A Theorist's Toolkit **(CMU 18-859T, Fall 2013)**

Lecture 20: Information Theory

November 13th, 2013

Lecturer: Ryan O'Donnell *Scribe: Kevin Su*

1 Introduction

Today, we cover some of the basics of information theory. Developed by Shannon in 1948, he was motivated by its applications to showing limits on the compressibility of data. Since then, information theory has found a wide range of applications, including coding theory, LP hierarchies, and quantum computing.

In this lecture, we'll cover the basic definitions of entropy, mutual information, and the Kullback-Leibler divergence. Along the way, we'll give some intuitive reasoning behind these values in addition to the formulas. Lastly, we'll end with an application to communication complexity.

2 Entropy

For information theory, the fundamental value we are interested in for a random variable $\mathbf{X}$ is the entropy of $\mathbf{X}$. We'll consider $\mathbf{X}$ to be a discrete random variable. The entropy can be thought of as any of the following intuitive definitions:

1. The amount of randomness in $\mathbf{X}$ (in bits)
2. Minimum number of random bits you need to generate a draw from $\mathbf{X}$ (on average)
3. Average number of bits you need to store a draw from $\mathbf{X}$ with compression
4. Minimum number of bits needed for Alice to communicate one draw from $\mathbf{X}$ to Bob
5. Number of yes/no questions needed on average to guess a draw from $\mathbf{X}$

There is one caveat - all of the above intuitive definitions are valid if we are in a "perfect world." That is, we'd like to avoid the annoyance with floors on fractions of bits and so forth. We deal with this later, and first we give the mathematical definition for entropy.

Definition 2.1. Entropy is defined as $\mathrm{H}(\mathbf{X}) = \sum_{x \in \mathrm{range}(\mathbf{X})} \mathrm{p}_{\mathbf{X}}(x) \cdot \log_2 \frac{1}{\mathrm{p}_{\mathbf{X}}(x)}$

where $\mathrm{p}_{\mathbf{X}}(x) = \mathbf{Pr}[X = x]$, which we use for notational convenience.

Example 2.1. Let us define $\mathbf{X}$ as follows:

$$\mathbf{X} = \begin{cases} \text{red} & \text{with probability } \frac{1}{2} \\ \text{green} & \text{with probability } \frac{1}{4} \\ \text{blue} & \text{with probability } \frac{1}{8} \\ \text{yellow} & \text{with probability } \frac{1}{8} \end{cases}$$

Note we use colors as the discrete values to avoid confusion with numbers. We could think about an efficient program to generate draws from $\mathbf{X}$. By efficient, we refer to the number of random bits used. For example, if we are given a random coin to flip (denote heads as H and tails as T), one might use the following procedure:

Flip 1		Flip 2		Flip 3		
H	$\to$	red				
T	$\to$	H	$\to$	green		
		T	$\to$	H	$\to$	blue
				H	$\to$	yellow

In other words, we flip a coin first. If it is heads, the outcome is red. Otherwise, we flip it again, and if it is heads, then we say the outcome is green. Otherwise, we flip it one last time, and we map heads to blue and tails to yellow. A quick verification shows this is a valid way of generating draws from $\mathbf{X}$.

On average, we use (note these values are the *exact* same as the terms of $\mathrm{H}(\mathbf{X})$):

$$\begin{aligned} \frac{1}{2} \cdot (1 \text{ flip}) + \frac{1}{4} \cdot (2 \text{ flips}) + \frac{1}{8} \cdot (3 \text{ flips}) + \frac{1}{8} \cdot (3 \text{ flips}) &= 1.75 \text{ flips} \\ &= \mathrm{H}(\mathbf{X}) \end{aligned}$$

This is perspective 2 from above. If we looked at perspective 3, we could store the outcomes as:

$$\begin{aligned} \text{red} &\Leftrightarrow H \\ \text{green} &\Leftrightarrow TH \\ \text{blue} &\Leftrightarrow TTH \\ \text{yellow} &\Leftrightarrow TTT \end{aligned}$$

The average number of bits needed to store a draw from $\mathbf{X}$ is 1.75 bits under this view as well. One note: this is a prefix-free encoding. In storing multiple draws, we do not need punctuation or delimiters, since no symbol is a prefix of another symbol. For example, the sequence $THTTHHH$ uniquely decodes to "green, blue, red, red."

Finally for view 5, if we wanted to ask yes/no questions to determine $x \sim \mathbf{X}$, we'd ask in order:

1. Is x red?
2. Is x green?
3. is x blue?

As expected, the average number of questions we ask is $\mathrm{H}(\mathbf{X}) = 1.75$. One comment: the questions are chosen in an ordering as to maximize the information gain here, where information gain is the reduction in entropy after asking a question.

