# Class Road Map - IT-I

<table>
<thead>
<tr>
<th>Class</th>
<th>Date</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>9/25</td>
<td>Overview, Communications, Information, Entropy</td>
</tr>
<tr>
<td>L2</td>
<td>9/30</td>
<td>Entropy, Mutual Information, KL-Divergence</td>
</tr>
<tr>
<td>L3</td>
<td>10/2</td>
<td>More KL, Jensen, more Venn, Log Sum, Data Proc. Inequality</td>
</tr>
<tr>
<td>L4</td>
<td>10/7</td>
<td>Data Proc. Ineq., thermodynamics, Stats, Fano</td>
</tr>
<tr>
<td>L5</td>
<td>10/9</td>
<td>M. of Conv, AEP,</td>
</tr>
<tr>
<td>L6</td>
<td>10/14</td>
<td><strong>AEP, Source Coding, Types</strong></td>
</tr>
<tr>
<td>L7</td>
<td>10/16</td>
<td></td>
</tr>
<tr>
<td>L8</td>
<td>10/21</td>
<td></td>
</tr>
<tr>
<td>L9</td>
<td>10/23</td>
<td></td>
</tr>
<tr>
<td>L10</td>
<td>10/28</td>
<td></td>
</tr>
<tr>
<td>L11</td>
<td>10/30</td>
<td></td>
</tr>
<tr>
<td>L12</td>
<td>11/4</td>
<td></td>
</tr>
<tr>
<td>LXX</td>
<td>11/6</td>
<td>In class midterm exam</td>
</tr>
<tr>
<td>LXX</td>
<td>11/11</td>
<td>Veterans Day holiday</td>
</tr>
<tr>
<td>L13</td>
<td>11/13</td>
<td></td>
</tr>
<tr>
<td>L14</td>
<td>11/18</td>
<td></td>
</tr>
<tr>
<td>L15</td>
<td>11/20</td>
<td></td>
</tr>
<tr>
<td>L16</td>
<td>11/25</td>
<td></td>
</tr>
<tr>
<td>L17</td>
<td>11/27</td>
<td></td>
</tr>
<tr>
<td>L18</td>
<td>12/2</td>
<td></td>
</tr>
<tr>
<td>L19</td>
<td>12/4</td>
<td></td>
</tr>
<tr>
<td>LXX</td>
<td>12/10</td>
<td>Final exam</td>
</tr>
</tbody>
</table>

**Finals Week:** December 9th–13th.
Read chapters 1 and 2 in our book (Cover & Thomas, “Information Theory”) (including Fano’s inequality).

Read chapters 3 and 4 in our book (Cover & Thomas, “Information Theory”).
Homework

- Homework 1 on our assignment dropbox (https://canvas.uw.edu/courses/1319497/assignments), was due Tuesday, Oct 8th, 11:55pm.

- Homework 2 on our assignment dropbox (https://canvas.uw.edu/courses/1319497/assignments), due Friday 10/16/2019, 11:45pm.

- Office hour Thurs evening.
I need to be at CMU (in Pittsburgh) this week, so Wednesday’s lecture (10/16/2019 will be youtube only (will be posted around Wednesday). Next Monday’s lecture will continue from the youtube lecture.
Towards AEP

- We wish to encode these $K^n$ outcomes with binary digit strings (i.e., code words) of length $m$. Hence, $M = 2^m$ possible code words.
- We can represent the encoder as follows:

  \[
  \begin{align*}
  \text{Source messages} & \quad \{X_1, X_2, \ldots, X_n\} \\
  \text{Encoder} & \quad \rightarrow \\
  \text{Code words} & \quad \{Y_1, Y_2, \ldots, Y_m\}
  \end{align*}
  \]

  \[
  \begin{align*}
  X_i & \in \{a_1, a_2, \ldots, a_K\} \\
  K^n & \text{ possible messages} \\
  n & \text{ source letters in each source msg} \\
  Y_i & \in \{0, 1\} \\
  2^m & \text{ possible messages} \\
  m & \text{ total bits}
  \end{align*}
  \]

- Example: English letters, would have $K = 26$ (alphabet size $K$), a “source message” consists of $n$ letters.
- Want to have a code word for every possible source message. Must have what condition? Num. Code Words $\geq$ Num. Messages, or:

  \[
  M = 2^m \geq K^n \quad \Rightarrow \quad m \geq n \log K \quad (6.17)
  \]
Towards AEP

- A question on rate: How many bits are used per source letter?
  \[
  R = \text{rate} = \frac{\log M}{n} = \frac{m}{n} \geq \log K \text{ bits per source letter} \quad (6.17)
  \]

  Not surprising, e.g., for English need \([\log K] = 5\) bits.

- Question: can we use fewer than this bits per source letter (on average) and still have “essentially” no error? Yes.

- How? One way: some source messages would not have a code.

  ![Diagram showing source messages and code words, with garbage assigned to a subset of the source messages.]

  I.e., code words only assigned to a subset of the source messages!
Towards AEP

- Any source message assigned to garbage would, if we wish to send that message, result in an error.
- Alternatively, and perhaps less distressingly, rather than throw some messages into the trash, we could assign to them long code words, and the non-garbage messages to short code words.

<table>
<thead>
<tr>
<th>Source messages</th>
<th>Short Code words</th>
<th>Long Code words</th>
</tr>
</thead>
</table>

- In either case, if $n$ gets big enough, we make the code such that the probability of getting one of those error source messages (or long-code-word source messages) very small!
Asymptotic Equipartition Property (AEP)

**Theorem 6.2.2 (AEP)**

If $X_1, X_2, \ldots, X_n$ are i.i.d. and $X_i \sim p(x)$ for all $i$, then

$$
-\frac{1}{n} \log p(X_1, X_2, \ldots, X_n) \xrightarrow{p} H(X) \quad (6.38)
$$

**Proof.**

$$
-\frac{1}{n} \log p(X_1, X_2, \ldots, X_n) = -\frac{1}{n} \log \prod_{i=1}^{n} p(X_i) = -\frac{1}{n} \sum_{i} \log p(X_i) \xrightarrow{p} -E \log p(X) \quad (6.40)
$$

$$
= H(X) \quad (6.41)
$$
Definition 6.2.2 (Typical Set)

The typical set $A^{(n)}_{\epsilon}$ w.r.t. $p(x)$ is the set of sequences $(x_1, x_2, \ldots, x_n) \in \mathcal{X}^n$ with the property that

$$2^{-n(H(X)+\epsilon)} \leq p(x_1, x_2, \ldots, x_n) \leq 2^{-n(H(X)-\epsilon)} \quad (6.38)$$

Equivalently, we may write $A^{(n)}_{\epsilon}$ as

$$A^{(n)}_{\epsilon} = \left\{ (x_1, x_2, \ldots, x_n) : \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| < \epsilon \right\} \quad (6.39)$$

- Typical set are those sequences with log probability within the range $-nH$

- $A^{(n)}_{\epsilon}$ has a number of interesting properties.
Typical Set $A^{(n)}_\epsilon$

Theorem 6.2.2 (Properties of $A^{(n)}_\epsilon$)

1. If $(x_1, x_2, \ldots, x_n) \in A^{(n)}_\epsilon$, then
   \[
   H(X) - \epsilon \leq -\frac{1}{n} \log p(x_1, x_2, \ldots, x_n) \leq H(X) + \epsilon
   \]  
   (6.38)

2. $p(A^{(n)}_\epsilon) = p\left(\left\{x : x \in A^{(n)}_\epsilon\right\}\right) > 1 - \epsilon$ for large enough $n$, for all $\epsilon > 0$.

3. **Upper bound:** $|A^{(n)}_\epsilon| \leq 2^{n(H(X)+\epsilon)}$, where $|A|$ is the number of elements in set $A$.

4. **Lower bound:** $|A^{(n)}_\epsilon| \geq (1 - \epsilon)2^{n(H(X)-\epsilon)}$ for large enough $n$.

- The typical set has, essentially, probability 1 (something typical will typically occur).
- All items in that set will have the same probability, $\approx 2^{-nH}$.
- The number of elements in that set is $\approx 2^{nH}$.
Binomial Distribution, when $n$ gets big, $p = 0.5$

$$\Pr(S_n = k) = \binom{n}{k} p^k q^{n-k}, \quad S_n = X_1 + X_2 + \cdots + X_n, \quad X_i \sim \text{Bernoulli}(p)$$

- What happens when $n$ gets big?
- Plot the probability of the normalized values, $S_n/n = k/n$, and see how the distribution changes when $n$ gets large.

![Graph showing the distribution changes with increasing $n$.](image-url)
Binomial Distribution, when $n$ gets big, $p = 0.9$

$$\Pr(S_n = k) = \binom{n}{k} p^k q^{n-k}, \quad S_n = X_1 + X_2 + \cdots + X_n, \quad X_i \sim \text{Bernoulli}(p)$$

- Plot the probability of the normalized values, $S_n/n = k/n$, and see how the distribution changes when $n$ gets large.
- By comparison, number of atoms in observable universe $\approx e^{187} \approx 10^{81}$. 
Typical Set $A_\epsilon^{(n)}$

Theorem 6.3.2 (Properties of $A_\epsilon^{(n)}$)

1. If $(x_1, x_2, \ldots, x_n) \in A_\epsilon^{(n)}$, then
   \[ H(X) - \epsilon \leq -\frac{1}{n} \log p(x_1, x_2, \ldots, x_n) \leq H(X) + \epsilon \]  
   (6.38)

2. $p(A_\epsilon^{(n)}) = p \left( \left\{ x : x \in A_\epsilon^{(n)} \right\} \right) > 1 - \epsilon$ for large enough $n$, for all $\epsilon > 0$.

3. Upper bound: $|A_\epsilon^{(n)}| \leq 2^{n(H(X) + \epsilon)}$, where $|A|$ is the number of elements in set $A$.

4. Lower bound: $|A_\epsilon^{(n)}| \geq (1 - \epsilon)2^{n(H(X) - \epsilon)}$ for large enough $n$

- The typical set has, essentially, probability 1 (something typical will typically occur).
- All items in that set will have the same probability, $\approx 2^{-nH}$.
- The number of elements in that set is $\approx 2^{nH}$.
Proof of Theorem 5.5.3.

1. This is a restatement of the AEP definition.
Proof of Theorem 5.5.3.

1. This is a restatement of the AEP definition.

2. Use the expanded definition of convergence in probability we saw earlier in class

\[ p(A_{\epsilon}^{(n)}) = p \left( \left| -\frac{1}{n} \sum_{i} \log p(x_i) - H \right| < \epsilon \right) > 1 - \delta \text{ for } n \text{ big enough} \tag{6.1} \]

and we can chose any \( \delta \) we wish, so choose \( \delta = \epsilon \), giving

\[ p(A_{\epsilon}^{(n)}) > 1 - \epsilon \text{ for } n \text{ big enough } \forall \epsilon \tag{6.2} \]

...
Proof of Theorem 5.5.3.

Upper bound size of $A^{(n)}_\varepsilon$

\[
\frac{1}{\varepsilon} = \sum_{x \in X} p(x)^2, \quad 2^n \left( H(X) + \varepsilon \right) \leq |A^{(n)}_\varepsilon|^2.
\] (6.4)

...
Proof of Theorem 5.5.3.

3. Upper bound size of $A_{\epsilon}^{(n)}$

\[1\]

(6.4)
Proof of Theorem 5.5.3.

3 Upper bound size of $A^{(n)}_\epsilon$

$$1 = \sum_x p(x)$$

(6.4)

...
Proof of Theorem 5.5.3.

3 Upper bound size of $A^{(n)}_{\epsilon}$

$$1 = \sum_{x} p(x) \geq \sum_{x \in A^{(n)}_{\epsilon}} p(x)$$

(6.4)
Proof of Theorem 5.5.3.

3 Upper bound size of \( A_\epsilon^{(n)} \)

\[
1 = \sum_{x} p(x) \geq \sum_{x \in A_\epsilon^{(n)}} p(x) \geq \sum_{x \in A_\epsilon^{(n)}} 2^{-n(H(X)+\epsilon)}
\]  
(6.3)

\[ (6.4) \]

...
Proof of Theorem 5.5.3.

Upper bound size of $A_{\epsilon}^{(n)}$

$$1 = \sum_{x} p(x) \geq \sum_{x \in A_{\epsilon}^{(n)}} p(x) \geq \sum_{x \in A_{\epsilon}^{(n)}} 2^{-n(H(X)+\epsilon)} \tag{6.3}$$

$$= |A_{\epsilon}^{(n)}|2^{-n(H(X)+\epsilon)} \tag{6.4}$$

...
Proof of Theorem 5.5.3.

3 Upper bound size of $A_{\epsilon}^{(n)}$

$$1 = \sum_{x} p(x) \geq \sum_{x \in A_{\epsilon}^{(n)}} p(x) \geq \sum_{x \in A_{\epsilon}^{(n)}} 2^{-n(H(X)+\epsilon)} \quad (6.3)$$

$$= |A_{\epsilon}^{(n)}| 2^{-n(H(X)+\epsilon)} \quad (6.4)$$

giving $|A_{\epsilon}^{(n)}| \leq 2^{n(H+\epsilon)}$. 

...
Proof of Theorem 5.5.3.

4. Lower bound size of $A^{(n)}_\epsilon$. For large enough $n$

\begin{align*}
\forall \epsilon > 0, \quad &\text{size of } A^{(n)}_\epsilon \leq 2^n (H(X) + \epsilon) \\
\Rightarrow &\text{lower bound size of } A^{(n)}_\epsilon
\end{align*}

(6.6)
Proof of Theorem 5.5.3.

4. Lower bound size of $A^{(n)}_\epsilon$. For large enough $n$

\[ 1 - \epsilon \]

(6.6)
Proof of Theorem 5.5.3.

4 Lower bound size of $A^{(n)}_\epsilon$. For large enough $n$

\[ 1 - \epsilon < p(A^{(n)}_\epsilon) \]

(6.6)

...
Proof of Theorem 5.5.3.

4 Lower bound size of $A^{(n)}_\epsilon$. For large enough $n$

$$1 - \epsilon < p(A^{(n)}_\epsilon) \leq \sum_{x \in A^{(n)}_\epsilon} 2^{-n(H(X) - \epsilon)}$$  \hspace{1cm} (6.5) \\
(6.6)

...
Proof of Theorem 5.5.3.

Lower bound size of $A_{\epsilon}^{(n)}$. For large enough $n$

$$1 - \epsilon < p(A_{\epsilon}^{(n)}) \leq \sum_{x \in A_{\epsilon}^{(n)}} 2^{-n(H(X) - \epsilon)}$$  \hfill (6.5)

$$= 2^{-n(H(X) - \epsilon)} |A_{\epsilon}^{(n)}|$$  \hfill (6.6)
Proof of Theorem 5.5.3.

4 Lower bound size of $A_\epsilon^{(n)}$. For large enough $n$

\[
1 - \epsilon < p(A_\epsilon^{(n)}) \leq \sum_{x \in A_\epsilon^{(n)}} 2^{-n(H(X) - \epsilon)} \tag{6.5}
\]

\[
= 2^{-n(H(X) - \epsilon)} |A_\epsilon^{(n)}| \tag{6.6}
\]

giving $|A_\epsilon^{(n)}| \geq (1 - \epsilon)2^{n(H(X) - \epsilon)}$
An important consequence of this is that we can compress data down to the entropy of the source.
Data Compression to the entropy of the source

- An important consequence of this is that we can compress data down to the entropy of the source.
- Idea: Consider $X_1, X_2, \ldots, X_n$ i.i.d. and $\sim p(x)$. 

Partition the set of sequences into two blocks:
- $A(n)$
- its complement, the non-typical sets $X_n \cap A(n)$

A partition, i.e., $A(n) = \emptyset$; and $A(n) \cup A(n)^c = X_n$: 
Data Compression to the entropy of the source

- An important consequence of this is that we can compress data down to the entropy of the source.
- Idea: Consider $X_1, X_2, \ldots, X_n$ i.i.d. and $\sim p(x)$.
- Partition the set of sequences into two blocks:
Data Compression to the entropy of the source

- An important consequence of this is that we can compress data down to the entropy of the source.
- Idea: Consider $X_1, X_2, \ldots, X_n$ i.i.d. and $\sim p(x)$.
- Partition the set of sequences into two blocks:
  - The typical sets $A^{(n)}_\epsilon$, ...
Data Compression to the entropy of the source

- An important consequence of this is that we can compress data down to the entropy of the source.
- Idea: Consider $X_1, X_2, \ldots, X_n$ i.i.d. and $\sim p(x)$.
- Partition the set of sequences into two blocks:
  - The typical sets $A^{(n)}_\epsilon$, ...
  - and its complement, the non-typical sets $\mathcal{X}^n \setminus A^{(n)}_\epsilon \triangleq A^{(n)}_\epsilon^c$
Data Compression to the entropy of the source

- An important consequence of this is that we can compress data down to the entropy of the source.
- Idea: Consider $X_1, X_2, \ldots, X_n$ i.i.d. and $\sim p(x)$.
- Partition the set of sequences into two blocks:
  - The typical sets $A^{(n)}_\varepsilon$, …
  - and its complement, the non-typical sets $\mathcal{X}^n \setminus A^{(n)}_\varepsilon \triangleq A^{(n)c}_\varepsilon$
- A partition, i.e., $A^{(n)}_\varepsilon \cap A^{(n)c}_\varepsilon = \emptyset$ and $A^{(n)}_\varepsilon \cup A^{(n)c}_\varepsilon = \mathcal{X}^n$:

\[ \mathcal{X}^n \text{ having } |\mathcal{X}^n| = K^n \text{ elements} \]
Typical Set Compression

- We index the elements in each of the sets, the typical set and the non-typical set, separately.
Typical Set Compression

- We index the elements in each of the sets, the typical set and the non-typical set, separately.

