EE514a – Information Theory I
Fall Quarter 2013

Prof. Jeff Bilmes

University of Washington, Seattle
Department of Electrical Engineering
Fall Quarter, 2013
http://j.ee.washington.edu/~bilmes/classes/ee514a_fall_2013/

Lecture 5 - Oct 9th, 2013
Class Road Map - IT-I

- L1 (9/26): Overview, Communications, Information, Entropy
- L2 (10/1): Props. Entropy, Mutual Information,
- L5 (10/9): AEP
- L6
- L7
- L8
- L9
- L10
- L11
- L12
- L13
- L14
- L15
- L16
- L17
- L18
- L19

Finals Week: December 12th–16th.
Read chapters 1 and 2 in our book (Cover & Thomas, “Information Theory”) (including Fano’s inequality).

Chapter 3 in our book (Cover & Thomas, “Information Theory”).

Section 11.1 (method of types).
Homework

- Homework 2 is posted on our assignment dropbox (https://canvas.uw.edu/courses/847774/assignments), due tonight, Tuesday, Oct 15th, at 11:45pm.
Announcements

- Office hours, every week, Tuesdays 4:30-5:30pm. Can also reach me at that time via a canvas conference.
Towards AEP

- Suppose we wish to encode these $K^n$ outcomes with binary digits of length $m$. Thus there are $M = 2^m$ possible code words.
- We can represent the encoder as follows:

  - **Source messages**: $\{X_1, X_2, \ldots, X_n\}$
  - **Code words**: $\{Y_1, Y_2, \ldots, Y_m\}$

  - $X_i \in \{a_1, a_2, \ldots, a_K\}$
  - $K^n$ possible messages
  - $n$ source letters in each source msg

  - $Y_i \in \{0, 1\}$
  - $2^m$ possible messages
  - $m$ total bits

- Example: English letters, would have $K = 26$ (alphabet size $K$), a “source message” consists of $n$ letters.
- We want to have a code word for every possible source message, must have what condition?

  $$M = 2^m \geq K^n \Rightarrow m \geq (\log K)n$$

  (5.1)
Towards AEP

- A question on rate: How many bits are used per source letter?
  
  \[ R = \text{rate} = \frac{\log M}{n} = \frac{m}{n} \geq \log K \text{ bits per source letter} \quad (5.1) \]

  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.

  \[ \text{Source messages} \rightarrow \text{garbage} \rightarrow \text{Code words} \]

- 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.

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 5.2.1 (AEP)**

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

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

(5.22)

**Proof.**

$$-\frac{1}{n} \log p(X_1, X_2, \ldots, X_n) = -\frac{1}{n} \log \prod_{i=1}^{n} p(X_i)$$

(5.23)

$$= -\frac{1}{n} \sum_{i} \log p(X_i) \xrightarrow{p} -E \log p(X)$$

(5.24)

$$= H(X)$$

(5.25)
**Typical Set**

**Definition 5.2.1 (Typical Set)**

The typical set $A_{\epsilon}^{(n)}$ 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

$$2^{-n(H(X)+\epsilon)} \leq p(x_1, x_2, \ldots, x_n) \leq 2^{-n(H(X)-\epsilon)}$$  \hspace{1cm} (5.22)

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

$$A_{\epsilon}^{(n)} = \left\{ (x_1, x_2, \ldots, x_n) : \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| < \epsilon \right\}$$  \hspace{1cm} (5.23)

- Typical set are those sequences with log probability within the range $-nH \pm n\epsilon$

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

Theorem 5.2.1 (Properties of $A^{(n)}_{\epsilon}$)

1. If $(x_1, x_2, \ldots, x_n) \in A^{(n)}_{\epsilon}$, then
   \[ H(X) - \epsilon \leq -\frac{1}{n} \log p(x_1, x_2, \ldots, x_n) \leq H(X) + \epsilon \] (5.22)

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 \( \approx 2^{nH} \).
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 $A_{\epsilon}^{(n)}$, 
  - and its complement, the non-typical sets $X^n \setminus A_{\epsilon}^{(n)} \triangleq A_{\epsilon}^{(n)c}$
- A partition, i.e., $A_{\epsilon}^{(n)} \cap A_{\epsilon}^{(n)c} = \emptyset$ and $A_{\epsilon}^{(n)} \cup A_{\epsilon}^{(n)c} = X^n$:

\[ X^n \text{ having } |X^n| = K^n \text{ elements} \]
Expected Length

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

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

(5.40)

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

(5.41)

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

(5.42)

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

(5.43)

\[
\leq n(H + \epsilon) + 2 + \epsilon n \log K + \epsilon 2
\]

(5.44)

\[
= n\left[H + \epsilon + \epsilon \log K + \frac{2}{n} + \frac{2\epsilon}{n}\right] = n(H + \epsilon')
\]

(5.45)
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 \( H(p) \).

- 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 \( p(\mathcal{X}^n) = 1 \).
Other high probable sets?

- We know that $p(X^n) = 1$.
- So $A^{(n)}_{\epsilon}$ is smaller, and $P(A^{(n)}_{\epsilon}) \approx 1$. 
We know that $p(X^n) = 1$.

So $A^{(n)}_{\epsilon}$ is smaller, and $P(A^{(n)}_{\epsilon}) \approx 1$.

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 $A^{(n)}_{\epsilon}$ essential (i.e., contribute significantly to the probability)?
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)?
- If so, maybe we can code for this still smaller set and achieve even better compression rate.
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?

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

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

(5.1)

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

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

$$p(B^{(n)}_\delta) \geq 1 - \delta \quad (5.1)$$

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

**Theorem 5.3.1**

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

$$p(B^{(n)}_\delta) > 1 - \delta,$$

then

$$\frac{1}{n} \log |B^{(n)}_\delta| > H - \delta' \text{ if } n \text{ is large enough} \quad (5.2)$$

$$\Rightarrow |B^{(n)}_\delta| > 2^{n(H-\delta')} \approx 2^{nH} \quad (5.3)$$
Other high probable sets?

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

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

(5.1)

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

**Theorem 5.3.1**

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

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

(5.2)

$$\Rightarrow |B^{(n)}_{\delta}| > 2^{n(H - \delta')} \approx 2^{nH}$$

(5.3)

- In other words, asymptotically $B^{(n)}_{\delta}$ is no smaller than $A^{(n)}_{\epsilon}$ and we are free to code for $A^{(n)}_{\epsilon}$.  

---

*Source Coding  | Types  | U. S. Coding*
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 $A_{\epsilon}^{(n)c}$. 
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 $A^{(n)c}_{\epsilon}$.
- 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 $A_\varepsilon(n)$ and one for the complement $A_\varepsilon(n)^c$.

- 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.

![Diagram of source messages and code words](image)
So, code is no longer one-to-one, and source sequences might map to the same code word.

\[ P = \text{probability of error} \]

We know

\[ p(A(n) \in \epsilon) > 1 - \epsilon \]  

(5.4)

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

\[ p(\text{error}) = p(A(n) \in \epsilon) \leq \epsilon \]  

(5.5)
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

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

(5.4)
Coding Strategy with errors

- So, code is no longer one-to-one, and source sequence might map to same code word.
- What is $P_e = \text{probability of error}$? We know
  \[ p(A^{(n)}_\epsilon) > 1 - \epsilon \]  
  (5.4)
- And error occurs when a sequence is not typical, so we can bound the error probability
  \[ p(\text{error}) = p(A^{(n)c}_\epsilon) \leq \epsilon \]  
  (5.5)
Coding Strategy with errors

- Recall typical: $\forall \epsilon > 0, \forall \delta > 0, \exists n_0 \text{ s.t. for } n > n_0,$
  \[
p\left\{ \left| \frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| < \epsilon \right\} > 1 - \delta \tag{5.6}
\]
Coding Strategy with errors

- Recall typical: \( \forall \epsilon > 0, \forall \delta > 0, \exists n_0 \text{ s.t. for } n > n_0, \)
  \[
p\left\{ \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| < \epsilon \right\} > 1 - \delta \tag{5.6}
\]

- Which is same as: \( \forall \epsilon > 0, \forall \delta > 0, \exists n_0 \text{ s.t. for } n > n_0, \)
  \[
p\left\{ \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| > \epsilon \right\} \leq \delta \tag{5.7}
\]
Coding Strategy with errors

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

  (5.6)

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

  (5.7)

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

Coding Strategy with errors

- Recall typical: \( \forall \epsilon > 0, \forall \delta > 0, \exists n_0 \text{ s.t. for } n > n_0, \)
  \[
  p\left\{ \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| < \epsilon \right\} > 1 - \delta
  \]
  (5.6)

- Which is same as: \( \forall \epsilon > 0, \forall \delta > 0, \exists n_0 \text{ s.t. for } n > n_0, \)
  \[
  p\left\{ \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| > \epsilon \right\} \leq \delta
  \]
  (5.7)

- 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 \( p(\text{error}) = p(A_{\epsilon}^{(n)c}) \leq \delta(n, \epsilon) \), or
  \[
  p(\text{error}) \to 0 \text{ as } n \to \infty
  \]
  (5.8)
Coding Strategy with errors

- Recall typical: \( \forall \epsilon > 0, \, \forall \delta > 0, \, \exists n_0 \text{ s.t. for } n > n_0, \)
  \[
  p\left\{ \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| < \epsilon \right\} > 1 - \delta \tag{5.6}
  \]

- Which is same as: \( \forall \epsilon > 0, \, \forall \delta > 0, \, \exists n_0 \text{ s.t. for } n > n_0, \)
  \[
  p\left\{ \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| > \epsilon \right\} \leq \delta \tag{5.7}
  \]

- 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 \( p(\text{error}) = p(A_{\epsilon}^{(n)c}) \leq \delta(n, \epsilon) \), or
  \[
  p(\text{error}) \to 0 \text{ as } n \to \infty \tag{5.8}
  \]

- 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

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

(5.9)

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

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

(5.9)

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
  \[
  2^{n(H - \alpha \epsilon)}
  \] (5.9)
  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
  \[ 2^{n(H - \alpha \epsilon)} \] 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.,
  \[ 2^{n(H - \alpha \epsilon)} 2^{-n(H - \epsilon)} = 2^{-n \epsilon (\alpha - 1)} \] (5.10)
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.,
  \[ 2^{n(H-\alpha\epsilon)}2^{-n(H-\epsilon)} = 2^{-n\epsilon(\alpha-1)} \]  
  \[ 2^{-n\epsilon(\alpha-1)} \]  
- 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 $n$. 

\[ \text{Prof. Jeff Bilmes} \]
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
  \[ 2^{n(H - \alpha \epsilon)} \]
  (5.9)
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.,
  \[ 2^{n(H - \alpha \epsilon)} 2^{-n(H - \epsilon)} = 2^{-n\epsilon(\alpha - 1)} \]
  (5.10)
- 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 $n \to \infty$. 
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.
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 you simultaneously decode) is long enough.
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 you simultaneously decode) is long enough.

- 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 you simultaneously decode) is long enough.

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 you simultaneously decode) is long enough.
- 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 $\epsilon' > 0$.
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 you simultaneously decode) is long enough.

- 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} (5.11)
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 you simultaneously decode) is long enough.