3 Some subtleties about entropy

In the previous example, we had the nice property that all probabilities were of the form $\frac{1}{2^k}$ where $k \in \mathbb{Z} \geq 0$. This meant that all log terms were of the form: $\log_2 \frac{1}{\mathrm{p}_{\mathbf{X}}(x)} = \log_2 2^k = k$. In a sense, this is what we mean by a perfect world - we want only integer values from the log terms. The formula without floors is more elegant, but we'll later see a relation between the elegant and inelegant quantities.

For example, consider the random variable $\mathbf{X}$ defined by:

$$\mathbf{X} = \begin{cases} \text{a} & \text{with probability } \frac{1}{2} \\ \text{b} & \text{with probability } \frac{1}{3} \\ \text{c} & \text{with probability } \frac{1}{6} \end{cases}$$

As before, we'd like to state that the average number of bits used is:

$$\frac{1}{2} \cdot (1 \,\text{bit}) + \frac{1}{3} \cdot (\log_2 3 \,\text{bits}) + \frac{1}{6} \cdot (\log_2 6 \,\text{bits})$$

However, $\log_2(3)$ and $\log_2(6)$ aren't integers, which is annoying. Of course, we could use the following storage scheme if we wanted to save a random draw:

$$\begin{aligned} \text{a} &\Leftrightarrow H \\ \text{b} &\Leftrightarrow TH \\ \text{c} &\Leftrightarrow TTH \end{aligned}$$

Here, instead of using $\log_2(3)$ and $\log_2(6)$ bits, we've used $\lceil \log_2 3 \rceil$ and $\lceil \log_2 6 \rceil$ bits respectively. In general, we can store/generate/communicate a draw from $\mathbf{X}$ using the following number of bits:

$$\begin{aligned} \sum_x \mathrm{p}_{\mathbf{X}}(x) \cdot \lceil \log_2 \frac{1}{\mathrm{p}_{\mathbf{X}}(x)} \rceil &\leq \sum_x \mathrm{p}_{\mathbf{X}}(x) \cdot (\log_2 \frac{1}{\mathrm{p}_{\mathbf{X}}(x)} + 1) \\ &= \sum_x \mathrm{p}_{\mathbf{X}}(x) \cdot \log_2 \frac{1}{\mathrm{p}_{\mathbf{X}}(x)} + \sum_x \mathrm{p}_{\mathbf{X}}(x) \cdot 1 \\ &= \mathrm{H}(\mathbf{X}) + 1 \end{aligned}$$

In constructing $\mathbf{X}$ as a codebook, we've achieved an encoding within 1 bit of the optimal. Since we know we can't do better than $\mathrm{H}(\mathbf{X})$, we know we're within $+1$ of the optimal. Optimal encoding can be done with Huffman encoding (recall you construct a priority queue of trees based on the frequencies of the elements, then merge the two least frequent and so forth to build a binary tree).

Lastly, we present some bounds on the entropy of $\mathbf{X}$. Let $|\operatorname{range}(\mathbf{X})| \leq N$, then:

$$0 \leq \mathrm{H}(\mathbf{X}) \leq \log_2(N)$$

The lower bound is an equality when $\mathbf{X}$ is constant, and the upper bound is an equality when $\mathbf{X}$ is the uniform distribution over $\operatorname{range}(\mathbf{X})$ (when $\operatorname{range}(\mathbf{X}) = N$), which can be proved using Jensen's inequality.

4 Relation between elegant and inelegant quantities

Before we deal with the issue at hand, it will help to analyze the entropy of the joint distribution of $(\mathbf{X}, \mathbf{Y})$, where $\mathbf{X}$ and $\mathbf{Y}$ are independent random variables. We can view $(\mathbf{X}, \mathbf{Y})$ as a random variable. For simplicity, we write it as $\mathbf{XY}$ (imagine concatenating the individual draws). Note the range of $\mathbf{XY}$ is just $\operatorname{range}(\mathbf{X}) \times \operatorname{range}(\mathbf{Y})$.

Theorem 4.1. *If* $\mathbf{X}$ *and* $\mathbf{Y}$ *are independent random variables, then* $\mathrm{H}(\mathbf{XY}) = \mathrm{H}(\mathbf{X}) + \mathrm{H}(\mathbf{Y})$.