- In the typical set, \( \exists |A_{\epsilon}^{(n)}| \leq 2^{n(H+\epsilon)} \) elements, requiring

\[
\lceil n(H + \epsilon) \rceil \leq n(H + \epsilon) + 1 \text{ bits.} \quad (6.7)
\]
Typical Set Compression

- We index the elements in each of the sets, the typical set and the non-typical set, separately.
- In the typical set, \( \exists |A_{\epsilon}^{(n)}| \leq 2^{n(H+\epsilon)} \) elements, requiring

\[
[n(H + \epsilon)] \leq n(H + \epsilon) + 1 \text{ bits.} \quad (6.7)
\]

- We use an extra bit at the beginning to indicate if it is typical or not, i.e., we use

\[
(b_0, b_1, b_2, \ldots, b_{\lfloor n(H+\epsilon) \rfloor}) \quad (6.8)
\]

which indexes which of the typical set elements it is. \( b_0 = 0 \) indicating that the set is typical.
Typical Set Compression

- We index the elements in each of the sets, the typical set and the non-typical set, separately.

- In the typical set, \( |A_\epsilon^{(n)}| \leq 2^{n(H+\epsilon)} \) elements, requiring
  
  \[
  \left\lfloor n(H + \epsilon) \right\rfloor \leq n(H + \epsilon) + 1 \text{ bits.} \tag{6.7}
  \]

- We use an extra bit at the beginning to indicate if it is typical or not, i.e., we use

  \[
  (b_0, b_1, b_2, \ldots, b_{\left\lfloor n(H+\epsilon) \right\rfloor}) \tag{6.8}
  \]

  which indexes which of the typical set elements it is. \( b_0 = 0 \) indicating that the set is typical.

- Total number of bits required is \( n(H + \epsilon) + 2 \) for a typical sequence.
Typical Set Compression

- We index the elements in each of the sets, the typical set and the non-typical set, separately.

\[(b_0, b_1, b_2, ..., b_{d \log |X|_n})\] (6.9)

where here \(b_0 = 1\) indicating atypicality.

Total number of bits for an atypical sequence is \(n \log K + 2\).

Note, this is our first code for the class!

This is called source coding or compression, and entails finding a sequence of bits for each source string so that average length is as short as possible.
Typical Set Compression

- We index the elements in each of the sets, the typical set and the non-typical set, separately.

- In the non-typical set, we index everything. I.e., we use
  \[ \lceil \log |\mathcal{X}|^n \rceil \leq n \log K + 1 \text{ bits.} \]
Typical Set Compression

- We index the elements in each of the sets, the typical set and the non-typical set, separately.

- In the non-typical set, we index everything. I.e., we use \( \lceil \log |X|^n \rceil \leq n \log K + 1 \) bits.

- I.e., we index everything with a bit vector of the form

  \[
  (b_0, b_1, b_2, \ldots, b_{\lceil \log |X|^n \rceil})
  \]  

  (6.9)

  where here \( b_0 = 1 \) indicating atypicality.
Typical Set Compression

- We index the elements in each of the sets, the typical set and the non-typical set, separately.
- In the non-typical set, we index everything. I.e., we use $\lceil \log |\mathcal{X}|^n \rceil \leq n \log K + 1$ bits.
- I.e., we index everything with a bit vector of the form

$$ (b_0, b_1, b_2, \ldots, b_{\lceil \log |\mathcal{X}|^n \rceil} ) $$

where here $b_0 = 1$ indicating atypicality.
- Total number of bits for an atypical sequence is $n \log K + 2$. 
Typical Set Compression

- We index the elements in each of the sets, the typical set and the non-typical set, separately.

- In the non-typical set, we index everything. I.e., we use $\lceil \log |\mathcal{X}|^n \rceil \leq n \log K + 1$ bits.

- I.e., we index everything with a bit vector of the form

$$\left( b_0, b_1, b_2, \ldots, b_{\lceil \log |\mathcal{X}|^n \rceil} \right)$$

where here $b_0 = 1$ indicating atypicality.

- Total number of bits for an atypical sequence is $n \log K + 2$.

- Note, this is our first code for the class! This is called source coding or compression, and entails finding a sequence of bits for each source string so that average length is as short as possible.
Typical Set Compression: Features of our code, and setup

- Code is 1-to-1, so easy to decode and encode, given code book (mapping).
Typical Set Compression: Features of our code, and setup

- Code is 1-to-1, so easy to decode and encode, given code book (mapping).
- Simple but brute force enumeration of atypical set $A_{\epsilon}(n)_c$ means more bits than necessary, since
  \[ |A_{\epsilon}(n)_c| = |\mathcal{X}^n| - |A_{\epsilon}(n)| = Kn - |A_{\epsilon}(n)| < Kn, \]
Typical Set Compression: Features of our code, and setup

- Code is 1-to-1, so easy to decode and encode, given code book (mapping).
- Simple but brute force enumeration of atypical set $A^{(n)c}_\epsilon$ means more bits than necessary, since

$$|A^{(n)c}_\epsilon| = |\mathcal{X}^n| - |A^{(n)}_\epsilon| = K^n - |A^{(n)}_\epsilon| < K^n,$$

but this is not going to matter, as we will see.
Code is 1-to-1, so easy to decode and encode, given code book (mapping).

Simple but brute force enumeration of atypical set $A_{\epsilon}^{(n)c}$ means more bits than necessary, since
\[ |A_{\epsilon}^{(n)c}| = |\mathcal{X}^n| - |A_{\epsilon}^{(n)}| = K^n - |A_{\epsilon}^{(n)}| < K^n, \]
but this is not going to matter, as we will see.

Typical sequences have a “short” description length, $\approx nH$. 

Typical Set Compression: Features of our code, and setup

- Code is 1-to-1, so easy to decode and encode, given code book (mapping).
- Simple but brute force enumeration of atypical set $A^{(n)c}_\epsilon$ means more bits than necessary, since $|A^{(n)c}_\epsilon| = |\mathcal{X}^n| - |A^{(n)}_\epsilon| = K^n - |A^{(n)}_\epsilon| < K^n$, but this is not going to matter, as we will see.
- Typical sequences have a “short” description length, $\approx nH$.
- Let $\ell(x_{1:n})$ be length of the codeword assigned to sequence $x_{1:n}$. 

Typical Set Compression: Features of our code, and setup

- Code is 1-to-1, so easy to decode and encode, given code book (mapping).

- Simple but brute force enumeration of atypical set $A^{(n)c}_\epsilon$ means more bits than necessary, since
  \[ |A^{(n)c}_\epsilon| = |\mathcal{X}^n| - |A^{(n)}_\epsilon| = K^n - |A^{(n)}_\epsilon| < K^n, \]
  but this is not going to matter, as we will see.

- Typical sequences have a “short” description length, $\approx nH$.

- Let $\ell(x_{1:n})$ be length of the codeword assigned to sequence $x_{1:n}$.

- $\ell(X_{1:n})$ is a random variable since $X_{1:n}$ is a random variable.
Typical Set Compression: Features of our code, and setup

- Code is 1-to-1, so easy to decode and encode, given code book (mapping).

- Simple but brute force enumeration of atypical set $A^{(n)c}_\epsilon$ means more bits than necessary, since $|A^{(n)c}_\epsilon| = |X^n| - |A^{(n)}_\epsilon| = K^n - |A^{(n)}_\epsilon| < K^n$, but this is not going to matter, as we will see.

- Typical sequences have a “short” description length, $\approx nH$.

- Let $\ell(x_{1:n})$ be length of the codeword assigned to sequence $x_{1:n}$.

- $\ell(X_{1:n})$ is a random variable since $X_{1:n}$ is a random variable.

- Thus, $E\ell(X_{1:n}) = \sum_{x_{1:n}} p(x_{1:n})\ell(x_{1:n})$ is the average, or expected, length of our code. We want this to be as short as possible.
Expected Length

Suppose that \( n \) is large enough so that \( p(A_{\epsilon}^{(n)}) > 1 - \epsilon \),
Expected Length

- Suppose that $n$ is large enough so that $p(A_{\epsilon}^{(n)}) > 1 - \epsilon$, 
Expected Length

- Suppose that $n$ is large enough so that $p(A^{(n)}_\epsilon) > 1 - \epsilon$, then

$$E\ell(X_{1:n}) = \sum_{x_{1:n}} p(x_{1:n})\ell(x_{1:n})$$ \hspace{1cm} (6.10)

$$\sum_{x_{1:n}} p(x_{1:n})\ell(x_{1:n})$$ \hspace{1cm} (6.11)

$$\sum_{x_{1:n}} p(x_{1:n})\ell(x_{1:n})$$ \hspace{1cm} (6.12)

$$\sum_{x_{1:n}} p(x_{1:n})\ell(x_{1:n})$$ \hspace{1cm} (6.13)

$$\sum_{x_{1:n}} p(x_{1:n})\ell(x_{1:n})$$ \hspace{1cm} (6.14)

$$\sum_{x_{1:n}} p(x_{1:n})\ell(x_{1:n})$$ \hspace{1cm} (6.15)
Expected Length

Suppose that \( n \) is large enough so that \( p(A_\epsilon^{(n)}) > 1 - \epsilon \), then

\[
E\ell(X_{1:n}) = \sum_{x_{1:n}} p(x_{1:n})\ell(x_{1:n})
\]

(6.10)

\[
= \sum_{x_{1:n} \in A_\epsilon^{(n)}} p(x_{1:n})\ell(x_{1:n}) + \sum_{x_{1:n} \in A_\epsilon^{(n)c}} p(x_{1:n})\ell(x_{1:n})
\]

(6.11)

\[
= p(A_\epsilon^{(n)}) [n(H + \epsilon) + 2] + p(A_\epsilon^{(n)c}) [n\log K + 2]
\]

(6.12)

\[
\leq [n(H + \epsilon) + 2] + p(A_\epsilon^{(n)c}) [n\log K + 2]
\]

(6.13)

\[
= n \cdot H + \epsilon \cdot n + \epsilon \cdot n \cdot \log K + 2 + \epsilon \cdot n \cdot \log K + 2
\]

(6.14)

\[
= n \cdot H + \epsilon \cdot n + \epsilon \cdot n \cdot \log K + 2 + \epsilon \cdot n \cdot \log K + 2
\]

(6.15)
Suppose that \( n \) is large enough so that \( p(A_{\varepsilon}^{(n)}) > 1 - \varepsilon \), then

\[
E\ell(X_{1:n}) = \sum_{x_{1:n}} p(x_{1:n})\ell(x_{1:n})
\]

\[
= \sum_{x_{1:n} \in A_{\varepsilon}^{(n)}} p(x_{1:n})\ell(x_{1:n}) + \sum_{x_{1:n} \in A_{\varepsilon}^{(n)c}} p(x_{1:n})\ell(x_{1:n})
\]  \hspace{1cm} (6.10)

\[
\leq \sum_{x_{1:n} \in A_{\varepsilon}^{(n)}} p(x_{1:n})[n(H + \varepsilon) + 2] + \sum_{x_{1:n} \in A_{\varepsilon}^{(n)c}} p(x_{1:n})[n \log K + 2]
\]  \hspace{1cm} (6.11)

\[
\leq n[H + \varepsilon + n \log K + 2]
\]  \hspace{1cm} (6.12)

\[
\leq n[H + \varepsilon + n \log K + 2]
\]  \hspace{1cm} (6.13)

\[
\leq n[H + \varepsilon + n \log K + 2]
\]  \hspace{1cm} (6.14)

\[
\leq n[H + \varepsilon + n \log K + 2]
\]  \hspace{1cm} (6.15)
Expected Length

Suppose that $n$ is large enough so that $p(A^{(n)}_{\epsilon}) > 1 - \epsilon$, then

$$E\ell(X_{1:n}) = \sum_{x_{1:n}} p(x_{1:n})\ell(x_{1:n})$$ (6.10)

$$= \sum_{x_{1:n} \in A^{(n)}_{\epsilon}} p(x_{1:n})\ell(x_{1:n}) + \sum_{x_{1:n} \in A^{(n)}_{\epsilon}^c} p(x_{1:n})\ell(x_{1:n})$$ (6.11)

$$\leq \sum_{x_{1:n} \in A^{(n)}_{\epsilon}} p(x_{1:n})[n(H + \epsilon) + 2] + \sum_{x_{1:n} \in A^{(n)}_{\epsilon}^c} p(x_{1:n})[n \log K + 2]$$ (6.12)

$$= p(A^{(n)}_{\epsilon})[n(H + \epsilon) + 2] + p(A^{(n)}_{\epsilon}^c)[n \log K + 2]$$ (6.13)

(6.15)
Expected Length

Suppose that $n$ is large enough so that $p(A_{\varepsilon}^{(n)}) > 1 - \varepsilon$, then

$$E\ell(X_{1:n}) = \sum_{x_{1:n}} p(x_{1:n})\ell(x_{1:n})$$  \hspace{1cm} (6.10)

$$= \sum_{x_{1:n} \in A_{\varepsilon}^{(n)}} p(x_{1:n})\ell(x_{1:n}) + \sum_{x_{1:n} \in A_{\varepsilon}^{(n)c}} p(x_{1:n})\ell(x_{1:n})$$  \hspace{1cm} (6.11)

$$\leq \sum_{x_{1:n} \in A_{\varepsilon}^{(n)}} p(x_{1:n})[n(H + \varepsilon) + 2] + \sum_{x_{1:n} \in A_{\varepsilon}^{(n)c}} p(x_{1:n})[n \log K + 2]$$  \hspace{1cm} (6.12)

$$= p(A_{\varepsilon}^{(n)})[n(H + \varepsilon) + 2] + p(A_{\varepsilon}^{(n)c})[n \log K + 2]$$  \hspace{1cm} (6.13)

$$\leq 1$$

(6.15)
**Expected Length**

Suppose that $n$ is large enough so that $p(A^{(n)}_\epsilon) > 1 - \epsilon$, then

$$E\ell(X_1:n) = \sum_{x_1:n} p(x_1:n)\ell(x_1:n)$$  \hspace{1cm} (6.10)

$$= \sum_{x_1:n \in A^{(n)}_\epsilon} p(x_1:n)\ell(x_1:n) + \sum_{x_1:n \in A^{(n)}_\epsilon^c} p(x_1:n)\ell(x_1:n)$$  \hspace{1cm} (6.11)

$$\leq \sum_{x_1:n \in A^{(n)}_\epsilon} p(x_1:n)[n(H + \epsilon) + 2] + \sum_{x_1:n \in A^{(n)}_\epsilon^c} p(x_1:n)[n \log K + 2]$$  \hspace{1cm} (6.12)

$$= \underbrace{p(A^{(n)}_\epsilon)}_{\leq 1} [n(H + \epsilon) + 2] + \underbrace{p(A^{(n)}_\epsilon^c)}_{< \epsilon} [n \log K + 2]$$  \hspace{1cm} (6.13)

$$\leq n(H + \epsilon) + 2 + \epsilon n \log K + 2$$  \hspace{1cm} (6.15)
Expected Length

- Suppose that $n$ is large enough so that $p(A^{(n)}_\epsilon) > 1 - \epsilon$, then

$$E\ell(X_{1:n}) = \sum_{x_{1:n}} p(x_{1:n})\ell(x_{1:n})$$  \hspace{1cm} (6.10)

$$= \sum_{x_{1:n} \in A^{(n)}_\epsilon} p(x_{1:n})\ell(x_{1:n}) + \sum_{x_{1:n} \in A^{(n)}_\epsilon^c} p(x_{1:n})\ell(x_{1:n})$$  \hspace{1cm} (6.11)

$$\leq \sum_{x_{1:n} \in A^{(n)}_\epsilon} p(x_{1:n})[n(H + \epsilon) + 2] + \sum_{x_{1:n} \in A^{(n)}_\epsilon^c} p(x_{1:n})[n \log K + 2]$$  \hspace{1cm} (6.12)

$$= \underbrace{p(A^{(n)}_\epsilon)}_{\leq 1} [n(H + \epsilon) + 2] + \underbrace{p(A^{(n)}_\epsilon^c)}_{< \epsilon} [n \log K + 2]$$  \hspace{1cm} (6.13)

$$\leq n(H + \epsilon) + 2 + \epsilon n \log K + \epsilon 2$$  \hspace{1cm} (6.14)

$$\leq n(H + \epsilon) + 2 + \epsilon n \log K + \epsilon 2$$  \hspace{1cm} (6.15)
**Expected Length**

