Information Theory: Principles and Applications

Tiago T. V. Vinhoza

April 9, 2010

Asymptotic Equipartition Property: Summary

- Definition of typical set:

$$2^{-n(H(X)+\epsilon)} \leq p_{\mathbf{X}^n}(\mathbf{x}^n) \leq 2^{-n(H(X)-\epsilon)}$$

- Size of typical set:

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

Source coding in the light of the AEP

- A source coder operating on strings of n source symbols need only provide a codeword for each string $\mathbf{x}^n$ in the typical set $A_\epsilon^{(n)}$.
- If a sequence $\mathbf{x}^n$ occurs that is not the typical set $A_\epsilon^{(n)}$, then a source coding failure is declared.
- The probability of failure can be made arbitrarily small by choosing a n large enough.
- Since $|A_\epsilon^{(n)}| \leq 2^{n(H(X)+\epsilon)}$, the number of source codewords that need to be provided is fewer than $2^{n(H(X)+\epsilon)}$.
- So, fixed length codewords of length $\lceil n(H(X)+\epsilon) \rceil$ is enough.

$$\overline{L} \leq H(X) + \epsilon + 1/n$$

Source coding theorem

- For any discrete memoryless source with entropy $H(X)$, any $\epsilon > 0$, any $\delta > 0$, and any sufficiently large n, there is a fixed-to-fixed-length source code with $P(\text{failure}) \leq \delta$ that maps blocks of n source symbols into fixed-length codewords of length $\overline{L} \leq H(X) + \epsilon + 1/n$ bps.
- Compare this result with $\log M$ for fixed-length source codes without failures.

Source coding theorem: converse

- Let $\mathbf{X}^n$ be a string of n discrete random variables X_i, $i = 1, \ldots, n$ each with entropy $H(X)$. For any $\nu > 0$, let $\mathbf{X}^n$ be encoded into fixed-length codewords of length $\lfloor n(H(X) - \nu) \rfloor$ bits. For any $\delta > 0$ and for all sufficiently large n,

$$P(\text{failure}) > 1 - \delta - 2^{-\nu n/2}$$

- Going from a fixed-length code with codeword lengths slightly larger than the entropy to a fixed-length code with codeword lengths slightly smaller than the entropy makes the probability of failure jump from almost 0 to almost 1.

Sources with dependent symbols

- AEP established that $nH(X)$ bits is enough, on average, to describe n independent and identically distributed random variables.
- What happens when the variables are dependent?
- What if the sequence of random variables form a stationary stochastic process?

Stochastic Processes

- A stochastic process is an indexed sequence of random variables.
- Characterized by the joint probability distribution $p_{X_1,\ldots,X_n}(x_1,\ldots,x_n)$. where $(x_1,\ldots,x_n) \in \mathcal{X}^n$

Stochastic Processes

- Stationarity: Joint probability distribution does not change with time-shifts.

$$p_{X_{1+d},\ldots,X_{n+d}}(x_1,\ldots,x_n) = p_{X_1,\ldots,X_n}(x_1,\ldots,x_n)$$

- for every shift d and for all where $x_1,\ldots,x_n \in \mathcal{X}$

Markov Process or Markov Chain

- Each random variable depends on the one preceding it and is conditionally independent of all other preceding random variables.

$$P(X_{n+1} = x_{n+1} | X_n = x_n, \ldots X_1 = x_1) = P(X_{n+1} = x_{n+1} | X_n = x_n)$$

- for all where $x_1, \ldots, x_{n+1} \in \mathcal{X}$
- Joint probability distribution

$$p_{X_1,\ldots,X_n}(x_1, \ldots, x_n) = p_{X_1}(x_1) p_{X_2|X_1=x_1}(x_2) p_{X_3|X_2=x_2}(x_3) \ldots p_{X_n|X_{n-}}$$

Markov Process or Markov Chain

- A Markov chain is irreducible if it is possible to go from any state to any other state in a finite number of steps
- A Markov chain is time invariant if the conditional probability does not depend on the time index n.

  $$P(X_{n+1} = a|X_n = b) = P(X_2 = a|X_1 = b)$$

  for all $a, b \in \mathcal{X}$.
- X_n is the state of the Markov chain in time n.