Proof. We can prove this by just writing out the definitions and playing with the log terms.

$$\begin{aligned}
\mathrm{H}(\mathbf{XY}) &= \sum_{x,y \in \operatorname{range}(\mathbf{XY})} \mathrm{p}_{\mathbf{XY}}(x,y) \cdot \log_2 \frac{1}{\mathrm{p}_{\mathbf{XY}}(x,y)} \\
&= \sum_x \sum_y \mathrm{p}_{\mathbf{X}}(x)\,\mathrm{p}_{\mathbf{Y}}(y) \cdot \log_2 \frac{1}{\mathrm{p}_{\mathbf{X}}(x)\,\mathrm{p}_{\mathbf{Y}}(y)} && \text{by independence of } \mathbf{X}, \mathbf{Y} \\
&= \sum_x \sum_y \mathrm{p}_{\mathbf{X}}(x)\,\mathrm{p}_{\mathbf{Y}}(y) \cdot \left[\log_2 \frac{1}{\mathrm{p}_{\mathbf{X}}(x)} + \log_2 \frac{1}{\mathrm{p}_{\mathbf{Y}}(y)} \right] && \text{break log} \\
&= \sum_x \mathrm{p}_{\mathbf{X}}(x) \log_2 \frac{1}{\mathrm{p}_{\mathbf{X}}(x)} \cdot \sum_y \mathrm{p}_{\mathbf{Y}}(y) + \sum_y \mathrm{p}_{\mathbf{Y}}(y) \log_2 \frac{1}{\mathrm{p}_{\mathbf{Y}}(y)} \cdot \sum_x \mathrm{p}_{\mathbf{Y}}(x) && \text{split summations} \\
&= \sum_x \mathrm{p}_{\mathbf{X}}(x) \log_2 \frac{1}{\mathrm{p}_{\mathbf{X}}(x)} \cdot 1 + \sum_y \mathrm{p}_{\mathbf{Y}}(y) \log_2 \frac{1}{\mathrm{p}_{\mathbf{Y}}(y)} \cdot 1 \\
&= \mathrm{H}(\mathbf{X}) + \mathrm{H}(\mathbf{Y}) && \square
\end{aligned}$$

In the previous section, we showed that we can encode an $x \in \mathbf{X}$ with $\mathrm{H}(\mathbf{X}) + 1$ bits on average. We present a motivation for dealing with the extra bit used. Consider the scenario where Alice is sending some number of independent copies of $\mathbf{X}$ to Bob, say $\mathbf{X}_1, \mathbf{X}_2, \ldots \mathbf{X}_n$. By induction, if she communicates them individually, we have:

$$\begin{aligned}
\mathrm{H}(\mathbf{X}_1 \mathbf{X}_2 \ldots \mathbf{X}_n) &= n \cdot \mathrm{H}(\mathbf{X}_i) && \text{by naïve transmission} \\
&\leq n \cdot (\mathrm{H}(\mathbf{X}) + 1) \\
&\leq n \cdot \mathrm{H}(\mathbf{X}) + n
\end{aligned}$$

What happened? The extra bit has blown up to an extra n bits. If we were transmitting ternary bits ($\mathbf{X}$ is the uniform distribution over $\{0, 1, 2\}$), then we know that $\mathrm{H}(\mathbf{X}) = \log_2 3 \approx 1.6$. But we've now used 2.6 bits per symbol, which is over 60% more information than the minimum.

The solution is to consider $\mathbf{X}_1, \ldots, \mathbf{X}_n$ as a single random variable (just juxtapose the individual values). Then from the previous section, we know we can send this with just the following number of bits (if we consider it as a joint random variable):

$$\mathrm{H}(\mathbf{X}_1\mathbf{X}_2 \ldots \mathbf{X}_n) + 1 = n \cdot \mathrm{H}(\mathbf{X}) + 1$$

If we look at the amortized cost over n symbols, we have for one symbol:

$$\frac{1}{n} \cdot \Big(n \,\mathrm{H}(\mathbf{X}) + 1 \Big) = \mathrm{H}(\mathbf{X}) + \frac{1}{n}$$

In some sense, as we send more and more symbols using this scheme, we have:

$$\lim_{n \to \infty} \mathrm{H}(\mathbf{X}) + \frac{1}{n} = \mathrm{H}(\mathbf{X})$$

For the ternary bit example, we can interpret this as sending the binary representation of the ternary number generated. The excess bit is some rounding error. Lastly, we note that this analysis is not done on a single copy of $\mathbf{X}$, but rather amortized over a large number of copies.

5 Mutual Information

Before, we had that $\mathbf{X}$ and $\mathbf{Y}$ were independent. What if they are dependent? In this section, we explore the notion of mutual information. How many bits on average are needed to generate a draw from $\mathbf{XY}$? We again denote this value as $\mathrm{H}(\mathbf{XY})$. This value could be smaller than $\mathrm{H}(\mathbf{X}) + \mathrm{H}(\mathbf{Y})$, which is the value if $\mathbf{X}$ and $\mathbf{Y}$ are independent.