Suppose that $n$ is large enough so that $p(A^{(n)}_\epsilon) > 1 - \epsilon$, then

$$E\ell(X_1:n) = \sum_{x_1:n} p(x_1:n)\ell(x_1:n)$$

$$= \sum_{x_1:n \in A^{(n)}_\epsilon} p(x_1:n)\ell(x_1:n) + \sum_{x_1:n \in A^{(n)}_\epsilon^c} p(x_1:n)\ell(x_1:n)$$

$$\leq \sum_{x_1:n \in A^{(n)}_\epsilon} p(x_1:n)[n(H + \epsilon) + 2] + \sum_{x_1:n \in A^{(n)}_\epsilon^c} p(x_1:n)[n \log K + 2]$$

$$= p(A^{(n)}_\epsilon)[n(H + \epsilon) + 2] + p(A^{(n)}_\epsilon^c)[n \log K + 2]$$

$$\leq n(H + \epsilon) + 2 + \epsilon n \log K + \epsilon 2$$

$$= n[H + \epsilon + \epsilon' \log K + \frac{2}{n} + \frac{2\epsilon}{n}]$$

where $\epsilon' = \epsilon n^{-1}$.
Expected Length

Suppose that $n$ is large enough so that $p(A^{(n)}_\epsilon) > 1 - \epsilon$, then

$$E\ell(X_{1:n}) = \sum_{x_{1:n}} p(x_{1:n})\ell(x_{1:n})$$

$$= \sum_{x_{1:n} \in A^{(n)}_\epsilon} p(x_{1:n})\ell(x_{1:n}) + \sum_{x_{1:n} \in A^{(n)}_\epsilon^c} p(x_{1:n})\ell(x_{1:n})$$

$$\leq \sum_{x_{1:n} \in A^{(n)}_\epsilon} p(x_{1:n})[n(H + \epsilon) + 2] + \sum_{x_{1:n} \in A^{(n)}_\epsilon^c} p(x_{1:n})[n \log K + 2]$$

$$= p(A^{(n)}_\epsilon) [n(H + \epsilon) + 2] + p(A^{(n)}_\epsilon^c) [n \log K + 2]$$

$$\leq n(H + \epsilon) + 2 + \epsilon n \log K + \epsilon 2$$

$$= n[H + \epsilon + \epsilon \log K + \frac{2}{n} + \frac{2\epsilon}{n}] = n(H + \epsilon')$$
Expected Length

- But $\epsilon' = \epsilon + \epsilon \log K + \frac{2}{n} + \frac{2\epsilon}{n}$ can be made as small as we wish by making $\epsilon$ small and $n$ large.
Expected Length

- But $\epsilon' = \epsilon + \epsilon \log K + \frac{2}{n} + \frac{2\epsilon}{n}$ can be made as small as we wish by making $\epsilon$ small and $n$ large.
- Thus, $n(H + \epsilon')$ can be made as close as we want to $nH$ by making $\epsilon$ small and $n$ large.
Expected Length

- But $\epsilon' = \epsilon + \epsilon \log K + \frac{2}{n} + \frac{2\epsilon}{n}$ can be made as small as we wish by making $\epsilon$ small and $n$ large.
- Thus, $n(H + \epsilon')$ can be made as close as we want to $nH$ by making $\epsilon$ small and $n$ large.
- We have just proven the following theorem:
Expected Length

- But $\epsilon' = \epsilon + \epsilon \log K + \frac{2}{n} + \frac{2\epsilon}{n}$ can be made as small as we wish by making $\epsilon$ small and $n$ large.
- Thus, $n(H + \epsilon')$ can be made as close as we want to $nH$ by making $\epsilon$ small and $n$ large.
- We have just proven the following theorem:

**Theorem 6.4.1**

Let $X_1:n$ be i.i.d. $\sim p(x)$, $\epsilon > 0$, then $\exists$ a code $f_n : \mathcal{X}^n \to$ binary strings and integer $n_\epsilon$, such that the mapping is one-to-one (so invertible w/o error), and

$$E\left[\frac{1}{n} \ell(X_1:n)\right] \leq H(X) + \epsilon$$  \hspace{1cm} (6.16)

for all $\epsilon > 0$ and for all $n \geq n_\epsilon$. 
Expected Length

- But $\epsilon' = \epsilon + \epsilon \log K + \frac{2}{n} + \frac{2\epsilon}{n}$ can be made as small as we wish by making $\epsilon$ small and $n$ large.
- Thus, $n(H + \epsilon')$ can be made as close as we want to $nH$ by making $\epsilon$ small and $n$ large.
- We have just proven the following theorem:

Theorem 6.4.1

Let $X_1:n$ be i.i.d. $\sim p(x)$, $\epsilon > 0$, then $\exists$ a code $f_n : X^n \rightarrow$ binary strings and integer $n_\epsilon$, such that the mapping is one-to-one (so invertible w/o error), and

$$E\left[\frac{1}{n}\ell(X_1:n)\right] \leq H(X) + \epsilon$$  \hspace{1cm} (6.16)

for all $\epsilon > 0$ and for all $n \geq n_\epsilon$.

- Thus, it takes at most $nH(X)$ bits to represent $X_1:n$ on average, or $H(X)$ bits per source alphabet symbol.
Shannon’s source coding theorem

The previous theorem is Shannon’s first theorem, stating that it is possible (using long block lengths) to compress down arbitrarily close to the entropy limit.
Shannon’s source coding theorem

- The previous theorem is Shannon’s first theorem, stating that it is possible (using long block lengths) to compress down arbitrarily close to the entropy limit.

- An instance of universal source coding, coding without explicitly using the distribution, since whatever happens, once $n$ gets large, is all that will happen.
Shannon’s source coding theorem

- The previous theorem is Shannon’s first theorem, stating that it is possible (using long block lengths) to compress down arbitrarily close to the entropy limit.

- An instance of universal source coding, coding without explicitly using the distribution, since whatever happens, once $n$ gets large, is all that will happen.

- Ex: online coding, code only those things that you encounter knowing that it must be typical if you encounter it, if $n$ is large enough. In such case, you don’t need $p(x)$, only $H(p)$. 

The previous theorem is Shannon’s first theorem, stating that it is possible (using long block lengths) to compress down arbitrarily close to the entropy limit.

An instance of universal source coding, coding without explicitly using the distribution, since whatever happens, once $n$ gets large, is all that will happen.

Ex: online coding, code only those things that you encounter knowing that it must be typical if you encounter it, if $n$ is large enough. In such case, you don’t need $p(x)$, only $H(p)$.

Ultimately, we need to prove that we can’t compress to lower than the entropy limit without incurring error, this is the converse of the theorem that we will prove soon.
Other high probable sets?

- We know that $p(X^n) = 1$. 

But is it smallest? Is there a smaller set than the typical one that has "all" of the probability? I.e., are all elements in $A(n)\in$ essential (i.e., contribute significantly to the probability)?

Answer, as we will see, is no. I.e., $A(n)\in$ is the smallest set that has "all" of the probability.
Other high probable sets?

- We know that \( p(\mathcal{X}^n) = 1 \).
- So \( A^{(n)}_\epsilon \) is smaller, and \( P(A^{(n)}_\epsilon) \approx 1 \).
Other high probable sets?

- We know that $p(\mathcal{X}^n) = 1$.
- So $A_{\epsilon}^{(n)}$ is smaller, and $P(A_{\epsilon}^{(n)}) \approx 1$.
- But is it smallest? Is there a smaller set than the typical one that has “all” of the probability? I.e., are all elements in $A_{\epsilon}^{(n)}$ essential (i.e., contribute significantly to the probability)?
Other high probable sets?

- We know that $p(X^n) = 1$.
- So $A_\epsilon^{(n)}$ is smaller, and $P(A_\epsilon^{(n)}) \approx 1$.
- But is it smallest? Is there a smaller set than the typical one that has “all” of the probability? I.e., are all elements in $A_\epsilon^{(n)}$ essential (i.e., contribute significantly to the probability)?
- If so, maybe we can code for this still smaller set and achieve even better compression rate.

Answer, as we will see, is no. I.e., $A_\epsilon^{(n)}$ is the smallest set that has “all” of the probability.
Other high probable sets?

- We know that \( p(X^n) = 1 \).
- So \( A_\varepsilon^{(n)} \) is smaller, and \( P(A_\varepsilon^{(n)}) \approx 1 \).
- But is it smallest? Is there a smaller set than the typical one that has “all” of the probability? I.e., are all elements in \( A_\varepsilon^{(n)} \) essential (i.e., contribute significantly to the probability)?
- If so, maybe we can code for this still smaller set and achieve even better compression rate.
- Answer, as we will see, is no. I.e., \( A_\varepsilon^{(n)} \) is the smallest set that has “all” of the probability.
Other high probable sets?

- Let $B_{\delta}^{(n)}$ be any set with the property

$$p(B_{\delta}^{(n)}) \geq 1 - \delta \quad (6.17)$$

$B_{\delta}^{(n)}$ could, say, contain the most likely sequences as well.
Other high probable sets?

- Let $B_\delta^{(n)}$ be any set with the property

$$p(B_\delta^{(n)}) \geq 1 - \delta$$

$B_\delta^{(n)}$ could, say, contain the most likely sequences as well.

**Theorem 6.4.2**

Let $X_1:n$ be an i.i.d. $\sim p(x)$ sequence. For $\delta < 1/2$ and any $\delta' > 0$, if $p(B_\delta^{(n)}) > 1 - \delta$, then

$$\frac{1}{n} \log |B_\delta^{(n)}| > H - \delta' \text{ if } n \text{ is large enough}$$

$$\Rightarrow |B_\delta^{(n)}| > 2^{n(H-\delta')} \approx 2^{nH}$$
Other high probable sets?

- Let $B_{\delta}^{(n)}$ be any set with the property

$$p(B_{\delta}^{(n)}) \geq 1 - \delta \tag{6.17}$$

$B_{\delta}^{(n)}$ could, say, contain the most likely sequences as well.

**Theorem 6.4.2**

Let $X_{1:n}$ be an i.i.d. $\sim p(x)$ sequence. For $\delta < 1/2$ and any $\delta' > 0,$ if

$$p(B_{\delta}^{(n)}) > 1 - \delta,$$ then

$$\frac{1}{n} \log |B_{\delta}^{(n)}| > H - \delta' \text{ if } n \text{ is large enough} \tag{6.18}$$

$$\Rightarrow |B_{\delta}^{(n)}| > 2^{n(H-\delta')} \approx 2^{nH} \tag{6.19}$$

- In other words, asymptotically $B_{\delta}^{(n)}$ is no smaller than $A_{\epsilon}^{(n)}$ and we are free to code for $A_{\epsilon}^{(n)}.$
Coding Strategy with errors

- Previous code was variable length, we had two lengths one for the typical set $A^{(n)}_{\epsilon}$ and one for the complement $A^{(n)\,c}_{\epsilon}$.
Coding Strategy with errors

- Previous code was variable length, we had two lengths one for the typical set $A_{\epsilon}^{(n)}$ and one for the complement $A_{\epsilon}^{(n)c}$.
- The code was guaranteed to have no errors!
Coding Strategy with errors

- Previous code was variable length, we had two lengths one for the typical set $A_{\epsilon}^{(n)}$ and one for the complement $A_{\epsilon}^{(n)c}$.
- The code was guaranteed to have no errors!
- Consider a variation of this code, to a fixed length code that might make errors.
Coding Strategy with errors

- Previous code was variable length, we had two lengths one for the typical set $A_{\epsilon}^{(n)}$ and one for the complement $A_{\epsilon}^{(n)c}$.
- The code was guaranteed to have no errors!
- Consider a variation of this code, to a fixed length code that might make errors.
- The typical sequences are coded using approximately $nH$ bits.
Coding Strategy with errors

- Previous code was variable length, we had two lengths one for the typical set $A_\epsilon(n)$ and one for the complement $A_\epsilon(n)^c$.
- The code was guaranteed to have no errors!
- Consider a variation of this code, to a fixed length code that might make errors.
- The typical sequences are coded using approximately $nH$ bits.
- The atypical sequences are arbitrarily mapped to one short codeword.
So, code is no longer one-to-one, and source sequence might map to same code word.
Coding Strategy with errors

- So, code is no longer one-to-one, and source sequence might map to same code word.
- What is $P_e =$ probability of error? We know

$$ p(A_{\epsilon}^{(n)}) > 1 - \epsilon $$  \hspace{0.5cm} (6.20)
So, code is no longer one-to-one, and source sequence might map to same code word.

What is $P_e = \text{probability of error}$? We know

$$p(A^{(n)}_\epsilon) > 1 - \epsilon$$

(6.20)

And error occurs when a sequence is not typical, so we can bound the error probability

$$p(\text{error}) = p(A^{(n)c}_\epsilon) \leq \epsilon$$

(6.21)
Coding Strategy with errors

- Recall typical: \( \forall \epsilon > 0, \forall \delta > 0, \exists n_0 \text{ s.t. for } n > n_0, \)

\[
\Pr \left\{ \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| < \epsilon \right\} > 1 - \delta \quad (6.22)
\]
Coding Strategy with errors

- Recall typical: \( \forall \epsilon > 0, \forall \delta > 0, \exists n_0 \) s.t. for \( n > n_0 \),

\[
\Pr \left\{ \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| < \epsilon \right\} > 1 - \delta \tag{6.22}
\]

- Which is same as: \( \forall \epsilon > 0, \forall \delta > 0, \exists n_0 \) s.t. for \( n > n_0 \),

\[
\Pr \left\{ \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| > \epsilon \right\} \leq \delta \tag{6.23}
\]
Coding Strategy with errors

- Recall typical: \( \forall \epsilon > 0, \forall \delta > 0, \exists n_0 \text{ s.t. for } n > n_0, \)
  \[
  \Pr \left\{ \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| < \epsilon \right\} > 1 - \delta \tag{6.22}
  \]

- Which is same as: \( \forall \epsilon > 0, \forall \delta > 0, \exists n_0 \text{ s.t. for } n > n_0, \)
  \[
  \delta(n, \epsilon) = \Pr \left\{ \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| > \epsilon \right\} \leq \delta \tag{6.23}
  \]

- We can think of this as a function \( \delta(n, \epsilon) \) with \( \lim_{n \to \infty} \delta(n, \epsilon) = 0 \) for all \( \epsilon > 0. \)
Recall typical: \( \forall \epsilon > 0, \forall \delta > 0, \exists n_0 \) s.t. for \( n > n_0 \),

\[
\Pr \left\{ \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| < \epsilon \right\} > 1 - \delta
\]

(6.22)

Which is same as: \( \forall \epsilon > 0, \forall \delta > 0, \exists n_0 \) s.t. for \( n > n_0 \),

\[
\Pr \left\{ \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| > \epsilon \right\} \leq \delta
\]

(6.23)

We can think of this as a function \( \delta(n, \epsilon) \) with \( \lim_{n \to \infty} \delta(n, \epsilon) = 0 \) for all \( \epsilon > 0 \).

Thus, we have \( \Pr(\text{error}) = \Pr(A_{\epsilon}^{(n) c}) \leq \delta(n, \epsilon) \), or

\[
\Pr(\text{error}) \to 0 \text{ as } n \to \infty
\]

(6.24)
Coding Strategy with errors

- Recall typical: \( \forall \epsilon > 0, \forall \delta > 0, \exists n_0 \text{ s.t. for } n > n_0, \)
  \[
  \Pr \left\{ \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| < \epsilon \right\} > 1 - \delta \quad (6.22)
  \]

- Which is same as: \( \forall \epsilon > 0, \forall \delta > 0, \exists n_0 \text{ s.t. for } n > n_0, \)
  \[
  \Pr \left\{ \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| > \epsilon \right\} \leq \delta \quad (6.23)
  \]

- We can think of this as a function \( \delta(n, \epsilon) \) with \( \lim_{n \to \infty} \delta(n, \epsilon) = 0 \) for all \( \epsilon > 0. \)

- Thus, we have \( \Pr(\text{error}) = \Pr(A^{(n)}_\epsilon) \leq \delta(n, \epsilon), \) or
  \[
  \Pr(\text{error}) \to 0 \text{ as } n \to \infty \quad (6.24)
  \]

- So, regardless of if we use a long codeword (and never have an error), or have errors, expected length is the same and the error probability goes to zero if we code the typical set.
Coding with fewer than $H$ bits, converse intuition

- Theorem says coding is error free if we use $n(H + \epsilon)$ bits per code word to code, for any $\epsilon > 0$. What if we use fewer?
Coding with fewer than $H$ bits, converse intuition

- Theorem says coding is error free if we use $n(H + \epsilon)$ bits per code word to code, for any $\epsilon > 0$. What if we use fewer?
- I.e., use $n(H - \alpha\epsilon)$ bits to code, with $\alpha > 1$. Thus, we have at most $2^{n(H-\alpha\epsilon)}$ code words. (6.25)
Coding with fewer than $H$ bits, converse intuition

- Theorem says coding is error free if we use $n(H + \epsilon)$ bits per code word to code, for any $\epsilon > 0$. What if we use fewer?
- I.e., use $n(H - \alpha \epsilon)$ bits to code, with $\alpha > 1$. Thus, we have at most
  
  $$2^{n(H-\alpha\epsilon)}$$  

  code words.

