Reading

- Read chapters 1, and 2 in C&T.
- Read chapter 3 in C&T, as we’ll start going over it on Tuesday.
- Read sections 11.1, 11.3 (method of types and universal source coding).
Announcements, Assignments, and Reminders

- Late policy: 10% every 24 hour period that you are late, and no more than 3 days late accepted.
- Lowest grade out of all HW grades is not counted towards final grade (so you can skip one HW with impunity).
- Please do use our discussion board (https://catalyst.uw.edu/gopost/board/karna/25503/) for all questions, so that all will benefit from them being answered.

Class Road Map

- L1 (1/3): Overview, Entropy
- L2 (1/5): Props. Entropy, Mutual Information, KL-Divergence
- L3 (1/10): KL-Divergence, Jensen, properties, Data Proc. Inequality
- L5 (1/17): Fano, AEP
- L6 (1/19): snow
- L6 (1/24): AEP, source coding
- L7 (1/26): Method of Types
- L8 (1/31):
- L9 (2/2):
- L11 (2/7):
- L12 (2/9):
- L13 (2/14): Midterm
- L14 (2/16):
- L15 (2/21):
- L16 (2/23):
- L175 (2/28):
- L18 (3/1):
- L19 (3/6):
- L20 (3/8):

Finals Week: March 12th–16th.
**Fano’s Inequality**

- Consider the following situation where we send $X$ through a noisy channel, receive $Y$, and do further processing:

  ![Noisy Channel Diagram](image)

  $\hat{X}$ is an estimate of $X$.

- An error if $X \neq \hat{X}$. How do we measure the error? With probability, $P_e \triangleq p(X \neq \hat{X})$.

- Intuitively, conditional entropy should tell us something about the error possibilities, in fact, we have

**Theorem 2.1 (Fano’s Inequality)**

\[
H(P_e) + P_e \log(|\mathcal{X}| - 1) \geq H(X|\hat{X}) \geq H(X|Y) \tag{1}
\]

**Aside: modes of convergence, expanded definition**

- Expanded description of convergence in probability.

- Let $X_1, X_2, \ldots$ be a sequence of i.i.d. r.v.s, with $EX = \mu < \infty$, with $S_n = \sum_{i=1}^{n} X_i$.

- Then, \( \forall \epsilon > 0, \forall \delta > 0, \exists n_0 \) such that for all $n > n_0$, we have

\[
p \left( \left| \frac{1}{n} S_n - \mu \right| > \epsilon \right) < \delta \text{ for } n > n_0 \tag{2}\]

- Equivalently

\[
p \left( \left| \frac{1}{n} S_n - \mu \right| \leq \epsilon \right) > 1 - \delta \text{ for } n > n_0 \tag{3}\]
Aside: modes of convergence, implications

- Some modes of convergence are stronger than others. That is
  \[ p \Rightarrow D \quad \forall r \geq 1 \]
  
- Also, if \( r > s \geq 1 \), then \( r \Rightarrow s \).
- Different versions of things like the law of large numbers differ only in the strength of their required modes of convergence.

Towards AEP

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

- Example: English letters, would have \( K = 26 \).
- We want to have a code word for every possible message, must have what condition?
  \[ 2^m \geq K^n \Rightarrow m \geq (\log K)n \]
WLLN and entropy

- Combining the above, we get
  \[ \frac{1}{n} \sum_{i=1}^{n} I(X_i) \xrightarrow{p} H(X) \quad (5) \]

- Thus, if \( n \) is big enough, we have that (this is where it gets cool 😊)
  \[ \frac{1}{n} \sum_{i=1}^{n} I(x_i) \approx H(X) \text{ when } \forall i, x_i \sim p(x) \quad (6) \]

- \[ \Rightarrow - \frac{1}{n} \sum_{i=1}^{n} \log p(x_i) \approx H(X) \quad (7) \]

- \[ \Rightarrow - \log \prod_{i=1}^{n} p(x_i) \approx nH(X) \quad (8) \]

- \[ \Rightarrow - \log p(x_1, x_2, \ldots, x_n) \approx nH(X) \quad (9) \]

- \[ \Rightarrow p(x_1, \ldots, x_n) \approx 2^{-nH(X)} \quad (10) \]

AEP: Almost all events are almost equally probable

- If \( X_1, X_2, \ldots, X_n \) are i.i.d. and \( X_i \sim p(x) \) for all \( i \), then
  - if \( n \) is large enough,
  - for any sample \( x_1, x_2, \ldots, x_n \)
  - The probability of the sample is essentially independent of the sample, i.e.,
    \[ p(x_1, \ldots, x_n) \approx 2^{-nH(X)} \quad (11) \]

  where \( H(X) \) is the entropy of \( p(x) \).