We can think of the "amount of savings" as the number of bits we save in generating $\mathbf{XY}$ using the dependency of $\mathbf{X}$ and $\mathbf{Y}$. Then using the independent case as an upper bound (since we can only save bits), we have the following:

Definition 5.1. The mutual information between $\mathbf{X}$ and $\mathbf{Y}$ is denoted as:

$$\mathrm{I}(\mathbf{X}; \mathbf{Y}) = \mathrm{H}(\mathbf{X}) + \mathrm{H}(\mathbf{Y}) - \mathrm{H}(\mathbf{XY})$$

We can write out the individual formulas and play with the log's to get the following:

$$\begin{aligned}
\mathrm{I}(\mathbf{X}; \mathbf{Y}) &= \mathrm{H}(\mathbf{X}) + \mathrm{H}(\mathbf{Y}) - \mathrm{H}(\mathbf{XY}) \\
&= \sum_{x,y} \mathrm{p}_{\mathbf{XY}}(x, y) \cdot \Big(\log_2 \frac{1}{\mathrm{p}_{\mathbf{X}}(x)} + \log_2 \frac{1}{\mathrm{p}_{\mathbf{Y}}(y)} - \log_2 \frac{1}{\mathrm{p}_{\mathbf{XY}}(x, y)} \Big) \\
&= \sum_{x,y} \mathrm{p}_{\mathbf{XY}}(x, y) \cdot \Big(\log_2 \frac{\mathrm{p}_{\mathbf{XY}}(x, y)}{\mathrm{p}_{\mathbf{X}}(x)\,\mathrm{p}_{\mathbf{Y}}(y)} \Big)
\end{aligned}$$

Note that if $\mathbf{X}$ and $\mathbf{Y}$ are independent, we have $\mathrm{p}_{\mathbf{XY}}(x, y) = \mathrm{p}_{\mathbf{X}}(x) \cdot \mathrm{p}_{\mathbf{Y}}(y)$, so the log term is $\log_2 1 = 0$, which implies that we can't save any bits in generating $\mathbf{XY}$.

Example 5.1. Consider the following joint distribution:

x	y	$\mathrm{p}_{\mathbf{XY}}(x,y)$
$\star$	$\star$	$\frac{1}{2}$
0	a	$\frac{1}{16}$
0	b	$\frac{1}{16}$
0	c	$\frac{1}{16}$
0	d	$\frac{1}{16}$
1	a	$\frac{1}{16}$
1	b	$\frac{1}{16}$
1	c	$\frac{1}{16}$
1	d	$\frac{1}{16}$

We can calculate the entropies of $\mathbf{X}$ and $\mathbf{Y}$ individually:

$$\begin{aligned}\mathrm{H}(\mathbf{X}) &= \frac{1}{2}\cdot 1 + \frac{1}{4}\cdot 2 + \frac{1}{4}\cdot 2 \\ &= 1.5 \\ \mathrm{H}(\mathbf{Y}) &= \frac{1}{2}\cdot 1 + \sum_{i=1}^{4}\frac{1}{8}\cdot 3 \\ &= 2\end{aligned}$$

And so we have $\mathrm{H}(\mathbf{X}) + \mathrm{H}(\mathbf{Y}) = 1.5 + 2 = 3.5$. However, we can also calculate $\mathrm{H}(\mathbf{XY})$:

$$\begin{aligned}\mathrm{H}(\mathbf{XY}) &= \frac{1}{2}\cdot 1 + \frac{1}{2}\cdot 4 \\ &= 2.5\end{aligned}$$

It follows then that $\mathrm{I}(\mathbf{X};\mathbf{Y}) = 3.5 - 2.5 = 1$, which means we save one bit when we generate $\mathbf{X}$ and $\mathbf{Y}$ together. Intuitively, we can think of flipping a coin as $H \to (\star, \star)$. If not, then the two variables are then independent. Therefore, we shared 1 bit in reducing the task into two independent tasks, which is our savings.