- $2^{n(H-\alpha\epsilon)}$ is the maximum number of code words.
Coding with fewer than $H$ bits, converse intuition

- Theorem says coding is error free if we use $n(H + \epsilon)$ bits per code word to code, for any $\epsilon > 0$. What if we use fewer?
- I.e., use $n(H - \alpha\epsilon)$ bits to code, with $\alpha > 1$. Thus, we have at most
  \[2^{n(H-\alpha\epsilon)}\]
  \[(6.25)\]
code words.
- $2^{n(H-\alpha\epsilon)}$ is the maximum number of code words.
- $2^{-n(H-\epsilon)}$ is the upper bound on the probability of a typical sequence...
Coding with fewer than $H$ bits, converse intuition

- Theorem says coding is error free if we use $n(H + \epsilon)$ bits per code word to code, for any $\epsilon > 0$. What if we use fewer?
- I.e., use $n(H - \alpha \epsilon)$ bits to code, with $\alpha > 1$. Thus, we have at most
  
  $$2^{n(H-\alpha \epsilon)}$$

  (6.25)

  code words.

- $2^{n(H-\alpha \epsilon)}$ is the maximum number of code words.
- $2^{-n(H-\epsilon)}$ is the upper bound on the probability of a typical sequence

The probability of sequences for which we can provide code words is no more than the product of the two, i.e.,

$$2^{n(H-\alpha \epsilon)} 2^{-n(H-\epsilon)} = 2^{-n\epsilon(\alpha-1)}$$

(6.26)
Coding with fewer than $H$ bits, converse intuition

- Theorem says coding is error free if we use $n(H + \epsilon)$ bits per code word to code, for any $\epsilon > 0$. What if we use fewer?
- I.e., use $n(H - \alpha \epsilon)$ bits to code, with $\alpha > 1$. Thus, we have at most

$$2^{n(H - \alpha \epsilon)}$$

(6.25)

code words.
- $2^{n(H - \alpha \epsilon)}$ is the maximum number of code words.
- $2^{-n(H - \epsilon)}$ is the upper bound on the probability of a typical sequence
- The probability of sequences for which we can provide code words is no more than the product of the two, i.e.,

$$2^{n(H - \alpha \epsilon)} 2^{-n(H - \epsilon)} = 2^{-n\epsilon(\alpha - 1)}$$

(6.26)
- For any $\alpha > 1$, this probability $\to 0$ as $n \to \infty$. Problem: probability of typicality shrinks exponentially faster than growth of number of code words, with $n$. 
Coding with fewer than $H$ bits, converse intuition

- Theorem says coding is error free if we use $n(H + \epsilon)$ bits per code word to code, for any $\epsilon > 0$. What if we use fewer?
- I.e., use $n(H - \alpha \epsilon)$ bits to code, with $\alpha > 1$. Thus, we have at most
  \[ 2^{n(H - \alpha \epsilon)} \]  
  code words.
- $2^{n(H - \alpha \epsilon)}$ is the maximum number of code words.
- $2^{-n(H - \epsilon)}$ is the upper bound on the probability of a typical sequence
- The probability of sequences for which we can provide code words is no more than the product of the two, i.e.,
  \[ 2^{n(H - \alpha \epsilon)} 2^{-n(H - \epsilon)} = 2^{-n\epsilon(\alpha - 1)} \]  
  (6.26)
- For any $\alpha > 1$, this probability $\to 0$ as $n \to \infty$. Problem: probability of typicality shrinks exponentially faster than growth of number of code words, with $n$.
- Thus, the error goes to 1 as $n \to \infty$.  

Prof. Jeff Bilmes
Shannon’s source coding theorem, intuitively

Given $n$ r.v.s each with entropy $H$ can be compressed into more than $nH$ bits with negligible risk of information loss, as $n \to \infty$. 
Shannon’s source coding theorem, intuitively

- Given $n$ r.v.s each with entropy $H$ can be compressed into more than $nH$ bits with negligible risk of information loss, as $n \to \infty$.
- Conversely, if the r.v.s are compressed into fewer than $nH$ bits, then it is virtually certain that information will be lost and errors will occur.
There exists a code that can achieve a compression rate of $H(X) + \epsilon'$ bits per source symbol for any $\epsilon' > 0$ as long as the block length $n$ (the length of source symbols that
Typical Set Source Coding/Compression (summary)

- There exists a code that can achieve a compression rate of $H(X) + \epsilon'$ bits per source symbol for any $\epsilon' > 0$ as long as the block length $n$ (the length of source symbols that
- Even though the source sequences are assumed i.i.d., it is necessary to encode them jointly in order to achieve this rate.
There exists a code that can achieve a compression rate of $H(X) + \epsilon'$ bits per source symbol for any $\epsilon' > 0$ as long as the block length $n$ (the length of source symbols that

even though the source sequences are assumed i.i.d., it is necessary to encode them jointly in order to achieve this rate.

We do this by using $n(H + \epsilon) + 2$ bits for every typical sequence, and we use $n \log K + 2$ bits for every atypical sequence. This code is 1-1 and guaranteed zero-error.
Typical Set Source Coding/Compression (summary)

- There exists a code that can achieve a compression rate of $H(X) + \epsilon'$ bits per source symbol for any $\epsilon' > 0$ as long as the block length $n$ (the length of source symbols that
- Even though the source sequences are assumed i.i.d., it is necessary to encode them jointly in order to achieve this rate.
- We do this by using $n(H + \epsilon) + 2$ bits for every typical sequence, and we use $n \log K + 2$ bits for every atypical sequence. This code is 1-1 and guaranteed zero-error.
- Alternatively, we can design for there to be an error whenever we receive an atypical source sequence. This code has a probability of error $P_e$ that is not zero, but $\rightarrow 0$ exponentially fast if $\epsilon' > 0$. 

In either case, the expected length is $E[1/n(X_1:n)] \leq H(X) + \epsilon' (6.27)$
Typical Set Source Coding/Compression (summary)

- There exists a code that can achieve a compression rate of $H(X) + \epsilon'$ bits per source symbol for any $\epsilon' > 0$ as long as the block length $n$ (the length of source symbols that
- Even though the source sequences are assumed i.i.d., it is necessary to encode them jointly in order to achieve this rate.
- We do this by using $n(H + \epsilon) + 2$ bits for every typical sequence, and we use $n \log K + 2$ bits for every atypical sequence. This code is 1-1 and guaranteed zero-error.
- Alternatively, we can design for there to be an error whenever we receive an atypical source sequence. This code has a probability of error $P_e$ that is not zero, but $\to 0$ exponentially fast if $\epsilon' > 0$.
- In either case, the expected length is

$$E[\frac{1}{n}\ell(X_1:n)] \leq H(X) + \epsilon \quad (6.27)$$
Typical Set Source Coding/Compression (summary)

- There exists a code that can achieve a compression rate of $H(X) + \epsilon'$ bits per source symbol for any $\epsilon' > 0$ as long as the block length $n$ (the length of source symbols that
- Even though the source sequences are assumed i.i.d., it is necessary to encode them jointly in order to achieve this rate.
- We do this by using $n(H + \epsilon) + 2$ bits for every typical sequence, and we use $n \log K + 2$ bits for every atypical sequence. This code is 1-1 and guaranteed zero-error.
- Alternatively, we can design for there to be an error whenever we receive an atypical source sequence. This code has a probability of error $P_e$ that is not zero, but $\to 0$ exponentially fast if $\epsilon' > 0$.
- In either case, the expected length is

$$E\left[\frac{1}{n} \ell(X_{1:n})\right] \leq H(X) + \epsilon$$  \hspace{1cm} (6.27)

- In the second case, however, $P_e \to 0$ as $n \to \infty$
Binomial Distribution, when $n$ gets big, $p = 0.5$

$$\Pr(S_n = k) = \binom{n}{k} p^k q^{n-k}, \quad S_n = X_1 + X_2 + \cdots + X_n, \quad X_i \sim \text{Bernoulli}(p)$$

- What happens when $n$ gets big?
- Plot the probability of the normalized values, $S_n/n = k/n$, and see how the distribution changes when $n$ gets large.
Typical sets and $p = 0.5$

- So, while all sequences are typical (they all have the same probability), the ones with $k = n/2$ ones eventually are all that happens (they have all the probability)
Typical sets and $p = 0.5$

- So, while all sequences are typical (they all have the same probability), the ones with $k = n/2$ ones eventually are all that happens (they have all the probability).
- A good way to explain this is with the method of types.
Overview: Method of types

- a refinement of the typical sequence approach (at least for discrete memory-less systems).
Overview: Method of types

- a refinement of the typical sequence approach (at least for discrete memory-less systems).

- Idea: $X_1, X_2, \ldots, X_n$ i.i.d. $\sim p(x)$, we partition the sequences into classes according to the sequences empirical distribution (histogram), i.e., the sequences type.

- Number of type classes grows sub-exponentially with $n$.

- Sequences of the same type are equiprobable.

- Number of sequences of a certain type class grows exponentially.

- Intersection of error events and type class events allows good bounds on the errors.

- We get Shannon's source coding theorem (and converse) in a formal but intuitive way.
Overview: Method of types

- a refinement of the typical sequence approach (at least for discrete memory-less systems).

- Idea: $X_1, X_2, \ldots, X_n$ i.i.d. $\sim p(x)$, we partition the sequences into classes according to the sequences empirical distribution (histogram), i.e., the sequences type.

- A “type” class is really a class (or set) of sequences of length $n$ sharing the same empirical histogram.
Overview: Method of types

- a refinement of the typical sequence approach (at least for discrete memory-less systems).
- Idea: $X_1, X_2, \ldots, X_n$ i.i.d. $\sim p(x)$, we partition the sequences into classes according to the sequences empirical distribution (histogram), i.e., the sequences type.
- A “type” class is really a class (or set) of sequences of length $n$ sharing the same empirical histogram.
- Number of type classes grows sub-exponentially with $n$
Overview: Method of types

- a refinement of the typical sequence approach (at least for discrete memory-less systems).
- Idea: $X_1, X_2, \ldots, X_n$ i.i.d. $\sim p(x)$, we partition the sequences into classes according to the sequences empirical distribution (histogram), i.e., the sequences type.
- A “type” class is really a class (or set) of sequences of length $n$ sharing the same empirical histogram.
- Number of type classes grows sub-exponentially with $n$
- Sequences of the same type are equiprobable.
Overview: Method of types

- a refinement of the typical sequence approach (at least for discrete memory-less systems).
- Idea: \( X_1, X_2, \ldots, X_n \) i.i.d. \( \sim p(x) \), we partition the sequences into classes according to the sequences empirical distribution (histogram), i.e., the sequences type.
- A “type” class is really a class (or set) of sequences of length \( n \) sharing the same empirical histogram.
- Number of type classes grows sub-exponentially with \( n \)
- Sequences of the same type are equiprobable.
- Number of sequences of a certain type class grows exponentially.
Overview: Method of types

- A refinement of the typical sequence approach (at least for discrete memory-less systems).
- Idea: $X_1, X_2, \ldots, X_n$ i.i.d. $\sim p(x)$, we partition the sequences into classes according to the sequences empirical distribution (histogram), i.e., the sequences type.
- A “type” class is really a class (or set) of sequences of length $n$ sharing the same empirical histogram.
- Number of type classes grows sub-exponentially with $n$
- Sequences of the same type are equiprobable.
- Number of sequences of a certain type class grows exponentially.
- Intersection of error events and type class events allows good bounds on the errors.
Overview: Method of types

- A refinement of the typical sequence approach (at least for discrete memory-less systems).
- Idea: \( X_1, X_2, \ldots, X_n \) i.i.d. \( \sim p(x) \), we partition the sequences into classes according to the sequences empirical distribution (histogram), i.e., the sequences type.
- A “type” class is really a class (or set) of sequences of length \( n \) sharing the same empirical histogram.
- Number of type classes grows sub-exponentially with \( n \)
- Sequences of the same type are equiprobable.
- Number of sequences of a certain type class grows exponentially.
- Intersection of error events and type class events allows good bounds on the errors.
- We get Shannon’s source coding theorem (and converse) in a formal but intuitive way.
Definition: the “type” of the sample

Let $X_1, X_2, \ldots, X_n \equiv X_{1:n}$ be a length-$n$ sample of a D-ary discrete random variable. So $x_i \in \mathcal{X}$ and alphabet size $D = |\mathcal{X}|$, and $\mathcal{X} = (a_1, a_2, \ldots, a_D)$. 

Define a statistic which is the empirical histogram of this sample. 

$$P_{x_1:n} = \frac{n(a_1 | x_1:n)}{n}, \frac{n(a_2 | x_1:n)}{n}, \ldots, \frac{n(a_D | x_1:n)}{n} \quad (6.28)$$

where $n(a_i | x_1:n)$ counts occurrence of symbol $a_i$ in sample $x_1:n$. 

Thus, $P_{x_1:n}$ is a probability mass function. $P_{x_1:n}$ is a histogram, or type, of the sample. 

$$P_{x_1:n}(a_i) = \frac{n(a_i | x_1:n)}{n}$$ for $a_i \in \mathcal{X}$. 

Prof. Jeff Bilmes
Definition: the “type” of the sample

Let $X_1, X_2, \ldots, X_n \equiv X_{1:n}$ be a length-$n$ sample of a $D$-ary discrete random variable. So $x_i \in \mathcal{X}$ and alphabet size $D = |\mathcal{X}|$, and $\mathcal{X} = (a_1, a_2, \ldots, a_D)$.

Define a statistic which is the empirical histogram of this sample.

$$P_{x_{1:n}} \triangleq \left( \frac{n(a_1|x_{1:n})}{n}, \frac{n(a_2|x_{1:n})}{n}, \ldots, \frac{n(a_D|x_{1:n})}{n} \right)$$ (6.28)

where $n(a_i|x_{1:n})$ counts occurrence of symbol $a_i$ in sample $x_{1:n}$. 

Definition: the “type” of the sample

- Let $X_1, X_2, \ldots, X_n \equiv X_{1:n}$ be a length-$n$ sample of a $D$-ary discrete random variable. So $x_i \in \mathcal{X}$ and alphabet size $D = |\mathcal{X}|$, and $\mathcal{X} = (a_1, a_2, \ldots, a_D)$.

- Define a statistic which is the empirical histogram of this sample.

$$P_{x_{1:n}} \triangleq \left( \frac{n(a_1|x_{1:n})}{n}, \frac{n(a_2|x_{1:n})}{n}, \ldots, \frac{n(a_D|x_{1:n})}{n} \right)$$ (6.28)

where $n(a_i|x_{1:n})$ counts occurrence of symbol $a_i$ in sample $x_{1:n}$.

- Thus, $P_{x_{1:n}}$ is a probability mass function.
Definition: the “type” of the sample

- Let \( X_1, X_2, \ldots, X_n \equiv X_{1:n} \) be a length-\( n \) sample of a \( D \)-ary discrete random variable. So \( x_i \in \mathcal{X} \) and alphabet size \( D = |\mathcal{X}| \), and \( \mathcal{X} = (a_1, a_2, \ldots, a_D) \).

- Define a statistic which is the empirical histogram of this sample.

\[
P_{x_{1:n}} \triangleq \left( \frac{n(a_1|x_{1:n})}{n}, \frac{n(a_2|x_{1:n})}{n}, \ldots, \frac{n(a_D|x_{1:n})}{n} \right) \tag{6.28}
\]

where \( n(a_i|x_{1:n}) \) counts occurrence of symbol \( a_i \) in sample \( x_{1:n} \).

- Thus, \( P_{x_{1:n}} \) is a probability mass function.

- \( P_{x_{1:n}} \) is a histogram, or type, of the sample.
Definition: the “type” of the sample

Let $X_1, X_2, \ldots, X_n \equiv X_{1:n}$ be a length-$n$ sample of a D-ary discrete random variable. So $x_i \in \mathcal{X}$ and alphabet size $D = |\mathcal{X}|$, and $\mathcal{X} = (a_1, a_2, \ldots, a_D)$.

Define a statistic which is the empirical histogram of this sample.

\[
P_{x_{1:n}} \triangleq \left( \frac{n(a_1|x_{1:n})}{n}, \frac{n(a_2|x_{1:n})}{n}, \ldots, \frac{n(a_D|x_{1:n})}{n} \right) \quad (6.28)
\]

where $n(a_i|x_{1:n})$ counts occurrence of symbol $a_i$ in sample $x_{1:n}$.

- Thus, $P_{x_{1:n}}$ is a probability mass function.
- $P_{x_{1:n}}$ is a histogram, or type, of the sample.
- $P_{x_{1:n}}(a) = \frac{n(a|x_{1:n})}{n}$ for $a \in \mathcal{X}$. 
Set of types

- Define $\mathcal{P}_n$ be the set of all possible types with denominator $n$. 

\[ \mathcal{P}_n = \{ P_n(x) : x \in \{0, 1\} \text{ for } X = \{0, 1\} \} \]

and there are a total of $n+1$ possible types (histograms) in this case.
Set of types

- Define $\mathcal{P}_n$ be the set of all possible types with denominator $n$.
- I.e., set of all possible histograms of sample of length $n$ for r.v. on domain $\mathcal{X}$.
Set of types

- Define \( \mathcal{P}_n \) be the set of all possible types with denominator \( n \)
- I.e., set of all possible histograms of sample of length \( n \) for r.v. on domain \( \mathcal{X} \).
- I.e., \( \mathcal{P}_n(\equiv \mathcal{P}_n(\mathcal{X}) \equiv \mathcal{P}_n(|\mathcal{X}|)) \)
Set of types

- Define $\mathcal{P}_n$ be the set of all possible types with denominator $n$
- I.e., set of all possible histograms of sample of length $n$ for r.v. on domain $\mathcal{X}$.
- I.e., $\mathcal{P}_n(\mathcal{X}) \equiv \mathcal{P}_n(|\mathcal{X}|)$ is set of types that could arise from sequences of length $n$ using symbols in alphabet $\mathcal{X}$. 

