10-704: Information Processing and Learning **Fall 2016**

Lecture 9: Sept 28

Lecturer: Siheng Chen

Note: *These notes are based on scribed notes from Spring15 offering of this course. LaTeX template courtesy of UC Berkeley EECS dept.*

9.1 Entropy Rate of Stochastic Processes

So far we have mostly talked about entropy of a random variable. We can extend that notion to a **stochastic process**, which is simply an indexed sequence of random variables.

Entropy of random variable X is $H(X)$, the joint entropy of $X_1 \dots X_n$ is then

$$\begin{aligned} H(X_1, \dots, X_n) &= \sum_{i=1}^{n} H(X_i | X_{i-1} \dots X_1) \quad \text{chain rule} \\ &\leq \sum_{i=1}^{n} H(X_i) \quad \text{since conditioning does not increase entropy} \\ &= nH(X) \quad \text{if the variables are identically distributed} \end{aligned}$$

If the random variables are also independent, then the joint entropy of n random variables increases with n. How does the joint entropy of a sequence of n random variables with possibly arbitrary dependencies scale?

To answer this, we consider a stochastic process which is an indexed sequence of random variables with possibly arbitrary dependencies. We define

Entropy rate of a stochastic process $\{X_i\} =: \mathcal{X}$ as

$$H(\mathcal{X}) := \lim_{n \to \infty} \frac{H(X_1, \dots, X_n)}{n}$$

i.e. the limit of the per symbol entropy, if it exists.

Stationary stochastic process: A stochastic process is stationary if the joint distribution of any subset of the sequence of random variables is invariant with respect to shifts:

$$p(X_1, \dots, X_n) = p(X_{1+l}, \dots, X_{n+l}) \qquad \forall l, \ \forall n$$

Theorem 9.1 *For a stationary stochastic process, the following limit always exists*

$$H(\mathcal{X}) := \lim_{n \to \infty} \frac{H(X_1, \dots, X_n)}{n}$$

i.e. limit of per symbol entropy, and and is equal to

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

i.e. the limit of the conditional entropy of last random variable given past.

We also consider **Markov processes** where k^{th} order Markov process satisfies

$$P(X_{n+1}|X_nX_{n-1}\dots X_1) = P(X_{n+1}|X_n\dots X_{n-k+1})$$

i.e. the dependence is only over last k variables in the sequence.

For stationary first order Markov processes:

$$H(\mathcal{X}) = \lim_{n\to\infty} H(X_n|X_{n-1}) = H(X_2|X_1)$$

Theorem 9.2 Burg's Maximum Entropy Theorem
The max entropy rate stochastic process $\{X_i\}$ satisfying the constraints

$$E[X_iX_{i+k}] = \alpha_k \qquad \text{for } k = 0, 1\dots p \quad \forall i \quad (\star)$$

is the Gauss-Markov process of the p^{th} order, having the form:

$$X_i = -\sum_{i=1}^{m} a_k X_{i-k} + Z_i,$$

where $Z_i \overset{iid}{\sim} \mathcal{N}(0,\sigma^2)$, a_k and σ^2 are parameters chosen such that constraints $\star$ are satisfied.

Note: The process $\{X_i\}$ is NOT assumed to be (1) zero-mean, (2) Gaussian or (3) stationary.
Note: The theorem states that $AR(m)$ auto-regressive Gauss-Markov process of order m arise as natural solutions when finding maximum entropy stochastic processes under second-order moment constraints up to lag m.

Proof: Let $\{X_i\}$ be any stochastic process that satisfies constraints $\star$, $\{Z_i\}$ be any Gaussian process that satisfies constraints $\star$, and $\{Z_i'\}$ be a p^{th} order Gauss-Markov process with the same some distribution for all orders up to p. (Existence of such a process will be established after the proof.)

Since the multivariate normal distribution maximizes entropy over all vector-valued random variables under a covariance constraint, we have:

$$\begin{aligned}
H(X_1,\dots,X_n) &\leq H(Z_1,\dots,Z_n) \\
&= H(Z_1,\dots,Z_p) + \sum_{i=p+1}^{n} H(Z_i|Z_{i-1},\dots,Z_1) \quad \text{(chain rule)} \\
&\leq H(Z_1,\dots,Z_p) + \sum_{i=p+1}^{n} H(Z_i|Z_{i-1},\dots,Z_{i-m}) \quad \text{(conditioning does not increase entropy)} \\
&= H(Z_1',\dots,Z_p') + \sum_{i=p+1}^{n} H(Z_i'|Z_{i-1}',\dots,Z_{i-m}') \\
&= H(Z_1',\dots,Z_p')
\end{aligned}$$

$$\Rightarrow \lim_{p\to\infty} \frac{1}{p} H(X_1\dots X_p) \leq \lim_{p\to\infty} \frac{1}{p} H(Z_1'\dots Z_p')$$