Some facts about mutual information:

$$0 \le \mathrm{I}(\mathbf{X};\mathbf{Y}) \le \mathrm{H}(\mathbf{X}) + \mathrm{H}(\mathbf{Y})$$

Note the left side follows from the case where $\mathbf{X}, \mathbf{Y}$ are independent, and the right side follows from the case where $\mathbf{XY}$ is constant. There is in fact a better upper bound:

$$\mathrm{I}(\mathbf{X};\mathbf{Y}) \le \min\Big\{\mathrm{H}(\mathbf{X}), \mathrm{H}(\mathbf{Y})\Big\}$$

The intuition for this bound is that the most bits you can save is the case where you only have to generate one of the variables. Equality of $\mathrm{I}(\mathbf{X};\mathbf{Y}) = \mathrm{H}(Y)$ follows if $\mathbf{Y}$ is determined by $\mathbf{X}$. In other words, if we generate $\mathbf{X}$, we don't have to spend any bits generating $\mathbf{Y}$.

Example 5.2. Let $\mathbf{X} = n$ independent bits, and $\mathbf{Y} =$ parity of $\mathbf{X}$.

Then we can generate $\mathbf{X}$ using n random bits, and we get $\mathbf{Y}$ for free. Then $\mathrm{I}(\mathbf{X};\mathbf{Y}) = 1$.

Example 5.3. Let $\mathbf{X} = n$ independent bits, and $\mathbf{Y}$ = parity of $\mathbf{X}$ and one more independent bit.

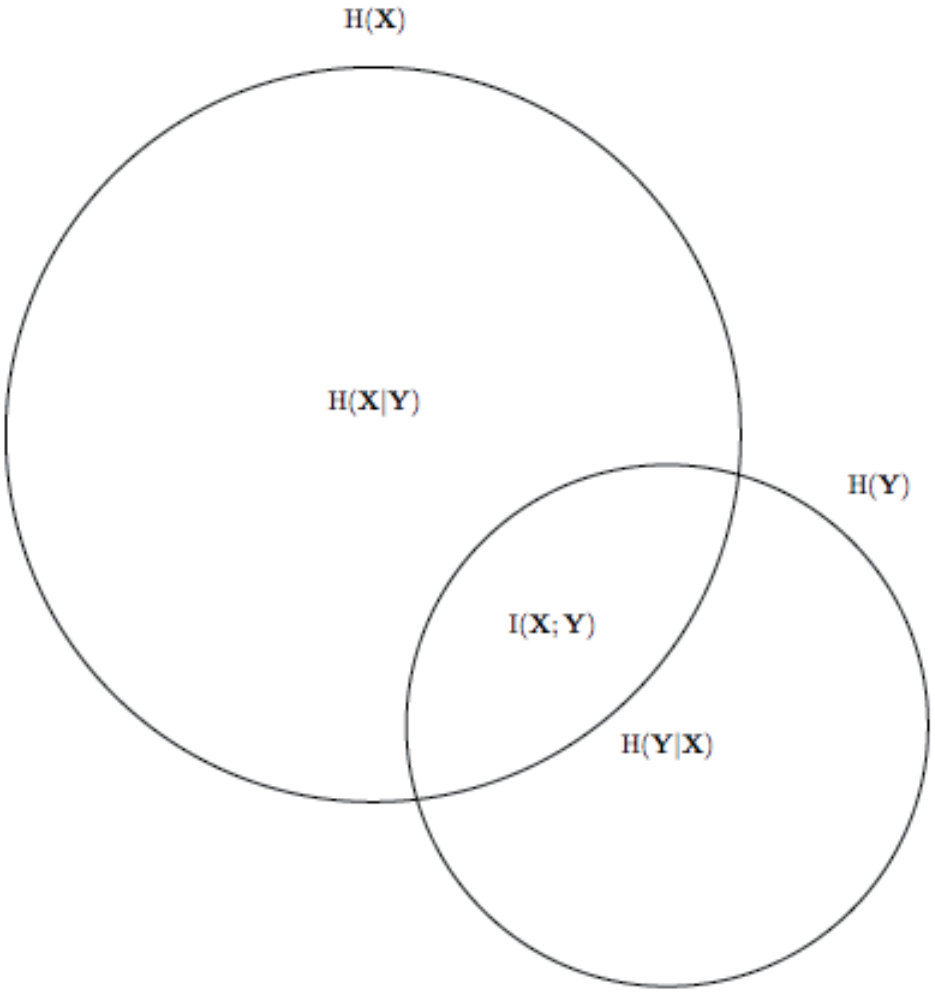
Then we can generate $\mathbf{X}$ using n random bits, and we get $\mathbf{Y}$ for free. But we still have to generate the last bit of $\mathbf{Y}$, so in total we have $\mathrm{H}(\mathbf{XY}) = n + 1$, and $\mathrm{I}(\mathbf{X}; \mathbf{Y}) = 1$. In this view, we first generate $\mathbf{X}$, then generate the remainder of $\mathbf{Y}$.