Example, $\mathcal{X} = \{0, 1\}$, then $\mathcal{P}_n(\mathcal{X}) = \{\theta^n_0, n^n_0, \theta^n_1, n^n_1, \ldots, \theta^n_0, n^n_1\}$. And there are a total of $n+1$ possible types (histograms) in this case. Note, $\mathcal{P}_n$ is a set of ordered lists. As usual curly braces {} designate sets, while parentheses () designated ordered lists.
Set of types

- Define $\mathcal{P}_n$ be the set of all possible types with denominator $n$
- I.e., set of all possible histograms of sample of length $n$ for r.v. on domain $\mathcal{X}$.
- I.e., $\mathcal{P}_n(\equiv \mathcal{P}_n(\mathcal{X}) \equiv \mathcal{P}_n(|\mathcal{X}|))$ is set of types that could arise from sequences of length $n$ using symbols in alphabet $\mathcal{X}$.
- E.x., $\mathcal{X} = \{0, 1\}$, then

$$\mathcal{P}_n(\mathcal{X}) = \left\{ \left( \frac{0}{n}, \frac{n}{n} \right), \left( \frac{1}{n}, \frac{n-1}{n} \right), \ldots, \left( \frac{n}{n}, \frac{0}{n} \right) \right\} \quad (6.29)$$

and there are a total of $n + 1$ possible types (histograms) in this case.
Set of types

- Define $\mathcal{P}_n$ be the set of all possible types with denominator $n$
- I.e., set of all possible histograms of sample of length $n$ for r.v. on domain $\mathcal{X}$.
- I.e., $\mathcal{P}_n(\equiv \mathcal{P}_n(\mathcal{X}) \equiv \mathcal{P}_n(|\mathcal{X}|))$ is set of types that could arise from sequences of length $n$ using symbols in alphabet $\mathcal{X}$.
- E.x., $\mathcal{X} = \{0, 1\}$, then

$$
\mathcal{P}_n(\mathcal{X}) = \left\{ \left( \frac{0}{n}, \frac{n}{n} \right), \left( \frac{1}{n}, \frac{n - 1}{n} \right), \ldots, \left( \frac{n}{n}, \frac{0}{n} \right) \right\}
$$

(6.29)

and there are a total of $n + 1$ possible types (histograms) in this case.

- Note, $\mathcal{P}_n$ is a set of ordered lists. As usual curly braces $\{\}$ designate sets, while parentheses $(\cdot)$ designated ordered lists.
Type class

For a given $P \in \mathcal{P}_n$, the set of length-$n$ sequences of type $P$ constitute what is called the type class of $P$. 

This is designated $T(P)$, i.e.,

$$T(P) = \{x_1 \ldots x_n : P_{x_1} = P\}$$ (6.30)
For a given $P \in \mathcal{P}_n$, the set of length-$n$ sequences of type $P$ constitute what is called the type class of $P$.

This is designated $T(P)$. I.e.,

$$
T(P) \triangleq \{ x_{1:n} \in \mathcal{X}^n : P_{x_{1:n}} = P \} 
$$

(6.30)

which is the set of all sequences of length $n$ having a certain histogram $P$. 
Notational Summary

- For sequences of length $n$, we have:
  1. the type (or histogram) of a sample $x_{1:n}$

$$P_{x_{1:n}} \triangleq \left( \frac{n(a_1|x_{1:n})}{n}, \frac{n(a_2|x_{1:n})}{n}, \ldots, \frac{n(a_D|x_{1:n})}{n} \right); \quad (6.31)$$
For sequences of length $n$, we have:

1) the type (or histogram) of a sample $x_{1:n}$

$$P_{x_{1:n}} \triangleq \left( \frac{n(a_1|x_{1:n})}{n}, \frac{n(a_2|x_{1:n})}{n}, \ldots, \frac{n(a_D|x_{1:n})}{n} \right);$$  \hspace{1cm} (6.31)

2) the set of all types (or histograms) $\mathcal{P}_n$;
Notational Summary

- For sequences of length $n$, we have:
  1) the type (or histogram) of a sample $x_{1:n}$

\[
P_{x_{1:n}} \triangleq \left( \frac{n(a_1|x_{1:n})}{n}, \frac{n(a_2|x_{1:n})}{n}, \ldots, \frac{n(a_D|x_{1:n})}{n} \right); \tag{6.31}
\]

  2) the set of all types (or histograms) $\mathcal{P}_n$;
  3) Some particular type $P \in \mathcal{P}_n$. 

For sequences of length $n$, we have:

1) the type (or histogram) of a sample $x_{1:n}$

$$P_{x_{1:n}} \triangleq \left( \frac{n(a_1|x_{1:n})}{n}, \frac{n(a_2|x_{1:n})}{n}, \ldots, \frac{n(a_D|x_{1:n})}{n} \right); \quad (6.31)$$

2) the set of all types (or histograms) $\mathcal{P}_n$;

3) Some particular type $P \in \mathcal{P}_n$.

4) Type class: For a given type $P$, the set of all sequences with that type $T(P) = \{x_{1:n} \in \mathcal{X}^n : P_{x_{1:n}} = P\}$. 
Example

Let $\mathcal{X} = \{1, 2, 3\}$ and $x_{1:5} = [1, 1, 3, 2, 1]$. Then $P_{x_{1:5}} = (3, 5, 1, 5, 1)$. And $T(P_{x_{1:5}})$ is the set of sequences of length 5 with three 1's, one 2, and one 3. I.e., $T(P_{x_{1:5}}) = \{[1, 1, 1, 2, 3], [1, 1, 1, 3, 2], \ldots, [3, 2, 1, 1, 1]\}$. How many types? I.e., What is $|P_n|$? Here, $|P_n| = 21$. Problem for you to think about. Turns out, in general, $|P_n| = \sqrt{n} + |\mathcal{X}|e - |\mathcal{X}|$. ◆
Example

- Let $\mathcal{X} = \{1, 2, 3\}$ and $x_{1:5} = [1, 1, 3, 2, 1]$.
- Then

$$P_{x_{1:5}} = \left(\frac{3}{5}, \frac{1}{5}, \frac{1}{5}\right) \hspace{1cm} (6.32)$$
Example

- Let $\mathcal{X} = \{1, 2, 3\}$ and $x_{1:5} = [1, 1, 3, 2, 1]$.
- Then
  \[
P_{x_{1:5}} = \left( \frac{3}{5}, \frac{1}{5}, \frac{1}{5} \right)
  \]  \hspace{1cm} (6.32)

- And $T(P_{x_{1:5}})$ is the set of sequences of length 5 with three 1’s, one 2, and one 3. I.e.,
  \[
  T(P_{x_{1:5}}) = \{[1, 1, 1, 2, 3], [1, 1, 1, 3, 2], \ldots, [3, 2, 1, 1, 1]\}
  \]  \hspace{1cm} (6.33)
Example

- Let $\mathcal{X} = \{1, 2, 3\}$ and $x_{1:5} = [1, 1, 3, 2, 1]$.
- Then
  \[
  P_{x_{1:5}} = \left(\frac{3}{5}, \frac{1}{5}, \frac{1}{5}\right)
  \]  
  \hspace{2cm} (6.32)

- And $T(P_{x_{1:5}})$ is the set of sequences of length 5 with three 1’s, one 2, and one 3. I.e.,
  \[
  T(P_{x_{1:5}}) = \{[1, 1, 1, 2, 3], [1, 1, 1, 3, 2], \ldots, [3, 2, 1, 1, 1]\}
  \]  
  \hspace{2cm} (6.33)

- How many types? I.e., What is $|\mathcal{P}_n|$?
Example

Let $\mathcal{X} = \{1, 2, 3\}$ and $x_{1:5} = [1, 1, 3, 2, 1]$.

Then

$$P_{x_{1:5}} = \left(\frac{3}{5}, \frac{1}{5}, \frac{1}{5}\right)$$ (6.32)

And $T(P_{x_{1:5}})$ is the set of sequences of length 5 with three 1’s, one 2, and one 3. I.e.,

$$T(P_{x_{1:5}}) = \{[1, 1, 1, 2, 3], [1, 1, 1, 3, 2], \ldots, [3, 2, 1, 1, 1]\}$$ (6.33)

How many types? I.e., What is $|\mathcal{P}_n|$? Here, $|\mathcal{P}_n| = 21$. 
Example

- Let $\mathcal{X} = \{1, 2, 3\}$ and $x_{1:5} = [1, 1, 3, 2, 1]$.
- Then
  \[ P_{x_{1:5}} = \left( \frac{3}{5}, \frac{1}{5}, \frac{1}{5} \right) \]  
  (6.32)
- And $T(P_{x_{1:5}})$ is the set of sequences of length 5 with three 1’s, one 2, and one 3. I.e.,
  \[ T(P_{x_{1:5}}) = \{ [1, 1, 1, 2, 3], [1, 1, 1, 3, 2], \ldots, [3, 2, 1, 1, 1] \} \]  
  (6.33)
- How may types? I.e., What is $|\mathcal{P}_n|$? Here, $|\mathcal{P}_n| = 21$.
- Problem for you to think about. Turns out, in general,
  \[ |\mathcal{P}_n| = \binom{n + |\mathcal{X}| - 1}{|\mathcal{X}| - 1} \]  
  (6.34)
Division of set of all sequences into type classes

\[ \mathcal{P}_n = \{ P_1, P_2, \ldots, P_{|\mathcal{P}_n|} \} \] is the set of all types,
Division of set of all sequences into type classes

- $\mathcal{P}_n = \{P_1, P_2, \ldots, P_{|\mathcal{P}_n|}\}$ is the set of all types,
- Thus, $\bigcup_{P \in \mathcal{P}_n} T(P) = \mathcal{X}^n$. 
Division of set of all sequences into type classes

- $\mathcal{P}_n = \{ P_1, P_2, \ldots, P_{|\mathcal{P}_n|} \}$ is the set of all types,
- Thus, $\bigcup_{P \in \mathcal{P}_n} T(P) = \mathcal{X}^n$.
- The space of all sequences.

$\mathcal{X}^n$: the set of all sequences of length $n$
Division of set of all sequences into type classes

- $\mathcal{P}_n = \{P_1, P_2, \ldots, P_{|\mathcal{P}_n|}\}$ is the set of all types,
- Thus, $\bigcup_{P \in \mathcal{P}_n} T(P) = \mathcal{X}^n$.
- The space of all sequences.

$\mathcal{X}^n$: partitioned into blocks within which all sequences have the same type
Proposition 6.5.1

\[ |P_n| \leq (n + 1)|X| \] (6.35)
Bound on number of type classes

Proposition 6.5.1

\[ |\mathcal{P}_n| \leq (n + 1)|X| \]  \hfill (6.35)

Proof.

- Note that numerator of each entry of a type may take on at most \((n + 1)\) possible values,
Proposition 6.5.1

\[ |\mathcal{P}_n| \leq (n + 1)|\mathcal{X}| \]  

(6.35)

Proof.

- Note that numerator of each entry of a type may take on at most \((n + 1)\) possible values,
- And there are \(|\mathcal{X}|\) numerators.
Bound on number of type classes

**Proposition 6.5.1**

\[ |\mathcal{P}_n| \leq (n + 1)|\mathcal{X}| \quad (6.35) \]

**Proof.**

- Note that numerator of each entry of a type may take on at most \((n + 1)\) possible values,
- And there are \(|\mathcal{X}|\) numerators.
- The numerator values interact (they must sum to \(n\)) but we can upper bound, pretending no interaction, leading to the upper bound.
Bound on number of type classes

Proposition 6.5.1

\[ |P_n| \leq (n + 1)|X| \] (6.35)

Proof.

- Note that numerator of each entry of a type may take on at most \((n + 1)\) possible values,
- And there are \(|X|\) numerators.
- The numerator values interact (they must sum to \(n\)) but we can upper bound, pretending no interaction, leading to the upper bound.
Bound on number of type classes

**Proposition 6.5.1**

\[ |\mathcal{P}_n| \leq (n + 1)^{|\mathcal{X}|} \]  

**Proof.**

- Note that numerator of each entry of a type may take on at most \((n + 1)\) possible values,
- And there are \(|\mathcal{X}|\) numerators.
- The numerator values interact (they must sum to \(n\)) but we can upper bound, pretending no interaction, leading to the upper bound.

- Key point: there are, thus, only a polynomial in \(n\) number of types of sequences of length \(n\).
Bound on number of type classes

**Proposition 6.5.1**

\[ |\mathcal{P}_n| \leq (n + 1)^{|\mathcal{X}|} \]  

(6.35)

**Proof.**

- Note that numerator of each entry of a type may take on at most \((n + 1)\) possible values,

- And there are \(|\mathcal{X}|\) numerators.

- The numerator values interact (they must sum to \(n\)) but we can upper bound, pretending no interaction, leading to the upper bound.

- Key point: there are, thus, only a **polynomial** in \(n\) number of types of sequences of length \(n\).

- However, \(\exists\) an exponential number of sequences of length \(n\), \(|\mathcal{X}|^n\).
Another summary thus far (don’t say we didn’t remind you)

- \( P_{x_1:n} \) is the type (empirical distribution) of the sequence \( x_{1:n} \), with 
  \[ P_{x_1:n}(a) = \frac{N(a|x_{1:n})}{n} \text{ for all } a \in \mathcal{X}. \]
Another summary thus far (don’t say we didn’t remind you)

- $P_{x_1:n}$ is the type (empirical distribution) of the sequence $x_{1:n}$, with $P_{x_1:n}(a) = \frac{N(a|x_{1:n})}{n}$ for all $a \in \mathcal{X}$.
- $\mathcal{P}_n(\mathcal{X})$ (or just $\mathcal{P}_n$) is the set of types with denominator $n$. 
Another summary thus far (don’t say we didn’t remind you)

- $P_{x_1:n}$ is the type (empirical distribution) of the sequence $x_{1:n}$, with $P_{x_1:n}(a) = N(a|x_{1:n})/n$ for all $a \in \mathcal{X}$.
- $\mathcal{P}_n(\mathcal{X})$ (or just $\mathcal{P}_n$) is the set of types with denominator $n$.
- $T(P)$, for type $P \in \mathcal{P}_n$, is the set of all sequences with type $P$, i.e., $T(P) = \{x_{1:n} \in \mathcal{X}^n : P_{x_1:n} = P\}$. 

Theorem: Number of types bounded by $\text{poly in } n$, $|\mathcal{P}_n| \leq (n+1)|\mathcal{X}|$.

Fact: number of sequences of length $n$ is exponential in $n$, $|\mathcal{X}|^n$. 

Prof. Jeff Bilmes
Another summary thus far (don’t say we didn’t remind you)

- $P_{x_1:n}$ is the type (empirical distribution) of the sequence $x_1:n$, with $P_{x_1:n}(a) = N(a|x_1:n)/n$ for all $a \in \mathcal{X}$.
- $\mathcal{P}_n(\mathcal{X})$ (or just $\mathcal{P}_n$) is the set of types with denominator $n$.
- $T(P)$, for type $P \in \mathcal{P}_n$, is the set of all sequences with type $P$, i.e., $T(P) = \{x_1:n \in \mathcal{X}^n : P_{x_1:n} = P\}$.
- Theorem: Number of types bounded by poly in $n$, $|\mathcal{P}_n| \leq (n + 1)|\mathcal{X}|$. 
Another summary thus far (don’t say we didn’t remind you)

- \( P_{x_1:n} \) is the type (empirical distribution) of the sequence \( x_{1:n} \), with \( P_{x_1:n}(a) = N(a|x_{1:n})/n \) for all \( a \in \mathcal{X} \).
- \( \mathcal{P}_n(\mathcal{X}) \) (or just \( \mathcal{P}_n \)) is the set of types with denominator \( n \).
- \( T(P) \), for type \( P \in \mathcal{P}_n \), is the set of all sequences with type \( P \), i.e., \( T(P) = \{ x_{1:n} \in \mathcal{X}^n : P_{x_{1:n}} = P \} \).
- Theorem: Number of types bounded by poly in \( n \), \(|\mathcal{P}_n| \leq (n + 1)|\mathcal{X}|\).
- Fact: number of sequences of length \( n \) is exponential in \( n \), \(|\mathcal{X}|^n \).
Probability depends only on the type

**Theorem 6.5.2**

- Let $X_1, X_2, \ldots, X_n$ be i.i.d. $\sim Q(x)$,
Probability depends only on the type

Theorem 6.5.2

1. Let $X_1, X_2, \ldots, X_n$ be i.i.d. $\sim Q(x)$,
2. with extension $Q^n(x_{1:n}) = \prod_i Q(x_i)$, with $Q$ otherwise arbitrary.

So, probability doesn't depend on the sequence, once we are given the type.