Existence: Does a p^{th} order Gaussian Markov process exists s.t. $(a_1 \dots a_p, \sigma^2)$ satisfy $\star$?

$$\begin{aligned} X_i X_{i-l} &= -\sum_{k=1}^{p} a_k X_{i-k} X_{i-l} + Z_i X_{i-l} \\ E[X_i X_{i-l}] &= -\sum_{k=1}^{p} a_k E[X_{i-k} X_{i-l}] + E[Z_i X_{i-l}] \end{aligned}$$

Let $R(l) = E[X_i X_{i-l}] = E[X_{i-l} X_i] = \alpha_l$ be the given $p+1$ constraints. Then we obtain *The Yule-Walker equations - p+1 equations in p+1 variables* $(a_1 \dots a_p, \sigma^2)$*:*

$$\begin{aligned} &\text{for } l = 0 && R(0) = -\sum_{k=1}^{p} a_k R(-k) + \sigma^2 \\ &\text{for } l > 0 && R(l) = -\sum_{k=1}^{p} a_k R(l-k) \quad (\text{since } Z_i \perp X_{i-l} \text{ for } l > 0.) \end{aligned}$$

The solution to the Yule-Walker equations will determine the p^{th} order Gaussian Markov process. ■

9.2 Data Compression / Source coding

Figure 9.1 shows a source coding schema.

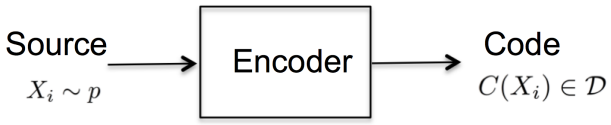

Figure 9.1: Coding schema.

Source code: A source code C is a mapping from the range of a random variable or a set of random variables to finite length strings of symbols from a $\mathcal{D}$-ary alphabet, that is

$$C : \mathcal{X} \to \mathcal{D}^\star,$$

and the code $C(X)$ for a symbol X is an element in $\mathcal{D}^\star$.

Instead of encoding an individual symbol, we can also encode blocks of symbols together. A length n **block code** encodes n length strings of symbols together and is denotes by $C(X_1, \cdots, X_n) =: C(X^n)$. We then define the extension of the code using concatenation as $C(X^n) = C(X_1)...C(X_n)$

Expected length of a source code denoted by $L(C)$ is given as follows:

$$L(C) = \sum_{x \in \mathcal{X}} p(x) l(x)$$

where $l(x)$ is the length of codeword $c(x)$ for a symbol $x \in \mathcal{X}$, and $p(x)$ is the probability of the symbol.

Several classes of symbol codes have appealing properties that are widely used. For example,

- Non-singularity : $\forall X_1 \neq X_2 \Rightarrow C(X_1) \neq C(X_2)$
- Unique-decodability : $\forall X_1^n \neq X_2^m \Rightarrow C(X_1^n) \neq C(X_2^m)$
- Self-punctuating (Prefix) : $\forall X_1 \neq X_2 \Rightarrow C(X_1) \notin \text{Prefix}(C(X_2))$

Note that unique decodability implies non-singularity and self-punctuating implies unique decodability. Self-punctuating codes are also called **instantaneous or prefix codes**. For unique decodability, we may need to see the entire sequence to decode it uniquely, but for instantaneous ones, you can decode a symbol as soon as you've seen its encoding.

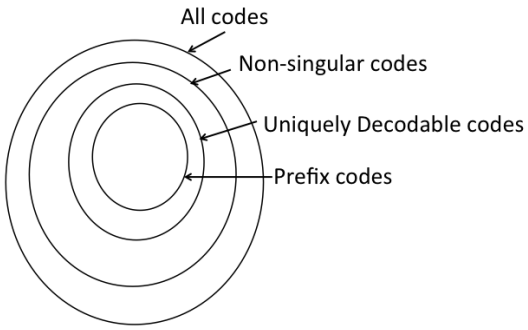

Figure 9.2: Codes.

9.2.1 Examples of Codes

- Encode a uniform distribution on 4 letters: We simply assign 2 bits per symbol. Hence, $\mathbb{E}l(x) = 2 = H(x)$.
- Uniform distribution on 7 letters: We assign 3 bits per letter. Hence $\mathbb{E}l(x) = 3 \leq H(x) + 1$
- $p(a) = \frac{1}{2}, p(b) = \frac{1}{4}, p(c) = \frac{1}{4}$. We code $a \to 0$, $b \to 10$, $c \to 11$. We have $\mathbb{E}l(x) = \frac{3}{2} = H(x)$. We note that is a prefix free code.
- Non-Singular Code: $p(a) = 0.5, p(b) = 0.25, p(c) = p(d) = \frac{1}{8}$, We code $a \to 0$, $b \to 1$, $c \to 00$, $d \to 01$. This is nonsingular but not very useful or practical since it is not uniquely decodable since the string $'01'$ can map to both ab or c . $\mathbb{E}l(x) = 1.25 \leq H(X) = 1.75$
- We code $a \to 10$, $b \to 00$, $c \to 11$, $d \to 110$. This is nonsingular and uniquely decodable but not prefix free since the code for c, $'11'$ is a prefix for the code for d,$'110'$. This would result in decoding "on the fly" impossible.
- We code $a \to 0$, $b \to 10$, $c \to 110$, $d \to 111$. Hence this code is prefix-free, and therefore non-singular and uniquely decodable.