Markov Process or Markov Chain

- A time invariant Markov chain is characterized by its initial state and a probability transition matrix $\mathbf{P}$, whose element (i, j) is given by

$$P(X_{n+1} = j | X_n = i)$$

- Stationary distributions

Entropy Rate

- Given a sequence of random variables $X_1, X_2, \ldots, X_n$.
- How does the entropy of the sequence grows with n?
- The entropy rate is defined as this rate of growth.

$$H(\mathcal{X}) = \lim_{n \to \infty} \frac{1}{n} H(X_1, X_2, \ldots, X_n)$$

when the limit exists.

Entropy Rate: Examples

- Typewriter with m equally likely output letters. After n keystrokes, we have m^n possible sequences. $H(X_1, \ldots, X_n) = \log m^n$.

$$H(\mathcal{X}) = \lim_{n\to\infty} \frac{1}{n} H(X_1, X_2, \ldots, X_n) = \lim_{n\to\infty} \frac{1}{n} \log m^n = \log m$$

- $X_1, X_2, \ldots$ are indepdendent and identically distributed random variables. $H(X_1, \ldots, X_n) = nH(X_1)$.

$$H(\mathcal{X}) = \lim_{n\to\infty} \frac{1}{n} H(X_1, X_2, \ldots, X_n) = H(X_1)$$

Entropy Rate

- Other definition of entropy rate:

$$H'(\mathcal{X}) = \lim_{n\to\infty} H(X_n|X_{n-1},\ldots,X_1)$$

  when the limit exists.
- For stationary stochastic processes $H(\mathcal{X}) = H'(\mathcal{X})$
- For a stationary Markov chain $H(\mathcal{X}) = H(X_2|X_1)$.

Why entropy rate is important?

- There is a version of the AEP for stationary ergodic sources.

$$-\frac{1}{n} p_{\mathbf{X}^n}(\mathbf{x}^n) \to H(\mathcal{X})$$

- Like the AEP presented last class: $2^{nH(\mathcal{X})}$ typical sequences with probability $2^{-nH(\mathcal{X})}$
- We can represent typical sequences of length n using $nH(\mathcal{X})$ bits.

Other Source Codes

- Shannon-Fano-Elias codes.
- Arithmetic codes.
- Lempel-Ziv codes.

Shannon-Fano-Elias Codes

- Simple encoding procedure based on the cumulative distribution function (CDF) to allot codewords.

$$F_X(x) = \sum_{a \leq x} p_X(a)$$

- Modified CDF

$$\overline{F}_X(x) = \sum_{a < x} p_X(a) + \frac{1}{2}; P(X = x)$$

- $\overline{F}_X(x)$ is known, x is known.

Shannon-Fano-Elias Codes

- From last class: We know that $l(x_i) = -\log p_X(x_i)$ gives good codes.
- Use binary expansion of $\overline{F}_X(x)$ as code for x. Rounding needed. We will round to $\sim -\log p_X(x_i)$.
- Use base 2 fractions.

$$z \in [0,1) \rightarrow z = \sum_{i=1}^{\infty} z_i 2^{-i}$$

- Taking the first k bits $\lfloor z \rfloor_k = z_1 z_2 \ldots z_k$, $z_i \in \{0,1\}$.
- Example: $2/3 = 0.10101010\ldots = 0.\overline{10} \rightarrow \lfloor 2/3 \rfloor_5 = 10101$

Shannon-Fano-Elias Codes

- Coding procedure

$$l(x_i) = \left\lceil \log \frac{1}{p_X(x_i)} \right\rceil + 1$$

$$\mathcal{C}(x_i) = \lfloor \overline{F}_X(x_i) \rfloor_{l(x_i)}$$

Shannon-Fano-Elias Codes

- Example:

	$p_X(x_i)$	$l(x_i)$	$\overline{F}_X(x_i)$	$\mathcal{C}(x_i)$
x_1	$1/3$	3	$1/6$	001
x_2	$1/6$	4	$5/12$	0110
x_3	$1/6$	4	$7/12$	1001
x_4	$1/3$	3	$5/6$	110

Dyadic Intervals

- A binary string can represent a subinterval of $[0,1)$
- From the usual binary representation of a number

$$z_1 z_2 \dots z_n \in \{0,1\}^m \rightarrow z = \sum_{i=1}^{m} z_i 2^{m-i} \in \{0,1,\dots,2^m - 1\}.$$

- We get

$$z_1 z_2 \dots z_n \rightarrow \left[\frac{z}{2^m}, \frac{z+1}{2^m}\right)$$

- Example: $110 \rightarrow [3/4, 7/8)$.
- Codewords of Shannon-Fano-Elias code are disjoint intervals.

