AICC II
Prof. Bixio Rimoldi

Benjamin Bovey

Semester of Spring 2019

19th February 2019

As opposed to the first AICC course, where we were mostly presented with tools, we will now see more applications of these tools for communication and computation. Mainly, we will see 3 applications in the first part of the semester:

- **Source coding** (compressing information)
- **Cryptography** (authentication / privacy / integrity of information)
- **Channel Coding** (dealing with noise and loss of information / protecting the information from natural damages)

What these three have in common is the idea of storing and communicating information. The notion of entropy, which will come up quite often, will also be important.

Basic probability review

Special case first: finite sample space Ω and uniform distribution. $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$. Events: $E \subseteq \Omega$. Then:

$$P(E) = \frac{|E|}{|\Omega|} \qquad \text{(uniform distribution)} \tag{1}$$

Definition 1 (conditional probability)**.** Let E, F be two events. Then, the probability that event E occurs knowing that F has occured:

$$P(E|F) = \frac{|E \cap F|}{|F|} \tag{2}$$

Intuitively, you restrict the sample space to F only, because you *know* that F has happened: this translates to the division by the cardinality of F instead of the cardinality of Ω. The intersection of E and F follows from the fact that the sample space is restricted to F: if there exists elements that are in E but not in F, they are outside of the new sample space F, which means that these elements CANNOT occur in conjunction with F. Therefore, we take the intersection of E and F to assure that these elements are not taken into account in the computation.

Theorem 1 (Law of total probability)**.** *Let E and F be two events in Ω, and let F^C denote the complement of F. Then:*

$$P(E) = P(E|F)P(F) + P(E|F^C)P(F^C) \tag{3}$$

This follows quite directly from the fact that $E = (E \cap F) \cup (E \cap F^C)$, so $P(E) = P(E \cap F) + P(E \cap F^C)$ ...

Remark (divide and conquer)*.* You can sometimes create a partition of your sample space, in a way that allows you to better apply the numbers you are given (p.27-28). This method is called *divide and conquer.*

Theorem 2 (Bayes)**.** *Bayes' theorem allows you to compute $p(F|E)$, given that you know $p(E|F), p(E)$ and $p(F)$:*

$$p(F|E) = \frac{p(E|F)p(F)}{p(E)} \tag{4}$$

Remark (application). This is useful, in real scenarios, when either one of $p(E|F)$ and $p(F|E)$ is easily observable, but the other isn't. For example, policemen may observe how many people are driving drunk knowing that they have had an accident (they just test the driver after the accident), but they cannot observe how many people are having an accident knowing that they are driving drunk (they can not really test drivers, and then let them drive drunk just to check if they have an accident or not).

Definition 2. A **random variable** X is a function $X : \Omega \to \mathbb{R}$. It is attached a *probability distribution function* $p_X(x)$, which represents the probability that X will take on the value x, that is, that the following event E occurs:

$$E = \{\omega \in \Omega : X(\omega) = x\} \tag{5}$$

Hence,

$$p_X(x) = p(E) = \sum_{\omega \in E} p(\omega) \tag{6}$$

The set of all possible values of X is sometimes called the *alphabet* of X, written with more curly letters like $\mathcal{A}$.

Theorem 3 (two random variables)**.** *Let $X : \Omega \to \mathbb{R}$ and $Y : \Omega \to \mathbb{R}$ be two random variables. The probability of the event $E = \{X = x\} \cap \{Y = y\}(= \{X = x \wedge Y = y\})$ is*

$$p_{X,Y}(x, y) = \sum_{\omega \in E} p(\omega) \tag{7}$$

We can compute p_X (or p_Y, similarly) from $p_{X,Y}$:

$$p_X(x) = \sum_{y} p_{X,Y}(x, y) \tag{8}$$

In one sense, we "fix in place" the value of x and "scroll through" all possible values of y, and add their probabilities up. Here, p_X is called the **marginal distribution** of $p_{X,Y}(x, y)$ with respect to x.