Alternatively, we could generate $n - 1$ bits of $\mathbf{X}$, then generate the parity bit of $\mathbf{Y}$, and finally the last bit of $\mathbf{Y}$. Note the final bit of $\mathbf{X}$ is determined by the $n - 1$ bits plus the parity bit. This is another view - generate $\mathbf{Y}$ first, then generate the remainder of $\mathbf{X}$.

This leads to another definition.

Definition 5.2. The entropy of $\mathbf{Y}$ conditioned on $\mathbf{X}$ is $\mathrm{H}(\mathbf{Y}|\mathbf{X}) := \mathrm{H}(Y) - \mathrm{I}(\mathbf{X}; \mathbf{Y})$.

In a Venn diagram, we can see that the conditional entropies are the "remaining" parts to generate if we generate the mutual information first. For example,



This leads to the following formula:

$$\mathrm{H}(\mathbf{Y}|\mathbf{X}) = \mathrm{H}(\mathbf{XY}) - \mathrm{H}(\mathbf{X})$$

Rearranging terms gives the following formula (called the chain rule):

$$\mathrm{H}(\mathbf{XY}) = \mathrm{H}(\mathbf{X}) + \mathrm{H}(\mathbf{Y}|\mathbf{X})$$

The last equation can be though of as first generating $\mathbf{X}$, then generating the remainder of $\mathbf{Y}$. The chain rule suggests the following equality, which we expect to hold (it does):

$$\mathrm{H}(\mathbf{Y}|\mathbf{X}) = \mathop{\mathbf{E}}_{x\sim\mathbf{X}}\Big[\mathrm{H}((\mathbf{Y}|\mathbf{X}=x))\Big]$$

It can be proved by just writing down the equations and playing with the terms. There is another definition that will be of use to us later:

Definition 5.3. The conditional mutual information of $\mathbf{X}, \mathbf{Y}$ given $\mathbb{Z}$ is denoted as:

$$\mathrm{I}(\mathbf{X}, \mathbf{Y}|\mathbf{Z}) = \mathrm{H}(\mathbf{X}|\mathbf{Z}) + \mathrm{H}(\mathbf{Y}|\mathbf{Z}) - \mathrm{H}(\mathbf{XY}|\mathbf{Z})$$

This leads to another formula for mutual information:

$$\begin{aligned}\mathrm{I}(\mathbf{X};\mathbf{Y}) &= \mathrm{H}(\mathbf{Y}) - \mathrm{H}(\mathbf{Y}|\mathbf{X}) \\ &= \mathrm{H}(\mathbf{X}) - \mathrm{H}(\mathbf{X}|\mathbf{Y}) && \text{by symmetry}\end{aligned}$$

The right hand side of the first equation can be thought of the amount of information you know about $\mathbf{X}$ as a result of knowing $\mathbf{Y}$, or the reduction in uncertainty. In communication complexity, if Bob already knows his own $\mathbf{Y}$, then this is the amount he knows about $\mathbf{X}$ *before* Alice sends any information (clearly if they are independent he knows nothing about $\mathbf{X}$).

6 Kullback–Leibler Divergence

We give one last definition here, that of the Kullback–Leibler Divergence. Intuitively, the KL divergence measures the difference between $\mathbf{X}$ and $\mathbf{Y}$. We first motivate it with an example.

Consider the random variable $\mathbf{E}$, generated by choosing a random letter out of an English novel. A first guess at the entropy of $\mathbf{E}$ might be $\log_2(26)$ given that there are 26 letters, but it turns out that some letters are more common than others. The entropy of $\mathbf{E}$ is actually closer to 4 (note $2^4 = 16$), which means many letters are "redundant."

We then play a guessing game where a player can ask yes/no questions in an attempt to determine the letter (as a side note, an optimal first guess is asking if $e \in \{C, H, U, M, P, G, I, F, T, S\}$). Therefore by the entropy of $\mathbf{E}$, a player must guess at least 4 questions on average.

We now consider a French player Pierre, who is an expert in the French version of the game. In other words, for a random variable $\mathbf{F}$ drawn from a French novel, he has an optimal strategy for determining the letter (side note, the optimal first question is asking if $f \in \{P, E, U, M, O, L\}$).

We are now interested in known the average number of questions Pierre will ask if he plays his strategy against $\mathbf{E}$. Clearly, he will ask on average more than $\mathrm{H}(\mathbf{E})$ questions (unless English and French have the same frequency of characters, which they don't). The Kullback–Leibler divergence is a measure of the penalty Pierre suffers as a result of playing his strategy for $\mathbf{F}$ against $\mathbf{E}$.