Arithmetic Codes

- Arithmetic Codes: invented by Elias, by Rissanen and by Pasco, and made practical by Witten et al in 1987.
- More practical than Huffman coding for large number of source symbols.
- Why? Huffman need to generate and store all codewords.
- Arithmetic Code generate codeword without needing to compute all the others.
- Protected by several US patents: not widely used.
- Original `bzip` used an arithmetic coder, its replacement `bzip2` employed a Huffman coder.
- Based on Shannon-Fano-Elias code.

Arithmetic Codes

- Example: Discrete memoryless source $X \in \{1, 2, 3, 4\}$
- $p_1 = 0.25$, $p_2 = 0.5$, $p_3 = 0.2$ and $p_4 = 0.05$.
- We want the binary codeword for 2313.

Lempel-Ziv Codes

- Do not require knowledge of the source statistics. They adapt so that the average codeword length $\overline{L}$ per source-symbol is minimized in some sense.
- Such algorithms are called *universal*.
- Widely used in practice.

Lempel-Ziv Codes: Algorithms

- LZ77: string-matching on a sliding window.
- Most popular LZ77 based compression method is called DEFLATE; it combines LZ77 with Huffman coding.
- LZ78: adaptive dictionary.
- UNIX `compress` is based on LZ78.
- A lot of variants: LZW, LZWA.

Lempel-Ziv Codes: LZ78 Example

- String: 1011010100010

Lempel-Ziv Codes: LZ78 Example

- String: 1011010100010
- Encoded String: 100011101100001000010

Communications Channel

- Channel: source of randomness (interference, fading, noise, etc.).
- Random nature of the channel is described by a probability distribution over the output of the channel.
- That distribution will often be dependent on the input chosen to be transmitted.
- Discrete case: Both input and output symbols belong to a finite alphabet.

Discrete Channel

- If we apply a sequence $x_1, x_2, \ldots, x_n$ from an alphabet $\mathcal{X}$ at the input of a channel, then at the output we will receive a sequence $y_1, y_2, \ldots, y_n$ belonging to an alphabet $\mathcal{Y}$.
- Usually the probability distribution over the outputs depend on the input and on the state of the channel.
- Some channels have memory. For example, the output symbol y_n might be dependent on previous inputs or outputs.
- Causal behavior: In general $y_1, y_2, \ldots, y_n$ do not need to consider inputs beyiond $x_1, y_2, \ldots, x_n$.

Discrete Channel

- Given an input alphabet $\mathcal{X}$, an output alphabet $\mathcal{Y}$ and a set of states $\mathcal{S}$, a *discrete channel* is defined as a system of conditional probability distributions

$$P(y_1, y_2, \ldots, y_n | x_1, x_2, \ldots, x_n; s)$$

  where $x_1, x_2, \ldots, x_n \in \mathcal{X}$, $y_1, y_2, \ldots, y_n \in \mathcal{Y}$ and $s \in \mathcal{S}$.
- $P(y_1, y_2, \ldots, y_n | x_1, x_2, \ldots, x_n; s)$ can be interpreted as the probability that the sequence $y_1, y_2, \ldots, y_n$ will appear at the output of the channel if the sequence $x_1, x_2, \ldots, x_n$ is applied at the input and the initial state of the channel is s.
- Initial state here is defined as the state before applying x_1 at the input.

Discrete Memoryless Channel

- A discrete channel is *memoryless* if
  - $P(y_1, y_2, \ldots, y_n | x_1, x_2, \ldots, x_n; s)$ does not depend on s so it can be written as $P(y_1, y_2, \ldots, y_n | x_1, x_2, \ldots, x_n)$
  - $P(y_1, y_2, \ldots, y_n | x_1, x_2, \ldots, x_n) = P(y_1|x_1)\ P(y_2|x_2)\ \ldots P(y_n|x_n)$. where $x_1, x_2, \ldots, x_n \in \mathcal{X}$, $y_1, y_2, \ldots, y_n \in \mathcal{Y}$ and $s \in \mathcal{S}$.

