

CS3236 Lecture Notes #3: Block Source Coding

Jonathan Scarlett

February 1, 2020

Useful references:

- Cover/Thomas Chapter 3
- MacKay Chapter 4
- Shannon's 1948 paper "A Mathematical Introduction to Communication"

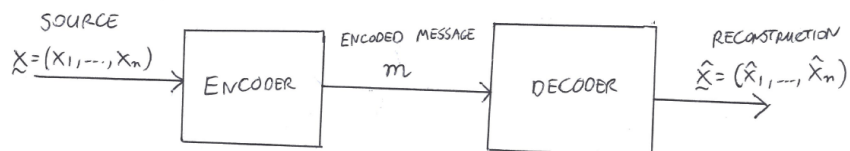
1 Setup

Introduction.

- In the previous lecture, we mapped individual source symbols $x \in \mathcal{X}$ to *variable-length* binary sequences one at a time (symbol coding), and briefly discussed mapping multiple at a time (block coding).
- In this lecture, we consider the following distinct setting:
 - We do not work symbol-by-symbol, but instead apply some encoding function to a *length- n block* X_1, \dots, X_n .
 - The output of the encoder is not a variable-length sequence, but instead an integer $m \in \{1, \dots, M\}$ for some M . For instance, we might store M on a computer as a *fixed-length* binary sequence of length $\log_2 M$.

Because each input sequence of length n is mapped to a binary output sequence with length $\log_2 M$, this is sometimes called *fixed-to-fixed* length source coding.

- An illustration:



- At the end of the last lecture, we briefly mentioned variable-length block coding methods, which are in fact more pertinent to practical compression methods. The fixed-length setting, on the other hand, provides a better warm-up for next lecture's topic of channel coding.
- *Key difference:* Consider the case that the X_i are binary (for simplicity of discussion). If $\log_2 M < n$ (i.e., $M < 2^n$), we clearly can't assign every sequence (X_1, \dots, X_n) a unique value of m . This means that *some* of the sequences must be decoded incorrectly ("errors").
 - In contrast, in the variable-length setting, we never have an error; some output sequences just come out longer than others.

Formal problem statement.

- The *source* is a sequence (X_1, \dots, X_n) . We focus on *discrete memoryless sources*:
 - Discrete: The alphabet \mathcal{X} is finite;
 - Memoryless: $P_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^n P_X(x_i)$, i.e., the source symbols are i.i.d. on some distribution P_X . (This is a restrictive assumption, but still an interesting problem to study)
- An *encoder* receives as input a sequence of source symbols $\mathbf{X} = (X_1, \dots, X_n) \in \mathcal{X}^n$, and maps it to an encoded message $m = f(\mathbf{X})$ in $\{1, \dots, M\}$.
- A *decoder* receives the encoded message m and maps it to an estimate $\hat{\mathbf{X}} = g(m)$ (in \mathcal{X}^n) of the source.
- An error is said to have occurred if $\hat{\mathbf{X}} \neq \mathbf{X}$, and the *error probability* is given by

$$P_e = \mathbb{P}[\hat{\mathbf{X}} \neq \mathbf{X}].$$

- The *rate* is defined to be

$$R = \frac{1}{n} \log_2 M,$$

and represents the number of bits per source symbol used to represent the encoded value m . The lower the rate, the more we have compressed the source sequence.

A fundamental trade-off.

- Clearly we would like P_e to be small.
- We also want M (or equivalently, R) to be small, so that we require less bits to store m .
- The length n also plays a fundamental role, and is referred to as the *block length*.
- Key question: What is the fundamental trade-off between error probability P_e , rate R , and block length n ? In particular, how low can the rate be while keeping the error probability small?
- **Fixed-Length Source Coding Theorem.** For any discrete memoryless source with per-symbol distribution P_X , we have the following:
 - (Achievability) If $R > H(X)$, then for any $\epsilon > 0$, there exists a (sufficiently large) block length n and a source code (i.e., encoder and decoder) of rate R such that $P_e \leq \epsilon$;
 - (Converse) If $R < H(X)$, then there exists $\epsilon > 0$ such every source code of rate R has $P_e > \epsilon$, regardless of the block length (i.e., P_e cannot be arbitrarily small).