Definition 6.1. The Kullback–Leibler divergence is the average number of extra bits needed to encode $\mathbf{X}$ using the optimal scheme for $\mathbf{Y}$. It is denoted $\mathrm{D}(\mathbf{X}||\mathbf{Y})$, and the value is:

$$\begin{aligned}\mathrm{D}(\mathbf{X}||\mathbf{Y}) &= \mathop{\mathbf{E}}_{x\sim\mathbf{X}}\left[\frac{1}{\log_2 \mathrm{p}_{\mathbf{Y}}(x)} - \frac{1}{\log_2 \mathrm{p}_{\mathbf{X}}(x)}\right] \\ &= \sum_i \log_2\left(\frac{\mathrm{p}_{\mathbf{X}}(i)}{\mathrm{p}_{\mathbf{Y}}(i)}\right)\cdot \mathrm{p}_{\mathbf{X}}(i)\end{aligned}$$

One additional note - we should be really careful with the base of the logarithm in the KL divergence. In general, when used in statistics, it uses the natural log, whereas in information theory, we use $\log_2$ (which yields an interpretation in bits).

If we consider compressing $\mathbf{XY}$ using a scheme where we compress $\mathbf{X}$ and $\mathbf{Y}$ by their own optimal encodings, we might expect the penalty to be related to the mutual information. In fact, it is an equality, which yields:

$$\mathrm{D}(\mathbf{XY}||\mathbf{X}\times\mathbf{Y}) = \mathrm{I}(\mathbf{X};\mathbf{Y})$$

Another note is that the KL divergence is *not* symmetric, and therefore *not* a distance. For example, in general,

$$\mathrm{D}(\mathbf{X}||\mathbf{Y}) \neq \mathrm{D}(\mathbf{Y}||\mathbf{X})$$

This doesn't really bother us too much, since the χ^2-divergence is not symmetric either.

Example 6.1. Consider the language Hawai'ian which only has 12 letters. If we let $\mathbf{H}$ denote a random letter chosen from a Haiwai'ian novel, we have the following:

$$\begin{aligned}\mathrm{D}(\mathbf{H}||\mathbf{E}) &\approx 2 \\ \mathrm{D}(\mathbf{E}||\mathbf{H}) &= \infty\end{aligned}$$

The last equality comes from the fact that you can't encode English with Haiwai'ian. Therefore, $\mathrm{D}(\mathbf{E}||\mathbf{F})$ is finite if $\mathrm{Support}(\mathbf{E}) \subseteq \mathrm{Support}(\mathbf{F})$ (recall a support of a distribution is the set of elements with non-zero probability).

In Lecture 7, we introduced the total variation distance and the χ^2-divergence. From before,

$$\begin{aligned}d_{TV}(\mathbf{X},\mathbf{Y}) &= \frac{1}{2}\sum_i \left|\,\mathrm{p}_{\mathbf{X}}(i) - \mathrm{p}_{\mathbf{Y}}(i)\right| \\ d_{\chi^2}(\mathbf{X},\mathbf{Y}) &= \mathop{\mathbf{Var}}_{\mathbf{Y}}[\mathbf{X}]\end{aligned}$$

We can relate the total variation distance to the KL divergence by the following inequality:

$$d_{TV}(\mathbf{X}, \mathbf{Y}) \lesssim \sqrt{d_{KL}(\mathbf{X}, \mathbf{Y})}$$

with a factor of $\approx \sqrt{\frac{1}{2}}$ on the right hand side.

Note this can be sharp. For example, consider the following:

$$\mathrm{D}\left(\mathrm{Bern}\left(\frac{1}{2}+\epsilon\right) || \mathrm{Bern}\left(\frac{1}{2}\right)\right) = \Theta(\epsilon^2)$$
$$d_{TV}\left(\mathrm{Bern}\left(\frac{1}{2}+\epsilon\right) || \mathrm{Bern}\left(\frac{1}{2}\right)\right) \approx \epsilon$$

The KL divergence has some relation to Chernoff bounds, in that we can measure with Chernoff bounds the probability that a variable varies to resemble another random variables distribution.

Finally, we give additional relations between the different distance and divergence measures:

$$d_{TV}(\mathbf{X}, \mathbf{Y}) \leq \frac{1}{2}\sqrt{d_{\chi^2}(\mathbf{X}, \mathbf{Y})}$$
$$d_{KL}(\mathbf{X}, \mathbf{Y}) \leq \log(1 + d_{\chi^2}(\mathbf{X}, \mathbf{Y}))$$
$$d_{H}(\mathbf{X}, \mathbf{Y}) \leq \sqrt{d_{KL}(\mathbf{X}, \mathbf{Y})}$$

Where d_H is the Hellinger distance. A full diagram and list can be found in [GS02].