21st February 2019

Definition 3 (expected value)**.** The **expected value**, or **mean** of a random variable $X : \Omega \to \mathbb{R}$, can be computed as

$$E[X] = \sum_{x \in \mathcal{A}(X)} x p_X(x) \quad \text{(requires } p_X\text{)}, \tag{9}$$

or as

$$E[X] = \sum_{\omega \in \Omega} X(\Omega) p(\omega). \tag{10}$$

One could say that the first way is "calculating over the codomain", and the second way is "calculating over the domain" (of X).

Remark. The expected value is a linear operation. Let $X_1, X_2, \ldots, X_n$ be random variables from Ω to $\mathbb{R}$, and let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be real numbers. Then

$$E \underbrace{\left[\sum_{i=1}^{n} X_i \lambda_i \right]}_{\text{random variable}} = \sum_{i=1}^{n} \lambda_i E[X_i] \tag{11}$$

Extending notions from events to random variables

The notion of independent events extends to random variables. Recall that two events E and F are independent iff $p(E|F) = p(E)$, which is equivalent to saying that $p(E \cap F) = p(E)p(F)$.

Similarly, two random variables may be independent if the value taken by one does not influence the value taken by the other.

Definition 4 (independence of random variables)**.** We say that two random variables $X, Y : \Omega \to \mathbb{R}$ are **independent** iff

$$p_{X,Y}(x, y) = p_X(x) p_Y(y) \tag{12}$$

From there, we can also extend the notion of conditional probability to random variables.

Definition 5. We define the **conditional probability of two random variables** $X, Y : \Omega \to \mathbb{R}$ as

$$p(X = x|Y = y) = \frac{p(X = x \wedge Y = y)}{p(Y = y)}. \tag{13}$$

Remark. The following statements are equivalent:

$$p_{X,Y}(x, y) = p_X(x) \tag{14}$$
$$p_{X|Y}(x|y) = p_X(x) \tag{15}$$
$$p_{Y|X}(y|x) = p_Y(y) \tag{16}$$

Remark (useful trick). If you are asked to check the independence of X and Y, you don't have to check the equality of (15) or (16). You just have to find the expression for the left-hand side function, and see if it depends on the other variable ($\implies$ they are NOT independent), or if it is just a function of one variable ($\implies$ they are independent).

Theorem 4 (consequence of the independence of random variables)**.** *In all cases,* $E[X + Y] \neq E[X] + E[Y]$*. However, if X and Y are* independent*, we also have that*

$$E[XY] = E[X]E[Y] \tag{17}$$

Source & Entropy

The main object that will interest us when studying source coding is the source itself. The question of the definition of a source took mathematicians and computer scientists a while to answer. We can loosely model a source as a black box that outputs *information*: we are not really interested in its inner mechanisms, but rather in what comes out of it, that is, *information*. This information could take many forms, for example sequences of symbols: since we are considering sources from a computer science point of view, we will be interested in sources that shite out sequences of numbers.
The notion of *entropy* comes into the frame when we realize that a symbol that can be *predicted* before it comes out of the source provides no new information. For example, if the sequence of numbers coming out of the source is `1, 1, 5, 5, 3, 3, 19, 19, 1, ...`, we quickly realize that we do not need to store the second number of each pair, as it brings no new information to the table.
An important observation that we can make at this point (it was initially made by Hartley in 1929) is that this link between information and entropy can be modeled very elegantly by random variables! If we think about it, the core idea of a random variable is that it gives you a number that you cannot certainly predict (hence, in fact, the name of *random* variable). If we choose to use this model, a source can be viewed as outputting a sequence of random variables.
Let us now consider a source outputting a sequence of random variables, call them $S_1, S_2, S_3, \ldots, S_n$. Another fundamental question we may ask ourselves is: <u>how much information is actually conveyed by each individual symbol?</u>
A partial answer was given by Hartley, that is, that this must depend on the alphabet of the random variable:

Definition 6. The **alphabet** of a random variable is the codomain of the random variable, that is, the set of all values that the random variable may take.

