } \sum_{i=1}^{k} \ell_C(x_i) = \ell \\ &= \sum_{l=k}^{kL_{\max}} A(l)2^{-\ell} \\ &\leq kL_{\max} \\ &= \beta k,\end{aligned}$$

since $A(\ell) \leq 2^\ell$: this holds since $\sum_{i=1}^k \ell_C(x_i) = \ell$, so we have ℓ bits, and due to unique decodability, we may allow up to 2^ℓ distinct codewords.

Thus, there exists a constant $\beta > 0$ such that $\alpha^k \leq \beta k \ \forall \ k$, which implies that $\alpha \leq 1$; otherwise, there would exist a k such that $\alpha^k > \beta k$. Thus, we have shown the result. □

3.2 Entropy lower bound on $\overline{L}(P, C)$

Definition 3.2. (Entropy). For a probability distribution P on $\mathcal{X}$, the entropy "in bits" of P is defined by

$$H(P) = \sum_{x \in \mathcal{X}} P(x) \log \frac{1}{P(x)}.$$

Remark 3.3. A note about notation: sometimes, we will see $H(X)$ being written for some random variable $X \sim P$. Note that $H(X) := H(P)$.

Theorem 3.4. *For any probability distribution P on $\mathcal{X}$ and for any UD code C on $\mathcal{X}$, the average length of the code satisfies*

$$H(P) \leq \overline{L}(C, P).$$

Proof. We can use our previous proof that for any PF code C, $H(P) \leq \overline{L}(C, P)$, except that we now use McMillan's theorem in place of Kraft's inequality. □

3.2.1 The information inequality: a brief digression

Remark 3.5. The above claim is called **the information inequality** and is one of the most important inequalities in information theory. It can be reformulated using the KL-divergence.

Definition 3.6. (Kullback-Leibler divergence). Let p and q be two probability distributions on $\mathcal{X}$ such that $q(x) = 0 \implies p(x) = 0$.

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

is called the **KL-divergence** or **relative entropy**, terms used in statistics and information theory, respectively.

Remark 3.7. Properties of $D(p \parallel q)$ include:

1. $D(p \parallel q) \geq 0 \ \forall \ p, q$ proper. This is the information inequality, which we already proved. Note that we are imposing a stronger condition than $\sum_{x \in \mathcal{X}} q(x) \leq 1$ by requiring q to be a probability distribution.

2. $D(p \parallel q) = 0 \iff p = q$. See Problem Set 1.

3. In general, $D(p \parallel q) \neq D(q \parallel p)$ and the triangle inequality does not hold, so $D(p \parallel q)$ is not a "true" distance metric.

4. In practice, the KL-divergence acts like a norm-squared. If p and q are "close," we have

$$D(p \parallel q) \approx \sum_{x \in \mathcal{X}} (p(x) - q(x))^2 p(x) = \|p - q\|_p^2.$$

   We will not go into defining a notion of "angle" between probability distributions, but for more information, look up "information geometry."

3.3 To what extent is $H(P)$ a lower bound?

We explore some questions regarding the tightness of the entropy lower bound on the expected codeword length.

- Can we achieve $H(P)$? When?
- If we cannot achieve $H(P)$, how close can we get?
- How do we construct the code of optimal expected length for a given distribution?

3.3.1 When and with what frequency can we achieve $H(P)$?

Remark 3.8. The inequality is tight, i.e., $H(P) = \overline{L}(C, P)$ for some P. For instance, take any dyadic distribution, i.e., $P(x) \in 2^{-\mathbb{Z}_+} \ \forall \ x \in \mathcal{X}$. We can have

$$\ell(x) = \log \frac{1}{P(x)} \geq 0 \ \forall \ x \in \mathcal{X}$$

and

$$\sum_{x \in \mathcal{X}} 2^{-\ell(x)} = \sum_{x \in \mathcal{X}} P(x) = 1.$$

Thus, we can achieve the entropy bound with a UD code after applying the existence part of Kraft's inequality. In fact, the entropy can be achieved if and only if $P(x) \in 2^{-\mathbb{Z}_+} \ \forall \ x \in \mathcal{X}$, and furthermore, we can achieve it with a PF code.

Example 3.9. The example we often return to is $P(x_1) = \frac{1}{2}, P(x_2) = \frac{1}{4}, P(x_3) = \frac{1}{8}, P(x_4) = \frac{1}{8}$ with $C(x_1) = 0, C(x_2) = 10, C(x_3) = 110, C(x_4) = 111$, which has $H(P) = \overline{L}(C, P) = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 = \frac{7}{4}$.

3.3.2 How close to $H(P)$ can we get?

Theorem 3.10.

$$\overline{L}^*(P) \leq H(P) + 1,$$

where we define $\overline{L}^*(P) = \min_{C \text{ UD on } \mathcal{X}} \overline{L}(C, P)$.

Proof. Given a probability distribution P on $\mathcal{X}$, let $\ell(x) := \lceil \log \frac{1}{P(x)} \rceil \ \forall \ x \in \mathcal{X}$. Note that we still have $\sum_{x \in \mathcal{X}} 2^{-\ell_C(x)} \leq 1$. Take a PF code for the length function, so we obtain an expected length

$$\sum_{x \in \mathcal{X}} \ell(x) P(x) = \sum_{x \in \mathcal{X}} \left\lceil \log \frac{1}{P(x)} \right\rceil P(x) \leq H(P) + 1$$

since $\left\lceil \log \frac{1}{P(x)} \right\rceil \leq \log \frac{1}{P(x)} + 1$. □

Lemma 3.11. *Let $X_1, \ldots, X_n$ be i.i.d. under the probability mass function P. Then,*

$$H(X_1, \ldots, X_n) = nH(X_1) = nH(P).$$

Proof. Problem Set 2. □

Corollary 3.12. *If an optimal UD code C is applied to a block of n i.i.d. symbols from P on $\mathcal{X}$, then we have from the previous two theorems that*

$$H(P^n) \leq \overline{L}(C^n, P^n) \leq H(P^n) + 1,$$

where $\overline{L}(C^n, P^n)$ is the expected length of encoding $X_1, \ldots, X_n \sim P \Longrightarrow (X_1, \ldots, X_n) \sim P^n$. So, the normalized expected length satisfies

$$\frac{H(P^n)}{n} \leq \frac{\overline{L}(C^n, P^n)}{n} \leq \frac{H(P^n)}{n} + \frac{1}{n}$$

$$\Longrightarrow H(P) \leq \frac{\overline{L}(C^n, P^n)}{n} \leq H(P) + \frac{1}{n},$$

and since $\frac{1}{n} \to 0$ as $n \to \infty$,

$$\Longrightarrow H(P) \leq \frac{\overline{L}(C^n, P^n)}{n} \leq H(P).$$

Hence, the normalized expected length of an optimal UD code for P^n tends to $H(P)$ as $n \to \infty$.

Remark 3.13. Note that $\overline{L}^*(P^n)$ is not achieved in general by $C^*(P)^n$.

Remark 3.14. $C^*(P^n) \neq C^*(P)^n$.