- Thus, there can only be \( 2^{nH} \) such samples, and it may be that \( 2^{nH} \ll K^n \)

- Those samples that will happen are called typical, and they are represented by \( A^{(n)}_\epsilon \).

- Thus, a large portion of \( \mathcal{X}^n \) essentially won’t happen, i.e., could be that \( 2^{nH} \approx |A^{(n)}_\epsilon| \ll |\mathcal{X}^n| = K^n \).
### AEP

**Theorem 2.2 (AEP)**

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

$$- \frac{1}{n} \log p(X_1, X_2, \ldots, X_n) \xrightarrow{p} H(X) \quad (12)$$

**Proof.**

$$- \frac{1}{n} \log p(X_1, X_2, \ldots, X_n) = - \frac{1}{n} \log \prod_{i=1}^{n} p(X_i) \quad (13)$$

$$= - \frac{1}{n} \sum_{i} \log p(X_i) \xrightarrow{p} -E \log p(X) \quad (14)$$

$$= H(X) \quad (15)$$

### Typical Set

**Definition 2.3 (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

$$2^{-n(H(X)+\epsilon)} \leq p(x_1, x_2, \ldots, x_n) \leq 2^{-n(H(X)-\epsilon)} \quad (16)$$

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

$$A_{\epsilon}^{(n)} = \left\{ (x_1, x_2, \ldots, x_n) : \left| - \frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| < \epsilon \right\} \quad (17)$$

- Typical set are those sequences with log probability within the range $-nH$
- $A_{\epsilon}^{(n)}$ has a number of interesting properties.
Typical Set $A_{\epsilon}^{(n)}$

**Theorem 2.4 (Properties of $A_{\epsilon}^{(n)}$)**

1. If $(x_1, x_2, \ldots, x_n) \in A_{\epsilon}^{(n)}$, then
   \[ H(X) - \epsilon \leq -\frac{1}{n} \log p(x_1, x_2, \ldots, x_n) \leq H(X) + \epsilon \quad (18) \]

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

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

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

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

---

Typical Set, example, $K = |\{0, 1\}|$

- Uniform distribution, $K = 2$, $X = \{0, 1\}$, Bernoulli trials, $p = 0.5$, entropy $H = 1$, and $|A_{\epsilon}^{(n)}| = 2^n H = 2^n = K^n$, so all sequences will occur with equal probability.

- Non-uniform distribution, $p = 0.1$, $1 - p = q = 0.9$, the entropy $H \approx 0.469$. Consider $n = 100$, then $K^{100} = 2^{100} \approx 10^{30}$, so representational capacity of the source strings is $10^{30}$.

- But $|A_{\epsilon}^{(n)}| = 2^n H \approx 10^{14} \ll 10^{30} \approx K^{100}$. So the number of typical sequences is much smaller than the number of possible sequences.

- Q: What is $10^{30} - 10^{14}$? A: $10^{30} - 10^{14} \approx 10^{26}$.

- Inefficiency: representational capacity is much larger than the things that occur. This means that things are inefficient, source alphabet is poor for compression.

- Thought question (i.e., what you might want to think about):
  Assume $\epsilon$ is very small, then where did all the mass of those $\approx 10^{30} - 10^{14}$ sequences go? We will answer this shortly.
Typical Sets are Typical

- Curiously,
  \[ p(A_\epsilon^{(n)}) > 1 - \epsilon \text{ for any } \epsilon > 0 \]  
  (19)

- So, \( A_\epsilon^{(n)} \) has pretty much all of the probability, and each element in \( A_\epsilon^{(n)} \) has the same probability, so
  \[ p(x) \approx 2^{-nH} \quad \forall x \in A_\epsilon^{(n)} \]  
  (20)

- Ex: Bernoulli trials: \( X_i \sim \text{Bernoulli}(p) \), with
  \[ p(X_i = 1) = p = 1 - p(X_i = 0), \text{ and } p > 0.5. \]

- Probability of \( n \) successive 1s is \( p^n \) and is the most likely sequence.
- Probability of each typical sequence is \( 2^{-nH} \).
- For \( n = 100 \), \( p = 0.9 = 1 - q \), most likely sequence has probability
  \[ p^n \approx 2.66 \times 10^{-5}, \text{ but a typical sequence has probability} \]
  \[ 2^{-nH} \approx 7.62 \times 10^{-15}. \]