Compare with sufficient statistics: all sequences with the same type have the same probability.
Theorem 6.5.2

- Let \( X_1, X_2, \ldots, X_n \) be i.i.d. \( \sim Q(x) \),
- with extension \( Q^n(x_1:n) = \prod_i Q(x_i) \), with \( Q \) otherwise arbitrary.
- **The probability of the sequence depends only on the type**

So, probability doesn’t depend on the sequence, once we are given the type.

Compare with sufficient statistics: all sequences with the same type have the same probability.
Probability depends only on the type

Theorem 6.5.2

Let $X_1, X_2, \ldots, X_n$ be i.i.d. $\sim Q(x)$,

with extension $Q^n(x_{1:n}) = \prod_i Q(x_i)$, with $Q$ otherwise arbitrary.

The probability of the sequence depends only on the type

restated, the probability is “independent” of the sequence given the type and $Q$
Probability depends only on the type

**Theorem 6.5.2**

- Let $X_1, X_2, \ldots, X_n$ be i.i.d. $\sim Q(x)$,
- with extension $Q^n(x_1:n) = \prod_i Q(x_i)$, with $Q$ otherwise arbitrary.
- The probability of the sequence depends only on the type
- restated, the probability is “independent” of the sequence given the type and $Q$
- That is

$$Q^n(x_1:n) = 2^{-n[H(P_{x_1:n})+D(P_{x_1:n}||Q)]} \quad (6.36)$$
Theorem 6.5.2

Let $X_1, X_2, \ldots, X_n$ be i.i.d. $\sim Q(x)$,

with extension $Q^n(x_{1:n}) = \prod_i Q(x_i)$, with $Q$ otherwise arbitrary.

The probability of the sequence depends only on the type

restated, the probability is “independent” of the sequence given the type and $Q$

That is

$$Q^n(x_{1:n}) = 2^{-n[H(P_{x_{1:n}}) + D(P_{x_{1:n}} || Q)]} \quad (6.36)$$
Probability depends only on the type

Theorem 6.5.2

- Let \( X_1, X_2, \ldots, X_n \) be i.i.d. \( \sim Q(x) \),
- with extension \( Q^n(x_1:n) = \prod_i Q(x_i) \), with \( Q \) otherwise arbitrary.
- The probability of the sequence depends only on the type
- restated, the probability is “independent” of the sequence given the type and \( Q \)
- That is

\[
Q^n(x_1:n) = 2^{-n \left[ H(P_{x_1:n}) + D(P_{x_1:n} || Q) \right]}
\] (6.36)

- So, probability doesn’t depend on the sequence, once we are given the type
Probability depends only on the type

Theorem 6.5.2

Let $X_1, X_2, \ldots, X_n$ be i.i.d. $\sim Q(x)$,

with extension $Q^n(x_1:n) = \prod_i Q(x_i)$, with $Q$ otherwise arbitrary.

The probability of the sequence depends only on the type

restated, the probability is "independent" of the sequence given the type and $Q$

That is

$$Q^n(x_1:n) = 2^{-n[H(P_{x_1:n}) + D(P_{x_1:n} || Q)]} \tag{6.36}$$

So, probability doesn’t depend on the sequence, once we are given the type

Compare with sufficient statistics
Probability depends only on the type

**Theorem 6.5.2**

- Let $X_1, X_2, \ldots, X_n$ be i.i.d. $\sim Q(x)$,
- with extension $Q^n(x_1:n) = \prod_i Q(x_i)$, with $Q$ otherwise arbitrary.
- The probability of the sequence depends only on the type
- restated, the probability is “independent” of the sequence given the type and $Q$
- That is

$$Q^n(x_1:n) = 2^{-n\left[H(P_{x_1:n}) + D(P_{x_1:n} \| Q)\right]} \quad (6.36)$$

- So, probability doesn’t depend on the sequence, once we are given the type
- Compare with sufficient statistics
- all sequences with the same type have the same probability.
Probability depends only on the type

Proof.

\[ Q^n(x_1:n) \]
Probability depends only on the type

Proof.

\[ Q^n(x_1:n) = \prod_{i=1}^{n} Q(x_i) \]

(6.40)
Probability depends only on the type

Proof.

\[
Q^n(x_1:n) = \prod_{i=1}^{n} Q(x_i) = \prod_{a \in \mathcal{X}} Q(a)^{n(a|x_1:n)}
\]  \hspace{1cm} (6.37)

\[
= \prod_{a \in \mathcal{X}} Q(a)^{n(a|x_1:n)}
\]  \hspace{1cm} (6.38)

\[
= \prod_{a \in \mathcal{X}} Q(a)^{n(a|x_1:n)}
\]  \hspace{1cm} (6.39)

\[
= \prod_{a \in \mathcal{X}} Q(a)^{n(a|x_1:n)}
\]  \hspace{1cm} (6.40)
Probability depends only on the type

Proof.

\[
Q^n(x_{1:n}) = \prod_{i=1}^{n} Q(x_i) = \prod_{a \in \mathcal{X}} Q(a)^{nP_{x_{1:n}}(a)}
\]

(6.37)

\[
= \prod_{a \in \mathcal{X}} Q(a)^{nP_{x_{1:n}}(a)}
\]

(6.40)
Probability depends only on the type

Proof.

\[
Q^n(x_{1:n}) = \prod_{i=1}^{n} Q(x_i) = \prod_{a \in \mathcal{X}} Q(a)^n(a|x_{1:n})
\]

\[
= \prod_{a \in \mathcal{X}} Q(a)^{nP_x(a)} = \prod_{a \in \mathcal{X}} 2^{nP_x(a) \log Q(a)}
\]

(6.37)

(6.38)

(6.40)
Probability depends only on the type

Proof.

\[ Q^n(x_1:n) = \prod_{i=1}^{n} Q(x_i) = \prod_{a \in \mathcal{X}} Q(a)^n(a|x_1:n) \]

\[ = \prod_{a \in \mathcal{X}} Q(a)^{nP_{x_1:n}(a)} = \prod_{a \in \mathcal{X}} 2^{\{nP_{x_1:n}(a) \log Q(a)\}} \]

\[ = \prod_{a \in \mathcal{X}} 2^{\left\{ P_{x_1:n}(a) \log Q(a) - P_{x_1:n}(a) \log P_{x_1:n}(a) + P_{x_1:n}(a) \log P_{x_1:n}(a) \right\}} \]

\[ = \prod_{a \in \mathcal{X}} 2^{nP_{x_1:n}(a) \log Q(a)} \]

(6.37)

(6.38)

(6.40)
Probability depends only on the type

Proof.

\[ Q^n(x_{1:n}) = \prod_{i=1}^{n} Q(x_i) = \prod_{a \in \mathcal{X}} Q(a)^{nP_{x_{1:n}}(a)} = \prod_{a \in \mathcal{X}} 2^{nP_{x_{1:n}}(a) \log Q(a)} \]  

(6.37)

\[ = \prod_{a \in \mathcal{X}} 2^{nP_{x_{1:n}}(a) \log Q(a) - P_{x_{1:n}}(a) \log P_{x_{1:n}}(a) + P_{x_{1:n}}(a) \log P_{x_{1:n}}(a)} \]

(6.38)

\[ = \prod_{a \in \mathcal{X}} 2^{nP_{x_{1:n}}(a) \log Q(a) - P_{x_{1:n}}(a) \log P_{x_{1:n}}(a)} \]

(6.39)

\[ = 2^n \sum_{a \in \mathcal{X}} (-P_{x_{1:n}}(a) \log \frac{P_{x_{1:n}}(a)}{Q(a)} + P_{x_{1:n}}(a) \log P_{x_{1:n}}(a)) \]

(6.40)
Probability depends only on the type

Proof.

\[ Q^n(x_1:n) = \prod_{i=1}^{n} Q(x_i) = \prod_{a \in \mathcal{X}} Q(a)^n(a|x_1:n) \]  
(6.37)

\[ = \prod_{a \in \mathcal{X}} Q(a)^n P_{x_1:n}(a) = \prod_{a \in \mathcal{X}} 2^{\{nP_{x_1:n}(a) \log Q(a)\}} \]  
(6.38)

\[ = \prod_{a \in \mathcal{X}} 2^{nP_{x_1:n}(a) \log Q(a) - P_{x_1:n}(a) \log P_{x_1:n}(a) + P_{x_1:n}(a) \log P_{x_1:n}(a)} = 0 \]

(6.39)

\[ = 2^{-n} \left( D(P_{x_1:n} \| Q) + H(P_{x_1:n}) \right) \]  
(6.40)
Corollary: If $Q$ is a rational distribution (i.e., a possible type) and if $x_1:n \in T(Q)$, then

$$Q^n(x_1:n) = 2^{-nH(Q)}$$  

(6.41)
Probability depends only on the type

- Corollary: If $Q$ is a rational distribution (i.e., a possible type) and if $x_{1:n} \in T(Q)$, then

$$Q^n(x_{1:n}) = 2^{-nH(Q)} \tag{6.41}$$

this should look familiar to us. Why?
Corollary: If \( Q \) is a rational distribution (i.e., a possible type) and if \( x_1:n \in T(Q) \), then
\[
Q^n(x_1:n) = 2^{-nH(Q)}
\] (6.41)

this should look familiar to us. Why?

Recall the probability of the typical sequence \( x_1, \ldots, x_n \),
\[
2^{-n(H(X)+\epsilon)} \leq p(x_1, x_2, \ldots, x_n) \leq 2^{-n(H(X)-\epsilon)}
\] (6.42)
Corollary: If $Q$ is a rational distribution (i.e., a possible type) and if $x_{1:n} \in T(Q)$, then

$$Q^n(x_{1:n}) = 2^{-nH(Q)} \quad (6.41)$$

this should look familiar to us. Why?

Recall the probability of the typical sequence $x_1, \ldots, x_n$,

$$2^{-n(H(X)+\varepsilon)} \leq p(x_1, x_2, \ldots, x_n) \leq 2^{-n(H(X)-\varepsilon)} \quad (6.42)$$

Hence, for $P \in \mathcal{P}_n$, $x_{1:n} \in T(P)$, $P^n(x_{1:n}) = 2^{-nH(P)}$. 
Probability depends only on the type

- Corollary: If $Q$ is a rational distribution (i.e., a possible type) and if $x_{1:n} \in T(Q)$, then

$$Q^n(x_{1:n}) = 2^{-nH(Q)} \tag{6.41}$$

this should look familiar to us. Why?

- Recall the probability of the typical sequence $x_1, \ldots, x_n$,

$$2^{-n(H(X)+\varepsilon)} \leq p(x_1, x_2, \ldots, x_n) \leq 2^{-n(H(X)-\varepsilon)} \tag{6.42}$$

- Hence, for $P \in \mathcal{P}_n$, $x_{1:n} \in T(P)$, $P^n(x_{1:n}) = 2^{-nH(P)}$.

- What if $Q$ was irrational?
Probability depends only on the type

- Corollary: If $Q$ is a rational distribution (i.e., a possible type) and if $x_1:n \in T(Q)$, then

$$Q^n(x_1:n) = 2^{-nH(Q)} \quad (6.41)$$

this should look familiar to us. Why?

- Recall the probability of the typical sequence $x_1, \ldots, x_n$,

$$2^{-n(H(X)+\epsilon)} \leq p(x_1, x_2, \ldots, x_n) \leq 2^{-n(H(X)-\epsilon)} \quad (6.42)$$

- Hence, for $P \in \mathcal{P}_n$, $x_1:n \in T(P)$, $P^n(x_1:n) = 2^{-nH(P)}$.

- What if $Q$ was irrational? Intuition: we could make $D(P_{x_1:n} \parallel Q)$ as small as we want, if we make $n$ large, since $P_{x_1:n}$ better approximates $Q$. Hence, even with irrational $Q$, the probability essentially depends only on the type.
We can easily express the size of the type class using multinomial coefficients.
Size of type class

- We can easily express the size of the type class using multinomial coefficients.
- I.e., the number of ways of choosing distinct alphabet symbols for every element of $x_{1:n}$. I.e., for $P \in \mathcal{P}_n$, we have

$$|T(P)| = \binom{n}{nP(a_1) \ nP(a_2) \ \cdots \ nP(a_n)}$$  \hspace{1cm} (6.43)
We can easily express the size of the type class using multinomial coefficients.

I.e., the number of ways of choosing distinct alphabet symbols for every element of $x_{1:n}$. I.e., for $P \in \mathcal{P}_n$, we have

$$|T(P)| = 
\begin{pmatrix}
nP(a_1) & nP(a_2) & \cdots & nP(a_n)
\end{pmatrix}^n \tag{6.43}$$

But we want bounds that are easier to mathematically manipulate than the multinomial.
Proposition 6.5.3

For any type $P \in \mathcal{P}_n$, we have

$$\frac{1}{(n+1)|\mathcal{X}|} 2^{nH(P)} \leq |T(P)| \leq 2^{nH(P)} \quad (6.44)$$

Proof.

$$1 \quad (6.46)$$
Size of type class

Proposition 6.5.3

For any type $P \in \mathcal{P}_n$, we have

$$\frac{1}{(n+1)|x|} 2^{nH(P)} \leq |T(P)| \leq 2^{nH(P)} \quad (6.44)$$

Proof.

$$1 \geq P^n(T(P)) \quad (6.46)$$
Proposition 6.5.3

*For any type* \( P \in \mathcal{P}_n \), *we have*

\[
\frac{1}{(n+1)|X|} 2^{nH(P)} \leq |T(P)| \leq 2^{nH(P)}
\]  
(6.44)

**Proof.**

\[
1 \geq P^n(T(P)) = \sum_{x_1:n \in T(P)} P^n(x_1:n)
\]
(6.46)
Size of type class

Proposition 6.5.3

For any type $P \in \mathcal{P}_n$, we have

$$\frac{1}{(n+1)|\mathcal{X}|} 2^{nH(P)} \leq |T(P)| \leq 2^{nH(P)} \quad (6.44)$$

Proof.

$$1 \geq P^n(T(P)) = \sum_{x_1:n \in T(P)} P^n(x_1:n) = \sum_{x_1:n \in T(P)} 2^{-nH(P)} \quad (6.45)$$

$$\sum_{x_1:n \in T(P)} 2^{-nH(P)} \leq 2^{nH(P)} \quad (6.46)$$
Proposition 6.5.3

For any type $P \in \mathcal{P}_n$, we have

$$\frac{1}{(n + 1)|\mathcal{X}|} 2^{nH(P)} \leq |T(P)| \leq 2^{nH(P)} \quad (6.44)$$

Proof.

$$1 \geq P^n(T(P)) = \sum_{x_{1:n}\in T(P)} P^n(x_{1:n}) = \sum_{x_{1:n}\in T(P)} 2^{-nH(P)} \quad (6.45)$$

$$= |T(P)| 2^{-nH(P)} \quad (6.46)$$
Size of type class

Proposition 6.5.3

For any type $P \in \mathcal{P}_n$, we have

$$\frac{1}{(n + 1)|X|} 2^{nH(P)} \leq |T(P)| \leq 2^{nH(P)} \quad (6.44)$$

Proof.

$$1 \geq P^n(T(P)) = \sum_{x_1:n \in T(P)} P^n(x_1:n) = \sum_{x_1:n \in T(P)} 2^{-nH(P)} \quad (6.45)$$

$$= |T(P)| 2^{-nH(P)} \quad (6.46)$$

Note that in the sums, $P_{x_1:n} = P$. 

□
Size of type class

**Proposition 6.5.3**

For any type $P \in \mathcal{P}_n$, we have

$$\frac{1}{(n + 1)|X|} 2^{nH(P)} \leq |T(P)| \leq 2^{nH(P)}$$  \hspace{1cm} (6.44)

**Proof.**

$$1 \geq P^n(T(P)) = \sum_{x_{1:n} \in T(P)} P^n(x_{1:n}) = \sum_{x_{1:n} \in T(P)} 2^{-nH(P)}$$  \hspace{1cm} (6.45)

$$= |T(P)| 2^{-nH(P)}$$  \hspace{1cm} (6.46)

Note that in the sums, $P_{x_{1:n}} = P$. This gives the upper bound.
Size of type class

**Proposition 6.5.3**

*For any type* $P \in \mathcal{P}_n$, we have

$$\frac{1}{(n+1)|X|} 2^{nH(P)} \leq |T(P)| \leq 2^{nH(P)}$$

(6.44)

**Proof.**

$$1 \geq P^n(T(P)) = \sum_{x_1:n \in T(P)} P^n(x_1:n) = \sum_{x_1:n \in T(P)} 2^{-nH(P)}$$

(6.45)

$$= |T(P)| 2^{-nH(P)}$$

(6.46)

Note that in the sums, $P_{x_1:n} = P$. This gives the upper bound.

Before doing the lower bound, let's do a lemma . . .
What type class has the highest probability, when the generating distribution is $P \in \mathcal{P}_n$?
Type class with highest probability

- What type class has the highest probability, when the generating distribution is $P \in \mathcal{P}_n$?
- Consider AEP, the typical sequences are the ones closest to the real distribution, and they have all the probability.
Type class with highest probability

- What type class has the highest probability, when the generating distribution is $P \in \mathcal{P}_n$?
- Consider AEP, the typical sequences are the ones closest to the real distribution, and they have all the probability.
- Thus, we’ll guess that $T(P)$ has the highest probability under distribution $P$. 