Example 3.15. $\mathcal{X} = \{A, B\}$, with $P(A) = \frac{1}{\pi}$, $P(B) = 1 - \frac{1}{\pi}$ and $P^2(AA) = \left(\frac{1}{\pi}\right)^2$, $P^2(AB) = P^2(BA) = \frac{1}{\pi}\left(1 - \frac{1}{\pi}\right)$, $P^2(BB) = \left(1 - \frac{1}{\pi}\right)^2$. $\overline{L}^*(P) = 1$ and $H(P) < 1$.

Question 3.16. How do we construct optimal codes? See the following discussion on Huffman codes.

4 Huffman codes

Question 4.1. How do we achieve $L^*(P)$ with a UD code in general?

4.1 Huffman encoding algorithm

Definition 4.2. (Algorithm). Input: $\mathcal{X}$ and P on $\mathcal{X}$. Output: Binary tree that gives a PF code.

1. Order $\{P(x)\}_{x\in\mathcal{X}}$ in decreasing order. Allocate a leaf for each element.
2. Merge the two smallest probabilities, and break ties arbitrarily (BTA). Create a vertex with the sum of the probabilities attached to it.
3. Consider the remaining leaves with this new vertex, and iterate until the tree is obtained.
4. Read the PF code from the tree (e.g., left = 0, right = 1).

Example 4.3. $\mathcal{X} = \{a, b, c, d, e, f\}$. $P = \{0.4, 0.2, 0.2, 0.1, 0.05, 0.05\}$. Note that

$$H(P) = 0.4 \log_2 2.5 + 2 \cdot 0.2 \log_2 5 + 0.1 \log 10 + 2 \cdot 0.05 \log 20 \approx 2.222.$$

Using the Huffman encoding algorithm, we find a few possible solutions. We obtain the PF code

$$C_1 = \{0, 100, 101, 110, 1110, 1111\},\ \overline{L}(C_1, P) = 2.3,$$

via the following merges:

- $\{e\}$ and $\{f\}$,
- $\{d\}$ and $\{e, f\}$,
- $\{b\}$ and $\{c\}$,
- $\{b, c\}$ and $\{d, e, f\}$,
- $\{a\}$ and $\{b, c, d, e, f\}$.

Similarly, we obtain the PF code

$$C_2 = \{0, 10, 110, 1110, 11110, 11111\},\ \overline{L}(C_2, P) = 2.3,$$

via the following merges:

- $\{e\}$ and $\{f\}$,
- $\{d\}$ and $\{e, f\}$,
- $\{c\}$ and $\{d, e, f\}$,
- $\{b\}$ and $\{c, d, e, f\}$,
- $\{a\}$ and $\{b, c, d, e, f\}$.

4.2 Properties of Huffman codes

Theorem 4.4. *The Huffman code achieves $\overline{L}^*(P)$ for any P.*

Proof. See TC Chapter 5.8 for the full proof. Some elements of the proof follow.

- The optimal UD code C for P on $\mathcal{X}$, with $|\mathcal{X}| = k$, where we let $P(1) \leq P(2) \leq \cdots \leq P(k)$ be the ordered probabilities, can be chosen such that:
  - (i) C is a PF code;
  - (ii) $\ell_i^* \geq \ell_j^*$ if $P(i) < P(j)$, where ℓ_i^* is the length of $C(i)$;
  - (iii) $\ell_1^* = \ell_2^*$;
  - (iv) $C(1)$ and $C(2)$ are siblings.
- More on(ii): Let C' be the code such that $C(i)$ and $C(j)$ are swapped, so $C'(j) = C(i)$ and $C'(i) = C(j)$. If $P(i) < P(j)$ and $\ell_i^* < \ell_j^*$, then

$$\overline{L}(C, P) - \overline{L}(C', P) = P(i)\ell_i^* + P(j)\ell_j^* - P(i)\ell_j^* - P(j)\ell_i^* = (P_j - P_i)(\ell_j - \ell_i) > 0.$$

  This contradicts the optimality of C. Thus, for an optimal code, $P(i) < P(j)$ implies $\ell_i^* \geq \ell_j^*$.

□

Question 4.5. How can we characterize P^n for large n? See the section on the asymptotic equipartition property (AEP).

5 Typical set and the asymptotic equipartition property

5.1 Asymptotic equipartition property (AEP)

We recap the Law of Large Numbers, which plays an important role in explaining the AEP.

Theorem 5.1. *(Weak Law of Large Numbers (LLN)). Let $Y_1, \ldots, Y_n$ be i.i.d. under P on some Lesbegue measurable set $\mathcal{Y}$ (e.g., $\mathcal{Y} \subset \mathbb{R}$) with $\mathbb{E}(Y_i) =: \mu < +\infty$. Then, for any $\varepsilon > 0$, we have convergence in probability:*

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} Y_i \in [\mu - \varepsilon,\ \mu + \varepsilon]\right) \xrightarrow{n\to\infty} 1.$$

Corollary 5.2. *Let $X_1, \ldots, X_n$ be i.i.d. under P on $\mathcal{X}$. ($X_1, \ldots, X_n$ are the random variables in our compression problem.) Let $f : \mathcal{X} \to \mathcal{Y}$ such that $\mathbb{E}(f(X_1)) < +\infty$. Then, $\forall\ \varepsilon > 0$,*

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} f(X_i) \in [\mathbb{E}(f(X_1)) - \varepsilon,\ \mathbb{E}(f(X_1)) + \varepsilon]\right) \xrightarrow{n\to\infty} 1.$$

In particular, take $f(x) = \log_2 \frac{1}{P(x)}$. Then,

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} \log_2 \frac{1}{P(x_i)} \in [H(P) - \varepsilon,\ H(P) + \varepsilon]\right) \xrightarrow{n\to\infty} 1.$$

Definition 5.3. (Typical set). Let P be a probability distribution on $\mathcal{X}$, $\varepsilon \in (0, 1)$, $n \geq 1$, $n \in \mathbb{Z}_+$. Define the typical set for these parameters to be

$$A_{\varepsilon,n}(P) := \left\{(x_1, \ldots, x_n) \in \mathcal{X}^n : \frac{1}{n}\sum_{i=1}^{n} \log_2 \frac{1}{P(x_i)} \in [H(P) - \varepsilon,\ H(P) + \varepsilon]\right\}.$$

Remark 5.4. Since $x = (x_1, \ldots, x_n)$, so $P^n((x_1, \ldots, x_n)) = P^n(x) = \prod_{i=1}^{n} P(x_i)$, we can write

$$\frac{1}{n}\sum_{i=1}^{n} \log \frac{1}{P(x_i)} = \frac{1}{n} \log \frac{1}{\prod_{i=1}^{n} P(x_i)} = \frac{1}{n} \log \frac{1}{P^n(x)}.$$

Thus, we can rewrite the condition for belonging in $A_{\varepsilon,n}(P)$:

$$\frac{1}{n}\sum_{i=1}^{n} \log \frac{1}{P(x_i)} \in [H(P) - \varepsilon,\ H(P) + \varepsilon] \iff P^n(x) \in \left[2^{-n(H(P)+\varepsilon)},\ 2^{-n(H(P)-\varepsilon)}\right].$$

Remark 5.5. (Binary case). Let $\mathcal{X} = \{0, 1\}$. $P(1) = p$, $P(0) = 1 - p$. Note that

$$\frac{1}{n}\sum_{i=1}^{n} \log \frac{1}{P(x_i)} \in [H(P) - \varepsilon, H(P) + \varepsilon]$$

and

$$\frac{1}{n}\sum_{i=1}^{n} x_i \in [p-\varepsilon, p+\varepsilon].$$

$$P^n(x) = \prod_{i=1}^{n} P(x_i) = \prod_{i=1}^{n} p^{x_i}(1-p)^{1-x_i} = p^{\sum_{i=1}^n x_i}(1-p)^{n-\sum_{i=1}^n x_i}.$$

Remark 5.6. If $p \in (0, 1/2)$ (e.g., $1/4$), then the maximal probability sequence is $(0, \ldots, 0)$ but is not typical for small enough $\varepsilon > 0$. See Problem Set 2.