Non-typical sequences are not typical

- Thus, \( p^n \gg 2^{-nH} \) and the most likely sequence is much more probable than a typical one.
- Typical set, essentially, has all the probability \( A_\epsilon^{(n)} > 1 - \epsilon \)
- But is the most likely sequence in the typical set? No, since the probability of the most likely sequence is not \( 2^{-nH} \)
- Again, for \( n = 100 \), \( p = 0.9 = 1 - q \), consider a sequence with ninety 1s and ten 0s, probability \( p^{90}(1-p)^{10} \approx 7.62 \times 10^{-15} \approx 2^{-nH} \).
- So, for \( n = 1 \), and \( p = 0.9 \),
  - most probable sequence has probability \( 2.66 \times 10^{-5} \)
  - but a typical sequence (such as above) has probability \( 7.62 \times 10^{-15} \ll 2.66 \times 10^{-5} \).
- Thus, this very improbable sequence is typical!
### Average probability of sequences

- What happens to the probability of the most probable sequence, on average (per symbol), as $n$ gets big?
- By the AEP, we have that if $(x_1, \ldots, x_n)$ is typical (i.e., $(x_1, \ldots, x_n) \in A^{(n)}_{\epsilon}$, for any $\epsilon > 0$), then
  \[
  -\frac{1}{n} \log p(x_1, \ldots, x_n) \xrightarrow{n \to \infty} H
  \]  
  (21)

- What happens to the probability of the most probable sequence, on average (per symbol), as $n$ gets big, for $p > 0.5$,
  \[
  -\frac{1}{n} \log p^n = -\log p = \log \frac{p}{p} \xrightarrow{n \to \infty} -\log p
  \]  
  (22)

- So average probability of the most probable sequence is quite different than the typical sequences.

### Are typical sequences most probable?

- Again, typical set has, essentially, all the probability $p(A^{(n)}_{\epsilon}) > 1 - \epsilon$.
- How can a sequence having the most probability not be typical, but a sequence with much lower probability be typical?
- There are exponentially many sequences that are typical, each with less probability than the high probability sequences.
- There are exponentially many more typical sequences then there are “high” probability sequences.
- The probability of each individual sequence goes to zero as $n \to \infty$.
- The size of the set of typical sequences grows fast enough, as $n \to \infty$ such that that the probability of $A^{(n)}_{\epsilon}$ goes to 1.
- The size of the set of highly probable sequences grows slow enough so that the probability of that set goes to zero, as $n \to \infty$. 

---
Binomial Distribution, what happens when \( n \) gets big?

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

- 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.

By comparison, number of atoms in the observable universe \( \approx e^{187} \).
Typical Set $A_e^{(n)}$

**Theorem 3.4 (Properties of $A_e^{(n)}$)**

1. If $(x_1, x_2, \ldots, x_n) \in A_e^{(n)}$, then
   \[
   H(X) - \epsilon \leq -\frac{1}{n} \log p(x_1, x_2, \ldots, x_n) \leq H(X) + \epsilon \quad (18)
   \]

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

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

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

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

**Proof of Theorem 3.4.**

1. This is a restatement of the AEP definition.

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

   \[
   p(A_e^{(n)}) = p \left( \left| -\frac{1}{n} \sum_i \log p(x_i) - H \right| < \epsilon \right) > 1 - \delta \text{ for } n \text{ big enough} \quad (23)
   \]

   and we can choose any $\delta$ we wish, so choose $\delta = \epsilon$, giving

   \[
   p(A_e^{(n)}) > 1 - \epsilon \text{ for } n \text{ big enough } \forall \epsilon \quad (24)
   \]

   ...
Theorem 3.4 Proofs

Proof of Theorem 3.4.

1. **Upper bound size of \( A_{\epsilon}^{(n)} \)**

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

(25)

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

(26)

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

...
Data Compression to the entropy of the source

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

  $\mathcal{A}$ having $|\mathcal{A}| = K^n$ elements

Typical Set Compression

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

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

  (29)
- We use an extra bit at the beginning to indicate if it is typical or not, i.e., we use

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

  (30)
  which indexes which of the typical set elements it is. $b_0 = 0$ indicating that the set is typical.
- Total number of bits required is $n(H + \epsilon) + 2$ for a typical sequence.
Typical Set Compression

- We index the elements in each of the sets, the typical set and the non-typical set, separately.
- In the non-typical set, we index everything. I.e., we use \([\log |\mathcal{X}|^n] \leq n \log K + 1\) bits.
- I.e., we index everything with a bit vector of the form
  \[(b_0, b_1, b_2, \ldots, b_{\lceil \log |\mathcal{X}|^n \rceil})\]  
  (31)
  where here \(b_0 = 1\) indicating atypicality.
- Total number of bits for an atypical sequence is \(n \log K + 2\).
- Note, this is our first code for the class! This is called source coding or compression, and entails finding a sequence of bits for each source string so that average length is as short as possible.