The proofs are respectively given in the next two sections.

2 Typical Sequences and the Asymptotic Equipartition Property

Definition.

- Recall that $\mathbf{X} = (X_1, \dots, X_n)$ is an i.i.d. sequence with each symbol distributed according to P_X . In the following, let $P_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^n P_X(x_i)$ be the PMF of \mathbf{X} .
- The *typical set* is defined as

$$\mathcal{T}_n(\epsilon) = \left\{ \mathbf{x} \in \mathcal{X}^n : 2^{-n(H(X)+\epsilon)} \leq P_{\mathbf{X}}(\mathbf{x}) \leq 2^{-n(H(X)-\epsilon)} \right\},$$

where $\epsilon > 0$ is a fixed (small) constant. (Note: This ϵ is not directly related to the ϵ in the theorem statement above, we just use the same symbol because both are arbitrarily small constants)

- As we will see shortly, it is called the typical set because for $\mathbf{X} \sim P_{\mathbf{X}}$ the probability that $\mathbf{X} \in \mathcal{T}_n(\epsilon)$ is very close to one.
- After analyzing its properties, we will give some intuition as to how one might have come up with this definition “from scratch”.

Properties.

- For any fixed $\epsilon > 0$, four key properties of the typical set are as follows (proofs below):

1. (Equivalent definition) We have $\mathbf{x} \in \mathcal{T}_n(\epsilon)$ if and only if

$$H(X) - \epsilon \leq \frac{1}{n} \sum_{i=1}^n \log_2 \frac{1}{P_X(x_i)} \leq H(X) + \epsilon$$

where x_i is the i -th entry of \mathbf{x} .

2. (High probability) $\mathbb{P}[\mathbf{X} \in \mathcal{T}_n(\epsilon)] \rightarrow 1$ as $n \rightarrow \infty$.
 3. (Cardinality upper bound) $|\mathcal{T}_n(\epsilon)| \leq 2^{n(H(X)+\epsilon)}$.
 4. (Cardinality lower bound) $|\mathcal{T}_n(\epsilon)| \geq (1 - o(1))2^{n(H(X)-\epsilon)}$, where $o(1)$ represents a term that vanishes as $n \rightarrow \infty$.
- Interpretation: With high probability (second property), a randomly drawn i.i.d. sequence \mathbf{X} will be one of roughly $2^{nH(X)}$ sequences (third and fourth properties), each of which has probability roughly $2^{-nH(X)}$ (definition of typical set).
 - We call this the *Asymptotic Equipartition Property*, because it states that asymptotically (as $n \rightarrow \infty$) the distribution is roughly uniform over $\mathcal{T}_n(\epsilon)$.

- Proofs:

1. Apply $\frac{1}{n} \log_2(\cdot)$ to the left, middle, and right of the condition $2^{-n(H(X)+\epsilon)} \leq P_{\mathbf{X}}(\mathbf{x}) \leq 2^{-n(H(X)-\epsilon)}$ defining $\mathcal{T}_n(\epsilon)$. The left and right clearly become $H(X) - \epsilon$ and $H(X) + \epsilon$ (since $\log_2(2^\alpha) = \alpha$), and the middle becomes $\frac{1}{n} \log_2 P_{\mathbf{X}}(\mathbf{x}) = \frac{1}{n} \log_2 \prod_{i=1}^n P_X(x_i) = \frac{1}{n} \sum_{i=1}^n \log_2 \frac{1}{P_X(x_i)}$.