Theorem 5.7. *(Asymptotic equipartition property (AEP)). Let $X_1, \ldots, X_n$ be i.i.d. under P.*

1. *$\forall\ \varepsilon > 0$, $P^n(A_{\varepsilon,n}(P)) \xrightarrow{n\to\infty} 1 \iff \mathbb{P}((X_1, \ldots, X_n) \in A_{\varepsilon,n}(P)) \xrightarrow{n\to\infty} 1$.*
2. *$\forall\ \varepsilon > 0$, $\forall\ n \geq 1$, $|A_{\varepsilon,n}(P)| \leq 2^{n(H(P)+\varepsilon)}$.*
3. *$\forall\ \varepsilon > 0$, n sufficiently large, $|A_{\varepsilon,n}(P)| \geq (1-\varepsilon)2^{n(H(P)-\varepsilon)}$.*

Proof. 1. Apply the Law of Large Numbers with the function $f(x) = \log \frac{1}{P(x)}$.

2.

$$\begin{aligned} 1 &\geq P^n(A_{\varepsilon,n}(P)) \\ &= \sum_{x \in A_{\varepsilon,n}(P)} P^n(x) \\ &\geq \sum_{x \in A_{\varepsilon,n}(P)} 2^{-n(H(P)+\varepsilon} \\ &= |A_{\varepsilon,n}(P)| 2^{-n(H(P)+\varepsilon)} \end{aligned}$$

3. $\forall \varepsilon > 0$, n sufficiently large,

$$\begin{aligned} 1-\varepsilon &\leq P^n(A_{\varepsilon,n}(P)) \\ &= \sum_{x \in A_{\varepsilon,n}(P)} P^n(x) \\ &\leq \sum_{x \in A_{\varepsilon,n}(P)} 2^{-n(H(P)-\varepsilon)} \\ &= |A_{\varepsilon,n}(P)| 2^{-n(H(P)-\varepsilon)} \end{aligned}$$

□

Remark 5.8. (Interpretation of AEP and visualization of a typical set). *** See figure. $X_1, \ldots, X_n$ are i.i.d. from P, while $(x_1, \ldots, x_n)$ are realizations of the random variables. P^n tends to be uniform on $A_{\varepsilon,n}(P)$ at about $2^{-n(H(P))}$ throughout, which has roughly $2^{nH(P)}$ elements. The volume of the probability distribution over the typical set is roughly 1, the size of the entire probability distribution. Note that this interpretation is another way of understanding entropy, beyond the sandwiching approach.

5.2 AEP encoder

How do we use AEP to show that there exists a code achieving $H(P)$?

Definition 5.9. (AEP code). Let $N = \lceil \log_2 |A_{\varepsilon,n}(P)| \rceil \approx nH(P)$, the number of bits needed to represent $A_{\varepsilon,n}(P)$, using the naive code (not exploiting information about the distribution). Index all elements of $A_{\varepsilon,n}(P)$ with N bits.

$$x \in A_{\varepsilon,n}(P) \longmapsto i(x) \in \{0,1\}^N.$$

Define

$$C_{\text{AEP}} : \mathcal{X}^n \to \{0,1\}^*,$$
$$x \longmapsto C_{\text{AEP}}(x).$$

We have two cases:

1. If $x \in A_{\varepsilon,n}(P)$, then $C_{\text{AEP}}(x) = (1, i(x))$. $i(x)$ is the index of x in $A_{\varepsilon,n}(P)$ and takes $N \approx nH(P)$ bits.
2. If $x \notin A_{\varepsilon,n}(P)$, then $C_{\text{AEP}}(x) = (0, b(x))$, where $b(x) = (b(x_1), \ldots, b(x_n))$, and $b(x_i)$ is the binary expansion of x_i with $\lceil \log_2 |\mathcal{X}| \rceil$ bits (so $n\lceil \log_2 |\mathcal{X}| \rceil$ overall).

Theorem 5.10. $\lim_{\varepsilon \to 0} \lim_{n \to \infty} \overline{L}(C_{AEP}, P) = H(P)$.

Proof. $\frac{\overline{L}(C_{\text{AEP}}, P^n)}{n} = \left[\frac{1}{n} + (H(P) + \varepsilon)\right](1 - \delta_n) + \left[\frac{1}{n} + \lceil \log |\mathcal{X}| \rceil\right] \delta_n = H(P) + \varepsilon.$ □

Remark 5.11. (Intuition behind the AEP code). We do not need to worry much about the second case, as the probability is vanishing. Note that if P is an equiprobable distribution, then the AEP encoder cannot compress, as in that case, everything will be in the typical set; otherwise, it does. *** Even though this coding scheme is not practically implementable (indexing elements of the typical set takes a very long time), and the decoding scheme is just a lookup table, which is not very clever with a large block length n, this code shows the existence of a code that achieves $H(P)$.

Remark 5.12. $P^n(\cdot)$ is a probability distribution on $\mathcal{X}^n$. The number of dimensions is $|\mathcal{X}|^n$.

Remark 5.13. We quickly recap what we have covered thus far.

- $H(P)$ is the fundamental limit for compressing an i.i.d. sequence (for the normalized expected length).
- Both Huffman on blocks and AEP codes are inefficient.

Question 5.14. How do we achieve $H(P)$ with an efficient code? We will discuss an efficient coding scheme when we introduce polar codes.

6 Information measures and estimation

6.1 Notation and facts

Recall the following definitions:

- Let X and Y be two random variables on the sets $\mathcal{X}$ and $\mathcal{Y}$. Their joint probability distribution $P_{X,Y}$ is defined on the set $\mathcal{X} \times \mathcal{Y}$, which denotes the set of all pairs (x, y), with $x \in \mathcal{X}$, $y \in \mathcal{Y}$. The marginal and conditional probabillities are given, respectively, by:
$$P_X(x) = \sum_{y \in \mathcal{Y}} P_{X,Y}(x, y)$$
and
$$P_{X|Y}(x|y) = \frac{P_{X,Y}(x, y)}{P_Y(y)}.$$

- X and Y are independent if
$$P_{X,Y}(x, y) = P_X(x) P_Y(y) \ \forall \ x \in \mathcal{X}, y \in \mathcal{Y},$$
or equivalently,
$$P_{X|Y}(x|y) = P_X(x) \ \forall \ x \in \mathcal{X}, y \in \mathcal{Y}.$$

- Entropy:
$$H(X) = H(P) = \sum_{x \in \mathcal{X}} P(x) \log_2 \frac{1}{P(x)},$$
where X is a random variable with probability distribution P on $\mathcal{X}$.

- Kullback-Leibler divergence or relative entropy:
$$D(P\|Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)},$$
where P and Q are two probability distributions on the same set $\mathcal{X}$.