It is clear that the bare minimum requirement when designing a practical code is that it needs to be uniquely decodable.

9.3 Shannon Source Coding Theorem (Achievability)

We now state and prove the Shannon source coding theorem:

Theorem 9.3 *Let x^n denote a sequence of n source symbols drawn iid from $p(X)$. $\exists$ a code C that maps sequences x^n into binary strings such that the mapping is one-to-one and for any $\epsilon \geq 0$, $\exists n_0$ such that $\forall n \geq n_0$ we have :*

$$\mathbb{E}[\frac{l(x^n)}{n}] \leq H(x) + \epsilon$$

Proof: Suppose I could find a set $A_\delta^{(n)} \subseteq \mathcal{X}^{(n)}$ with $|A_\delta^{(n)}| \leq 2^{n(H(x)+\delta)}$ but also with $\mathbb{P}\big[A_\delta^{(n)}\big] \geq 1-\delta$. We shall revisit this construction.

We present a coding scheme: If $x^{(n)} \in A_\delta^{(n)}$, then code by indexing into this set using $1 + \lceil n(H+\delta) \rceil$ bits. If $x^{(n)} \notin A_\delta^{(n)}$, we then code with $1 + \lceil n \log |\mathcal{X}| \rceil$ bits.

We now write out the expectation of the length of the codeword.

$$\mathbb{E}\big[l(x^{(n)}\big] = \sum_{x^{(n)}} p(x^{(n)}) l(x^{(n)} \leq \sum_{x^{(n)} \in A_\delta^{(n)}} p(x^{(n)})(2 + n(H+\delta)) + \sum_{x^{(n)} \notin A_\delta^{(n)}} p(x^{(n)})(2 + n \log |\mathcal{X}|)$$

$$\leq 2 + n(H+\delta) + n \log |\mathcal{X}| (1 - \mathbb{P}(A_\delta^n)$$

$$\leq 2 + n(H+\delta) + n\delta \log |\mathcal{X}| \equiv n(H+\epsilon) \text{ where } \epsilon = \frac{2}{n} + \delta(1 + \log |\mathcal{X}|)$$

We now consider how to build $A_\delta^{(n)}$, often called a typical set. Let:

$$A_\delta^{(n)} = \{(x_1 \cdots x_n) | H(x^{(n)}) - \delta \leq -\frac{1}{n} \log p(x_1 \cdots x_n)\} \leq H(x) + \delta\}.$$

Clearly this is well defined.

Claim 9.4 *If $X_1 \cdots X_n \rightsquigarrow p$, then*

$$-\frac{1}{n} \log p(x_1, c \ldots x_n) \to H(X)$$

as $n \to \infty$

Proof: This essentially holds by the weak law of large numbers.

$$\frac{1}{n} \log p(x_1 \cdots x_n) = -\frac{1}{n} \sum_i \log p(x_i) \to \mathbb{E}_{x \rightsquigarrow p} - \log p(x) = H(x)$$

■

The finite sample version holds by Hoeffding's inequality. If $p(x) \geq p_{min}$, then $0 \leq -\log p(x) \leq \log p_{min} \triangleq \gamma$ and as a result :

$$\mathbb{P}\big[\frac{1}{n}|\sum_i -\log p(x_i) - H(X)| \geq \epsilon\big] \leq 2e^{\frac{-2n\epsilon^2}{\gamma^2}}$$

or with probability $1-\delta$

$$|\frac{1}{n}\sum_i -\log(p(x_i)) - H(x)| \leq \sqrt{\frac{\gamma^2}{2n}\log(\frac{2}{\delta})}$$

Hence we simply set $A_{\epsilon,\delta}^{(n)}$ with $\epsilon = \sqrt{\frac{\gamma^2}{2n}\log(\frac{2}{\delta})}$ and we have $\mathbb{P}\big[A_{\epsilon,\delta}^{(n)}\big] \geq 1-\delta$

Hence, we have proven the achievability part of the Source Coding Theorem. Recapping, we use the Law of large numbers that most of the probability mass concentrates on a few sequences characterized by the entropy. We use the code that assigns short sequences to these sequences in the typical set and long sequences to the rest that are not in the typical set. ■

In the next class, we will talk about Necessity i.e. any uniquely decodable source code cannot compress below entropy.