Typical Set Compression: Features of our code, and setup

- Code is 1-to-1, so easy to decode and encode, given code book (mapping).
- Simple but brute force enumeration of atypical set \(A_e^{(n)c}\) means more bits than necessary, since \(|A_e^{(n)c}| = |\mathcal{X}^n| - |A_e^{(n)}| = K^n - |A_e^{(n)}|\), but this is not going to matter, we will see.
- typical sequences have a “short” description length, \(\approx nH\).
- Let \(\ell(x_{1:n})\) be length of the codeword assigned to sequence \(x_{1:n}\).
- \(\ell(X_{1:n})\) is a random variable since \(X_{1:n}\) is a random variable.
- Thus, \(E\ell(X_{1:n}) = \sum_{x_{1:n}} p(x_{1:n})\ell(x_{1:n})\) is the average, or expected, length of our code. We want this to be as short as possible.
Expected Length

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

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

(32)

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

(33)

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

(34)

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

(35)

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

(36)

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

(37)

Thus, it takes at most $nH(X)$ bits to represent $X_{1:n}$ on average, or $H(X)$ bits per source alphabet symbol.

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

Thus, $n(H + \epsilon')$ can be made as close as we want to $nH$ by making $\epsilon$ small and $n$ large.

We have just proven the following theorem:

**Theorem 4.1**

Let $X_{1:n}$ be i.i.d. $\sim p(x)$, $\epsilon > 0$, then $\exists$ a code $f_n : \mathcal{X}^n \rightarrow$ binary strings and integer $n_\epsilon$, such that the mapping is one-to-one (so invertible w/o error), and

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

(38)

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

Thus, it takes at most $nH(X)$ bits to represent $X_{1:n}$ on average, or $H(X)$ bits per source alphabet symbol.
Shannon’s source coding theorem

- The previous theorem is Shannon’s first theorem, stating that it is possible (using long block lengths) to compress down to the entropy limit.
- An instance of universal source coding, coding without explicitly using the distribution, since whatever happens, once \( n \) gets large, is all that will happen.
- Ex: online coding, code only those things that you encounter knowing that it must be typical if you encounter it, if \( n \) is large enough. In such case, you don’t need \( p(x) \), only \( H(p) \).
- Ultimately, we need to prove that we can’t compress to lower than the entropy limit without incurring error, this is the converse of the theorem that we will prove soon.

Other high probable sets?

- We know that \( p(X^n) = 1 \).
- 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 $B_{\delta}^{(n)}$ be any set with the property
  \[ p(B_{\delta}^{(n)}) \geq 1 - \delta \]  
  (39)
  $B_{\delta}^{(n)}$ could, say, contain the most likely sequences as well.

**Theorem 4.2**

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

\[
\frac{1}{n} \log |B_{\delta}^{(n)}| > H - \delta' \text{ if } n \text{ is large enough}
\]  
(40)
\[
|B_{\delta}^{(n)}| > 2^{n(H-\delta')} \approx 2^{nH}
\]  
(41)

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

Coding Strategy with errors

- Previous code was variable length, we had two lengths one for the typical set $A_{\epsilon}^{(n)}$ 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.
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
  \[
  p(A^{(n)}_\epsilon) > 1 - \epsilon \tag{42}
  \]
- And error occurs when a sequence is not typical, so we can bound the error probability
  \[
  p(\text{error}) = p(A^{(n)c}_\epsilon) \leq \epsilon \tag{43}
  \]
- Recall typical: \( \forall \epsilon > 0, \forall \delta > 0, \exists n_0 \text{ s.t. for } n > n_0, \)
  \[
  p\left\{ \left| \frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| < \epsilon \right\} > 1 - \delta \tag{44}
  \]
- Which is same as: \( \forall \epsilon > 0, \forall \delta > 0, \exists n_0 \text{ s.t. for } n > n_0, \)
  \[
  p\left\{ \left| \frac{1}{n} \log p(x_1, \ldots, x_n) - H \right| > \epsilon \right\} \leq \delta \tag{45}
  \]
- 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 \( p(A^{(n)c}_\epsilon) \leq \delta(n, \epsilon), \) or
  \[
  p(\text{error}) \to 0 \text{ as } n \to \infty \tag{46}
  \]
- 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.
Coded 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.,
  $$2^{n(H - \alpha \epsilon)} 2^{-n(H - \epsilon)} = 2^{-n\epsilon(\alpha - 1)}$$
- For any $\alpha > 1$, this probability $\to 0$ as $n \to \infty$. Problem: probability shrinks exponentially faster than number of code words grows.
- 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 $n \to \infty$.
- Conversely, if the r.v.s are compressed into fewer than $nH$ bits, then it is virtually certain that information will be loss and errors will occur.
Overview: Method of types

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