Lecture 7 The typical set and the Source Coding Theorem

September 20th, 2022

Outline

Review

Theorem (Shannon's source coding theorem)

Let X be an random variable with entropy $H(X) = H$ bits. Given $\epsilon > 0$ and $0 < \delta < 1$, there exists a positive integer N_0 such that for $N > N_0$,

$$|\frac{1}{N} H_\delta(X^N) - H| < \epsilon.$$

Outline

Why does increasing N help? Let's examine long strings from X^N.

Why does increasing N help? Let's examine long strings from X^N.

Let us consider the case of coin flip problem introduced in last lecture, where $N = 100$ and $p_1 = 0.1$.

Why does increasing N help? Let's examine long strings from X^N.

Let us consider the case of coin flip problem introduced in last lecture, where $N = 100$ and $p_1 = 0.1$.

The probability of a string **x** that contains r 1s and $N - r$ 0s is

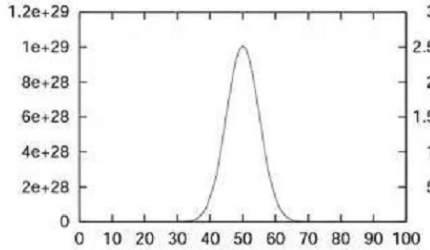
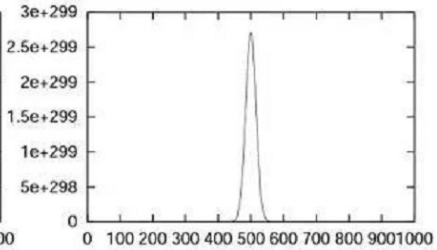
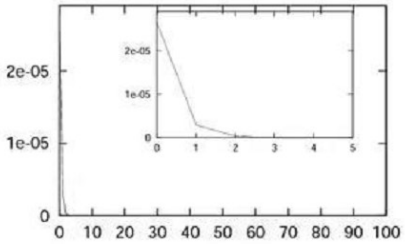
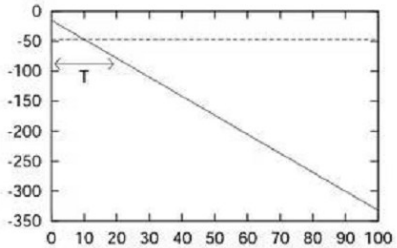
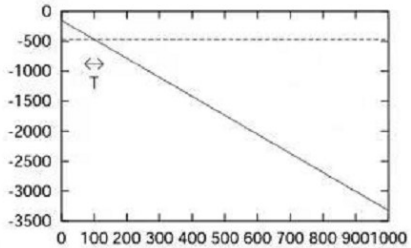
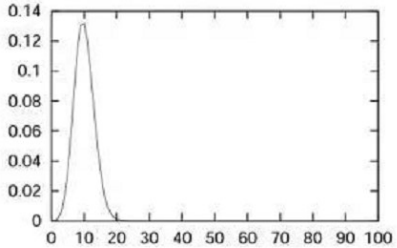
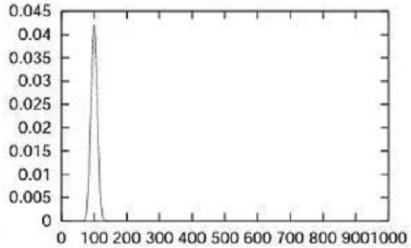
$$P(\mathbf{x}) = p_1^r (1 - p_1)^{N-r}.$$

The number of strings that contain r 1s is

$$n(r) = \binom{N}{r}.$$

So the number of 1s, r, has a binomial distribution:

$$P(r) = \binom{N}{r} p_1^r (1 - p_1)^{N-r}.$$

	$N = 100$	$N = 1000$
$n(r) = \binom{N}{r}$		
$P(\mathbf{x}) = p_1^r(1-p_1)^{N-r}$		
$\log_2 P(\mathbf{x})$		
$n(r)P(\mathbf{x}) = \binom{N}{r}p_1^r(1-p_1)^{N-r}$		
	r	r

Let us define typicality for an arbitrary ensemble X with alphabet $\mathcal{X}$. Our definition of a typical string involve the string's probability.

Let us define typicality for an arbitrary ensemble X with alphabet $\mathcal{X}$. Our definition of a typical string involve the string's probability.

A long string of N symbols will usually contain about $p_1 N$ occurrences of the first symbol, $p_2 N$ occurrences of the second, etc.

Let us define typicality for an arbitrary ensemble X with alphabet $\mathcal{X}$. Our definition of a typical string involve the string's probability.