Note: Non-negative integers $m$ and $n$, then $m! = n^m n!$, since if $m > n$, then $m! n! = m(m-1)...(n+1)$, and if $m < n$, then $m! n! = 1n! (n-1)...(m+1)$, and if $m = n$ then obvious.
Type class with highest probability

- What type class has the highest probability, when the generating distribution is $P \in \mathcal{P}_n$?
- Consider AEP, the typical sequences are the ones closest to the real distribution, and they have all the probability.
- Thus, we’ll guess that $T(P)$ has the highest probability under distribution $P$.
- I.e., our lemma becomes
Type class with highest probability

- What type class has the highest probability, when the generating distribution is $P \in \mathcal{P}_n$?
- Consider AEP, the typical sequences are the ones closest to the real distribution, and they have all the probability.
- Thus, we’ll guess that $T(P)$ has the highest probability under distribution $P$.
- I.e., our lemma becomes

**Lemma 6.5.4**

For $P \in \mathcal{P}_n$, then $T(P)$ has the highest probability. That is

$$P^n(T(P)) \geq P^n(T(\hat{P})), \quad \forall \hat{P} \in \mathcal{P}_n$$  \hspace{1cm} (6.47)
Type class with highest probability

- What type class has the highest probability, when the generating distribution is \( P \in \mathcal{P}_n \)?
- Consider AEP, the typical sequences are the the ones closest to the real distribution, and they have all the probability.
- Thus, we’ll guess that \( T(P) \) has the highest probability under distribution \( P \).
- I.e., our lemma becomes

**Lemma 6.5.4**

For \( P \in \mathcal{P}_n \), then \( T(P) \) has the highest probability. That is

\[
P^n(T(P)) \geq P^n(T(\hat{P})), \quad \forall \hat{P} \in \mathcal{P}_n
\]  

(6.47)

- Note: Non-negative integers \( m \) and \( n \), then \( \frac{m!}{n!} \geq n^{m-n} \) since
Type class with highest probability

- What type class has the highest probability, when the generating distribution is $P \in \mathcal{P}_n$?
- Consider AEP, the typical sequences are the ones closest to the real distribution, and they have all the probability.
- Thus, we’ll guess that $T(P)$ has the highest probability under distribution $P$.
- I.e., our lemma becomes

**Lemma 6.5.4**

For $P \in \mathcal{P}_n$, then $T(P)$ has the highest probability. That is

$$P^n(T(P)) \geq P^n(T(\hat{P})), \quad \forall \hat{P} \in \mathcal{P}_n$$  \hspace{1cm} (6.47)

- Note: Non-negative integers $m$ and $n$, then $\frac{m!}{n!} \geq n^{m-n}$ since if $m > n$, then $\frac{m!}{n!} = m(m-1) \ldots (n+1) \geq n^{m-n}$,
Type class with highest probability

- What type class has the highest probability, when the generating distribution is $P \in \mathcal{P}_n$?
- Consider AEP, the typical sequences are the ones closest to the real distribution, and they have all the probability.
- Thus, we’ll guess that $T(P)$ has the highest probability under distribution $P$.
- I.e., our lemma becomes

\begin{equation}
\text{Lemma 6.5.4}
\end{equation}

\[
\text{for } P \in \mathcal{P}_n, \text{ then } T(P) \text{ has the highest probability. That is }
\]

\[
P^n(T(P)) \geq P^n(T(\hat{P})), \quad \forall \hat{P} \in \mathcal{P}_n
\]  

\begin{align}
\text{Note: Non-negative integers } m \text{ and } n, \text{ then } \frac{m!}{n!} \geq n^{m-n} \text{ since if } \\
m > n, \text{ then } \frac{m!}{n!} = m(m-1) \ldots (n+1) \geq n^{m-n}, \text{ and if } m < n, \\
\text{then } \frac{m!}{n!} = \frac{1}{n(n-1) \ldots (m+1)} \geq \frac{1}{n^{n-m}},
\end{align}
Type class with highest probability

- What type class has the highest probability, when the generating distribution is \( P \in \mathcal{P}_n \)?
- Consider AEP, the typical sequences are the the ones closest to the real distribution, and they have all the probability.
- Thus, we’ll guess that \( T(P) \) has the highest probability under distribution \( P \).
- I.e., our lemma becomes

**Lemma 6.5.4**

For \( P \in \mathcal{P}_n \), then \( T(P) \) has the highest probability. That is

\[
P^n(T(P)) \geq P^n(T(\hat{P})), \quad \forall \hat{P} \in \mathcal{P}_n
\]  

(6.47)

- Note: Non-negative integers \( m \) and \( n \), then \( \frac{m!}{n!} \geq n^{m-n} \) since if \( m > n \), then \( \frac{m!}{n!} = m(m - 1) \ldots (n+1) \geq n^{m-n} \), and if \( m < n \), then \( \frac{m!}{n!} = \frac{1}{n(n-1) \ldots (m+1)} \geq \frac{1}{n^{n-m}} \), and if \( m = n \) then obvious.
Type class with highest probability

Proof of Lemma 6.5.4.

\[
\frac{P^n(T(P))}{P^n(T(\hat{P}))}
\]  

(6.53)
Type class with highest probability

Proof of Lemma 6.5.4.

\[
\frac{P^n(T(P))}{P^n(T(\hat{P}))} = \frac{|T(P)| \prod_{a \in \mathcal{X}} P(a)^{nP(a)}}{|T(\hat{P})| \prod_{a \in \mathcal{X}} \hat{P}(a)^{nP(a)}}
\]  

(6.48)

\[
\prod_a \left[ \frac{P^n(a)}{\hat{P}^n(a)} \right] = \left[ \frac{\prod_a P(a)^{nP(a)}}{\prod_a \hat{P}(a)^{nP(a)}} \right]
\]

(6.53)
Type class with highest probability

Proof of Lemma 6.5.4.

\[
\frac{P^n(T(P))}{P^n(T(\hat{P}))} = \frac{|T(P)| \prod_{a \in \mathcal{X}} P(a)^nP(a)}{|T(\hat{P})| \prod_{a \in \mathcal{X}} P(a)^n\hat{P}(a)} \quad (6.48)
\]

\[
= \frac{\binom{nP(a_1)}{n} \binom{nP(a_2)}{n} \cdots \binom{nP(a_n)}{n}}{\binom{n\hat{P}(a_1)}{n} \binom{n\hat{P}(a_2)}{n} \cdots \binom{n\hat{P}(a_n)}{n}} \prod_{a \in \mathcal{X}} P(a)^nP(a) \quad (6.49)
\]

\[
= \frac{1}{1} \quad (6.53)
\]
Type class with highest probability

Proof of Lemma 6.5.4.

\[
\frac{P^n(T(P))}{P^n(T(\hat{P})))} = \frac{|T(P)| \prod_{a \in \mathcal{X}} P(a)^{nP(a)}}{|T(\hat{P})| \prod_{a \in \mathcal{X}} P(a)^{n\hat{P}(a)}}
\]

(6.48)

\[
= \frac{\binom{nP(a_1)}{nP(a_2)} \cdots \binom{nP(a_n)}{nP(a_1)} \prod_{a \in \mathcal{X}} P(a)^{nP(a)}}{\binom{n\hat{P}(a_1)}{n\hat{P}(a_2)} \cdots \binom{n\hat{P}(a_n)}{n\hat{P}(a_1)} \prod_{a \in \mathcal{X}} P(a)^{n\hat{P}(a)}}
\]

(6.49)

\[
= \prod_{a \in \mathcal{X}} \frac{[n\hat{P}(a)]!}{[nP(a)]!} P(a)^{n(P(a) - \hat{P}(a))}
\]

(6.50)

(6.53)
Type class with highest probability

**Proof of Lemma 6.5.4.**

\[
\frac{P^n(T(P))}{P^n(T(\hat{P}))} = \frac{|T(P)| \prod_{a \in \mathcal{X}} P(a)^{nP(a)}}{|T(\hat{P})| \prod_{a \in \mathcal{X}} P(a)^{n\hat{P}(a)}}
\]

(6.48)

\[
= \left( \frac{nP(a_1)}{n\hat{P}(a_1)} \frac{nP(a_2)}{n\hat{P}(a_2)} \cdots \frac{nP(a_n)}{n\hat{P}(a_n)} \right) \prod_{a \in \mathcal{X}} P(a)^{nP(a)}
\]

(6.49)

\[
= \prod_{a \in \mathcal{X}} \left[ \frac{[n\hat{P}(a)]!}{nP(a)!} \right] P(a)^{nP(a)-\hat{P}(a)}
\]

(6.50)

\[
\geq \prod_{a \in \mathcal{X}} (nP(a))^{n(\hat{P}(a)-P(a))} P(a)^{nP(a)-\hat{P}(a)}
\]

(6.51)

(6.53)
Type class with highest probability

Proof of Lemma 6.5.4.

\[
\frac{P^n(T(P))}{P^n(T(\hat{P}))} = \frac{|T(P)| \prod_{a \in \mathcal{X}} P(a)^{nP(a)}}{|T(\hat{P})| \prod_{a \in \mathcal{X}} P(a)^{n\hat{P}(a)}}
\]

(6.48)

\[
= \frac{\binom{nP(a_1)}{P(a_2)} \ldots \binom{nP(a_n)}{P(a_n)}}{\binom{n\hat{P}(a_1)}{\hat{P}(a_2)} \ldots \binom{n\hat{P}(a_n)}{\hat{P}(a_n)}} \prod_{a \in \mathcal{X}} P(a)^{n\hat{P}(a)}
\]

(6.49)

\[
= \prod_{a \in \mathcal{X}} \frac{[n\hat{P}(a)]!}{[nP(a)]!} P(a)^{nP(a) - \hat{P}(a)}
\]

(6.50)

\[
\geq \prod_{a \in \mathcal{X}} (nP(a))^{n(\hat{P}(a) - P(a))} P(a)^{nP(a) - \hat{P}(a)}
\]

(6.51)

\[
= \prod_{a \in \mathcal{X}} n^{n(\hat{P}(a) - P(a))}
\]

(6.53)
Type class with highest probability

Proof of Lemma 6.5.4.

\[
\frac{P^n(T(P))}{P^n(T(\hat{P}))} = \frac{|T(P)| \prod_{a \in \mathcal{X}} P(a)^{nP(a)}}{|T(\hat{P})| \prod_{a \in \mathcal{X}} P(a)^{n\hat{P}(a)}}
\]

(6.48)

\[
= \frac{\left( nP(a_1) \quad nP(a_2) \quad \ldots \quad nP(a_n) \right) \prod_{a \in \mathcal{X}} P(a)^{nP(a)}}{\left( n\hat{P}(a_1) \quad n\hat{P}(a_2) \quad \ldots \quad n\hat{P}(a_n) \right) \prod_{a \in \mathcal{X}} P(a)^{n\hat{P}(a)}}
\]

(6.49)

\[
= \prod_{a \in \mathcal{X}} \frac{[n\hat{P}(a)]!}{[nP(a)]!} P(a)^{nP(a) - \hat{P}(a)}
\]

(6.50)

\[
\geq \prod_{a \in \mathcal{X}} \left( nP(a) \right)^{n(\hat{P}(a) - P(a))} P(a)^{nP(a) - \hat{P}(a)}
\]

(6.51)

\[
= \prod_{a \in \mathcal{X}} n^{n(\hat{P}(a) - P(a))} = n^n \left[ \sum_{a \in \mathcal{X}} \hat{P}(a) - \sum_{a \in \mathcal{X}} P(a) \right]
\]

(6.52)

(6.53)
Proof of Lemma 6.5.4.

\[
\frac{P^n(T(P))}{P^n(T(\hat{P}))} = \frac{|T(P)| \prod_{a \in \mathcal{X}} P(a)^{nP(a)}}{|T(\hat{P})| \prod_{a \in \mathcal{X}} P(a)^{n\hat{P}(a)}} = \frac{\binom{nP(a_1)}{nP(a_2)} \cdots \binom{nP(a_n)}{nP(a_n)}}{\binom{n\hat{P}(a_1)}{n\hat{P}(a_2)} \cdots \binom{n\hat{P}(a_n)}{n\hat{P}(a_n)}} \prod_{a \in \mathcal{X}} P(a)^{nP(a)} (6.48)
\]

\[
= \prod_{a \in \mathcal{X}} \left[ \frac{[n\hat{P}(a)]!}{[nP(a)]!} \right] P(a)^{nP(a) - \hat{P}(a)} = \prod_{a \in \mathcal{X}} (nP(a))^{n(\hat{P}(a) - P(a))} P(a)^{nP(a) - \hat{P}(a)} (6.49)
\]

\[
\geq \prod_{a \in \mathcal{X}} (nP(a))^{n(\hat{P}(a) - P(a))} P(a)^{nP(a) - \hat{P}(a)} (6.50)
\]

\[
= \prod_{a \in \mathcal{X}} n^{n(\hat{P}(a) - P(a))} = n^n \left[ \sum_{a \in \mathcal{X}} \hat{P}(a) - \sum_{a \in \mathcal{X}} P(a) \right] (6.51)
\]

\[
= n^n (1 - 1) (6.52)
\]

\[
= n^n (1 - 1) (6.53)
\]
Proof of Lemma 6.5.4.

\[
\frac{P^n(T(P))}{P^n(T(\hat{P}))} = \frac{|T(P)| \prod_{a \in \mathcal{X}} P(a)^{nP(a)}}{|T(\hat{P})| \prod_{a \in \mathcal{X}} P(a)^{n\hat{P}(a)}} \tag{6.48}
\]

\[
= \frac{nP(a_1) \cdots nP(a_n)}{n\hat{P}(a_1) \cdots n\hat{P}(a_n)} \prod_{a \in \mathcal{X}} P(a)^{nP(a)} \prod_{a \in \mathcal{X}} P(a)^{n\hat{P}(a)} \tag{6.49}
\]

\[
= \prod_{a \in \mathcal{X}} \frac{[nP(\hat{P}(a))]!}{[nP(a)]!} P(a)^{nP(a) - \hat{P}(a)} \tag{6.50}
\]

\[
\geq \prod_{a \in \mathcal{X}} (nP(a))^{n(\hat{P}(a) - P(a))} P(a)^{nP(a) - \hat{P}(a)} \tag{6.51}
\]

\[
= \prod_{a \in \mathcal{X}} n^{nP(a) - P(a)} = n^n \left[ \sum_{a \in \mathcal{X}} \hat{P}(a) - \sum_{a \in \mathcal{X}} P(a) \right] \tag{6.52}
\]

\[
= n^n(1-1) = 1 \tag{6.53}
\]
Type class with highest probability

Proof of Lemma 6.5.4.

\[
\frac{P^n(T(P))}{P^n(T(\hat{P}))} = \frac{|T(P)| \prod_{a \in \mathcal{X}} P(a)^{nP(a)}}{|T(\hat{P})| \prod_{a \in \mathcal{X}} P(a)^{n\hat{P}(a)}}
\]

\[
= \frac{(nP(a_1) \cdotnP(a_2) \cdots \cdotnP(a_n)) \prod_{a \in \mathcal{X}} P(a)^{nP(a)}}{(n\hat{P}(a_1) \cdotn\hat{P}(a_2) \cdots \cdotn\hat{P}(a_n)) \prod_{a \in \mathcal{X}} P(a)^{n\hat{P}(a)}}
\]

\[
= \prod_{a \in \mathcal{X}} \frac{[n\hat{P}(a)]!}{[nP(a)]!} P(a)^{nP(a) - \hat{P}(a)}
\]

\[
\geq \prod_{a \in \mathcal{X}} (nP(a))^{n(\hat{P}(a) - P(a))} P(a)^{nP(a) - \hat{P}(a)}
\]

\[
= \prod_{a \in \mathcal{X}} n^{n(\hat{P}(a) - P(a))} = n^{n[\sum_{a \in \mathcal{X}} \hat{P}(a) - \sum_{a \in \mathcal{X}} P(a)]}
\]

\[
= n^{n(1-1)} = 1
\]

thus, \( P^n(T(P)) \geq P^n(T(\hat{P})) \).
Proof of lower bound

Proof of lowerbound of theorem 6.5.3.