2. Since \mathbf{X} is an i.i.d. sequence, $\frac{1}{n} \sum_{i=1}^n \log_2 \frac{1}{P_X(X_i)}$ is an i.i.d. sum of random variables. The mean of each such random variable is $\mathbb{E}[\log_2 \frac{1}{P_X(X)}] = H(X)$. Therefore, due to property 1, we see that property 2 simply follows from the law of large numbers.¹
3. By the definition of the typical set, if $\mathbf{x} \in \mathcal{T}_n(\epsilon)$ then $P_{\mathbf{X}}(\mathbf{x}) \geq 2^{-n(H(X)+\epsilon)}$. Since any probability is at most one, we have

$$\begin{aligned}
1 &\geq \mathbb{P}[\mathbf{X} \in \mathcal{T}_n(\epsilon)] \\
&= \sum_{\mathbf{x} \in \mathcal{T}_n(\epsilon)} P_{\mathbf{X}}(\mathbf{x}) \\
&\geq \sum_{\mathbf{x} \in \mathcal{T}_n(\epsilon)} 2^{-n(H(X)+\epsilon)} \\
&= |\mathcal{T}_n(\epsilon)| \cdot 2^{-n(H(X)+\epsilon)}.
\end{aligned}$$

Re-arranging gives the third property.

4. By the definition of the typical set, if $\mathbf{x} \in \mathcal{T}_n(\epsilon)$ then $P_{\mathbf{X}}(\mathbf{x}) \leq 2^{-n(H(X)-\epsilon)}$. Writing property 2 as $\mathbb{P}[\mathbf{X} \in \mathcal{T}_n(\epsilon)] = 1 - o(1)$, we obtain

$$\begin{aligned}
1 - o(1) &= \mathbb{P}[\mathbf{X} \in \mathcal{T}_n(\epsilon)] \\
&= \sum_{\mathbf{x} \in \mathcal{T}_n(\epsilon)} P_{\mathbf{X}}(\mathbf{x}) \\
&\leq \sum_{\mathbf{x} \in \mathcal{T}_n(\epsilon)} 2^{-n(H(X)-\epsilon)} \\
&= |\mathcal{T}_n(\epsilon)| \cdot 2^{-n(H(X)-\epsilon)}.
\end{aligned}$$

Re-arranging gives the fourth property.

Implication.

- The above suggests a very simple source coding scheme:
 - Map each typical sequence to a unique integer in $\{1, \dots, M-1\}$;
 - Map each non-typical sequence to a “dummy value” M .

The decoder lets $\hat{\mathbf{X}}$ be arbitrary for $m = M$, whereas for $m < M$ it simply outputs the corresponding typical sequence.

- Clearly this scheme is possible if $M = |\mathcal{T}_n(\epsilon)| + 1$, and yields error probability $P_e \leq \mathbb{P}[\mathbf{X} \notin \mathcal{T}_X]$, which is arbitrarily small for sufficiently large n by property 2 above.
- Substituting property 3 above gives $M = 2^{n(H(X)+\epsilon)} + 1$. Since ϵ may be arbitrarily small and the rate is $\frac{1}{n} \log_2 M$, we deduce that *we can get arbitrarily small error probability with a rate arbitrarily close to $H(X)$* .
 - Note that this only holds as $n \rightarrow \infty$. The closer we take the rate to $H(X)$ (smaller ϵ) and the smaller we take the error probability, the higher we might need to take the block length n .

¹The *law of large numbers* states that the average of n i.i.d. random variables is arbitrarily close to its mean with probability approaching one. See the prerequisite material document for a more formal statement.

A possible thought process behind deriving $\mathcal{T}_n(\epsilon)$.

- Since we only have a finite number of messages $\{1, \dots, M\}$ to work with, it makes sense to assign them only to the most probable sequences, i.e., those such that

$$P_{\mathbf{X}}(\mathbf{x}) \geq \gamma$$

for some $\gamma > 0$. How high can we make γ while still ensuring the set has high probability?

- After staring at this for a while, one becomes tempted to take the log (to simplify the product in $P_{\mathbf{X}}(\mathbf{x}) = \prod_{i=1}^n P_X(x_i)$) to get the equivalent condition

$$\sum_{i=1}^n \log_2 P_X(x_i) \geq \log_2 \gamma.$$

- Recognizing $\sum_{i=1}^n \log_2 P_X(X_i)$ as a sum of independent random variables with mean $H(X)$, one realizes that $\log_2 \gamma$ should be chosen as roughly $-nH(X)$ by the law of large numbers. With some re-arranging, the original condition $P_{\mathbf{X}}(\mathbf{x}) \geq \gamma$ reduces to $P_{\mathbf{X}}(\mathbf{x}) \gtrsim 2^{-nH(X)}$.
- Since the law of large numbers works for deviations on both sides of the mean, one then realizes that things also work out if we use the two-sided version $\mathcal{T}_n(\epsilon)$.