Indeed, the bigger the alphabet, the more information it can carry, as there are more possibilities for each symbol, and therefore less predictability. With basic combinatorics, we can see that there are $|\mathcal{A}|^n$ possible length-n sequences $(s_1, s_2, \ldots, s_n)$. Therefore, the amount of information carried by S_i is $\log |\mathcal{A}|$.

Example 1. Imagine this is the sequence of good days (1) and bad days(0) in London during a year:

$$(s_1, s_2, \ldots, s_{365}) = (0, 1, 1, 0, 1, \ldots, 0, 1, 1).$$

There is a lot of entropy, it is unpredictable. It would therefore be hard to optimize the storage of this information better. San Diego:

$$(\overbrace{0, 0, 0, \ldots, 0}^{\text{24 zeros}}, 1, \overbrace{0, 0, 0, \ldots, 0, 0}^{\text{340 zeros}}).$$

This is a very predictable sequence, which would allow us to just store $(24, 1, 340)$ rather than storing all 365 bits.

Entropy redefined by Shannon

Shannon, in 1948, gave a new formula for the amount of information carried by the random variable $S \in \mathcal{A}$. He said that this amount *is* in fact the entropy itself, and gave this formula for the entropy $H(S)$:

$$H(S) = -\sum_{s \in \operatorname{supp}(P_S)} p_S(s) \cdot \log_b\big(p_S(s)\big) \tag{18}$$

COMMENTS

- $s \in \operatorname{supp}(S)$ is needed because $\log(0)$ is not defined
- if we are using the convention $0 \cdot \log(0) = 0$, then we can simplify the notation like this: $H(S) = -\sum_{s\in\mathcal{A}} p_S(s) \log_b\big(p_S(s)\big)$
- when $b = 2$ (default) then the unit is the bit. $H(S) = H_2(S)$
- if we rewrite the formula as $H(S) = \sum\limits_{s\in\mathcal{A}} p_S(s) \underbrace{\Big(-\log_b\big(p_S(s)\big)\Big)}_{\text{rand. var. } X} = E[X]$, because this is the expression of the expected value of a random variable X.

EXAMPLE Let the random variable $S \in \mathcal{A}$ be uniformly distributed. Then

$$\begin{aligned} H(S) &= -\sum_{s\in\operatorname{supp}(P_S)} \underbrace{p_S(s)}_{\frac{1}{|\mathcal{A}|}} \log_b\big(\underbrace{p_S(s)}_{\frac{1}{|\mathcal{A}|}}\big) \\ &= \end{aligned}$$

So, Hartley and Shannon agree when the random variable has a uniform distribution.

Entropy extends to any number of random variables.

26th February 2019

We saw last time that a source can mathematically be modeled as one or more random variables, each being described by its probability mass function. We saw that the entropy is a number which represents the ultimate amount of bits (not necessarily, but generally, binary) needed to represent a random variable (and therefore a source).

Example 2. $S \in \{0,1\} = \mathcal{A}, p_S(0) = P$

$$\begin{aligned} H(S) &= -\sum_{s\in\mathcal{A}} p_S(0) \log_b\big(p_S(0)\big) \\ &= \underbrace{-p\log p - (1-p)\log(1-p)}_{h(p) \text{ function of } p} \end{aligned}$$

IT inequality

Many results in information theory are the consequence of the following inequality.

Theorem 5 (IT inequality)**.** *Let $r > 0$. Then*

$$\log_b(r) \le (r-1)\log_b(e) \tag{19}$$

with the equality iff $r = 1$.

Theorem 6 (Entropy bounds)**.** *Let $s \in \mathcal{A}$. Then*

$$0 \le H(s) \le \log|\mathcal{A}| \tag{20}$$

with the first inequality holding iff $S = const.$, and the second inequality holding iff $p_S(s) = \frac{1}{|\mathcal{A}|}$.