- Information inequality: $D(P\|Q) \geq 0$.

6.2 Information measures

Definition 6.1. (Entropy for joint and conditional distributions).

- $H(X, Y) = H(P_{X,Y}) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} P_{X,Y}(x, y) \log_2 \frac{1}{P_{X,Y}(x,y)}$, where $P_{X,Y}$ is a probability distribution on $\mathcal{X} \times \mathcal{Y}$.

- $H(X|Y = y) \triangleq H(P_{X|Y=y}) \triangleq \sum_{x \in \mathcal{X}} P_{X|Y}(x|y) \log_2 \frac{1}{P_{X|Y}(x|y)}$, where $P_{X|Y}(x|y) := \frac{P_{X,Y}(x,y)}{P_Y(y)}$ and $P_Y(y) = \sum_{x \in \mathcal{X}} P_{X,Y}(x, y)$.

- $H(X|Y) \triangleq \sum_{y \in \mathcal{Y}} H(X|Y = y) P_Y(y)$.

Definition 6.2. (Chain rule). $H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$. The chain rule is the equivalent of Bayes' rule for entropy.

Theorem 6.3. *Note that if $X \perp\!\!\!\perp Y$, then $H(X|Y) = H(X)$.*

Theorem 6.4. $H(X_1, \ldots, X_n) = \sum_{i=1}^{n} H(X_i|X_1, \ldots, X_{i-1})$.

Proof. See the proof for independent $X_1, \ldots, X_n$ in Problem Set 2. □

Lemma 6.5. *A few more basic inequalities follow:*

- $0 \leq H(X) \leq \log_2 |\mathcal{X}|$. *Note that the bounds are tight. The first inequality holds since X can be a constant, and the second inequality holds since X can be uniformly distributed.*
- $0 \leq H(X|Y) \leq \log_2 |\mathcal{X}|$. *Again, the bounds are tight. The first inequality holds since X can be a deterministic function of Y, and the second inequality holds since we can have $X \perp\!\!\!\perp Y$ and X uniform.*

Proof. We prove the first point below.

- $H(X) \geq 0$ is trivial. To prove $H(X) \leq \log_2 |\mathcal{X}|$, we can show that $H(U_{\mathcal{X}}) - H(X) \geq 0$, where $U_{\mathcal{X}}$ is the uniform distribution on $\mathcal{X}$.

$$\begin{aligned} H(U_{\mathcal{X}}) - H(P) &= \log_2 |\mathcal{X}| - \sum_{x \in \mathcal{X}} P(x) \log_2 \frac{1}{P(x)} \\ &= \sum_{x \in \mathcal{X}} P(x) \log_2 |\mathcal{X}| - \sum_{x \in \mathcal{X}} P(x) \log_2 \frac{1}{P(x)} \\ &= \sum_{x \in \mathcal{X}} P(x) \log_2 \frac{P(x)}{1/|\mathcal{X}|} \\ &= D(P \| U_{\mathcal{X}}) \geq 0. \end{aligned}$$

□

Definition 6.6. (Mutual information). The mutual information of random variables X and Y is given by

$$\begin{aligned} I(X;Y) &\triangleq H(X) + H(Y) - H(X,Y) \\ &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X), \end{aligned}$$

where the second and third inequalities hold automatically from the chain rule.

Remark 6.7. The semicolon between X and Y is used whenever we call the mutual information between two random variables X and Y. $I(X;Y) = I(Y;X)$.

Theorem 6.8. *(Conditioning reduces entropy). $H(X|Y) \leq H(X)$. This theorem is equivalent to stating that the mutual information is always positive, i.e., $I(X;Y) \geq 0$ for any two random variables X and Y.*

Proof. Claim: $I(X;Y) = D(P_{X,Y} \parallel P_X P_Y)$. We prove the claim:

$$\begin{aligned}
I(X;Y) &= H(X) - H(X|Y) \\
&= \sum_{y\in\mathcal{Y}} P_Y(y) \sum_{x\in\mathcal{X}} P_X(x) \log_2 \frac{1}{P_X(x)} - \sum_{y\in\mathcal{Y}} P_Y(y) \sum_{x\in\mathcal{X}} P_{X|Y}(x|y) \log_2 \frac{1}{P_{X|Y}(x|y)} \\
&= \sum_{x\in\mathcal{X},y\in\mathcal{Y}} P_{X,Y}(x,y) \log_2 \frac{1}{P_X(x)} - \sum_{x\in\mathcal{X},y\in\mathcal{Y}} P_{X,Y}(x,y) \log_2 \frac{1}{P_{X|Y}(x|y)} \\
&= \sum_{x\in\mathcal{X},y\in\mathcal{Y}} P_{X,Y}(x,y) \log_2 \frac{P_{X|Y}(x|y)}{P_X(x)} \\
&= \sum_{x\in\mathcal{X},y\in\mathcal{Y}} P_{X,Y}(x,y) \log_2 \frac{P_{X|Y}(x|y)P_Y(y)}{P_X(x)P_Y(y)} \\
&= D(P_{X,Y} \| P_X P_Y) \\
&\geq 0.
\end{aligned}$$

$I(X;Y) = 0 \iff X \perp\!\!\!\perp Y$. □

Remark 6.9. If X, Y are random variables on a common field, $\mathrm{Cov}(X,Y) = \mathbb{E}(XY) - \mathbb{E}(X)\mathbb{E}(Y)$. Mutual information often acts as a better measure than correlation. In some sense, $I(X;Y)$ measures the distance of X and Y from being independent, thus the "dependence" of X and Y. Mutual information is used throughout machine learning, computer vision, etc. for its many nice properties.

Remark 6.10. (Picture). *** See figure.

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

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

6.3 Rényi entropy

Definition 6.11. (Rényi entropy of parameter $s \geq 0$). We define the Rényi entropy as

$$H_s(P) = \begin{cases} \log_2 |\{x \in \mathcal{X} : P_X(x) > 0\}| & \text{for } s = 0 \\ \frac{1}{1-s} \log_2 \left(\sum_{x\in\mathcal{X}} P_X^s(x)\right) & \text{for } s \in (0,1) \cap (1,\infty) \\ H(X) & \text{for } s = 1 \\ \min_{x\in\mathcal{X}} \log_2 \frac{1}{P_X(x)} & \text{for } s = \infty. \end{cases}$$

In more compact form,

$$H_s(P) := \frac{1}{1-s} \log_2 \left(\sum_{x\in\mathcal{X}} P^s(x) \right).$$

Example 6.12. Certain cases of the Rényi entropy have special names.

- $s = 0$: $H_0(P) = \log_2 |\mathcal{X}|$ is the **Hartley entropy**.

- $s = 1$: $H_1(P) = H(P)$, the **Shannon entropy**.
- $s = 2$: $H_2(P) = \log \frac{1}{\|P\|_2^2}$ is the **collision entropy**.
- $s = \infty$: $H_\infty(P) = \min_{x \in \mathcal{X}} \log \frac{1}{P(x)}$ is the **min entropy**.