(Optional) Alternative “one-sided typicality” proof.

- It is, in fact, not hard to see that we could get to the same “ R arbitrarily close to $H(X)$ ” result using a one-sided typicality notion like that in the above thought process. Specifically, consider the definition

$$\mathcal{T}'_n(\epsilon) = \left\{ \mathbf{x} \in \mathcal{X}^n : P_{\mathbf{X}}(\mathbf{x}) \geq 2^{-n(H(X)+\epsilon)} \right\}.$$

We still have $\mathbf{X} \in \mathcal{T}'_n(\epsilon)$ with probability approaching one, and the same upper bound on the total number of sequences satisfying it (by the same proofs as above).

- This variation arguably makes *more sense*, as it encodes all sequences whose value of $P_{\mathbf{X}}(\mathbf{x})$ is sufficiently high – it seems strange to ignore those that are the most probable!
- Nevertheless, two-sided typicality is more common in information theory proofs, and in certain other settings it is actually useful for the mathematical analysis.

3 Fano’s Inequality and a Converse Bound

Fano’s inequality.

- In the following, X denotes a generic random variable (or vector), and \hat{X} can be thought of as any estimate of X . At this stage, these should not be thought of as necessarily directly related to the definitions in the previous sections.
- Fano’s inequality relates two fundamental quantities:
 - The conditional entropy $H(X|\hat{X})$;

- The “error probability” $P_e = \mathbb{P}[\hat{X} \neq X]$.

Intuitively, if $H(X|\hat{X})$ is “large”, then \hat{X} does not reveal much information about X , so P_e must not be too small either (if it were very small, then knowing \hat{X} would tell us a lot about X !).

Similarly, if P_e is small then $H(X|\hat{X})$ should be small too. As an extreme example, if $P_e = 0$ then $\hat{X} = X$ and therefore $H(X|\hat{X}) = 0$.

- **Claim (Fano’s Inequality).** For any discrete random variables X and \hat{X} on a common finite alphabet \mathcal{X} , we have

$$H(X|\hat{X}) \leq H_2(P_e) + P_e \log_2 (|\mathcal{X}| - 1),$$

where $H_2(\alpha) = \alpha \log_2 \frac{1}{\alpha} + (1 - \alpha) \log_2 \frac{1}{1 - \alpha}$ is the binary entropy function.

- Intuition. To resolve the uncertainty in X given \hat{X} , we can first ask whether the two are equal, which bears uncertainty $H_2(P_e)$. In the case that they differ, which only occurs a fraction P_e of the time, the remaining uncertainty is at most $\log_2 (|\mathcal{X}| - 1)$, since the uniform distribution maximizes entropy.
- Formal proof. Defining the error indicator random variable $E = \mathbb{1}\{X \neq \hat{X}\}$, we have

$$\begin{aligned} H(X|\hat{X}) &\stackrel{(a)}{=} H(X, E|\hat{X}) \\ &\stackrel{(b)}{=} H(E|\hat{X}) + H(X|\hat{X}, E) \\ &\stackrel{(c)}{\leq} H(E) + H(X|\hat{X}, E) \\ &\stackrel{(d)}{=} H_2(P_e) + P_e H(X|\hat{X}, E = 1) + (1 - P_e) H(X|\hat{X}, E = 0) \\ &\stackrel{(e)}{\leq} H_2(P_e) + P_e \log_2 (|\mathcal{X}| - 1), \end{aligned}$$

where:

- (a) holds since E is a deterministic function of (X, \hat{X}) . More formally, the chain rule gives $H(X, E|\hat{X}) = H(X|\hat{X}) + H(E|X, \hat{X})$, and then we have $H(E|X, \hat{X}) = 0$.
- (b) follows from the chain rule.
- (c) holds since conditioning reduces entropy.
- (d) uses $H(E) = H_2(P_e)$ for the first term (recall that $H_2(p)$ is defined to be the entropy of a Bernoulli(p) random variable) and the definition of conditional entropy for the second term.
- (e) follows since X has no uncertainty given \hat{X} when $E = 0$, and takes one of $|\mathcal{X}| - 1$ values given \hat{X} when $E = 1$.

