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 \( n \):
  \[
  \binom{n}{nt} = \frac{n!}{(nt)!(n(1-t))!} \\
  \approx \frac{n^n e^{-n}}{(nt)^n t^n e^{-nt} (n(1-t))^{n(1-t)} e^{-n(1-t)}} \\
  = 2^{nH(t)}
  \]

• Key approximation: \( \binom{n}{nt} \approx 2^{nH(t)} \)

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

• 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

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

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

Claim For any fixed $\epsilon$, let

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

$A_\epsilon$ contains nearly all the sequences:

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

Proof:

$$|A_\epsilon^c| = \sum_{|t-1/2| > \epsilon, nt \in \mathbb{Z}} \binom{n}{nt} \leq n2^{nH(t)+O(\log n)}$$
where \( \overline{H(t)} = \max_{|t-1/2|>\epsilon} H(t) \leq H(1/2 - \epsilon) \).

For \( n \) large enough, \( 2^n >> |A^c_\epsilon| \).

**True or False?**

For large enough \( n \),

- \( \left( \begin{array}{c} n \\ n/2 \end{array} \right) \div 2^n \)
- \( \left( \begin{array}{c} n \\ n/2 \end{array} \right) >> |A^c_\epsilon| \)
- \( \left( \begin{array}{c} n \\ n/2 \end{array} \right) \)
- \( \frac{\left( \begin{array}{c} n \\ n/2 \end{array} \right)}{2^n} \rightarrow 1 \)
Fair Coin Toss

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

- All the sequences have the same probability.
- Since $\frac{|A_\epsilon|}{2^n} \rightarrow 1$,
  
  $$P(X_1^n \in A_\epsilon) \rightarrow 1$$

- 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. $nt \in \mathbb{Z}$,

$$P(T = t) = \binom{n}{nt} p^{nt} (1 - p)^{n(1-t)}$$

$$= 2^n(-t \log t - (1-t) \log(1-t)) + O(\log n)$$

$$= 2^n(t \log p + (1-t) \log(1-p))$$

$$= 2^{-nD(t||p)} + O(\log n)$$

$$\equiv 2^{-nD(t||p)}$$

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

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

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 $x^n_1$ with $T(x^n_1) < p$ has higher probability than any individual sequence in the typical set.

Define

$$\{x^n_1 : T(x^n_1) < p + \epsilon\} = A(0, p + \epsilon)$$

as the "high probability set".

$$|A(0, p + \epsilon)| \triangleq |A(p - \epsilon, p + \epsilon)|$$

$$P(A(0, p + \epsilon)) \triangleq P(A(p - \epsilon, p + \epsilon))$$
Source Coding and AEP

Definition
A **source code** \( C \) of a random variable \( X \) is a mapping from \( X \) to \( D^* \), 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

\[
L(C) = \sum_{x \in X} P_X(x)l(x)
\]

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 $|A^{(n)}_\epsilon| \leq 2^n(H+\epsilon)$, use $n(H+\epsilon) + 1$ bits to index all sequences in $A^{(n)}_\epsilon$.

- Use an extra bit to indicate $A^{(n)}_\epsilon$.

\[
E(l(X^n_1)) = \sum_{x^n_1} P(x^n_1) l(x^n_1)
\]
\[
= P(A^{(n)}_\epsilon)[n(H+\epsilon) + 2] + P(A^{(n)}_\epsilon^c)[n \log |\mathcal{X}| + 2]
\]
\[
\leq n(H+\epsilon) + n\epsilon \log |\mathcal{X}| + 2
\]
\[
= n(H + \epsilon')
\]

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

**Theorem**

\[
\frac{1}{n} E[l(X^n_1)] \leq H(X) + \epsilon
\]
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:

<table>
<thead>
<tr>
<th>( x )</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C(x) )</td>
<td>1</td>
<td>11</td>
<td>10</td>
<td>101</td>
</tr>
</tbody>
</table>

Prefix code

Example The following code is uniquely decodable,

<table>
<thead>
<tr>
<th>$x$</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C(x)$</td>
<td>10</td>
<td>00</td>
<td>11</td>
<td>110</td>
</tr>
</tbody>
</table>

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:

\[
\text{minimize } \sum_{x \in \mathcal{X}} P_X(x) l(x)
\]

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

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