- 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

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

(5.11)

- In the second case, however, $P_e \to 0$ as $n \to \infty$. 

Prof. Jeff Bilmes
EE514a/Fall 2013/Information Theory I – Lecture 5 - Oct 9th, 2013
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).
- While the type with $k = n/2$ is smaller, it is not asymptotically smaller.
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.
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

$\mathcal{X} = (a_1, a_2, \ldots, a_D)$. 

\[
P_{x_{1:n}}(a) = \frac{n(a|_{x_{1:n}})}{n} \quad \text{for} \quad a \in \mathcal{X}.
\]
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.

\[
P_{x_{1:n}} \triangleq \left( \frac{n(a_1|x_{1:n})}{n}, \frac{n(a_2|x_{1:n})}{n}, \ldots, \frac{n(a_D|x_{1:n})}{n} \right) \tag{5.12}
\]

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.

$$P_{x_{1:n}} \triangleq \left( \frac{n(a_1 | x_{1:n})}{n}, \frac{n(a_2 | x_{1:n})}{n}, \ldots, \frac{n(a_D | x_{1:n})}{n} \right) \quad (5.12)$$

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.

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

(5.12)

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.

\[
P_{X_{1:n}} \triangleq \left( \frac{n(a_1|x_{1:n})}{n}, \frac{n(a_2|x_{1:n})}{n}, \ldots, \frac{n(a_D|x_{1:n})}{n} \right) \tag{5.12}
\]

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 $n$.
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 $\mathcal{X}$. 

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., $\mathcal{P}_n(\equiv \mathcal{P}_n(\mathcal{X}) \equiv \mathcal{P}_n(|\mathcal{X}|)) = \mathcal{P}_n(\mathcal{D})$
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 $\mathcal{X}$. 

\[ \text{E.x., } \mathcal{X} = \{0, 1\}, \text{ then } \mathcal{P}_n(X) = \{ \left( \begin{array}{c} 0 \frac{n}{n} \end{array} \right), \left( \begin{array}{c} 1 \frac{n}{n} \end{array} \right), \ldots, \left( \begin{array}{c} n \frac{0}{n} \end{array} \right) \} \] (5.13)
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

$$\mathcal{P}_n(\mathcal{X}) = \left\{ \left( \frac{0}{n}, \frac{n}{n} \right), \left( \frac{1}{n}, \frac{n-1}{n} \right), \ldots, \left( \frac{n}{n}, \frac{0}{n} \right) \right\} \quad (5.13)$$

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

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

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

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

(5.14)

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

Prof. Jeff Bilmes
Notational Summary

For sequences of length $n$, we have: $\emptyset$
For sequences of length \( n \), we have:

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

\[
P_{x_{1:n}} \triangleq \left( \frac{n(a_1 | x_{1:n})}{n}, \frac{n(a_2 | x_{1:n})}{n}, \ldots, \frac{n(a_D | x_{1:n})}{n} \right); \tag{5.15}
\]
Notational Summary

- For sequences of length $n$, we have: n
- 1) the type (or histogram) of a sample $x_{1:n}$
  \[
P_{x_{1:n}} \triangleq \left( \frac{n(a_1|x_{1:n})}{n}, \frac{n(a_2|x_{1:n})}{n}, \ldots, \frac{n(a_D|x_{1:n})}{n} \right);
\]  (5.15)
- 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 $x_{1:n}$

$$P_{x_{1:n}} \triangleq \left( \frac{n(a_1|x_{1:n})}{n}, \frac{n(a_2|x_{1:n})}{n}, \ldots, \frac{n(a_D|x_{1:n})}{n} \right); \quad (5.15)$$

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

Notational Summary

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

$$P_{x_{1:n}} \triangleq \left( \frac{n(a_1|x_{1:n})}{n}, \frac{n(a_2|x_{1:n})}{n}, \ldots, \frac{n(a_D|x_{1:n})}{n} \right); \quad (5.15)$$

  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 $P_{x_{1:5}} = (3, 5, 1, 5, 1)$ (5.16)

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., $T(P_{x_{1:5}}) = \{[1, 1, 1, 2, 3], [1, 1, 1, 3, 2], \ldots, [3, 2, 1, 1, 1]\}$ (5.17)

How many types? I.e., What is $|P_n|$?

Here, $|P_n| = 21$. (5.18)

Problem for you to think about. Turns out, in general, $|P_n| = (n + |X| - 1)^{|X| - 1}$ (5.18)
Example

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

Then

$$P_{x_{1:5}} = \left( \frac{3}{5}, \frac{1}{5}, \frac{1}{5} \right)$$ (5.16)

How many types? I.e., What is $|P_n|$?

Here, $|P_n| = 21$. Problem for you to think about. Turns out, in general,

$$|P_n| = \binom{n + |\mathcal{X}| - 1}{|\mathcal{X}| - 1}$$ (5.18)
Example

- Let $\mathcal{X} = \{1, 2, 3\}$ and $x_{1:5} = [1, 1, 3, 2, 1]$.
- Then
  \[ P_{x_{1:5}} = \left( \frac{3}{5}, \frac{1}{5}, \frac{1}{5} \right) \]  (5.16)
- 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.,
  \[ T(P_{x_{1:5}}) = \{ [1, 1, 1, 2, 3], [1, 1, 1, 3, 2], \ldots, [3, 2, 1, 1, 1] \} \]  (5.17)

\[ \forall x_{1:5} \in T(P_{y_{1:5}}), \quad P_{x_{1:5}} = P_{y_{1:5}} \]
Example

- Let $\mathcal{X} = \{1, 2, 3\}$ and $x_{1:5} = [1, 1, 3, 2, 1]$.
- Then
  \[
  P_{x_{1:5}} = \left( \frac{3}{5}, \frac{1}{5}, \frac{1}{5} \right) \tag{5.16}
  \]
  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.,
  \[
  T(P_{x_{1:5}}) = \{ [1, 1, 1, 2, 3], [1, 1, 1, 3, 2], \ldots, [3, 2, 1, 1, 1] \} \tag{5.17}
  \]
- 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

$$P_{x_{1:5}} = \left(\frac{3}{5}, \frac{1}{5}, \frac{1}{5}\right) \quad (5.16)$$

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.,

$$T(P_{x_{1:5}}) = \{[1, 1, 1, 2, 3], [1, 1, 1, 3, 2], \ldots, [3, 2, 1, 1, 1]\} \quad (5.17)$$

How may 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
  \[ P_{x_{1:5}} = \left( \frac{3}{5}, \frac{1}{5}, \frac{1}{5} \right) \]  
  (5.16)

- 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.,
  \[ T(P_{x_{1:5}}) = \{[1, 1, 1, 2, 3], [1, 1, 1, 3, 2], \ldots, [3, 2, 1, 1, 1]\} \]  
  (5.17)

- How many types? I.e., What is $|\mathcal{P}_n|$? Here, $|\mathcal{P}_n| = 21$.
- Problem for you to think about. Turns out, in general,
  \[ |\mathcal{P}_n| = \binom{n + |\mathcal{X}| - 1}{|\mathcal{X}| - 1} \]  
  (5.18)
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, $\bigcup_{P \in \mathcal{P}_n} T(P) = \mathcal{X}^n$.

$p_1 \neq p_2 \implies T(p_1) \cap T(p_2) = \emptyset$
Division of set of all sequences into type classes

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

\( \chi^n \): the set of all sequences of length n

A particular sequence \( \chi_{1:n} \)
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 5.4.1

\[ |\mathcal{P}_n| \leq (n + 1)|\mathcal{X}| \]  \hspace{1cm} (5.19)
Bound on number of type classes

Proposition 5.4.1

\[ |P_n| \leq (n + 1)^{|X|} \]  \hspace{1cm} (5.19)

Proof.

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

Proposition 5.4.1

\[ |P_n| \leq (n + 1)|\mathcal{X}| \quad (5.19) \]

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.
Proposition 5.4.1

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

(5.19)

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 5.4.1

\[ |P_n| \leq (n + 1)|\mathcal{X}| \quad (5.19) \]

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 5.4.1

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

(5.19)

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 5.4.1

\[ |\mathcal{P}_n| \leq (n + 1)|\mathcal{X}| \] (5.19)

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
  \[ P_{x_1:n}(a) = \frac{\mathbb{P}(a|x_{1:n})}{n} \text{ for all } a \in \mathcal{X}. \]
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 $P_{x_1:n}(a) = \frac{N(a|x_{1:n})}{n}$ for all $a \in \mathcal{X}$.
- $\mathcal{P}_n(\mathcal{X})$ (or just $\mathcal{P}_n$) is the set of types with denominator $n$.
- $T(P)$, for type $P \in \mathcal{P}_n$, is the set of all sequences with type $P$, i.e., $T(P) = \{x_{1:n} \in \mathcal{X}^n : P_{x_1:n} = P\}$.
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 $n$, $|\mathcal{X}|^n$. 
Theorem 5.4.2

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

**Theorem 5.4.2**

- 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.
Theorem 5.4.2

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

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

The probability of the sequence depends only on the type
Theorem 5.4.2

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 $Q$
Theorem 5.4.2

- 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

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

(5.20)
Theorem 5.4.2

- 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

\[
Q^n(x_1:n) = 2^{-n}[H(P_{x_1:n}) + D(P_{x_1:n} || Q)] \tag{5.20}
\]
Theorem 5.4.2

- 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

$$Q^n(x_1:n) = 2^{-n[H(P_{x1:n})+D(P_{x1:n}||Q)]} \quad (5.20)$$

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

**Theorem 5.4.2**

- 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

$$Q^n(x_{1:n}) = 2^{-n[H(P_{x_{1:n}}) + D(P_{x_{1:n}} || Q)]} \quad (5.20)$$

- 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 5.4.2

- 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

$$Q^n(x_1:n) = 2^{-n[H(P_{x_1:n}) + D(P_{x_1:n} || Q)]}$$ (5.20)

- 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.

\[ Q^n(x_1:n) \]

(5.24)
Probability depends only on the type

Proof.

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

\[
Q^n(x_1:n) = \prod_{i=1}^{n} Q(x_i) = \prod_{a \in \mathcal{X}} Q(a)^{n(a|_1:n)}
\] (5.21)

\[
2^{-n(\mathcal{D}(P|x_1:n)||Q) + H(P|x_1:n))}
\] (5.24)
Probability depends only on the type

Proof.

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

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

(5.21)

(5.24)
Probability depends only on the type

Proof.