Implication for source coding.

- **Theorem.** In the block source coding problem with a discrete memoryless source P_X , if $R < H(X)$, then $P_e = \mathbb{P}[\hat{\mathbf{X}} \neq \mathbf{X}]$ cannot be made arbitrarily small as $n \rightarrow \infty$.
 - Holds for any code design! Results of this type are called *converse bounds* or *impossibility results*. (The entropy bound of the previous lecture was also of this type).
 - This is a statement of mathematical impossibility *regardless of computation, storage, etc.*

- Proof: Start with Fano's inequality with $(\mathbf{X}, \hat{\mathbf{X}})$ playing the role of the generic variables (X, \hat{X}) :

$$H(\mathbf{X}|\hat{\mathbf{X}}) \leq H_2(P_e) + P_e \log_2(|\mathcal{X}^n| - 1)$$

where $\mathcal{X}^n = \mathcal{X} \times \dots \times \mathcal{X}$ (n times) is the set of all length- n sequences with symbols in \mathcal{X} . For convenience, we upper bound $\log_2(|\mathcal{X}^n| - 1) \leq \log_2 |\mathcal{X}^n| = n \log_2 |\mathcal{X}|$, and also $H_2(P_e) \leq 1$ (binary entropy is at most one bit), to obtain

$$H(\mathbf{X}|\hat{\mathbf{X}}) \leq P_e \cdot n \log_2 |\mathcal{X}| + 1$$

This is a weakened form of Fano's inequality.

Recall the definition of mutual information, $I(\mathbf{X}; \hat{\mathbf{X}}) = H(\mathbf{X}) - H(\mathbf{X}|\hat{\mathbf{X}})$. Upper bounding $H(\mathbf{X}|\hat{\mathbf{X}})$ according to the previous display equation gives

$$I(\mathbf{X}; \hat{\mathbf{X}}) \geq H(\mathbf{X}) - P_e \cdot n \log_2 |\mathcal{X}| - 1.$$

On the other hand, the definition of mutual information (in the “other” form) gives

$$\begin{aligned} I(\mathbf{X}; \hat{\mathbf{X}}) &= H(\hat{\mathbf{X}}) - H(\hat{\mathbf{X}}|\mathbf{X}) \\ &\stackrel{(a)}{\leq} H(\hat{\mathbf{X}}) \\ &\stackrel{(b)}{\leq} nR \end{aligned}$$

where (a) uses the non-negativity of (conditional) entropy, and (b) uses the fact that $\hat{\mathbf{X}}$ takes on one of $M = 2^{nR}$ values (and entropy is always upper bounded by log of the number of values). Combining the previous two equations with $H(\mathbf{X}) = nH(X)$ (easily verified by the i.i.d. assumption on \mathbf{X} , i.e., the memoryless property), we get

$$nR \geq nH(X) - P_e \cdot n \log_2 |\mathcal{X}| - 1,$$

or equivalently,

$$P_e \geq \frac{1}{\log_2 |\mathcal{X}|} \left(H(X) - R - \frac{1}{n} \right).$$

Therefore, if $R < H(X)$ then P_e cannot tend to zero as $n \rightarrow \infty$.

- **A minor technical detail:** On Page 2, we stated the source coding theorem for arbitrary n , not only $n \rightarrow \infty$. However, the result for $n \rightarrow \infty$ implies the result for arbitrary n . Indeed, the only way to get arbitrarily small error probability at finite n is to have $P_e = 0$. But if we can achieve $P_e = 0$ at some rate with finite block length, we can also achieve it as $n \rightarrow \infty$ by simply using that code many times in succession.