Information Processed by a Channel

- Let the input uncertainty be $H(X)$, $H(Y)$ is the output uncertainty and the conditional uncertainties $H(X|Y)$ and $H(Y|X)$. We define the information processed by the channel as

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

- The information processed by a channel depends on the input distribution $p_X(x)$.
- We may vary the input distribution until the information reaches a maximum; the maximum information is called the channel capacity.

$$C = \max_{p_X(x)} I(X;Y).$$

Channel Capacity

- Properties of channel capacity
  - $C \geq 0$, since $I(X;Y) \geq 0$.
  - $C \leq \log |\mathcal{X}|$, since $C = \max I(X;Y) \leq \max H(X) = \log |\mathcal{X}|$
  - $C \leq \log |\mathcal{Y}|$, for the same reason.
  - $I(X;Y)$ is a continuous function on $p_X(x)$.
  - $I(X;Y)$ is a concave function of $p_X(x)$.
- Global maximum.
- Convex optimization techniques.
- Blahut-Arimoto algorithm

Classification of Channels

- A channel is *lossless* if $H(X|Y) = 0$ for all input distributions.
- Input is determined from the output and no transmission errors can occur.
- A channel is *deterministic* if $P(Y = y_i|X = x_j) = 1 \text{ or } 0$ for all i, j. The output is determined by the input, that is, $H(Y|X) = 0$ for all input distributions.
- A channel is *noiseless* is is lossless and deterministic.
- A channel is *useless* (or zero-capacity) if $I(X;Y) = 0$ for all input distributions. Input X and output Y are independent.

Symmetric Channels

- A channel is *symmetric* if the rows of the channel transition matrix are permutations of each other, and the column are permutations of each other

$$P(Y|X) = \begin{bmatrix} 1/3 & 1/3 & 1/6 & 1/6 \\ 1/6 & 1/6 & 1/3 & 1/3 \end{bmatrix}$$

$$P(Y|X) = \begin{bmatrix} 1/2 & 1/3 & 1/6 \\ 1/6 & 1/2 & 1/3 \\ 1/3 & 1/6 & 1/2 \end{bmatrix}$$

- The entry at the i-th row and j-th column denotes the conditional probability $P(Y = y_j|X = x_i)$ that y_j is received given that x_i was sent.

Symmetric Channels

- A channel is *weakly symmetric* if the rows of the channel transition matrix are permutations of each other, and the sums of the columns are equal.

$$P(Y|X) = \begin{bmatrix} 1/3 & 1/6 & 1/2 \\ 1/3 & 1/2 & 1/6 \end{bmatrix}$$

Binary Symmetric Channels

- It is the basic example of a noisy communication system
- Binary input and binary output. The output is equal to the input with probability $1 - p$. With probability p a 0 is received as 1, and vice-versa.

$$P(Y|X) = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}$$

Binary Erasure Channel

- Bits are lost instead of being flipped.
- A fraction α of bits is lost and the receiver knows that a bit was supposed to arrive.
- Packet communications

$$P(Y|X) = \begin{bmatrix} 1-\alpha & \alpha & 0 \\ 0 & \alpha & 1-\alpha \end{bmatrix}$$

Channel Capacity: Toy Examples

- Noiseless Binary Channel
  - One error-free bit can be transmitted per use of the channel.
  - $C = 1$ bit, and is achieved with uniform input distribution.
- Lossless channel
  - Input can be determined from the output. Every transmitted bit can be recovered without error.
  - For our example, $C = 1$ bit, and is achieved with uniform input distribution.
- Noisy Typewriter
  - Channel input is either received unchanged at the output with probability 1/2 or it is transformed to the next letter with probability 1/2. That is, if A is transmitted, we can receive A or B. Each with probability 1/2.
  - Input has 26 symbols. If we use alternate input symbols (A, C, E), we can transmit 13 symbols without error.

$$C = \max H(Y) - H(Y|X) = \max H(Y) - 1 = \log 26 - 1 = \log 13.$$

Channel Capacity for BSC

- Bounding the mutual information for the BSC:

$$
\begin{aligned}
I(X;Y) &= H(Y) - H(Y|X) \\
&= H(Y) - \sum_{x \in \mathcal{X}} H(Y|X = x) p_X(x) \\
&= H(Y) - \sum_{x \in \mathcal{X}} H(p) p_X(x) \\
&= H(Y) - H(p) \\
&\leq 1 - H(p)
\end{aligned}
$$

- Equality is achieved when the input distribution is uniform.

$$C = 1 - H(p)$$

Channel Capacity for BEC

- $C = 1 - \alpha$.
- This result is somewhat intuitive: since a fraction α of the input bits is erased, we can recover (at most) $1 - \alpha$ of the bits.

Why the channel capacity is important?

- Shannon proved that the channel capacity is the maximum number of bits that can be reliably transmitted over the channel.
- Reliably = probability of error can be made arbitrarily small.
- Channel coding theorem.