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

(5.21)

(5.22)

(5.23)

(5.24)
Probability depends only on the type

**Proof.**

\[
Q^n(x_1:n) = \prod_{i=1}^{n} Q(x_i) = \prod_{a \in \mathcal{X}} Q(a)^{n(a|x_1:n)} = \prod_{a \in \mathcal{X}} 2^{\{nP_{x_1:n}(a) \log Q(a)\}}
\]

(5.22)

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

(5.24)
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} (5.21)

\[ = \prod_{a \in \mathcal{X}} Q(a)^{nP_{x_{1:n}}(a)} = \prod_{a \in \mathcal{X}} 2\left\{nP_{x_{1:n}}(a) \log Q(a)\right\} \]  \hspace{1cm} (5.22)

\[ = \prod_{a \in \mathcal{X}} 2^n \left\{ P_{x_{1:n}}(a) \log Q(a) - P_{x_{1:n}}(a) \log P_{x_{1:n}}(a) + P_{x_{1:n}}(a) \log P_{x_{1:n}}(a) \right\} = 0 \]

\[ = 2^n \sum_{a \in \mathcal{X}} \left( -P_{x_{1:n}}(a) \log \frac{P_{x_{1:n}}(a)}{Q(a)} + P_{x_{1:n}}(a) \log P_{x_{1:n}}(a) \right) \]  \hspace{1cm} (5.23)

\[ = 2^n \sum_{a \in \mathcal{X}} \left( -P_{x_{1:n}}(a) \log \frac{P_{x_{1:n}}(a)}{Q(a)} + P_{x_{1:n}}(a) \log P_{x_{1:n}}(a) \right) \]  \hspace{1cm} (5.24)
Probability depends only on the type

Proof.

\[ Q^n(x_{1:n}) = \prod_{i=1}^{n} Q(x_i) = \prod_{a \in \mathcal{X}} Q(a)^{n(a|x_{1:n})} \]  
\[ = \prod_{a \in \mathcal{X}} Q(a)^{nP_{x_{1:n}}(a)} = \prod_{a \in \mathcal{X}} 2^{nP_{x_{1:n}}(a) \log Q(a)} \]  
\[ = \prod_{a \in \mathcal{X}} 2^{nP_{x_{1:n}}(a) \log Q(a) - P_{x_{1:n}}(a) \log P_{x_{1:n}}(a) + P_{x_{1:n}}(a) \log P_{x_{1:n}}(a)} = 0 \]  
\[ = 2^n \sum_{a \in \mathcal{X}} (-P_{x_{1:n}}(a) \log \frac{P_{x_{1:n}}(a)}{Q(a)} + P_{x_{1:n}}(a) \log P_{x_{1:n}}(a)) \]  
\[ = 2^{-n} \left( D(P_{x_{1:n}}||Q) + H(P_{x_{1:n}}) \right) \rightarrow \bar{J} n H(Q) \]
Corollary: If \( Q \) is a rational distribution (i.e., a possible type) and if \( x_{1:n} \in T(Q) \), then

\[
Q^n(x_{1:n}) = 2^{-nH(Q)}
\]  

(5.25)
Corollary: If \( Q \) is a rational distribution (i.e., a possible type) and if \( x_{1:n} \in T(Q) \), then

\[
Q^n(x_{1:n}) = 2^{-nH(Q)}
\]  

(5.25)

a result already familiar to us.
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

$$Q^n(x_{1:n}) = 2^{-nH(Q)} \quad (5.25)$$

a result already familiar to us.

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

$$Q^n(x_{1:n}) = 2^{-nH(Q)}$$ (5.25)

a result already familiar to us.

What if $Q$ was irrational? Intuition: we could make $D(P_{x_1:n} \| Q)$ as small as we want, if we make $n$ large.
Size of type class

- 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

$$|T(P)| = \left( \begin{array}{c} nP(a_1) \\nP(a_2) \\n\vdots \\nP(a_n) \end{array} \right)^n$$  (5.26)
Size of type class

- We can easily express the size of the type class using multinominal 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

$$|T(P)| = \binom{n}{nP(a_1) \ nP(a_2) \ \cdots \ nP(a_n)}$$  (5.26)

- But we want bounds that are easier to mathematically manipulate than the multinominal.
Proposition 5.4.3

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

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

(5.27)

Proof.

$$1$$

(5.29)
Proposition 5.4.3

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

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

(5.27)

Proof.

\[
1 \geq P^n(T(P))
\]

(5.29)
Proposition 5.4.3

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

$$\frac{1}{(n + 1)|x|}2^{nH(P)} \leq |T(P)| \leq 2^{nH(P)} \quad (5.27)$$

Proof.

$$1 \geq P^n(T(P)) = \sum_{x_1:n \in T(P)} P^n(x_{1:n}) \quad (5.29)$$
Proposition 5.4.3

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

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

(5.27)

Proof.

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

(5.28)

(5.29)
Proposition 5.4.3

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

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

(5.27)

Proof.

\[
1 \geq P^n(T(P)) = \sum_{x_{1:n} \in T(P)} P^n(x_{1:n}) = \sum_{x_{1:n} \in T(P)} 2^{-nH(P)}
\]

(5.28)

\[
= |T(P)| 2^{-nH(P)}
\]

(5.29)
Proposition 5.4.3

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

$$\frac{1}{(n + 1)|\mathcal{X}|}2^{nH(P)} \leq |T(P)| \leq 2^{nH(P)}$$  \hspace{1cm} (5.27)

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} (5.28)

$$= |T(P)|2^{-nH(P)}$$ \hspace{1cm} (5.29)

Note that in the sums, $P_{x_1:n} = P$.  \hspace{1cm} \square
Size of type class

Proposition 5.4.3

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

$$\frac{1}{(n + 1)|x|}2^{nH(P)} \leq |T(P)| \leq 2^{nH(P)} \quad (5.27)$$

Proof.

$$1 \geq P^n(T(P)) = \sum_{x_1:n \in T(P)} P^n(x_1:n) = \sum_{x_1:n \in T(P)} 2^{-nH(P)} \quad (5.28)$$

$$= |T(P)|2^{-nH(P)} \quad (5.29)$$

Note that in the sums, $P_{x_1:n} = P$. This gives the upper bound.
Proposition 5.4.3

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

$$\frac{1}{(n+1)|x|} 2^{nH(P)} \leq |T(P)| \leq 2^{nH(P)} \quad (5.27)$$

Proof.

$$1 \geq P^n(T(P)) = \sum_{x_{1:n} \in T(P)} P^n(x_{1:n}) = \sum_{x_{1:n} \in T(P)} 2^{-nH(P)} \quad (5.28)$$

$$= |T(P)| 2^{-nH(P)} \quad (5.29)$$

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 . . .
Type class with highest probability

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 $m! \geq n^m$ since if $m > n$, then $m! = m(m-1)...(n+1) \geq n^m$, and if $m < n$, then $m! = 1(n-1)...(m+1) \geq n^{m-n}$, and if $m = n$ then obvious.
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
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 5.4.4**

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} (5.30)
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 5.4.4**

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

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

(5.30)

- 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

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

(5.30)

**Lemma 5.4.4**

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

- Note: Non-negative integers \( m \) and \( n \), then \( \frac{m!}{n!} \geq n^{m-n} \) since if \( m > n \), then \( \frac{m!}{n!} = m(m-1)\ldots(n+1) \geq n^{m-n} \),
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 5.4.4**

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

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

(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 $\frac{m!}{n!} = \frac{1}{n(n-1) \ldots (m+1)} \geq \frac{1}{n^{n-m}}$)
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 5.4.4**

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

$$P^n(T(P)) \geq P^n(T(\hat{P})), \ \forall \hat{P} \in \mathcal{P}_n$$  \hspace{1cm} (5.30)

- 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.
Proof of Lemma 5.4.4.

\[
\frac{P^n(T(P))}{P^n(T(\hat{P}))}
\]
Type class with highest probability

Proof of Lemma 5.4.4.

\[
\frac{P^n(T(P))}{P^n(T(\hat{P}))} = \frac{|T(P)| \prod_{a \in \mathcal{X}} P(a)^n P(a)}{|T(\hat{P})| \prod_{a \in \mathcal{X}} P(a)^n \hat{P}(a)}
\] (5.31)

\[
\geq \prod_{a \in \mathcal{X}} \frac{n P(a)}{n \hat{P}(a)}
\] (5.32)

\[
= \prod_{a \in \mathcal{X}} \frac{n \hat{P}(a) - P(a)}{n P(a) - \hat{P}(a)}
\] (5.33)

\[
= \prod_{a \in \mathcal{X}} \frac{\sum_{a \in \mathcal{X}} \hat{P}(a) - \sum_{a \in \mathcal{X}} P(a)}{\sum_{a \in \mathcal{X}} P(a) - \hat{P}(a)}
\] (5.35)

\[
= \frac{1}{n}
\] (5.36)
Type class with highest probability

Proof of Lemma 5.4.4.

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

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

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

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

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

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

Thus,

\[P^n(T(P)) \geq P^n(T(\hat{P}))
\]
Type class with highest probability

Proof of Lemma 5.4.4.

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

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

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

(5.31)

(5.32)

(5.33)

(5.34)

(5.35)

(5.36)
Proof of Lemma 5.4.4.

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

(5.31)

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

(5.32)

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

(5.33)

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

(5.34)

(5.36)
Type class with highest probability

Proof of Lemma 5.4.4.

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

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

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

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

\[
= \prod_{a \in \mathcal{X}} n^{n(\hat{P}(a) - P(a))}
\]
Type class with highest probability

Proof of Lemma 5.4.4.

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

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

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

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

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

(5.31) \hspace{1cm} (5.32) \hspace{1cm} (5.33) \hspace{1cm} (5.34) \hspace{1cm} (5.35) \hspace{1cm} (5.36)
Type class with highest probability

Proof of Lemma 5.4.4.

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

(5.31)

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

(5.32)

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

(5.33)

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

(5.34)

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

(5.35)

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

(5.36)
Type class with highest probability

Proof of Lemma 5.4.4.

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

(5.31)

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

(5.32)

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

(5.33)

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

(5.34)

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

(5.35)

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

(5.36)
Type class with highest probability

Proof of Lemma 5.4.4.

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

(5.31)

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

(5.32)

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

(5.33)

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

(5.34)

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

(5.35)

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

(5.36)

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

Proof of lowerbound of theorem 5.4.3.

1

\[
1 = \sum_{Q \in \mathcal{P}_n} P_n(T(Q)) \leq \sum_{Q \in \mathcal{P}_n} \max_{R \in \mathcal{P}_n} P_n(T(R)) \tag{5.37}
\]

\[
1 = \sum_{Q \in \mathcal{P}_n} P_n(T(P)) \leq (n+1)|X|P_n(T(P)) \tag{5.38}
\]

\[
1 = (n+1)|X|T(P) \leq 2^{nH(P)} \tag{5.39}
\]

\[
1 = (n+1)|X||T(P)|2^{-nH(P)} \tag{5.40}
\]

which gives us our result that.

\[
|T(P)| \geq 1 \frac{(n+1)|X|}{2^n} \tag{5.42}
\]
Proof of lower bound

Proof of lower bound of theorem 5.4.3.

\[
1 = \sum_{Q \in \mathcal{P}_n} P^n(T(Q))
\]
Proof of lower bound of theorem 5.4.3.