Remark 6.13. (Intuition behind "collision" entropy). Note that for X, Y i.i.d. from P,

$$\mathbb{P}(X = Y) = \sum_z \mathbb{P}(X = z)\mathbb{P}(Y = z) = \sum_{z \in \mathcal{X}} P^2(z) = \|P\|_2^2.$$

Remark 6.14. Properties of Rényi entropy of parameter $s \geq 0$:

- $0 \leq H_s(P) \leq \log_2 |\mathcal{X}|$. The first inequality is tight iff P is deterministic (except $s = 0$), and the second inequality is tight iff P is uniform (except $s = 0$).
- $H_s(P) \leq H_t(P) \ \forall \ s \geq t$.
- In fact, $0 \leq H_\infty(P) \leq H_2(P) \leq H_1(P) \leq H_0(P) \leq \log_2 |\mathcal{X}|$.
- The chain rule applies to the Rényi entropy only with $s = 1$.

6.4 Estimation

In this section, we will use the following notation.

- $X_1 \oplus X_2$ denotes $X_1 + X_2 \pmod 2$.
- $H(p)$ refers to the entropy of the distribution $\text{Ber}(p)$.

Let X and Y be two random variables defined on the finite sets $\mathcal{X}$ and $\mathcal{Y}$ under the distributions P_X and P_Y, respectively. Let $(X, Y) \sim P_{XY}$. The optimal guess for the value of X would be

$$X^* = \arg\max_{x \in \mathcal{X}} P_X(x).$$

The probability of making an error with this guess is defined by

$$\begin{aligned} \mathbb{P}_e(X) &= 1 - \max_{x \in \mathcal{X}} P_X(x) \\ &= 1 - 2^{-H_\infty(X)}, \end{aligned}$$

since

$$\begin{aligned} \mathbb{P}_e(X) &= 1 - 2^{\log_2 \max_{x \in \mathcal{X}} P_X(x)} \\ &= 1 - 2^{-\min_{x \in \mathcal{X}} \log_2 \frac{1}{P_X(x)}} \\ &= 1 - 2^{-H_\infty(P_X)}. \end{aligned}$$

Now, upon observing $Y = y$, the decision that minimizes the probability of guessing X incorrectly is

$$X^*(y) = \arg\max_{x \in \mathcal{X}} P_{X|Y}(x|y),$$

and

$$\begin{aligned}\mathbb{P}_e(X|Y=y) &= 1 - \max_{x\in\mathcal{X}} P_{X|Y}(x|y) \\ &= 1 - 2^{-H_\infty(X|Y=y)}.\end{aligned}$$

Let

$$\mathbb{P}_e(X|Y) = \sum_{y\in\mathcal{Y}} \mathbb{P}_e(X|Y=y)\mathbb{P}(Y=y).$$

By Jensen's inequality[1], and since $1 - 2^{-x} \le \ln 2 \cdot x$,

$$\begin{aligned}\mathbb{P}_e(X|Y) &= 1 - 2^{-H_\infty(X|Y)} \\ &= 1 - 2^{-\sum_{y\in\mathcal{Y}} H_\infty(X|Y=y)P_Y(y)} \\ &\le 1 - 2^{-\sum_{y\in\mathcal{Y}} H(X|Y=y)P_Y(y)} \\ &= 1 - 2^{-H(X|Y)} \\ &\le \ln 2 \cdot H(X|Y) \\ &\le H(X|Y).\end{aligned}$$

Thus,

$$\mathbb{P}_e(X|Y) \le H_\infty(X|Y) \le H(X|Y).$$

Hence,

$$H(X|Y) \text{ small} \implies \mathbb{P}_e(X|Y) \text{ small}.$$

We now would like to play the following guessing game: let X_1 and X_2 be i.i.d. on $\{0,1\}$ under the distribution $\mathrm{Ber}(p)$, where $0 < p < \frac{1}{2}$. The optimal guessing strategy for guessing X_1 given the outcome of X_2 (say $X_2 = a$, where $a \in \{0,1\}$) has error probability $\mathbb{P}_e(X_1|X_2 = a) = p$ because X_1 and X_2 are independent random variables. Assume that the game allows us to take any one-to-one map $(X_1, X_2) \to (Y_1, Y_2)$ and query either Y_1 or Y_2 but not both. We are expected to guess the other. Do we have a chance of decreasing our conditional error probability?

Example 6.15. Let $f(X_1, X_2) = (X_1 \oplus X_2, X_2)$, and query $Y_1 = X_1 \oplus X_2$ to guess Y_2. Say $Y_1 = 0$. Then, guess $Y_2 = 0$ since

$$\mathbb{P}(X_2 = 0|X_1 \oplus X_2 = 0) = \frac{(1-p)^2}{p^2 + (1-p)^2} > \frac{p^2}{p^2 + (1-p)^2} = \mathbb{P}(X_2 = 1|X_1 \oplus X_2 = 0).$$

The conditional error probability becomes

$$\mathbb{P}(X_2 = 1|X_1 \oplus X_2 = 0) = \frac{p^2}{p^2 + (1-p)^2} < p.$$

Note that although the above mapping decreases one of the conditional error probabilities, it increases the other conditional error probability, and the resulting average error probability remains p.

[1](Jensen's inequality). Let X be a random variable and $f : \mathcal{X} \subseteq \mathbb{R} \to \mathbb{R}$ be a concave function. Then, $\mathbb{E}(f(X)) \le f(\mathbb{E}(X))$.

Remark 6.16. Note that $H(f(X)) \leq H(X)$ for any random variable X and function f, with equality only when f is a one-to-one map (see Problem Set 3). We have that $H(X_1, X_2) = 2H(p)$ since X_1 and X_2 are independent and $H(X_1|X_2) = H(X_1)$. However, since

$$H(X_1 \oplus X_2) = H(2p(1-p)) > H(p),$$

for $p \neq 0, \frac{1}{2}, 1$, we have

$$\begin{aligned} H(X_1|X_1 \oplus X_2) &= H(X_1, X_1 \oplus X_2) - H(X_1 \oplus X_2) \\ &= H(X_1, X_2) - H(X_1 \oplus X_2) \\ &< H(p). \end{aligned}$$

Now, we are ready to discuss polar codes.

7 Polar codes

Our goal with polar codes is to achieve $H(P)$ for lossless compression over DMS *efficiently*. We will use the convention $+$ to denote $\oplus$.

7.1 Introduction

We consider a DMS $(X_1, \ldots, X_n)$ drawn i.i.d from Ber(p). Recall that our conclusion with AEP was that only $nH(P)$ bits are needed to encode the DMS.

$\underline{n = 1}$:
We have lossy compression as we can only store X_1.

$\underline{n = 2}$:
Let $X^2 = (X_1, X_2)$ and $U^2 = (U_1, U_2) = (X_1 + X_2, X_2)$. Now, imagine that you decide to store a function $f(X_1, X_2)$ valued in $\{0, 1\}$. Assume that $f(X_1, X_2) = X_1 + X_2$. In other words, we are creating a dependency between (X_1, X_2) and X_2, stored in f.

Question 7.1. Upon observing $X_1 + X_2 = Y$, can we guess "better" than by simply observing X_1 and guessing X_2? What does it mean to do "better?"