1

\[ (6.58) \]

which gives us our result that.

\[ |T(P)| \leq (n + 1) |X|^{2nH(P)} \] (6.59)

Prof. Jeff Bilmes
EE514a/Fall 2019/Info. Theory I – Lecture 6 - Oct 14th, 2019
Proof of lower bound

Proof of lowerbound of theorem 6.5.3.

\[ 1 = \sum_{Q \in \mathcal{P}_n} P^n(T(Q)) \]

(6.58)
Proof of lower bound

Proof of lowerbound of theorem 6.5.3.

\[ 1 = \sum_{Q \in \mathcal{P}_n} P^n(T(Q)) \leq \sum_{Q \in \mathcal{P}_n} \max_{R \in \mathcal{P}_n} P^n(T(R)) \]  \hspace{1cm} (6.54)

\[ (6.58) \]
Proof of lower bound

Proof of lowerbound of theorem 6.5.3.

\[
1 = \sum_{Q \in \mathcal{P}_n} P^n(T(Q)) \leq \sum_{Q \in \mathcal{P}_n} \max_{R \in \mathcal{P}_n} P^n(T(R)) \tag{6.54}
\]

\[
= \sum_{Q \in \mathcal{P}_n} P^n(T(P))
\]

\[
= (n+1) |X| 2^n H(P) \tag{6.58}
\]
Proof of lower bound of theorem 6.5.3.

\[ 1 = \sum_{Q \in \mathcal{P}_n} P^n(T(Q)) \leq \sum_{Q \in \mathcal{P}_n} \max_{R \in \mathcal{P}_n} P^n(T(R)) \]  

(6.54)

\[ = \sum_{Q \in \mathcal{P}_n} P^n(T(P)) \leq (n + 1)|X| P^n(T(P)) \]  

(6.55)

\[ = (n + 1) |X| \]  

(6.58)
Proof of lower bound of theorem 6.5.3.

\[ 1 = \sum_{Q \in \mathcal{P}_n} P^n(T(Q)) \leq \sum_{R \in \mathcal{P}_n} \max_{Q \in \mathcal{P}_n} P^n(T(R)) \]  

(6.54)

\[ = \sum_{Q \in \mathcal{P}_n} P^n(T(P)) \leq (n + 1)|\mathcal{X}| P^n(T(P)) \]  

(6.55)

\[ = (n + 1)|\mathcal{X}| \sum_{x_1:n \in T(P)} P^n(x_1:n) \]  

(6.58)
Proof of lower bound

Proof of lower bound of theorem 6.5.3.

\[
1 = \sum_{Q \in \mathcal{P}_n} P^n(T(Q)) \leq \sum_{Q \in \mathcal{P}_n} \max_{R \in \mathcal{P}_n} P^n(T(R)) \tag{6.54}
\]

\[
= \sum_{Q \in \mathcal{P}_n} P^n(T(P)) \leq (n + 1)|\mathcal{X}| P^n(T(P)) \tag{6.55}
\]

\[
= (n + 1)|\mathcal{X}| \sum_{x_1:n \in T(P)} P^n(x_1:n) = (n + 1)|\mathcal{X}| \sum_{x_1:n \in T(P)} 2^{-nH(P)} \tag{6.56}
\]

\[
\leq (n + 1)|\mathcal{X}| \sum_{x_1:n \in T(P)} 2^{-nH(P)} \tag{6.58}
\]
Proof of lower bound of theorem 6.5.3.

\[ 1 = \sum_{Q \in \mathcal{P}_n} P^n(T(Q)) \leq \sum_{Q \in \mathcal{P}_n} \max_{R \in \mathcal{P}_n} P^n(T(R)) \quad (6.54) \]

\[ = \sum_{Q \in \mathcal{P}_n} P^n(T(P)) \leq (n + 1)|\mathcal{X}| P^n(T(P)) \quad (6.55) \]

\[ = (n + 1)|\mathcal{X}| \sum_{x_{1:n} \in T(P)} P^n(x_{1:n}) = (n + 1)|\mathcal{X}| \sum_{x_{1:n} \in T(P)} 2^{-nH(P)} \quad (6.56) \]

\[ = (n + 1)|\mathcal{X}| |T(P)| 2^{-nH(P)} \quad (6.57) \]

\[ (6.58) \]
Proof of lower bound of theorem 6.5.3.

\[
1 = \sum_{Q \in \mathcal{P}_n} P^n(T(Q)) \leq \sum_{Q \in \mathcal{P}_n} \max_{R \in \mathcal{P}_n} P^n(T(R)) 
\]

\[
= \sum_{Q \in \mathcal{P}_n} P^n(T(P)) \leq (n + 1)|\mathcal{X}| P^n(T(P)) 
\]

\[
= (n + 1)|\mathcal{X}| \sum_{x_1:n \in T(P)} P^n(x_1:n) = (n + 1)|\mathcal{X}| \sum_{x_1:n \in T(P)} 2^{-nH(P)} 
\]

\[
= (n + 1)|\mathcal{X}| |T(P)| 2^{-nH(P)} 
\]

which gives us our result that.

\[
|T(P)| \geq \frac{1}{(n + 1)|\mathcal{X}|} 2^{nH(P)} 
\]
For binary case, \( \mathcal{X} = \{0, 1\} \) we have the interesting bound

\[
\frac{1}{(n + 1)^2} 2^{nH\left(\frac{k}{n}\right)} \leq \binom{n}{k} \leq 2^{nH\left(\frac{k}{n}\right)}
\]  

(6.60)
Combinatorial Bounds

For binary case, $\mathcal{X} = \{0, 1\}$ we have the interesting bound

$$\frac{1}{(n+1)^2} 2^{nH\left(\frac{k}{n}\right)} \leq \binom{n}{k} \leq 2^{nH\left(\frac{k}{n}\right)}$$

(6.60)

Information theory can often be used to produce bounds on combinatorial functions. See chapter 16 in 1st edition of book or chapter 17 in 2nd edition.
Combinatorial Bounds

- For binary case, \( \mathcal{X} = \{0, 1\} \) we have the interesting bound
  \[
  \frac{1}{(n+1)^2} 2^{nH\left(\frac{k}{n}\right)} \leq \binom{n}{k} \leq 2^{nH\left(\frac{k}{n}\right)}
  \]  
  (6.60)

- Information theory can often be used to produce bounds on combinatorial functions. See chapter 16 in 1st edition of book or chapter 17 in 2nd edition.

- The bound in fact can be tightened a bit in the binary case:
  \[
  \frac{1}{(n+1)} 2^{nH\left(\frac{k}{n}\right)} \leq \binom{n}{k}
  \]  
  (6.61)

(exercise: show this)
How probable is each type class?

- Notation: \( a_n \hat{=} b_n \) if \( \lim_{n \to \infty} \frac{1}{n} \log \frac{a_n}{b_n} = 0 \).
How probable is each type class?

- Notation: \( a_n \asymp b_n \) if \( \lim_{n \to \infty} \frac{1}{n} \log \frac{a_n}{b_n} = 0 \).
How probable is each type class?

- Notation: \( a_n \approx b_n \) if \( \lim_{n \to \infty} \frac{1}{n} \log \frac{a_n}{b_n} = 0 \).

**Theorem 6.5.5**

For any \( P \in \mathcal{P}_n \), and any distribution \( Q \), the probability of type class \( T(P) \) under \( Q^n \) is such that \( Q^n(T(P)) \approx 2^{-nD(P||Q)} \). Specifically,

\[
\frac{1}{(n + 1)|X|} 2^{-nD(P||Q)} \leq Q^n(T(P)) \leq 2^{-nD(P||Q)}
\]

(6.62)
How probable is each type class?

- Notation: \( a_n \doteq b_n \) if \( \lim_{n \to \infty} \frac{1}{n} \log \frac{a_n}{b_n} = 0 \).

**Theorem 6.5.5**

For any \( P \in \mathcal{P}_n \), and any distribution \( Q \), the probability of type class \( T(P) \) under \( Q^n \) is such that \( Q^n(T(P)) \doteq 2^{-nD(P||Q)} \). Specifically,

\[
\frac{1}{(n+1)|X|} 2^{-nD(P||Q)} \leq Q^n(T(P)) \leq 2^{-nD(P||Q)}
\]

(6.62)

Note: so any type less close than the “closest” type to \( Q \) will decrease in probability exponentially (in \( n \)) faster than the most probable type.
How probable is each type class?

- Notation: $a_n \overset{d}{=} b_n$ if $\lim_{n \to \infty} \frac{1}{n} \log \frac{a_n}{b_n} = 0$.

**Theorem 6.5.5**

For any $P \in \mathcal{P}_n$, and any distribution $Q$, the probability of type class $T(P)$ under $Q^n$ is such that $Q^n(T(P)) \overset{d}{=} 2^{-nD(P||Q)}$. Specifically,

$$\frac{1}{(n + 1)|X|}2^{-nD(P||Q)} \leq Q^n(T(P)) \leq 2^{-nD(P||Q)} \quad (6.62)$$

Note: so any type less close than the “closest” type to $Q$ will decrease in probability exponentially (in $n$) faster than the most probable type.

**Proof.**

$$Q^n(T(P)) \quad (6.64)$$
How probable is each type class?

- Notation: \( a_n \doteq b_n \) if \( \lim_{n \to \infty} \frac{1}{n} \log \frac{a_n}{b_n} = 0. \)

**Theorem 6.5.5**

For any \( P \in \mathcal{P}_n \), and any distribution \( Q \), the probability of type class \( T(P) \) under \( Q^n \) is such that

\[
Q^n(T(P)) \doteq 2^{-nD(P||Q)}.
\]

Specifically,

\[
\frac{1}{(n + 1)|X|} 2^{-nD(P||Q)} \leq Q^n(T(P)) \leq 2^{-nD(P||Q)}
\]

(6.62)

Note: so any type less close than the “closest” type to \( Q \) will decrease in probability exponentially (in \( n \)) faster than the most probable type.

**Proof.**

\[
Q^n(T(P)) = \sum_{x_{1:n} \in T(P)} Q^n(x_{1:n})
\]

(6.64)
How probable is each type class?

- Notation: \( a_n = b_n \) if \( \lim_{n \to \infty} \frac{1}{n} \log \frac{a_n}{b_n} = 0 \).

**Theorem 6.5.5**

For any \( P \in \mathcal{P}_n \), and any distribution \( Q \), the probability of type class \( T(P) \) under \( Q^n \) is such that \( Q^n(T(P)) \sim 2^{-nD(P||Q)} \). Specifically,

\[
\frac{1}{(n+1)|x|} 2^{-nD(P||Q)} \leq Q^n(T(P)) \leq 2^{-nD(P||Q)}
\]  

(6.62)

Note: so any type less close than the “closest” type to \( Q \) will decrease in probability exponentially (in \( n \)) faster than the most probable type.

**Proof.**

\[
Q^n(T(P)) = \sum_{x_{1:n} \in T(P)} Q^n(x_{1:n}) = \sum_{x_{1:n} \in T(P)} 2^{-n(D(P||Q)+H(P))}
\]  

(6.63)

\[
(6.64)
How probable is each type class?

- Notation: \( a_n \equiv b_n \) if \( \lim_{n \to \infty} \frac{1}{n} \log \frac{a_n}{b_n} = 0 \).

**Theorem 6.5.5**

For any \( P \in \mathcal{P}_n \), and \textit{any} distribution \( Q \), the probability of type class \( T(P) \) under \( Q^n \) is such that \( Q^n(T(P)) \equiv 2^{-nD(P||Q)} \). Specifically,

\[
\frac{1}{(n + 1)|X|} 2^{-nD(P||Q)} \leq Q^n(T(P)) \leq 2^{-nD(P||Q)} \tag{6.62}
\]

Note: so any type less close than the “closest” type to \( Q \) will decrease in probability exponentially (in \( n \)) faster than the most probable type.

**Proof.**

\[
Q^n(T(P)) = \sum_{x_{1:n} \in T(P)} Q^n(x_{1:n}) = \sum_{x_{1:n} \in T(P)} 2^{-n(D(P||Q)+H(P))} \tag{6.63}
\]

\[
= |T(P)| 2^{-n(D(P||Q)+H(P))} \tag{6.64}
\]
How probable is each type class?

- Notation: \( a_n \sim b_n \) if \( \lim_{n \to \infty} \frac{1}{n} \log \frac{a_n}{b_n} = 0 \).

**Theorem 6.5.5**

For any \( P \in \mathcal{P}_n \), and any distribution \( Q \), the probability of type class \( T(P) \) under \( Q^n \) is such that \( Q^n(T(P)) \sim 2^{-nD(P||Q)} \). Specifically,

\[
\frac{1}{(n+1)|X|} 2^{-nD(P||Q)} \leq Q^n(T(P)) \leq 2^{-nD(P||Q)}
\]

(6.62)

Note: so any type less close than the “closest” type to \( Q \) will decrease in probability exponentially (in \( n \)) faster than the most probable type.

**Proof.**

\[
Q^n(T(P)) = \sum_{x_{1:n} \in T(P)} Q^n(x_{1:n}) = \sum_{x_{1:n} \in T(P)} 2^{-n(D(P||Q)+H(P))}
\]

(6.63)

\[
= |T(P)| 2^{-n(D(P||Q)+H(P))}
\]

(6.64)

and then use \( \frac{1}{(n+1)|X|} 2^{nH(P)} \leq |T(P)| \leq 2^{nH(P)} \)
Summary of basic theorems

- Number of types with denominator $n$

$$|P_n| \leq (n + 1)|\mathcal{X}|$$  \hspace{1cm} (6.65)
Summary of basic theorems

- Number of types with denominator \( n \)

\[
|\mathcal{P}_n| \leq (n + 1)|\mathcal{X}|
\]  

(6.65)

- \( p(x_1:n) \) depends only on the type (prob. indep. of sample given type)

\[
Q^n(x_{1:n}) = 2^{-n[H(P_{x_1:n}) + D(P_{x_1:n} || Q)]}
\]  

(6.66)
Summary of basic theorems

- Number of types with denominator $n$
  
  $$|\mathcal{P}_n| \leq (n + 1)|\mathcal{X}| \quad (6.65)$$

- $p(x_{1:n})$ depends only on the type (prob. indep. of sample given type)
  
  $$Q^n(x_{1:n}) = 2^{-n}[H(P_{x_{1:n}}) + D(P_{x_{1:n}} || Q)] \quad (6.66)$$

- Size of the type class
  
  $$|T(P)| = 2^{nH(P)} \quad (6.67)$$
Summary of basic theorems

- Number of types with denominator $n$
  \[
  |\mathcal{P}_n| \leq (n + 1)|\mathcal{X}|
  \]  
  (6.65)

- $p(x_{1:n})$ depends only on the type (prob. indep. of sample given type)
  \[
  Q^n(x_{1:n}) = 2^{-n[H(P_{x_{1:n}}) + D(P_{x_{1:n}} || Q)]}
  \]  
  (6.66)

- Size of the type class
  \[
  |T(P)| \doteq 2^{nH(P)}
  \]  
  (6.67)

- Probability of a type class
  \[
  Q^n(T(P)) \doteq 2^{-nD(P || Q)}
  \]  
  (6.68)
Types with the most probability

Q: Which types will have the most probability?

The property $Q_n(T(P)) = 2^n D(P||Q)$ says that the ones that are farther away will have exponentially smaller probability than the others, as $n \to \infty$. This suggests that "typical set of sequences" applies here as well, in fact

**Definition 6.5.6 (typical set of sequences)**

Let $X_1, X_2, ..., X_n$ be i.i.d. $\forall i, x_i \sim Q(x_i)$. The typical set is defined as

$$T_{\varepsilon}Q = \{ x_1^n : \max_i D(P(x_1^n)||Q) \leq \varepsilon \}$$

(6.69)

Intuitively, these are sequences that come from types that are $\varepsilon$-close to $Q$ in the KL-sense.
Types with the most probability

Q: Which types will have the most probability?
A: Clearly, the ones that are closest to the true distribution.
Types with the most probability

Q: Which types will have the most probability?
A: Clearly, the ones that are closest to the true distribution.

The property $Q^n(T(P)) \overset{!}{=} 2^{-nD(P||Q)}$ says that the ones that are farther away will have exponentially smaller probability than the others, as $n \to \infty$. 
Types with the most probability

Q: Which types will have the most probability?
A: Clearly, the ones that are closest to the true distribution.

The property $Q^n(T(P)) \triangleq 2^{-nD(P||Q)}$ says that the ones that are farther away will have exponentially smaller probability than the others, as $n \to \infty$.

This suggests that “typical set of sequences” applies here as well,
Types with the most probability

Q: Which types will have the most probability?
A: Clearly, the ones that are closest to the true distribution.

The property $Q^n(T(P)) \overset{\cdot}{=} 2^{-nD(P\|Q)}$ says that the ones that are farther away will have exponentially smaller probability than the others, as $n \to \infty$.

This suggests that “typical set of sequences” applies here as well,

in fact
Types with the most probability

Q: Which types will have the most probability?
A: Clearly, the ones that are closest to the true distribution.

The property \(Q^n(T(P)) \equiv 2^{-nD(P||Q)}\) says that the ones that are farther away will have exponentially smaller probability than the others, as \(n \to \infty\).

This suggests that “typical set of sequences” applies here as well, in fact

**Definition 6.5.6 (typical set of sequences)**

Let \(X_1, X_2, \ldots, X_n\) be i.i.d. \(\forall i, x_i \sim Q(x)\). Then the typical set is defined as

\[
T_Q^\epsilon = \{x_{1:n} : D(P_{x_{1:n}}\|Q) \leq \epsilon\} \tag{6.69}
\]
Types with the most probability

Q: Which types will have the most probability?
A: Clearly, the ones that are closest to the true distribution.

The property $Q^n(T(P)) = 2^{-nD(P\|Q)}$ says that the ones that are farther away will have exponentially smaller probability than the others, as $n \to \infty$.

This suggests that “typical set of sequences” applies here as well, in fact

**Definition 6.5.6 (typical set of sequences)**

Let $X_1, X_2, \ldots, X_n$ be i.i.d. $\forall i, x_i \sim Q(x)$. Then the typical set is defined as

$$T^\varepsilon_Q = \{x_{1:n} : D(P_{x_{1:n}} \| Q) \leq \varepsilon\}$$

Intuitively, these are sequences that come from types that are $\varepsilon$-close to $Q$ in the KL-sense.