Example 3. Let S be your 4-digit lock number. Then $S = \{0, 1, \ldots, 9999\}$. Let's say you choose your lock number at random: then $H(s) = \log 10^4$. However, if your grandma always chooses 0000, then s is a constant, and $H(S) = 0$. A random (least previsible) choice has the most entropy, and a constant (most previsible) choice has the least entropy (zero). This makes sense, and it also means that the random lock number carries the most information, and the constant one carries the least information.

Let's apply this to sources.

Example 4. Let $S_1, S_2, \ldots, S_n$ be a sequence of coin flips. Then $S_i \in \{0, 1\}$, and $P_S(s) = \frac{1}{2}$.
Intuitively, we are flipping n coins, and it should take n bits to describe the result (a length-n bitstring).

$$\underbrace{P_{S_1,S_2,\ldots,S_n}(s_1, s_2, \ldots, s_n)}_{\text{abbreviate as } P(s_1,\ldots,s_n)} = \prod_{i=1}^{n} P(s_i) = \left(\frac{1}{2}\right)^n$$
$$H(S_1, S_2, \ldots, S_n) = \log(|\mathcal{A}|^n) = n \log 2 = n$$

PERSONAL NOTE: recall that when we write $P(s_1, s_2, \ldots, s_n)$, we mean the probability of getting the sequence $(s_1, s_2, \ldots, s_n) \in \mathcal{A}^n$. Then the cardinality of the alphabet is $|\mathcal{A}^n| = |\mathcal{A}|^n$.

Source coding

The setup we have is the following: let's say we have a source emitting $S_i \in \mathcal{A}$ towards an encoder with an *encoding map* (a function $\mathcal{A} \to \mathcal{C}$) called Γ. We're going to map each element of the alphabet into a codeword, for example, each letter of the word `dinner`. For example: d$\to$ 000, i$\to$ 010, ...
The encoder is specified by

- an input alphabet $\mathcal{A}$, which is the alphabet of the source
- an output alphabet $\mathcal{C}$
- the encoding map $\Gamma : \mathcal{A} \to \mathcal{C}$

The code is a set of codewords, which are the output of the Γ map. The Γ map is always one-to-one and onto (bijective), but this doesn't mean that we can necessarily go back from the code words to the words, since we usually concatenate the output.

Example 5. Let $\mathcal{A} = \{\texttt{H}, \texttt{E}, \texttt{L}, \texttt{O}\}$ and $\mathcal{C} = \{01, 10, 0, 11\}$ (Γ maps them in the written order). Then $\Gamma : \mathcal{A} \to \mathcal{C}$ encodes the word `HELLO` to the bitstring `01100011`. The conversion is easy in this direction, but when trying to decode the message, we run into a difficulty: the bitstring could either be interpreted as `01,10,0,0,11`, which would give back the correct message `HELLO`, or as `0,11,0,0,0,11`, which would give the incorrect message `LOLLLO`.

Definition 7. We say that a code is **uniquely decodable** if each concatenation of codewords has a unique parsing into codewords, that is, if we can be sure of getting the correct message when decoding a sequence of codewords. The kinds of encodings that give uniquely decodable codes are the ones that are most interesting in information encoding.

Example 6. Let's have a look at a few different Γ mappings, and check whether they are uniquely decodable or not:

$\mathcal{A}$	Γ_O	Γ_A	Γ_B	Γ_C
a	00	0	0	0
b	01	01	10	01
c	10	10	110	011
d	11	11	1110	0111

FINISH THIS EXAMPLE

Definition 8. A code is said to be **prefix-free** if no codeword is the prefix of a longer codeword. Prefix-free codes are preferred in encoding information. They are also called **instantaneous codes**, since they allow instantaneous decoding.

Remark. Prefixes seem like a good idea, but the problem with codes that are not prefix-free is that when you concatenate words, you need to have all of them to be able to decode it correctly, as shown in the next example.