7 Application in communication complexity

Information theory is a new tool being used in communication complexity, particularly in the last 3 years. Prior to this, combinatorics used to be a big tool in proving bounds. We can show bounds by providing bounds on the amount of information that Alice and Bob need to exchange to compute $f(x, y)$. For example, for the equality function, Alice would need to transmit at least $\mathrm{H}(\mathbf{X})$ bits for Bob to know if his number was equal (assuming $\mathbf{X}$ and $\mathbf{Y}$ are identical and independent).

We can think about distributional communication complexity as computing $f(x, y)$ over the distributions $\mathbf{X}, \mathbf{Y}$. We can think of this as the information complexity in computing $f(x, y)$ with high probability. This leads to the following definition, which originally appeared in a paper by Chakrabarti, Shi, Wirth, and Yao.

Definition 7.1. Let π be a deterministic protocol. Then the information cost is denoted as:

$$\mathrm{IC}_{\mu_{\mathbf{XY}}}(\pi) := \mathrm{I}(\pi; \mathbf{Y}|\mathbf{X}) + \mathrm{I}(\pi; \mathbf{X}|\mathbf{Y})$$

where $\mu_{\mathbf{XY}}$ is the joint distribution of $\mathbf{XY}$. [CSWY01]

Imagine Alice drawing a value from $\mathbf{X}$, and Bob drawing a value from $\mathbf{Y}$. An intuition is the following: given the transcript of the protocol, we want to know what Alice learns about Bob's value (from conversation π). She (or he) then uses this information to compute $f(x, y)$ with

high probability. The information cost of π under $\mu_{\mathbf{XY}}$ is then just the amount of information exchanged.

Therefore, if we wish to bound the error, we can write it as follows:

$$\mathrm{IC}^{\epsilon}_{\mu_{\mathbf{XY}}}(f) = \inf\left\{ \mathrm{IC}_{\mu_{\mathbf{XY}}}(\pi) : \underset{(x,y)\sim\mu_{\mathbf{XY}}}{\mathbf{Pr}}\left[\pi(x,y) \neq f(x,y)\right] \leq \epsilon \right\}$$

Note here the infimum is actually necessary, since it need not be a finite protocol in the set (we might have a decreasing information cost that converges). We easily have the following fact by definition:

$$\mathrm{D}^{\epsilon}_{\mu}(f) \geq \mathrm{IC}^{\epsilon}_{\mu}(f)$$

A key question is is there a converse? Recall that $\mathrm{DISJ}_n = \bigwedge_{i=1}^{n} \mathrm{NAND}(x_i, y_i)$. A recent development is the following theorem by Bar-Yossef, Jayram, Kumar, and Sivakumar:

Theorem 7.2. *There exists a μ, which is a distribution on two bits such that:*

$$\mathrm{IC}^{\epsilon}_{\mu^{\otimes n}}(\mathrm{DISJ}_n) \geq n \cdot \mathrm{IC}^{\epsilon}_{\mu}(\mathrm{NAND})$$

where $\mu^{\otimes n}$ denotes the distribution on drawing two n-bit strings using μ. [BYJKS02]

There is still a lot of work needed to understand if we can compress long conversations (protocols) into brief conversations that contain all of the relevant information. In fact, by generalizing the proof on disjointness, we have:

$$\mathrm{IC}^{\epsilon}_{\mu^{\otimes n}}(f^n) = n \cdot \mathrm{IC}^{\epsilon}_{\mu}(f)$$

If the points are drawn from the distribution μ, then the left term is the best information cost of any deterministic protocol which computes each of $f(x_1, y_1), f(x_2, y_2), \ldots, f(x_n, y_n)$ with error at most ϵ. The proof is not too difficult. From this, Braverman and Rao showed in 2010 the following [BR11]:

$$\lim_{n\to\infty} \frac{\mathrm{D}^{\epsilon}_{\mu^{\otimes n}}(f^n)}{n} = \mathrm{IC}^{\epsilon}_{\mu}(f)$$

8 Recommended reading

For further reading, the following book is recommended: [CT12].

References

[BR11] Mark Braverman and Anup Rao. Information equals amortized communication. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 748–757. IEEE, 2011.

[BYJKS02] Ziv Bar-Yossef, TS Jayram, Ravi Kumar, and D Sivakumar. An information statistics approach to data stream and communication complexity. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 209–218. IEEE, 2002.

[CSWY01] Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 270–278. IEEE, 2001.

[CT12] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley, 2012.

[GS02] Alison L Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *International statistical review*, 70(3):419–435, 2002.