LECTURE 5

Last time:

- Types of convergence
- Weak Law of Large Numbers
- Strong Law of Large Numbers
- Asymptotic Equipartition Property

Lecture outline

- Continue on AEP
- Codes
- Kraft inequality
- optimal codes.

Reading: Scts. 5.1-5.4.

Continue the Coin Toss Example

- Stirling's Formula:

$$n! \approx n^n e^{-n}\sqrt{2\pi n}$$

- Count the number of possible sequences of length n:

$$\begin{aligned}\binom{n}{nt} &= \frac{n!}{(nt)!(n(1-t))!} \\ &\approx \frac{n^n e^{-n}}{(nt)^{nt}e^{-nt}(n(1-t))^{n(1-t)}e^{-n(1-t)}} \\ &\doteq 2^{nH(t)}\end{aligned}$$

- Key approximation: $\binom{n}{nt} \doteq 2^{nH(t)}$

$$\lim_{n\to\infty} \frac{\log_2 \binom{n}{nt}}{n} = H(t)$$

- To be precise

$$\binom{n}{nt} = 2^{nH(t)+O(\log(n))}$$

Number of Possible Sequences

- Let the number of 0's in a sequence x_1^n be m, define function $T(x_1^n) = \frac{m}{n}$ as the fraction of 0's.
- For a subset $S \subset [0,1]$, define

$$A(S) = \{x_1^n : T(x_1^n) \in S\}.$$

$$|A(S)| = \sum_{t \in S, nt \in \mathcal{Z}} \binom{n}{nt}$$

Claim For any fixed ϵ, let

$$A_\epsilon = A(1/2 - \epsilon, 1/2 + \epsilon),$$

A_ϵ contains nearly all the sequences:

$$\frac{|A_\epsilon|}{2^n} \to 1$$

Proof:

$$\begin{aligned} |A_\epsilon^c| &= \sum_{|t-1/2|>\epsilon, nt \in \mathcal{Z}} \binom{n}{nt} \\ &\leq n 2^{n\overline{H(t)} + O(\log n)} \end{aligned}$$

where $\overline{H(t)} = \max_{|t-1/2|>\epsilon} H(t) \leq H(1/2 - \epsilon)$.

For n large enough, $2^n >> |A_\epsilon^c|$.

True or False?

For large enough n,

- $\binom{n}{n/2} \doteq 2^n$
- $\binom{n}{n/2} >> |A_\epsilon^c|$
- $\dfrac{\binom{n}{n/2}}{2^n} \to 1$

Fair Coin Toss

Let $P(X_i = 0) = p = 1/2$,

- All the sequences have the same probability.
- Since $\frac{|A_\epsilon|}{2^n} \to 1$,

$$P(X_1^n \in A_\epsilon) \to 1$$

- Two different ways to define the typical set.

Coin Toss with Probability p

- For a sequence x_1^n, let $T(x_1^n)$ be the fraction of 0's. $T(X_1^n)$ is a r.v.

- For any t s.t. $nt \in \mathcal{Z}$,

$$\begin{aligned} P(T=t) &= \binom{n}{nt} p^{nt}(1-p)^{n(1-t)} \\ &= 2^{n(-t\log t-(1-t)\log(1-t))+O(\log n)} \\ &\quad 2^{n(t\log p+(1-t)\log(1-p))} \\ &= 2^{-nD(t||p)+O(\log n)} \\ &\doteq 2^{-nD(t||p)} \end{aligned}$$

- $P(|T-p| \le \epsilon) \to 1$. Proof by summing over the probability $P(T=t)$ for all t with $|t-p| > \epsilon$, and show

$$P(|T-p| > \epsilon) << 1$$

  for large enough n.

- Typical set for a distribution $\{p, 1-p\}$ is $A_e^{(n)} = A(p-\epsilon, p+\epsilon)$

Corollary

- For any distribution q, the typical set is defined as $A(q-\epsilon, q+\epsilon)$.

Consider an i.i.d. sequence generalized according to a distribution p. It is typical w.r.t. a second distribution q with probability $2^{-nD(q||p)}$.

- High probability sets and the typical set: consider $p < \frac{1}{2}$, a sequence x_1^n with $T(x_1^n) < p$ has higher probability than any individual sequence in the typical set.

Define

$$\{x_1^n : T(x_1^n) < p + \epsilon\} = A(0, p + \epsilon)$$

as the "high probability set".

$$\begin{aligned} |A(0, p+\epsilon)| &\doteq |A(p-\epsilon, p+\epsilon)| \\ P(A(0, p+\epsilon)) &\doteq P(A(p-\epsilon, p+\epsilon)) \end{aligned}$$

Source Coding and AEP

Definition
A **source code** C of a random variable X is a mapping from $\mathcal{X}$ to $\mathcal{D}^*$, the set of finite length strings of symbols from a D-ary alphabet.