Example 7. Here's an example of why prefix codes are not the best in terms of decoding. The following code is uniquely decodable, but it uses a prefix:

$\mathcal{A}$	Γ
`a`	0
`b`	00001

If the decoder receives the sequence `00`, it cannot instantaneously determine wether this is the start of a `b` or two concatenated `a`, and so it has to wait until it has the full string of bits to be able to decode it correctly.

Theorem 7 (Kraft-McMillan 1). *If a D-ary code is uniquely decodable, then its codeword lengths $l_1, l_2, \dots, l_M$ satisfy*

$$D^{-l_1} + \cdots + D^{-l_M} \leq 1 \quad \textit{(Kraft's inequality)} \tag{21}$$

Remark. Kraft's sum is only about lengths!

Example 8. Let's look at the following code:

$\mathcal{A}$	$\mathcal{C}$
`a`	`01`
`b`	`0101`

This is a 2-ary (binary) code with lengths 2 and 4, so Kraft's sum gives

$$2^{-2} + 2^{-4} = \frac{1}{4} + \frac{1}{16} = \frac{5}{16} \leq 1,$$

and Kraft's inequality is satisfied, however the code is clearly <u>not</u> uniquely decodable.

Summary.
- $\mathcal{C}$ is uniquely decodable $\implies$ it satisfies the kraft inequality
  contrapositive: $\mathcal{C}$ does not satisfy the Kraft inequality $\implies$ $\mathcal{C}$ is NOT uniquely decodable.
- the converse is NOT true.

28th February 2019

Desired properties of an encoding map $\Gamma : \mathcal{A} \to \mathcal{C}$:

- $\mathcal{C}$ be uniquely decodable
- $\mathcal{C}$ be prefix-free

Theorem 8 (Kraft-McMillan 2). *If $l_1, \dots, l_m$ satisfy Kraft's inequality for some positive integer D, then there exists a D-ary prefix-free code that has codeword lengths $l_1, \dots, l_m$.*

Definition 9. We define the expected length $L(S, \Gamma)$ as

$$L(S, \Gamma) = \sum_{S \in \mathcal{A}} p_S(s) \underbrace{l\big(\Gamma(s)\big)}_{\text{shorthand } l(s)}.$$

Sometimes we write

$$L(S, \Gamma) = \sum_i p_i l_i.$$

This is a rather intuitive concept, as the expected length should indeed depend on the length of each codeword, and on its probability of appearing in the code.

Theorem 9. *Let Γ be the encoding map of a D-ary code for the source S. If the code is uniquely decodable, then*

$$H_D(S) \leq L(S, \Gamma). \tag{22}$$

The entropy is a lower bound to the average length.

Definition 10. We say that a probability distribution is **diadic** iff

$$p_i = D^{-l_i}$$

Theorem 10. *It is possible to make the codewords lengths l_i equal the entropy iff the code is diadic.*

CLEAR UP THE THING ABOUT THE -log(pi) THAT I DONT UNDERSTAND

Remark. Most probability distributions are not diadic. Then, $-\log(p_i)$ is not an integer, and we can't make the length equal that number.

Definition 11 (Shannon-Fano code)**.** A code $\mathcal{C}$ for which $l_i = \lceil -\log p_i \rceil$ is called a **Shannon-Fano code**.

Remark. The Shannon-Fano code is not optimal (WHY? COMPLETE THIS).

Definition 12 (Huffman code)**.** The Huffman code is the optimal code (there doesn't exist a better one). The Huffman code on $\mathcal{A} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$ is:

$\mathcal{A}$	Γ
a	000
b	001
c	01
d	1

We can compute the expected length and the entropy (in bits):

$$\begin{aligned} L(S, \Gamma_H) &= 0.15 + 0.15 + 0.2 + 0.8 = 1.3 \\ H_2(S) &= \ldots = 1.022 \end{aligned}$$