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)) (5.37)

(5.41)
Proof of lower bound

Proof of lower bound of theorem 5.4.3.

\[ 1 = \sum_{Q \in \mathcal{P}_n} P^n(T(Q)) \leq \sum_{Q \in \mathcal{P}_n} \max_{R \in \mathcal{P}_n} P^n(T(R)) \]  \hspace{1cm} (5.37) 

\[ = \sum_{Q \in \mathcal{P}_n} P^n(T(P)) \]  \hspace{1cm} (5.41)
Proof of lower bound of theorem 5.4.3.

\[
1 = \sum_{Q \in \mathcal{P}_n} P^n(T(Q)) \leq \sum_{Q \in \mathcal{P}_n} \max_{R \in \mathcal{P}_n} P^n(T(R)) \quad (5.37)
\]

\[
= \sum_{Q \in \mathcal{P}_n} P^n(T(P)) \leq (n + 1)^{|X|} P^n(T(P)) \quad (5.38)
\]

\[
= (n + 1)^{|X|} \sum_{x_1: n} P^n(T(P)) \quad (5.41)
\]
Proof of lower bound

Proof of lowerbound of theorem 5.4.3.

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

\[(5.37)\]

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

\[(5.38)\]

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

\[(5.41)\]
Proof of lower bound

Proof of lower bound of theorem 5.4.3.

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

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

\[ \geq (n + 1) |\mathcal{X}| 2^{-nH(P)} \]
Proof of lower bound of theorem 5.4.3.

\[ 1 = \sum_{Q \in \mathcal{P}_n} P^n(T(Q)) \leq \sum_{Q \in \mathcal{P}_n} \max_{R \in \mathcal{P}_n} P^n(T(R)) \]  \hspace{1cm} (5.37)

\[ = \sum_{Q \in \mathcal{P}_n} P^n(T(P)) \leq (n + 1)|\mathcal{X}| P^n(T(P)) \]  \hspace{1cm} (5.38)

\[ = (n + 1)|\mathcal{X}| \sum_{x_1:n \in T(P)} P^n(x_1:n) = (n + 1)|\mathcal{X}| \sum_{x_1:n \in T(P)} 2^{-nH(P)} \]  \hspace{1cm} (5.39)

\[ = (n + 1)|\mathcal{X}| |T(P)| 2^{-nH(P)} \]  \hspace{1cm} (5.40)

\[ = (n + 1)|\mathcal{X}| |T(P)| 2^{-nH(P)} \]  \hspace{1cm} (5.41)
Proof of lower bound of theorem 5.4.3.

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

(5.37)

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

(5.38)

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

(5.39)

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

(5.40)

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

(5.41)

which gives us our result that.

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

(5.42)
For binary case, \( \mathcal{X} = \{0, 1\} \) we have the interesting bound

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

(5.43)
For binary case, \( \mathcal{X} = \{0, 1\} \) we have the interesting bound

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

(5.43)

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

$$\frac{1}{(n+1)^2} 2^{nH\left(\frac{k}{n}\right)} \leq \binom{n}{k} \leq 2^{nH\left(\frac{k}{n}\right)} \quad (5.43)$$

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:

$$\frac{1}{(n+1)} 2^{nH\left(\frac{k}{n}\right)} \leq \binom{n}{k} \quad (5.44)$$

(exercise: show this)
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: $a_n \equiv b_n$ if $\lim_{n \to \infty} \frac{1}{n} \log \frac{a_n}{b_n} = 0$. 

Theorem 5.4.5

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

$$Q_n(T(P)) = 2^{-nD(P||Q)}.$$

Specifically,

$$\frac{1}{(n+1)|X|} 2^{-nD(P||Q)} \leq Q_n(T(P)) \leq 2^{-nD(P||Q)} \quad (5.45)$$

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. $Q_n(T(P)) = \sum_{x_1:n \in T(P)} Q_n(x_1:n) = \sum_{x_1:n \in T(P)} 2^{-n\left(D(P||Q) + H(P)\right)} \quad (5.46) = |T(P)| 2^{-n\left(D(P||Q) + H(P)\right)} \quad (5.47)$

and then use $\frac{1}{(n+1)|X|} 2^{-nH(P)} \leq |T(P)| \leq 2^{nH(P)}$
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 5.4.5**

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,

\[
\frac{1}{(n+1)|X|} 2^{-nD(P||Q)} \leq Q^n(T(P)) \leq 2^{-nD(P||Q)} \tag{5.45}
\]
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 5.4.5**

*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,*

\[ \frac{1}{(n+1)|\mathcal{X}|} 2^{-nD(P||Q)} \leq Q^n(T(P)) \leq 2^{-nD(P||Q)} \quad (5.45) \]

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

**Theorem 5.4.5**

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

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

(5.45)

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.**

\[
Q^n(T(P))
\]

(5.47)
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 5.4.5**

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

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

(5.45)

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.**

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

(5.47)
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 5.4.5**

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

\[
\frac{1}{(n+1)|X|} 2^{-nD(P||Q)} \leq Q^n(T(P)) \leq 2^{-nD(P||Q)}
\] (5.45)

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.**

\[
Q^n(T(P)) = \sum_{x_1:n \in T(P)} Q^n(x_1:n) = \sum_{x_1:n \in T(P)} 2^{-n(D(P||Q) + H(P))}
\] (5.46)

(5.47)
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 5.4.5**

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

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

(5.45)

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.**

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

(5.46)

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

(5.47)
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 5.4.5**

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

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

(5.45)

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.**

\[
Q^n(T(P)) = \sum_{x_1:n \in T(P)} Q^n(x_1:n) = \sum_{x_1:n \in T(P)} 2^{-n(D(P||Q)+H(P))}
\]  

(5.46)

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

(5.47)

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 $n$

$$|\mathcal{P}_n| \leq (n + 1)|\mathcal{X}|$$  (5.48)
Summary of basic theorems

- Number of types with denominator $n$
  \[ |\mathcal{P}_n| \leq (n + 1)|\mathcal{X}| \quad (5.48) \]

- $p(x_1:n)$ depends only on the type (prob. indep. of sample given type)
  \[ Q^n(x_1:n) = 2^{-n[H(P_{x_1:n}) + D(P_{x_1:n}||Q)]} \quad (5.49) \]
Summary of basic theorems

- Number of types with denominator $n$

\[ |\mathcal{P}_n| \leq (n + 1)|\mathcal{X}| \quad (5.48) \]

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

\[ Q^n(x_1:n) = 2^{-n}[H(P_{x_1:n}) + D(P_{x_1:n} || Q)] \quad (5.49) \]

- Size of the type class

\[ |T(P)| = 2^{nH(P)} \quad (5.50) \]
Summary of basic theorems

- Number of types with denominator $n$
  \[ |\mathcal{P}_n| \leq (n + 1)|\mathcal{X}| \]  
  (5.48)

- $p(x_1:n)$ depends only on the type (prob. indep. of sample given type)
  \[ Q^n(x_1:n) = 2^{-n}[H(P_{x_1:n}) + D(P_{x_1:n}||Q)] \]  
  (5.49)

- Size of the type class
  \[ |T(P)| \doteq 2^{nH(P)} \]  
  (5.50)

- Probability of a type class
  \[ Q^n(T(P)) \doteq 2^{-nD(P||Q)} \]  
  (5.51)
Types with the most probability

Q: Which types will have the most probability?

Clearly, the ones that are closest to the true distribution. The property

$$Q_n(T(P)) = 2^n - nD(P || Q)$$

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 5.4.6 (typical set of sequences)

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

$$T_\epsilon Q = \{x_1: n: D(P_{x_1^n} || Q) \leq \epsilon\}$$ (5.52)

Intuitively, these are sequences that come from types that are $$\epsilon$$-close to $$Q$$ in the KL-sense.
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 $Q^n(T(P)) = 2^{-nD(P||Q)}$ says that the ones that are farther away will have exponentially smaller probability than the others, as $n \to \infty$. 

Intuitively, these are sequences that come from types that are $\epsilon$-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.
- 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,
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,
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 $Q^n(T(P)) \overset{d}{=} 2^{-nD(P\|Q)}$ 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 5.4.6 (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

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

(5.52)
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 5.4.6 (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

$$T_Q^\epsilon = \{x_1:n : D(P_{x1:n}||Q) \leq \epsilon\}$$

Intuitively, these are sequences that come from types that are $\epsilon$-close to $Q$ in the KL-sense.
Theorem 5.4.7

Let $X_1, X_2, \ldots, X_n$ be i.i.d. $\forall i, x_i \sim Q(x)$. Then the probability of the complement of the typical set $\bar{T}_Q^\epsilon$ has expression:

$$Q(\bar{T}_Q^\epsilon) = Q(\{x_1:n : D(P_{x_1:n} || Q) > \epsilon\}) \leq 2^{-n(\epsilon - |X|\log(n+1)/n)} \quad (5.53)$$

and therefore,

$$D(P_{X_1:n} || Q) \xrightarrow{p} 0 \text{ as } n \to \infty \quad (5.54)$$

Intuitively, this means that types that are more than $\epsilon$ away from $Q$ have decreasing probability.
Probability of typical set

**Theorem 5.4.7**

Let \( X_1, X_2, \ldots, X_n \) be i.i.d. \( \forall i, x_i \sim Q(x) \). Then the probability of the complement of the typical set \( \bar{T}_Q^\epsilon \) has expression:

\[
Q(\bar{T}_Q^\epsilon) = Q(\{x_1:n : D(P_{x_1:n} \mid\mid Q) > \epsilon\}) \leq 2^{-n(\epsilon - |X| \log(n+1)/n)} \tag{5.53}
\]

and therefore,

\[
D(P_{X_1:n} \mid\mid Q) \xrightarrow{p} 0 \text{ as } n \to \infty \tag{5.54}
\]

- Intuitively, this means that types that are more than \( \epsilon \) away from \( Q \) have decreasing probability.
- Moreover, the typical set, which ends up for large \( n \) being the only thing that occurs without vanishingly small probability, is such that the KL divergence gets between the type and \( Q \) quickly gets arbitrarily small.
Probability of typical set

Proof of Theorem 5.4.7.

\[ Q(\bar{T}_Q^\epsilon) \]

(5.58)
Probability of typical set

Proof of Theorem 5.4.7.

\[ Q(\widetilde{T}_Q^\epsilon) = 1 - Q^n(T_Q^\epsilon) \]

(5.58)
Proof of Theorem 5.4.7.

\[ Q(\bar{T}_Q^\epsilon) = 1 - Q^n(T_Q^\epsilon) = \sum_{P \in \mathcal{P}_n : D(P||Q) > \epsilon} Q^n(T(P)) \]  \hspace{1cm} (5.55)

\[ \leq \sum_{P \in \mathcal{P}_n : D(P||Q) > \epsilon} D(P||Q) > \epsilon \]  \hspace{1cm} (5.56)

\[ \leq (n+1)|X|^2 - n \epsilon = 2 - n \epsilon (\epsilon - |X| \log(n+1) / n) \]  \hspace{1cm} (5.57)

\[ (5.58) \]
Proof of Theorem 5.4.7.

\[ Q(\bar{T}_Q^\epsilon) = 1 - Q^n(T_Q^\epsilon) = \sum_{P \in \mathcal{P}_n : D(P \| Q) > \epsilon} Q^n(T(P)) \] (5.55)

\[ \leq \sum_{P \in \mathcal{P}_n : D(P \| Q) > \epsilon} 2^{-nD(P \| Q)} \] (5.56)

\[ \leq (n + 1)|X|^2 - n\epsilon \] (5.58)
Proof of Theorem 5.4.7.