Remark 7.2. *** See picture. Note that $H(X_1, X_2) = H(X_1 + X_2, X_2) = H(X_1) + H(X_2)$ because there is a one-to-one map $(X_1, X_2) \iff (X_1 + X_2, X_2)$. If we use the chain rule in the opposite direction, we have

$$H(X_1, X_2) = H(X_1 + X_2, X_2) = H(X_1 + X_2) + H(X_2|X_1 + X_2)$$

$$\implies 2H(p) \equiv H(X_1, X_2) = H(U_1, U_2) = H(U_1) + H(U_2|U_1).$$

It is easy to see that $H(U_1) > H(p)$, so there exists a $\delta > 0$ such that $H(U_1) = H(p) + \delta$ and $H(U_2|U_1) = H(p) - \delta$. Thus, we can always guess U_2 given U_1 with probability better than p. We can write $U^2 = G_2 X^2$, where $G_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$.

$\underline{n = 4}$:
We now extend our analysis to $n = 4$; from here, we hope to see the general picture. Let $X^4 = (X_1, X_2, X_3, X_4)$ be drawn i.i.d. from Ber(p), and take

$$\begin{aligned} U^4 &= (U_1, U_2, U_3, U_4) \\ &= (X_1 + X_2 + X_3 + X_4, X_2 + X_4, X_3 + X_4, X_4) \\ &= G_4 X^4, \end{aligned}$$

where

$$G_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Define intermediate vectors V^2 and W^2 such that

$$\begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = G_2 \begin{pmatrix} X_1 \\ X_3 \end{pmatrix}, \ \begin{pmatrix} W_1 \\ W_2 \end{pmatrix} = G_2 \begin{pmatrix} X_2 \\ X_4 \end{pmatrix}.$$

Thus, we have

$$\begin{pmatrix} U_1 \\ U_2 \end{pmatrix} = G_2 \begin{pmatrix} V_1 \\ W_1 \end{pmatrix}, \ \begin{pmatrix} U_3 \\ U_4 \end{pmatrix} = G_2 \begin{pmatrix} V_2 \\ W_2 \end{pmatrix}.$$

From here, note that

$$H(U_1) + H(U_2|U_1) = H(U_1, U_2) = H(V_1, W_1) = 2H(V_1),$$

since (U_1, U_2) can be obtained from (V_1, W_1), and V_1 and W_1 are i.i.d. Similarly,

$$H(U_3|U^2) + H(U_4|U^3) = H(U_3, U_4|U^2) = H(V_2, W_2|V_1, W_1) = 2H(V_2|V_1).$$

Finally,

$$H(V_1) + H(V_2|V_1) = H(V_2, V_1) = H(X_3, X_1) = 2H(p).$$

These equations imply that there exist two values $\delta_1, \delta_2 > 0$ such that $H(U_1) = H(V_1) + \delta_1$, $H(U_2|U_1) = H(V_1) - \delta_1$, $H(U_3|U^2) = H(V_2|V_1) + \delta_2$, and $H(U_4|U^3) = H(V_2|V_1) - \delta_2$. From case $n = 2$, we know that $H(V_1) = H(p) + \delta$ and $H(V_2|V_1) = H(p) - \delta$, so we obtain the following equations:

$$H(U_1) = H(p) + \delta + \delta_1, \ H(U_2|U_1) = H(p) + \delta - \delta_1,$$

$$H(U_3|U^2) = H(p) - \delta + \delta_2, \ H(U_4|U^3) = H(p) - \delta - \delta_2.$$

Intuition, 02/23, end of class. Fire analogy. ***

$\underline{n = 8}$:

$$G_8 = \begin{pmatrix} G_4 & G_4 \\ 0 & G_4 \end{pmatrix}.$$

$\underline{n = 2^m}$:

$$G_n = \begin{pmatrix} G_{n/2} & G_{n/2} \\ 0 & G_{n/2} \end{pmatrix} = G_2^{\otimes m},$$

$$G_2 \otimes G_2 = G_2^{\otimes 2} = \begin{pmatrix} G_2 & G_2 \\ 0 & G_2 \end{pmatrix},$$

$$G_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

7.2 Polar codes ***

Definition 7.3. (G_n). We define the $n \times n$ matrix G_n, where n is a power of 2, as

$$G_n = \begin{pmatrix} G_{n/2} & G_{n/2} \\ 0 & G_{n/2} \end{pmatrix},$$

with

$$G_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Example 7.4. (Tensor operation.) $G_4 = G_2 \otimes G_2 = G_2^{\otimes 2}$.

Remark 7.5. We can equivalently define $G_{2^m} = G_2^{\otimes m}$.

Let n be a power of 2. For polar codes, we start with a source $(X_1, \dots, X_n)$ with each element drawn i.i.d. from $\text{Ber}(p)$ and

$$\begin{pmatrix} U_1 \\ \vdots \\ U_n \end{pmatrix} = G_m \begin{pmatrix} X_1 \\ \vdots \\ X_n \end{pmatrix}.$$

We wish to take a subset of rows of G_m to get a flat matrix.

Definition 7.6. We define the notation

$$U^i = \begin{pmatrix} U_1 \\ \vdots \\ U_i \end{pmatrix}.$$

Remark 7.7. $H(X|Y) = 0$ if and only if X is a determinstic function of Y.

Remark 7.8. Look at $H(U_i|U^{i-1})$. If, for some i, $H(U_i|U^{i-1}) \approx 0$, then we can remove row i from G_m since U_i is a deterministic function of U^{i-1}, so we do not need to store U_i.

Remark 7.9. G_m is invertible, so by the chain rule, we have

$$nH(p) = H(X^n) = H(U^n) = \sum_{i=1}^{n} H(U_i|U^{i-1}).$$

We refer to this equation as the **"balance equation."** We will now show that when n is large, $H(U_i|U^{i-1})$ tends to either 0 or 1.

Theorem 7.10. *Let $(X_1, \dots, X_n) = X^n \overset{i.i.d.}{\sim} Ber(p)$, where n is a power of 2. Let $U^n = G_n X^n$, where $G_n = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{\otimes \log_2 n}$. Let $\varepsilon \in (0, 1/2)$. Then,*

1. $\frac{1}{n} |\{i \in [n] \ : \ H(U_i|U^{i-1}) \in (\varepsilon, 1-\varepsilon)\}| \xrightarrow{n\to\infty} 0$.
2. *Going to the extremes is called **polarization**. Moreover, this is still true for $\varepsilon = \varepsilon_n = 2^{-n^{0.49}}$.*

Definition 7.11. Let $\varepsilon \in (0, 1)$ and n be a power of 2. Define

$$R_{\varepsilon,n}(p) := \left\{ i \in [n] \ : \ H(U_i|U^{i-1}) \geq \varepsilon \right\}.$$

Corollary 7.12.

$$\frac{1}{n} |R_{\varepsilon,n}(p)| \to H(p).$$

7.3 Use of the theorem ***

7.3.1 Construction of polar codes

7.3.2 Decoding polar codes

7.3.3 Complexity

7.4 Polar compressor and decompressor ***

Definition 7.13. (Polar compressor). Given $X^n \overset{i.i.d.}{\sim} \mathrm{Ber}(p)$, compute $U^n = G_n X^n$ and store $U^n[R_{\varepsilon,n}(p)] = \{U_i : i \in R_{\varepsilon,n}(p)\}$.

Definition 7.14. (Polar decompressor). If $i \notin R_{\varepsilon,n}(p)$, $P_\ell(U_i|U^{i-1}) \leq H(U_i|U^{i-1}) < \varepsilon$. Sequentially assign U_i for $i \notin R_{\varepsilon,n}(p)$ to its most likely value given the past.

$$\hat{U}_i := \arg\max_{U_i \in [0,1]} \mathbb{P}(U_i = u_i \mid U^{i-1} = u^{i-1}).$$

This gives $\hat{U}^n$, from which we compute $\hat{X}^n = G_n^{-1} U_n$.