- The same definition applies for sequence of r.v.s, X_1^n.
- x or x_1^n are called *source symbol (string)*, D is the set of *coded symbols*. $C(x)$ is called the *codeword* corresponding to x.
- We allow different codewords to have different length, denote $l(x)$ as the length of $C(x)$.

Definition
The expected length of a code $L(C)$ is given by

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

Goal For a given source, find a code to minimize the expected length (per source symbol).

Data Compression by AEP

- Use $n \log |\mathcal{X}| + 1$ bits to describe (index) any sequence in $\mathcal{X}^n$.
- Since $|A_\epsilon^{(n)}| \leq 2^{n(H+\epsilon)}$, use $n(H+\epsilon)+1$ bits to index all sequences in $A_\epsilon^{(n)}$.
- Use an extra bit to indicate $A_\epsilon^{(n)}$.

$$
\begin{aligned}
E(l(X_1^n)) &= \sum_{x_1^n} P(x_1^n) l(x_1^n) \\
&= P(A_\epsilon^{(n)})[n(H+\epsilon)+2] \\
&\quad + P(A_\epsilon^{(n)c})[n \log |\mathcal{X}| + 2] \\
&\leq n(H+\epsilon) + n\epsilon \log |\mathcal{X}| + 2 \\
&= n(H+\epsilon')
\end{aligned}
$$

As $n \to \infty$, ϵ' can be made arbitrarily small.

Theorem

$$
\frac{1}{n} E\left[l(X_1^n)\right] \leq H(X) + \epsilon
$$

Concatenation

Definition The *extension* of a code C is the a code for finite strings of $\mathcal{X}$ given by the concatenation of the individual codewords

$$C(x_1, x_2, \ldots, x_n) = C(x_1)C(x_2)\ldots C(x_n)$$

- A code is called **non-singular** if

$$x_i \neq x_j \Rightarrow C(x_i) \neq C(x_j)$$

- A code is called **uniquely decodable** if its extension is non-singular

Example:

x	a	b	c	d
$C(x)$	1	11	10	101

Prefix code

Example The following code is uniquely decodable,

x	a	b	c	d
$C(x)$	10	00	11	110

consider a coded string 11000000000000010.

Definition A code is called a *prefix code* or *instantaneous code* is no codeword is a prefix of any other codeword.

- Self-punctuating.
- Can decode without reference of the future.

Kraft's Inequality

Theorem For any prefix code over an alphabet of size D, let the codeword length be $l_1, l_2, \ldots,$, we have

$$\sum_{i=1}^{\infty} D^{-l_i} \leq 1$$

Conversely, for any given set of codeword lengths that satisfy the inequality, we can construct a prefix code with these codeword lengths.

Proof

- Construct a D-ary tree.
- Prefix code means each codeword is a leaf, no codeword can be the descendent of any other codeword.
- Assign weight D^{-l_i} to each codeword.

Consider a codeword $y_1 y_2 \ldots y_{l_i}$, where $y_j \in \{0, \ldots, D-1\}$. Let

$$0.y_1 y_2 \ldots y_{l_i} = \sum_{j=1}^{l_i} y_j D^{-j} \in [0,1]$$

.

This codeword corresponds to an interval

$$\left(0.y_1 y_2 \ldots y_{l_i}, 0.y_1 y_2 \ldots y_{l_i} + \frac{1}{D^{l_i}}\right)$$

Prefix code implies the intervals are disjoint.

Optimal codes

Optimal code is defined as code with smallest possible $L(C)$ with respect to P_X

Optimization:

minimize $\sum_{x\in\mathcal{X}} P_X(x)l(x)$

subject to $\sum_{x\in\mathcal{X}} D^{-l(x)} \leq 1$

and $l(x)$s are integers

Optimal codes

Let us relax the integer constraint and replace the first constraint by equality to obtain a lower bound. Use Lagrange multipliers, define

$J = \sum_{x \in \mathcal{X}} P_X(x) l(x) + \lambda \sum_{x \in \mathcal{X}} D^{-l(x)}$

and set $\frac{\partial J}{\partial l(i)} = 0$

$$P_X(i) - \lambda \log(D) D^{-l(i)} = 0$$

equivalently $D^{-l(i)} = \frac{P_X(i)}{\lambda \log(D)}$

solve for $\lambda = \frac{1}{\log(D)}$, yielding $l(i) = -\log_D(P_X(i))$

The expected codeword length

$$\begin{aligned} L(C) &= E[l(X)] = E[-\log_D P_X(X)] \\ &= H_D(X) \\ &= \frac{H(X)}{\log_2 D} \end{aligned}$$