\[ Q(\tilde{T}_Q^\epsilon) = 1 - Q^n(T_Q^\epsilon) = \sum_{P \in \mathcal{P}_n : D(P || Q) > \epsilon} Q^n(T(P)) \]  

(5.55)

\[ \leq \sum_{P \in \mathcal{P}_n : D(P || Q) > \epsilon} 2^{-nD(P || Q)} \]  

(5.56)

\[ \leq \sum_{P \in \mathcal{P}_n : D(P || Q) > \epsilon} 2^{-n\epsilon} \]  

(5.57)

\[ \leq \left( \frac{n+1}{n} \right) |X|^2 - n\epsilon \log\left( \frac{n+1}{n} \right) \]  

(5.58)
### Proof of Theorem 5.4.7.

\[
Q(T_Q^\epsilon) = 1 - Q^n(T_Q^\epsilon) = \sum_{P \in P_n : D(P \| Q) > \epsilon} Q^n(T(P)) = 1 - \sum_{P \in P_n} Q^n(T(P))
\]  

(5.55)

\[
\leq \sum_{P \in P_n : D(P \| Q) > \epsilon} 2^{-nD(P \| Q)} \leq \sum_{P \in P_n : D(P \| Q) > \epsilon} 2^{-n\epsilon}
\]  

(5.56)

\[
\leq (n + 1)|\mathcal{X}|2^{-n\epsilon}
\]  

(5.57)

\[
\leq (n + 1)|\mathcal{X}|2^{-n\epsilon}
\]  

(5.58)
Proof of Theorem 5.4.7.

\[
Q(\bar{T}^\epsilon_Q) = 1 - Q^n(T_Q^\epsilon) = \sum_{P \in \mathcal{P}_n : D(P \| Q) > \epsilon} Q^n(T(P)) \quad (5.55)
\]

\[
\leq \sum_{P \in \mathcal{P}_n : D(P \| Q) > \epsilon} 2^{-n D(P \| Q)} \quad (5.56)
\]

\[
\leq \sum_{P \in \mathcal{P}_n : D(P \| Q) > \epsilon} 2^{-n \epsilon} \quad (5.57)
\]

\[
\leq (n + 1)^{|\mathcal{X}|} 2^{-n \epsilon} = 2^{-n \left( \epsilon - |\mathcal{X}| \frac{\log(n+1)}{n} \right)} \quad (5.58)
\]
Proof of Theorem 5.4.7.

\[ Q(\bar{T}_Q^\epsilon) = 1 - Q^n(T_Q^\epsilon) = \sum_{P \in \mathcal{P}_n : D(P||Q) > \epsilon} Q^n(T(P)) \] (5.55)

\[ \leq \sum_{P \in \mathcal{P}_n : D(P||Q) > \epsilon} 2^{-nD(P||Q)} \] (5.56)

\[ \leq \sum_{P \in \mathcal{P}_n : D(P||Q) > \epsilon} 2^{-n\epsilon} \] (5.57)

\[ \leq (n + 1)|\mathcal{X}| 2^{-n\epsilon} = 2^{-n\left(\epsilon - |\mathcal{X}| \log(n+1) \right)} \] (5.58)

and the r.h.s. \( \to 0 \) as \( n \to \infty \),
Proof of Theorem 5.4.7.

\[ Q(\bar{T}^\epsilon_Q) = 1 - Q^n(T^\epsilon_Q) = \sum_{P \in \mathcal{P}_n : D(P||Q) > \epsilon} Q^n(T(P)) \]  \hspace{1cm} (5.55)

\[ \leq \sum_{P \in \mathcal{P}_n : D(P||Q) > \epsilon} 2^{-nD(P||Q)} \]  \hspace{1cm} (5.56)

\[ \leq \sum_{P \in \mathcal{P}_n : D(P||Q) > \epsilon} 2^{-n\epsilon} \]  \hspace{1cm} (5.57)

\[ \leq (n + 1)|\mathcal{X}|2^{-n\epsilon} = 2^{-n\left(\epsilon - |\mathcal{X}|\log(n+1)\right)} \]  \hspace{1cm} (5.58)

and the r.h.s. \(\rightarrow 0\) as \(n \rightarrow \infty\), and thus the probability of the typical set \(\rightarrow 1\) as \(n \rightarrow \infty\).
Also, $D(P_{X_1:n} \| Q) \to 0$ with probability 1.

Note this is very much like the AEP we saw before.
Also, $D(P_{X_1:n} \parallel Q) \to 0$ with probability 1.

If not, i.e., if $p(D(P_{X_1:n} \parallel Q) \to \gamma) > 0$ with $\gamma > 0$, then, say $T_{Q}^{\gamma_1 c}$ with $\gamma_1 = \gamma/2$ would not converge to 0 with $n \to \infty$. 
Also, $D(P_{X_1:n} \parallel Q) \to 0$ with probability 1.

If not, i.e., if $p(D(P_{X_1:n} \parallel Q) \to \gamma) > 0$ with $\gamma > 0$, then, say $T_{Q}^{\gamma_1 c}$ with $\gamma_1 = \gamma/2$ would not converge to 0 with $n \to \infty$.

Note this is very much like the AEP we saw before.
Also, $D(P_{X_1:n} \| Q) \to 0$ with probability 1.

If not, i.e., if $p(D(P_{X_1:n} \| Q) \to \gamma) > 0$ with $\gamma > 0$, then, say $T_{Q}^{\gamma_1 c}$ with $\gamma_1 = \gamma/2$ would not converge to 0 with $n \to \infty$.

Note this is very much like the AEP we saw before.

the type of the sequence gets closer and closer to $Q$ as $n$ grows.
How often does an atypical event occur

- Since $p(T^c_Q) \leq 2^{-n \left( \epsilon + |X| \log \frac{n+1}{n} \right)}$ is an exponentially decreasing sequence in $n$, it is summable,
How often does an atypical event occur

- Since \( p(\mathcal{T}_Q^\epsilon) \leq 2^{-n\left(\epsilon + |\mathcal{X}| \log \frac{n+1}{n}\right)} \) is an exponentially decreasing sequence in \( n \), it is summable,

- and in fact

\[
\sum_{n=1}^{\infty} p(D(P_{x_1:n} \| Q) > \epsilon) = E \left[ \sum_{n=1}^{\infty} 1\{D(P_{X_1:n} \| Q) > \epsilon\} \right] \quad (5.59)
\]
How often does an atypical event occur

- Since \( p(\bar{T}^\epsilon_Q) \leq 2^{-n\left(\epsilon + |\mathcal{X}| \log \left(\frac{n+1}{n}\right)\right)} \) is an exponentially decreasing sequence in \( n \), it is summable,
- and in fact

\[
\infty > \sum_{n=1}^{\infty} p(D(P_{X_1:n} \| Q) > \epsilon) = E \left[ \sum_{n=1}^{\infty} 1\{D(P_{X_1:n} \| Q) > \epsilon\} \right] \quad (5.59)
\]

- So expected number of times the event \( D(P_{X_1:n} \| Q) > \epsilon \) occurs is finite, out of an infinite set of possible times.
How often does an atypical event occur

- Since \( p(\bar{T}^\epsilon_Q) \leq 2^{-n\left(\epsilon + |X| \log \left(\frac{n+1}{n}\right)\right)} \) is an exponentially decreasing sequence in \( n \), it is summable,
- and in fact

\[
\infty > \sum_{n=1}^{\infty} p(D(P_{X_{1:n}} \parallel Q) > \epsilon) = E \left[ \sum_{n=1}^{\infty} 1\{D(P_{X_{1:n}} \parallel Q) > \epsilon\} \right] \tag{5.59}
\]

- So expected number of times the event \( D(P_{X_{1:n}} \parallel Q) > \epsilon \) occurs is finite, out of an infinite set of possible times.
- This has probability 0 by the Borel-Cantelli lemma (see the book Billingsly, Probability & Measure, page 59 for more details).
Universal Source Coding

- If we know $p(x)$, then we will be able to develop a code to compress sources generated by $p(x)$. Huffman, Lempel-Ziv, etc. are codes that, as we will soon see, do that.
If we know $p(x)$, then we will be able to develop a code to compress sources generated by $p(x)$. Huffman, Lempel-Ziv, etc. are codes that, as we will soon see, do that.

What if we don't know $p(x)$?
Universal Source Coding

- If we know $p(x)$, then we will be able to develop a code to compress sources generated by $p(x)$. Huffman, Lempel-Ziv, etc. are codes that, as we will soon see, do that.
- What if we don't know $p(x)$?
- Q: do there exist codes that can compress without knowing $p(x)$ and that do so down to the entropy limit?
Universal Source Coding

- If we know $p(x)$, then we will be able to develop a code to compress sources generated by $p(x)$. Huffman, Lempel-Ziv, etc. are codes that, as we will soon see, do that.
- What if we don’t know $p(x)$?
- Q: do there exist codes that can compress without knowing $p(x)$ and that do so down to the entropy limit?
- Q: can we compress down to the rate $R$ (in units of bits per source symbol) if $R > H(Q)$? (this is Shannon’s source coding theorem)
Universal Source Coding

- If we know $p(x)$, then we will be able to develop a code to compress sources generated by $p(x)$. Huffman, Lempel-Ziv, etc. are codes that, as we will soon see, do that.
- What if we don't know $p(x)$?
- Q: do there exist codes that can compress without knowing $p(x)$ and that do so down to the entropy limit?
- Q: can we compress down to the rate $R$ (in units of bits per source symbol) if $R > H(Q)$? (this is Shannon’s source coding theorem)
- What happens if $R < H(Q)$? (this is the converse of Shannon’s source coding theorem)
Universal Source Coding

- If we know $p(x)$, then we will be able to develop a code to compress sources generated by $p(x)$. Huffman, Lempel-Ziv, etc. are codes that, as we will soon see, do that.
- What if we don't know $p(x)$?
- Q: do there exist codes that can compress without knowing $p(x)$ and that do so down to the entropy limit?
- Q: can we compress down to the rate $R$ (in units of bits per source symbol) if $R > H(Q)$? (this is Shannon’s source coding theorem)
- What happens if $R < H(Q)$? (this is the converse of Shannon’s source coding theorem)
- We’ll formally prove this theorem using the method of types.
Universal Source Coding: intuitive idea from AEP

- Basic idea is similar to the typical set $A_c^{(n)}$ we’ve already seen: when $n$ is long enough, the only sequences that occur (with non-vanishingly small probability) will be typical.
Universal Source Coding: intuitive idea from AEP

- Basic idea is similar to the typical set $A_c^{(n)}$ we’ve already seen: when $n$ is long enough, the only sequences that occur (with non-vanishingly small probability) will be typical.