Remark 7.15. $G_n^{-1} = G_n$. We check this:

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Corollary 7.16.

$$\mathbb{P}(\hat{X}^n \neq X^n) \xrightarrow{n\to\infty} 0 \text{ as fast as } 2^{-n^{0.49}}.$$

$$\mathbb{P}(\hat{X}^n \neq X^n) \leq n\varepsilon.$$

Remark 7.17. This also gives a lossless compressor achieving $H(p)$ for norm exp length.

Main improvement: The encoding and decoding complexity of polar codes is $\mathcal{O}(n \log_2 n)$.

8 Lempell-Ziv LZ77 algorithm

We cover the notation that we will use in this section.

8.1 Notation and algorithm

- $\mathcal{X}$ denotes the source alphabet.
- $x_m^n := (x_m, x_{m+1}, \ldots, x_n)$.

For example, one potential source string from the alphabet $\mathcal{X} = \{a, b, c, d\}$ could be:

a c d b c d a c b a b a c d b c a b a b d c.

For this string, $x_2^5 =$ (c d b c).

Definition 8.1. (Algorithm).

1. Set the window to have the first w symbols, where $w \in \mathbb{Z}_+,\ w \geq 1$.
2. Encode the window symbols using $\lceil \log_2 |\mathcal{X}| \rceil$ bits for each symbol using a basic binary encoder for the $|\mathcal{X}|$ symbols.
3. Set the "pointer" $p = w$. The pointer keeps track of the index of the last symbol that has been encoded.
4. Find the largest $n \geq 2$ such that there exists $u \in \{1, \ldots, w\}$ with $x_{p+1}^{p+n} = x_{p+1-u}^{p+n-u}$. If there is no such (n, u), then set and encode $n = 1$ and then encode x_{p+1} using the basic binary code. If such an $n \geq 2$ exists, then encode n using unary-binary code (to be defined later), using $\lceil \log_2 w \rceil$ bits. Thus, encode (n, u) if $n \geq 2$ and encode $(n = 1, x_{p+1})$ if $n = 1$.
5. Set $p = p + n$, and go back to step 4.

Definition 8.2. (Unary-binary code). A prefix-free code for the positive integers.

1. Begin the codeword with $\lfloor \log_2(n) \rfloor$ zeros.
2. Append the base 2 expansion of n to the prefix from step 1.

Example 8.3. Let $\mathcal{X} = \{a, b, c, d\}$. We examine the example given above.

1. Set the window to be the first w symbols. For instance, we can take $w = 16$.
2. Assume that we have the following $\log_2 |\mathcal{X}| = 2$ bit code for each letter in our alphabet:
   - $a \to 00$
   - $b \to 01$
   - $c \to 10$
   - $d \to 11$,

n	0-prefix	Base 2 expansion	Code
1		1	1
2	0	10	010
3	0	11	011
4	00	100	00100
5	00	101	00101
6	00	110	00110
7	00	111	00111
8	000	1000	0001000
⋮	⋮	⋮	⋮

Table 1: Unary binary code. ***

so $x = 00101101....$

3. Set the pointer: $p = 16$.

4. We find a match (a b a): $(n, u) = (3, 7)$. We encode n as 011 and u as 0111 using unary binary code. So now, x so far is 00101101...0110111....

8.2 Achieving $H(P)$ ***

Question 8.4. Why does Lempell-Ziv achieve $H(P)$ for a DMS?

Remark 8.5. Let w be the window size, and let $a \in \mathcal{X}$. We want to compress $x_1, x_2, \ldots$ from DMS(p).

$$N_a = |\{i \in [w] \ : \ x_i = a\}| .$$

Note that

$$\mathbb{E}(N_a) = wp(a),$$

so by LLN,

$$N_a \approx wp(a).$$

Let $a^n = \in \mathcal{X}^n$ for $n \geq 1$. Assuming that w is large enough,

$$N_{a^n} \approx wp^n(a^n).$$

For typical a^n, and assuming n is large enough,

$$N_{a^n} \approx w2^{-nH(p)}.$$

When is $N_{a^n} \geq 1$, i.e., when do we start seeing a^n in the window?

$$\begin{aligned} w2^{-nH(p)} &\approx 1 \\ \iff w &\approx 2^{nH(p)} \\ \iff \log w &\approx nH(p) \\ \iff \frac{\log w}{H(p)} &\approx n. \end{aligned}$$

How many bits are required?

- For the $u \in \{1, \ldots, w\}$, we require $\log_2(w)$ bits.
- For encoding n with binary unary code, we need $2\log_2(n) + 1 \approx 2\log\left(\frac{\log w}{H(p)}\right) + 1$ bits.
- Together, we require (order of magnitude) $\log(w) + 2\log\log(w)$ bits.

With this section, we conclude our discussion of lossless compression.

9 Information transmission and channel coding ***

9.1 Roadmap of digital communication ***

9.2 Summary of lossless compression

9.2.1 Huffman codes

- One shot problem: $H(P) \leq \overline{L}^*(P) \leq H(P) + 1$.
- DMS: $\frac{\overline{L}^*(P^n)}{n} \to H(P)$.
- However, encoding and decoding requires indexing, which is infeasible for efficient algorithms.

9.2.2 AEP codes

- AEP is more of a proof technique to show that good source codes exist (good in the sense of entropy achieving).
- Encoding and decoding requires indexing the typical set, which is not efficient or practical.

9.2.3 Source polar codes

- Efficient but almost lossless.

Remark 9.1. Note that almost lossless codes can be turned into fully lossless codes. After compressing x^n to y^m, the encoder tries decoding y^m to see if it can get x^n back. If x^n is successfully recovered from y^m, then the encoder stores y^m (i.e., the encoder stores the compressed version). If x^n is not successfully recovered from y^m, then the encoder stores x^n instead (i.e., if there is error in recovering, the encoder does not compress). Since the code is almost lossless, the probability of bad x^n must be vanishing. Therefore, encoder performance is not affected. So in the case of source polar codes, this encoding scheme achieves

$$\frac{1}{n}\left(\left(1 - e^{-n^{0.49}}\right) nH(P) + o(n) + e^{-n^{0.49}} n\right) \to H(P).$$

9.3 Notation

In these notes,

- $\mathcal{X}$ is a finite set that denotes the input alphabet.
- $\mathcal{Y}$ is also a finite set, and it denotes the output alphabet.
- m denotes a message to be transmitted.
- M denotes the total number of messages.
- $G^{\perp} = H$ is known as the **parity check matrix**.

9.4 Channel coding ***

9.5 Discrete memoryless channel (DMC) ***

9.6 (M, n)-channel code ***

9.7 Source-channel duality ***

We cover the duality between data transmission and compression.

First, consider the binary symmetric channel BSC(p). Let $\mathcal{X} = \mathcal{Y} = \{0, 1\}$, where $\mathcal{X}$ is the input alphabet and $\mathcal{Y}$ is the output alphabet. Let $x \in \{0, 1\}$ be the input to BSC(p). Then, the output is $Y = x + Z$, where $Z \sim \text{Ber}(p)$. Note that BSC(p) flips the input with probability p. BSC(p) acts on blocks of n inputs independently on each input.