- If we encounter such a sequence, it “must” be typical since the only things that occur are typical.
Universal Source Coding: intuitive idea from AEP

- Basic idea is similar to the typical set $A_c^{(n)}$ we’ve already seen: when $n$ is long enough, the only sequences that occur (with non-vanishingly small probability) will be typical.
- If we encounter such a sequence, it “must” be typical since the only things that occur are typical.
- Thus, we only encode the things we see, and we count them along the way.
Universal Source Coding: intuitive idea from AEP

- Basic idea is similar to the typical set $A_{\epsilon}^{(n)}$ we’ve already seen: when $n$ is long enough, the only sequences that occur (with non-vanishingly small probability) will be typical.
- If we encounter such a sequence, it “must” be typical since the only things that occur are typical.
- Thus, we only encode the things we see, and we count them along the way.
- In the end, we’ll need at most $|A_{\epsilon}^{(n)}|$ code words for which we can index with $nH$ bits.
Universal Source Coding: intuitive idea from AEP

- Basic idea is similar to the typical set $A_{\epsilon}^{(n)}$ we’ve already seen: when $n$ is long enough, the only sequences that occur (with non-vanishingly small probability) will be typical.

- If we encounter such a sequence, it “must” be typical since the only things that occur are typical.

- Thus, we only encode the things we see, and we count them along the way.

- In the end, we’ll need at most $|A_{\epsilon}^{(n)}|$ code words for which we can index with $nH$ bits.

- We want to formalize Shannon’s theorem and its converse using the method of types.
Universal Source Coding: intuitive idea from types

- There are $2^{nH(P)}$ sequences of type $P$. 

Universal Source Coding: intuitive idea from types

- There are $2^{nH(P)}$ sequences of type $P$.
- Thus, we use $nH(P)$ bits to represent such sequences.
Universal Source Coding: intuitive idea from types

- There are $2^{nH(P)}$ sequences of type $P$.
- Thus, we use $nH(P)$ bits to represent such sequences.
- If $R > H(P)$, we are easily able to use $nR$ bits to represent such sequences.
Universal Source Coding: intuitive idea from types

- There are $2^{nH(P)}$ sequences of type $P$.
- Thus, we use $nH(P)$ bits to represent such sequences.
- If $R > H(P)$, we are easily able to use $nR$ bits to represent such sequences.
- As $n$ gets big, only the types $P$ that are "close" to $Q$ will occur.
Universal Source Coding: intuitive idea from types

- There are $2^{nH(P)}$ sequences of type $P$.
- Thus, we use $nH(P)$ bits to represent such sequences.
- If $R > H(P)$, we are easily able to use $nR$ bits to represent such sequences.
- As $n$ gets big, only the types $P$ that are “close” to $Q$ will occur.
- There are an exponential (in $n$) number of sequences.
Universal Source Coding: intuitive idea from types

- There are $2^{nH(P)}$ sequences of type $P$.
- Thus, we use $nH(P)$ bits to represent such sequences.
- If $R > H(P)$, we are easily able to use $nR$ bits to represent such sequences.
- As $n$ gets big, only the types $P$ that are “close” to $Q$ will occur.
- There are an exponential (in $n$) number of sequences.
- There are only a polynomial (in $n$) number of types.
Universal Source Coding: intuitive idea from types

- There are $2^{nH(P)}$ sequences of type $P$.
- Thus, we use $nH(P)$ bits to represent such sequences.
- If $R > H(P)$, we are easily able to use $nR$ bits to represent such sequences.
- As $n$ gets big, only the types $P$ that are “close” to $Q$ will occur.
- There are an exponential (in $n$) number of sequences.
- There are only a polynomial (in $n$) number of types.
- Thus, one type must eventually get “all” of the probability.
Universal Source Coding: intuitive idea from types

- There are $2^{nH(P)}$ sequences of type $P$.
- Thus, we use $nH(P)$ bits to represent such sequences.
- If $R > H(P)$, we are easily able to use $nR$ bits to represent such sequences.
- As $n$ gets big, only the types $P$ that are “close” to $Q$ will occur.
- There are an exponential (in $n$) number of sequences.
- There are only a polynomial (in $n$) number of types.
- Thus, one type must eventually get “all” of the probability.
- If $P \approx Q$, and $H(Q) < R$, then all types that actually “occur” can be represented in $R$ bits per source symbol.
Universal Source Coding: intuitive idea from types

- There are $2^{nH(P)}$ sequences of type $P$.
- Thus, we use $nH(P)$ bits to represent such sequences.
- If $R > H(P)$, we are easily able to use $nR$ bits to represent such sequences.
- As $n$ gets big, only the types $P$ that are “close” to $Q$ will occur.
- There are an exponential (in $n$) number of sequences.
- There are only a polynomial (in $n$) number of types.
- Thus, one type must eventually get “all” of the probability.
- If $P \approx Q$, and $H(Q) < R$, then all types that actually “occur” can be represented in $R$ bits per source symbol.
- If $P \approx Q$, and $H(Q) > R$, then types that occur can not be represented in $R$ bits per source symbol.
Recall from earlier our \( x \) to \( y \) encoder setup.

Source messages:
\[ \{X_1, X_2, \ldots, X_n\} \]
- \( X_i \in \{a_1, a_2, \ldots, a_K\} \)
- \( K^n \) possible messages
- \( n \) source letters in each source msg

Code words:
\[ \{Y_1, Y_2, \ldots, Y_m\} \]
- \( Y_i \in \{0, 1\} \)
- \( 2^m \) possible messages
- \( m \) total bits
(M, n) codes

- Fixed rate block code of rate $R$. 

There are $M$ code words, $M$ = number of possible messages. There are $n$ source symbols encoded at a time in each code word. An encoder maps from length-$n$ strings of source symbols to length-$m$ bit strings.

The rate $R$ of the code depends on $M$ and $n$

$$R = \frac{\log M}{n} = \log \left( \frac{\text{# of code words}}{\text{# of source symbols}} \right)$$ (5.60)
(M, n) codes

- Fixed rate block code of rate $R$.
- There are $M$ code words, $M =$ number of possible messages.
\((M,n)\) codes

- Fixed rate block code of rate \(R\).
- There are \(M\) code words, \(M = \) number of possible messages.
- There are \(n\) source symbols encoded at a time in each code word.
\((M, n)\) codes

- Fixed rate block code of rate \(R\).
- There are \(M\) code words, \(M = \) number of possible messages.
- There are \(n\) source symbols encoded at a time in each code word.
- An encoder maps from length-\(n\) strings of source symbols to length-\(m\) bit strings.

\[
\begin{align*}
\text{Encoder} \quad & \quad \begin{cases}
(x^{(1)}_1, x^{(1)}_2, \ldots, x^{(1)}_n) \\
(x^{(2)}_1, x^{(2)}_2, \ldots, x^{(2)}_n) \\
(x^{(3)}_1, x^{(3)}_2, \ldots, x^{(3)}_n) \\
\vdots \\
(y^{(1)}_1, y^{(2)}_1, \ldots, y^{(1)}_m) \\
(y^{(2)}_1, y^{(2)}_2, \ldots, y^{(2)}_m) \\
(y^{(3)}_1, y^{(3)}_2, \ldots, y^{(3)}_m) \\
\vdots \\
(y^{(M)}_1, y^{(M)}_2, \ldots, y^{(M)}_m)
\end{cases}
\end{align*}
\]

The rate \(R\) of the code depends on \(M\) and \(n\):

\[
R = \frac{\log M}{n} = \log \left( \frac{\text{# of code words}}{\text{# of source symbols}} \right)
\]
**$(M,n)$ codes**

- Fixed rate block code of rate $R$.
- There are $M$ code words, $M =$ number of possible messages.
- There are $n$ source symbols encoded at a time in each code word.
- An encoder maps from length-$n$ strings of source symbols to length-$m$ bit strings.

The rate $R$ of the code depends on $M$ and $n$

$$R = \frac{\log M}{n} = \frac{\log(\text{# of code words})}{\text{# of source symbols}}$$  \hspace{1cm} (5.60)

In the diagram:
- $n$ source symbols
  - $(x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)})$
  - $(x_1^{(2)}, x_2^{(2)}, \ldots, x_n^{(2)})$
  - $(x_1^{(3)}, x_2^{(3)}, \ldots, x_n^{(3)})$
  - $\vdots$
- $m$ bits in each code word
  - $(y_1^{(1)}, y_2^{(1)}, \ldots, y_m^{(1)})$
  - $(y_1^{(2)}, y_2^{(2)}, \ldots, y_m^{(2)})$
  - $(y_1^{(3)}, y_2^{(3)}, \ldots, y_m^{(3)})$
  - $\vdots$
  - $(y_1^{(M)}, y_2^{(M)}, \ldots, y_m^{(M)})$
- $M$ code words
- Encoder
Fixed rate block code of rate $R$

- An $(M, n)$ code is one that uses $M$ code words for $n$ source symbols.
Fixed rate block code of rate $R$

- An $(M, n)$ code is one that uses $M$ code words for $n$ source symbols.
- Such a code thus has rate of $R = \frac{\log M}{n}$ bits per source symbol. So we need $\log M = nR$ bits to index this code.
Fixed rate block code of rate $R$

- An $(M, n)$ code is one that uses $M$ code words for $n$ source symbols.
- Such a code thus has rate of $R = \frac{\log M}{n}$ bits per source symbol. So we need $\log M = nR$ bits to index this code.
- Then a code is defined as follows:
Fixed rate block code of rate $R$

- An $(M, n)$ code is one that uses $M$ code words for $n$ source symbols.
- Such a code thus has rate of $R = \frac{\log M}{n}$ bits per source symbol. So we need $\log M = nR$ bits to index this code.
- Then a code is defined as follows:

**Definition 5.5.1 (fixed rate block code of rate $R$)**

Let $X_1, X_2, \ldots, X_n \sim Q$, i.i.d. but $Q$ unknown. We have encoder and decoder functions as follows:

\begin{align*}
\text{Encoder:} & \quad f_n : X^n \rightarrow \{1, 2, \ldots, 2^{nR}\} \\
\text{Decoder:} & \quad \phi_n : \{1, 2, \ldots, 2^{nR}\} \rightarrow X^n
\end{align*}

(5.62)
**Fixed rate block code of rate** $R$

- An $(M, n)$ code is one that uses $M$ code words for $n$ source symbols.
- Such a code thus has rate of $R = \frac{\log M}{n}$ bits per source symbol. So we need $\log M = nR$ bits to index this code.
- Then a code is defined as follows:

**Definition 5.5.1 (fixed rate block code of rate $R$)**

Let $X_1, X_2, \ldots, X_n \sim Q$, i.i.d. but $Q$ unknown. We have encoder and decoder functions as follows:

Encoder: $f_n : \mathcal{X}^n \rightarrow \{1, 2, \ldots, 2^{nR}\}$

$$\text{Encoder: } f_n : \mathcal{X}^n \rightarrow \{1, 2, \ldots, 2^{nR}\} \quad (5.61)$$

$$\text{Decoder: } \phi_n : \{1, 2, \ldots, 2^{nR}\} \rightarrow \mathcal{X}^n \quad (5.62)$$
Fixed rate block code of rate $R$

- An $(M, n)$ code is one that uses $M$ code words for $n$ source symbols.
- Such a code thus has rate of $R = \frac{\log M}{n}$ bits per source symbol. So we need $\log M = nR$ bits to index this code.
- Then a code is defined as follows:

**Definition 5.5.1 (fixed rate block code of rate $R$)**

Let $X_1, X_2, \ldots, X_n \sim Q$, i.i.d. but $Q$ unknown. We have encoder and decoder functions as follows:

Encoder: $f_n : \mathcal{X}^n \rightarrow \{1, 2, \ldots, 2^{nR}\}$ \hspace{1cm} (5.61)

Decoder: $\phi_n : \{1, 2, \ldots, 2^{nR}\} \rightarrow \mathcal{X}^n$ \hspace{1cm} (5.62)
Fixed rate block code of rate $R$

- An $(M, n)$ code is one that uses $M$ code words for $n$ source symbols.
- Such a code thus has rate of $R = \frac{\log M}{n}$ bits per source symbol. So we need $\log M = nR$ bits to index this code.
- Then a code is defined as follows:

**Definition 5.5.1 (fixed rate block code of rate $R$)**

Let $X_1, X_2, \ldots, X_n \sim Q$, i.i.d. but $Q$ unknown. We have encoder and decoder functions as follows:

Encoder: $f_n : \mathcal{X}^n \rightarrow \{1, 2, \ldots, 2^{nR}\}$  \hspace{1cm} (5.61)

Decoder: $\phi_n : \{1, 2, \ldots, 2^{nR}\} \rightarrow \mathcal{X}^n$  \hspace{1cm} (5.62)

and probability of error

$$P_e^{(n)} = Q^n(\{x_{1:n} : \phi_n(f_n(x_{1:n})) \neq x_{1:n}\})$$  \hspace{1cm} (5.63)
**Fixed rate block code of rate** $R$

- An $(M, n)$ code is one that uses $M$ code words for $n$ source symbols.
- Such a code thus has rate of $R = \frac{\log M}{n}$ bits per source symbol. So we need $\log M = nR$ bits to index this code.
- Then a code is defined as follows:

**Definition 5.5.1 (fixed rate block code of rate $R$)**

Let $X_1, X_2, \ldots, X_n \sim Q$, i.i.d. but $Q$ unknown. We have encoder and decoder functions as follows:

Encoder: $f_n : \mathcal{X}^n \rightarrow \{1, 2, \ldots, 2^{nR}\}$ \hspace{1cm} (5.61)

Decoder: $\phi_n : \{1, 2, \ldots, 2^{nR}\} \rightarrow \mathcal{X}^n \hspace{1cm} (5.62)$

and probability of error

$$P_e^{(n)} = Q^n(\{x_{1:n} : \phi_n(f_n(x_{1:n})) \neq x_{1:n}\}) \hspace{1cm} (5.63)$$

- Notation: $(M, n) = (2^{nR}, n)$ designates a series (in $n$) of such codes.
Definition 5.5.2 (Universal rate $R$ block code)

A rate $R$ block code for a source is universal if the functions $f_n$ and $\phi_n$ do not depend on the source distribution $Q$ and if

$$P_e^{(n)} \to 0 \text{ as } n \to \infty \text{ whenever } H(Q) < R \quad (5.64)$$
Definition 5.5.2 (Universal rate $R$ block code)

A rate $R$ block code for a source is universal if the functions $f_n$ and $\phi_n$ do not depend on the source distribution $Q$ and if

$$P_e^{(n)} \to 0 \text{ as } n \to \infty \text{ whenever } H(Q) < R$$ (5.64)

- So we require the “ability to code” at rate $R$, which really means code without error, or the error goes to zero for larger block length.
Definition 5.5.2 (Universal rate $R$ block code)

A rate $R$ block code for a source is universal if the functions $f_n$ and $\phi_n$ do not depend on the source distribution $Q$ and if

$$P_e^{(n)} \to 0 \text{ as } n \to \infty \text{ whenever } H(Q) < R \quad (5.64)$$

- So we require the “ability to code” at rate $R$, which really means code without error, or the error goes to zero for larger block length.
- We next state and prove one of Shannon’s main theorems.
Universal Code

Definition 5.5.2 (Universal rate $R$ block code)

A rate $R$ block code for a source is universal if the functions $f_n$ and $\phi_n$ do not depend on the source distribution $Q$ and if

$$P_e^{(n)} \to 0 \text{ as } n \to \infty \text{ whenever } H(Q) < R \quad (5.64)$$

- So we require the “ability to code” at rate $R$, which really means code without error, or the error goes to zero for larger block length.
- We next state and prove one of Shannon’s main theorems.
- If $R > H(Q)$, then there exists a sequence (in $n$) of codes with the error of becoming vanishingly small.
Definition 5.5.2 (Universal rate $R$ block code)

A rate $R$ block code for a source is universal if the functions $f_n$ and $\phi_n$ do not depend on the source distribution $Q$ and if

$$P_e^{(n)} \to 0 \quad \text{as} \quad n \to \infty \quad \text{whenever} \quad H(Q) < R$$

(5.64)

- So we require the “ability to code” at rate $R$, which really means code without error, or the error goes to zero for larger block length.
- We next state and prove one of Shannon’s main theorems.
- If $R > H(Q)$, then there exists a sequence (in $n$) of codes with the error of becoming vanishingly small.
- Conversely, if $R < H(Q)$, then the error goes to 1.
Source Coding Theorem

Theorem 5.5.3 (Shannon’s Source Coding Theorem)

∃ a sequence \( (2^nR, n) \) of universal source codes such that \( P_e^{(n)} \rightarrow 0 \) for all source distributions \( Q \) such that \( H(Q) < R \).

Proof.

- Fix \( R > H(Q) \) to be strictly greater than entropy.
Theorem 5.5.3 (Shannon’s Source Coding Theorem)

\[ \exists \text{ a sequence } (2^{nR}, n) \text{ of universal source codes such that } P_e^{(n)} \to 0 \text{ for all source distributions } Q \text{ such that } H(Q) < R. \]

Proof.

- Fix \( R > H(Q) \) to be strictly greater than entropy.
- Define a rate for \( n \) that is “fixed up” with a polynomial factor. I.e.,

\[
R_n \triangleq R - |X| \frac{\log(n + 1)}{n} < R \quad (5.65)
\]
Source Coding Theorem

Theorem 5.5.3 (Shannon’s Source Coding Theorem)

\[ \exists \text{ a sequence } (2^{nR}, n) \text{ of universal source codes such that } P_c(n) \to 0 \text{ for all source distributions } Q \text{ such that } H(Q) < R. \]

Proof.

- Fix \( R > H(Q) \) to be strictly greater than entropy.
- Define a rate for \( n \) that is “fixed up” with a polynomial factor. I.e.,

\[
R_n \triangleq R - |\mathcal{X}| \frac{\log(n+1)}{n} < R \tag{5.65}
\]

- Define set of sequences that have entropy less than this rate.

\[
A_n \triangleq \{ x_{1:n} \in \mathcal{X}^n : H(P_{x_{1:n}}) \leq R_n \} \tag{5.66}
\]

\[
= \bigcup_{P \in \mathcal{P}_n} T(P) : H(P) \leq R_n \tag{5.67}
\]

\ldots
Source Coding Theorem

... Proof of theorem 5.5.3 continued.

- Then

\[
|A_n| = \sum_{P \in P_n} H(P) \leq R_n \leq \sum_{P \in P_n} H(P) \leq 2^n R_n \leq (n+1)|X| 2^{nR_n} \tag{5.69}
\]

(5.70)

Since \( |A_n| \leq 2^n R_n \), we can index \( A_n \) with \( nR_n \) bits.

Let the encoder be:

\[
f_n(x_1:n) =
\begin{cases}
\text{lexicographic index of } x_1:n \text{ in } A_n \\
0 & \text{if } x_1:n \in A_n \text{ (i.e., if } H(P_{x_1:n}) \leq R_n) \\
\text{else} & \text{(i.e., if } H(P_{x_1:n}) > R_n) 
\end{cases}
\tag{5.71}
\]
Proof of theorem 5.5.3 continued.

Then $|A_n|$
Source Coding Theorem

... Proof of theorem 5.5.3 continued.

Then \(|A_n| = \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} T(P)|.

(5.70)
Proof of theorem 5.5.3 continued.

Then $|A_n| = \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} T(P) \leq \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} 2^{nH(P)}$  

(5.68)

Since $|A_n| \leq 2^n (R_n + \log(n+1))$ we can index $A_n$ with $nR$ bits.

Let the encoder be:

$$f_n(x_1:n) = \begin{cases} \text{lexicographic index of } x_1:n \text{ in } A_n & \text{if } x_1:n \in A_n \text{ (i.e., if } H(P_{x_1:n}) \leq R_n) \\ 0 & \text{else (i.e., if } H(P_{x_1:n}) > R_n) \end{cases}$$

(5.71)
Source Coding Theorem

... Proof of theorem 5.5.3 continued.

Then

\[ |A_n| = \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} T(P) \leq \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} 2^{nH(P)} \]  (5.68)

\[ \leq \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} 2^{nR_n} \]  (5.70)
Then $|A_n| = \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} T(P) \leq \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} 2^{nH(P)} \leq \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} 2^{nR_n} \leq (n + 1)|\mathcal{X}|2^{nR_n}$. 

(5.70)
Source Coding Theorem

... Proof of theorem 5.5.3 continued.

Then \(|A_n| = \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} T(P) \leq \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} 2^{nH(P)} \tag{5.68}\)

\[
\leq \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} 2^{nR_n} \leq (n + 1)|\mathcal{X}|2^{nR_n} \tag{5.69}
\]

\[
= 2^n(R_n + |\mathcal{X}| \frac{\log(n+1)}{n}) \tag{5.70}
\]
Source Coding Theorem

... Proof of theorem 5.5.3 continued.

Then

\[
|A_n| = \sum_{P \in \mathcal{P}_n \mid H(P) \leq R_n} T(P) \leq \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} 2^{nH(P)} \tag{5.68}
\]

\[
\leq \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} 2^{nR_n} \leq (n + 1)|\mathcal{X}|2^{nR_n} \tag{5.69}
\]

\[
= 2^{n(R_n + |\mathcal{X}| \frac{\log(n+1)}{n})} = 2^{nR} \tag{5.70}
\]
Source Coding Theorem

...Proof of theorem 5.5.3 continued.

Then $|A_n| = \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} T(P) \leq \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} 2^{nH(P)}$ (5.68)

\[ \leq \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} 2^{nR_n} \leq (n + 1)|\mathcal{X}|2^{nR_n} \] (5.69)

\[ = 2^n(R_n + |\mathcal{X}| \frac{\log(n+1)}{n}) = 2^nR \] (5.70)

Since $|A_n| \leq 2^nR$, we can index $A_n$ with $nR$ bits.

...
Source Coding Theorem

... Proof of theorem 5.5.3 continued.

1. Then \(|A_n| = \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} T(P) \leq \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} 2^n H(P)\) (5.68)

\[
\leq \sum_{P \in \mathcal{P}_n : H(P) \leq R_n} 2^{nR_n} \leq (n + 1)|\mathcal{X}| 2^{nR_n} \quad (5.69)
\]

\[
= 2^n (R_n + |\mathcal{X}| \frac{\log(n+1)}{n}) = 2^n R \quad (5.70)
\]

Since \(|A_n| \leq 2^n R\), we can index \(A_n\) with \(nR\) bits.