Assume that $p \in (0, 1/2)$. The idea of channel coding is to only consider a subset of possible sequences x^n to be transmitted and to choose these sequences such that they are far apart. Let M be the number of codewords. The upper bound is 2^n.

The codeword will be of length n, and as n tends to infinity, we want to be able to tell them apart. Two codewords is trivial—we can take the vector of all 0s and the vector of all 1s. For a few finite number of codewords, it is not too difficult to pick them far apart.

Question 9.2. How fast can M grow with n such that the probability of not recovering any codeword in our codebook is vanishing as $n \to \infty$? We can actually achieve $M = 2^{nR}$, where $R < 1 - H(p)$. The larger M we can tolerate, the better for us, as we can pack more codewords.

Let M be the number of codewords that we want to transmit, and let $k := \log_2 M$. Assume that $k \in \mathbb{Z}_+$. Take a linear code, i.e., each codeword is the output of a linear map $C : \mathbb{F}_2^k \to \mathbb{F}_2^n$. So

$$x^n = Au^k,$$

where $A \in \mathbb{R}^{n \times k}$, i.e., the codebook is the linear space spanned by the columns of A.

If we use a linear code on BSC(p), we receive at the output

$$Y^n = x^n + Z^n,$$

$$Y^n = An^k + Z^n.$$

Applying $A^\perp = B \in \mathbb{R}^{(n-k) \times k}$, we get

$$BY^n = 0 + BZ^n,$$

since there exists $A^\perp$ such that $A^\perp A = 0 \in \mathbb{R}^{k \times k}$.

Hence, if we can reconstruct Z^n with $p \to 1$ as $n \to \infty$ from BZ^n, then we can recover(?) x^n with $p \to 1$ as $n \to \infty$.

$$m = n - k, m = nH(p) + o(n)$$

Definition 9.3. (Program).

- Take B as the polar code matrix for DMS(p).
- Compute $A = B^\perp$, and define the codebook as all elements in the image of A.

- $M = 2^{nR}$, where $R = 1 - H(p) - o(1)$.

Midterm might have Q teasing transmission with compression ***.

Remark 9.4. Last time, we covered source compression, which was almost lossless and with a linear code. We can design a matrix A such that for Z^n i.i.d. Ber(p), we have

1. $P_e(Z^n|AZ^n) \xrightarrow{n\to\infty} 0$
2. $m := \text{rank}(A) = nH(p) + o(n)$.

Note that $Z^m = AZ^n$, since $A \in \mathbb{R}^{m\times n}$. We can apply this to a BSC(p):

$$Y^n = x^n + Z^n \xrightarrow{A} \hat{Y}^n = AZ^n \to Z^n \to x^n.$$

$\ker(A) = (1 - H(p))n + o(n)$ is the dimension of the code.

9.8 Channel coding theorem

Definition 9.5. Let $\mathcal{X}$ and $\mathcal{Y}$ be two finite sets called, respectively, the input and output alphabets. Let $P_{Y|X}$ be a one-shot channel from $\mathcal{X}$ to $\mathcal{Y}$, i.e., $P_{Y|X}(\cdot|x)$ in a probability distribution on $\mathcal{Y}$ for all $x \in \mathcal{X}$. We can represent $P_{Y|X}$ as a stochastic matrix of dimension $|\mathcal{X}| \times |\mathcal{Y}|$. (Recall that a stochastic matrix is a matrix with nonnegative entries and rows that sum up to 1.)

The one-shot channel is (*** See Table on p. 3 in Notes 8). Now, we extend the one-shot channel to a memoryless channel over blocks of n symbols:

$$\forall n \geq 1,\ x^n \in \mathcal{X}^n,\ y^n \in \mathcal{Y}^n,\ P_{Y^n|X^n}(y^n|x^n) := \prod_{i=1}^{n} P_{Y|X}(y_i|x_i).$$

This is called a discrete memoryless channel (DMC). Often, we simply call this a "channel."

Definition 9.6. $M, n \geq 1$ positive integers. An (M, n)-channel code (or, simply, code) is a pair of an encoder E and a decoder D such that

$$E : [M] \to \mathcal{X}^n,$$

$$D : \mathcal{Y}^n \to [M].$$

Thus, the input and output symbols do not need to be the same.

Definition 9.7. (Error probability). Define the error probability for a given message $m \in [M]$ as

$$P_{e,m} := \sum_{y^n \in \mathcal{Y}^n,\ D(y^n)\neq m} P_{Y^n|X^n}(y^n|E(m)).$$

Note that this is defined for a specific code (E, D).

The average error probability is defined as

$$P_{e,\text{avg}} = \frac{1}{M} \sum_{m\in[M]} P_{e,m}.$$

The maximum error probability is

$$P_{e,\max} = \max_{m\in[M]} P_{e,m}.$$

We eventually want this error probability to go to 0.

Definition 9.8. (Rate). A rate $R \in \mathbb{R}_+$ is achievable for reliable communication over a DMC if there exists an (M, n)-code such that

1. $\frac{\log_2 M}{n} \xrightarrow{n\to 0} R$. $M = 2^{nR+o(n)}$.
2. $P_{e,\max} \xrightarrow{n\to\infty} 0$.

Definition 9.9. (Capacity). The capacity of a DMC $(P_{Y|X})$ is denoted $C(P_{Y|X})$ and is the supremum of all achievable rates.

Theorem 9.10. *((Shannon 1948) Channel coding theorem). For any DMC $(P_{Y|X})$,*

$$C(P_{Y|X}) = \max_{P_X \text{ p.d. on } \mathcal{X}} I(P_X; P_{Y|X}).$$

(Alternatively, we can express the RHS as $I(X;Y)$ as this implies that we are given joint distribution $P_{X,Y} = P_X P_{Y|X}$.) This is perhaps the most important theorem in information theory.

Remark 9.11. (Combinatorial side comment).

$$\binom{n}{\lceil np\rceil} = 2^{nH(p)+o(n)}.$$

Recall the theorem from the one-shot formula:

$$\begin{aligned} I(P_X, P_{Y|X}) &:= \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} P_{Y|X}(y|x)P_X(x)\log_2 \frac{P_{Y|X}(y|x)}{\sum_{u\in\mathcal{X}} P_{Y|X}(y|u)P_X(u)} = D(P_{X,Y}\|P_X P_Y) \\ &= I(X;Y) \text{ for } (X,Y)\sim P_X P_{Y|X} \\ &= H(X) - H(X|Y). \end{aligned}$$

Example 9.12. 1. BSC(p): $Y = X + Z$, where $Z \sim \text{Ber}(p)$. $I(X;Y) = H(Y) - H(Y|X) = H(X+Z) - H(Z) \le 1 - H(P)$ if X uniform on $\{0,1\}$.

Remark 9.13.

$$H(X+Z|X) = H(Z|X) = H(Z).$$
$$P_{X+Z}(u|x) = P_Z(u+x).$$

Definition 9.14. (Binary erasure channel). BEC(p), where $p \in [0,1]$.

$$I(X;Y) = H(Y) - H(Y|X) \le 1 - \sum_{i\in\{0,1\}} H(Y|X=i)P(X=i) = 1-p.$$

$C(\text{BEC}(p)) = 1-p$.