2. Let the encoder be:

\[
f_n(x_{1:n}) = \begin{cases} 
\text{lexicographic index} & \text{if } x_{1:n} \in A_n \\
\text{of } x_{1:n} \text{ in } A_n & (\text{i.e., if } H(P_{x_{1:n}}) \leq R_n) \\
0 & \text{else} \\
& (\text{i.e., if } H(P_{x_{1:n}}) > R_n)
\end{cases} \quad (5.71)
\]

...
Source Coding Theorem

... Proof of theorem 5.5.3 continued.

- Note: $f_n(\cdot)$ does not depend on the source distribution, only on the ordering and on $R_n$. 

...
Source Coding Theorem

... Proof of theorem 5.5.3 continued.

- Note: $f_n(\cdot)$ does not depend on the source distribution, only on the ordering and on $R_n$.
- Error occurs if $x_{1:n} \notin A_n$. 
... Proof of theorem 5.5.3 continued.

- Note: $f_n(\cdot)$ does not depend on the source distribution, only on the ordering and on $R_n$.
- Error occurs if $x_{1:n} \notin A_n$.
- We can represent this by placing types within the probability simplex, to indicate which types may be encoded. E.g., if $|\mathcal{X}| = 3$, then

![Diagram showing types within the probability simplex](image)

(1,0,0)
(0,1,0)
(0,0,1)
Proof of theorem 5.5.3 continued.

- Within the simplex, each point is potentially a type (the points with rational values with denominator $n$ and numerator between 0 and $n$).
Within the simplex, each point is potentially a type (the points with rational values with denominator \( n \) and numerator between 0 and \( n \)).

Yellow region corresponds to types \( P \in P_n \) whose sequences can be encoded correctly, as the rate constraint is satisfied.
Source Coding Theorem

... Proof of theorem 5.5.3 continued.

- Within the simplex, each point is potentially a type (the points with rational values with denominator $n$ and numerator between 0 and $n$).
- Yellow region corresponds to types $P \in P_n$ whose sequences can be encoded correctly, as the rate constraint is satisfied.
- Light blue corresponds to types whose sequences will result in an error.

$H(P) < R_n$

$H(P) > R_n$

$H(P) = R_n$

Set of sequences that are encoded correctly.
... Proof of theorem 5.5.3 continued.

- We upper bound $P_{e}^{(n)}$ and show it $\to 0$ as $n \to \infty$ when $R > H(Q)$.
Source Coding Theorem

... Proof of theorem 5.5.3 continued.

- We upper bound $P_e^{(n)}$ and show it $\to 0$ as $n \to \infty$ when $R > H(Q)$.
- An error occurs when the sequence is not in $A_n$, thus

\[
\leq (n+1)|X| \max_P: H(P) > Rn Q_n(T(P)) \tag{5.72}
\]

\[
\leq (n+1)|X|^2 - n \min_P: H(P) > Rn D(P||Q) \tag{5.74}
\]

So we have $R_n \uparrow R \Rightarrow R_n < R$ for all $n$, and $H(Q) < R_n$. Thus, for some $n_0$, $\forall n > n_0$, we have $H(Q) < R_n$. ...
Source Coding Theorem

... Proof of theorem 5.5.3 continued.

- We upper bound $P_e(n)$ and show it $\to 0$ as $n \to \infty$ when $R > H(Q)$.
- An error occurs when the sequence is not in $A_n$, thus

$$P_e(n)$$

(5.74)
... Proof of theorem 5.5.3 continued.

- We upper bound $P_e^{(n)}$ and show it $\to 0$ as $n \to \infty$ when $R > H(Q)$.
- An error occurs when the sequence is not in $A_n$, thus

$$P_e^{(n)} = 1 - Q^n(A_n)$$

(5.74)
... Proof of theorem 5.5.3 continued.

- We upper bound $P_e^{(n)}$ and show it $\to 0$ as $n \to \infty$ when $R > H(Q)$.
- An error occurs when the sequence is not in $A_n$, thus

$$P_e^{(n)} = 1 - Q^n(A_n) = \sum_{P: H(P) > R_n} Q^n(T(P))$$ (5.72)

$$\leq (n + 1)|X|$$ (5.73)

$$\leq (n + 1)|X|^2 - n [\min_{P: H(P) > R_n} D(P || Q)]$$ (5.74)

So we have $R_n \uparrow R \Rightarrow R_n < R$ for all $n$, and $H(Q) < R$.
Source Coding Theorem

... Proof of theorem 5.5.3 continued.

- We upper bound $P_e^{(n)}$ and show it $\to 0$ as $n \to \infty$ when $R > H(Q)$.
- An error occurs when the sequence is not in $A_n$, thus

$$P_e^{(n)} = 1 - Q^n(A_n) = \sum_{P: H(P) > R_n} Q^n(T(P))$$  \hspace{1cm} (5.72)

$$\leq (n + 1)|\mathcal{X}| \max_{P: H(P) > R_n} Q^n(T(P))$$  \hspace{1cm} (5.73)

$$\leq (n + 1)|\mathcal{X}| \max_{P: H(P) > R_n} Q^n(T(P))$$  \hspace{1cm} (5.74)

...
Source Coding Theorem

... Proof of theorem 5.5.3 continued.

- We upper bound $P_e(n)$ and show it $\to 0$ as $n \to \infty$ when $R > H(Q)$.
- An error occurs when the sequence is not in $A_n$, thus

$$P_e(n) = 1 - Q^n(A_n) = \sum_{P:H(P)>R_n} Q^n(T(P))$$  \hspace{1cm} (5.72)

$$\leq (n + 1)|X| \max_{P:H(P)>R_n} Q^n(T(P))$$  \hspace{1cm} (5.73)

$$\leq (n + 1)|X| 2^{-n \min_{P:H(P)>R_n} D(P||Q)}$$  \hspace{1cm} (5.74)
Proof of theorem 5.5.3 continued.

- We upper bound $P_e^{(n)}$ and show it $\rightarrow 0$ as $n \rightarrow \infty$ when $R > H(Q)$.
- An error occurs when the sequence is not in $A_n$, thus

$$P_e^{(n)} = 1 - Q^n(A_n) = \sum_{P: H(P) > R_n} Q^n(T(P))$$

$$\leq (n + 1)|\mathcal{X}| \max_{P: H(P) > R_n} Q^n(T(P))$$

$$\leq (n + 1)|\mathcal{X}| 2^{-n \left[ \min_{P: H(P) > R_n} D(P||Q) \right]}$$

- So we have $R_n \uparrow R \Rightarrow R_n < R$ for all $n$, and $H(Q) < R$. 

...
Proof of theorem 5.5.3 continued.

- We upper bound $P_e^{(n)}$ and show it $\to 0$ as $n \to \infty$ when $R > H(Q)$.
- An error occurs when the sequence is not in $A_n$, thus

$$P_e^{(n)} = 1 - Q^n(A_n) = \sum_{P: H(P) > R_n} Q^n(T(P)) \leq (n + 1)|\mathcal{X}| \max_{P: H(P) > R_n} Q^n(T(P)) \leq (n + 1)|\mathcal{X}| 2^{-n \left[ \min_{P: H(P) > R_n} D(P \| Q) \right]}$$

So we have $R_n \uparrow R \Rightarrow R_n < R$ for all $n$, and $H(Q) < R$.

- Thus, for some $n_0$, $\forall n > n_0$, we have $H(Q) < R_n$.
... Proof of theorem 5.5.3 continued.

- In Eq. (5.74) we chose $P : H(P) > R_n$ for the current $n$ (assuming there is one)
### Source Coding Theorem

... Proof of theorem 5.5.3 continued.

<table>
<thead>
<tr>
<th>$R_{n_0-2}$</th>
<th>$R_{n_0-1}$</th>
<th>$R_{n_0}$</th>
<th>$R_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In Eq. (5.74) we chose $P : H(P) > R_n$ for the current $n$ (assuming there is one)

$\Rightarrow H(P) > R_n > H(Q)$
In Eq. (5.74) we chose $P : H(P) > R_n$ for the current $n$ (assuming there is one)

$\implies H(P) > R_n > H(Q)$

$\implies P \neq Q$
... Proof of theorem 5.5.3 continued.

<table>
<thead>
<tr>
<th>$R_{n_0-2}$</th>
<th>$R_{n_0-1}$</th>
<th>$R_{n_0}$</th>
<th>$R_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H(Q)$</td>
<td></td>
<td></td>
<td>$H(P)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$R$</td>
</tr>
</tbody>
</table>

- In Eq. (5.74) we chose $P : H(P) > R_n$ for the current $n$ (assuming there is one)
- $\Rightarrow H(P) > R_n > H(Q)$
- $\Rightarrow P \neq Q$
- $\Rightarrow D(P||Q) > 0$ for the chosen $P$
In Eq. (5.74) we chose $P : H(P) > R_n$ for the current $n$ (assuming there is one)

$\Rightarrow H(P) > R_n > H(Q)$

$\Rightarrow P \neq Q$

$\Rightarrow D(P\|Q) > 0$ for the chosen $P$

Thus, we get

$$P_e^{(n)} \leq (n + 1)|\mathcal{X}| 2^{-n \left[ \min_{P : H(P) > R_n} D(P\|Q) \right]} \quad (5.75)$$
In Eq. (5.74) we chose $P : H(P) > R_n$ for the current $n$ (assuming there is one)

$\Rightarrow H(P) > R_n > H(Q)$

$\Rightarrow P \neq Q$

$\Rightarrow D(P\|Q) > 0$ for the chosen $P$

Thus, we get

$$P_e^{(n)} \leq (n + 1)|\mathcal{X}| 2^{-n \left[ \min_{P : H(P) > R_n} D(P\|Q) \right]}$$

(5.75)
In Eq. (5.74) we chose $P : H(P) > R_n$ for the current $n$ (assuming there is one)

$\Rightarrow H(P) > R_n > H(Q)$

$\Rightarrow P \neq Q$

$\Rightarrow D(P \parallel Q) > 0$ for the chosen $P$

Thus, we get

$$P_e^{(n)} \leq (n + 1)|\mathcal{X}| \underbrace{2^{-n \left[ \min_{P : H(P) > R_n} D(P \parallel Q) \right]}}_{\text{poly in } n} \underbrace{\exp. \text{ decreasing as } n \to \infty}_{(5.75)}$$
**Source Coding Theorem**

... Proof of theorem 5.5.3 continued.

<table>
<thead>
<tr>
<th>$R_{n_0-2}$</th>
<th>$R_{n_0-1}$</th>
<th>$R_{n_0}$</th>
<th>$R_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H(Q)$</td>
<td></td>
<td></td>
<td>$H(P)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$R$</td>
</tr>
</tbody>
</table>

- In Eq. (5.74) we chose $P : H(P) > R_n$ for the current $n$ (assuming there is one)
- $\Rightarrow H(P) > R_n > H(Q)$
- $\Rightarrow P \neq Q$
- $\Rightarrow D(P\|Q) > 0$ for the chosen $P$
- Thus, we get

$$P_e^{(n)} \leq (n + 1)|X| \cdot 2^{-n \left[ \min_{P: H(P) > R_n} D(P\|Q) \right]}$$  \hspace{1cm} (5.75)

- Which implies that $P_e^{(n)} \rightarrow 0$ as $n \rightarrow \infty$.