A long string of N symbols will usually contain about $p_1 N$ occurrences of the first symbol, $p_2 N$ occurrences of the second, etc.

The probability of this string is roughly

$$p(\mathbf{x})_{typ} = P(x_1)P(x_2)P(x_3)\ldots P(x_N) \approx p_1^{p_1 N} p_2^{p_2 N} \cdots p_I^{p_I N}$$

so that the information cotent of atypical string is

$$\log_2 \frac{1}{P(\mathbf{x})} \approx N \sum_i p_i \log_2 \frac{1}{p_i} = NH.$$

Definition

We call the set typical elements *the typical set*, $T_{N\beta}$:

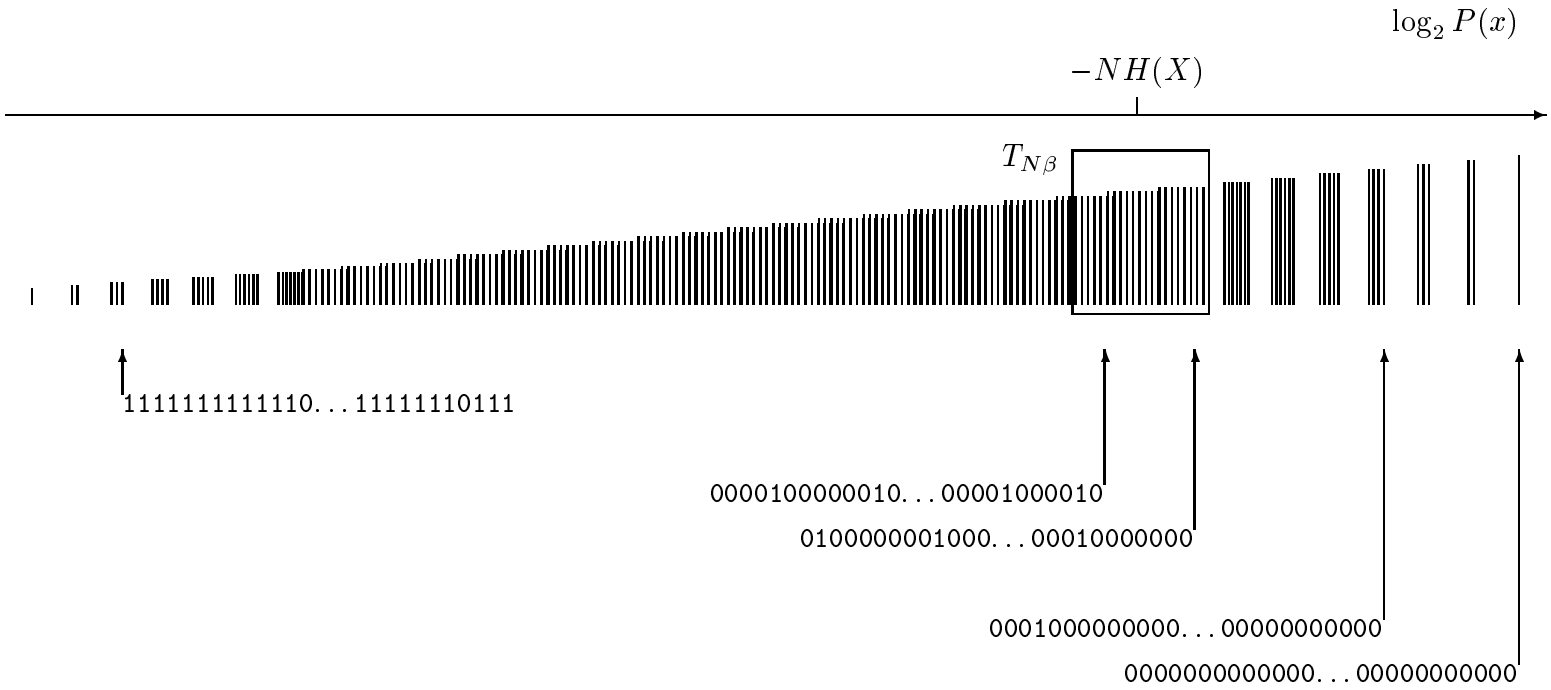
$$T_{N\beta} := \{\mathbf{x} \in \mathcal{X}^N : |\frac{1}{N} \log_2 \frac{1}{P(\mathbf{x})} - H| < \beta\}.$$

Asymptotic equipartition property

For an ensemble of N independent identically distributed random variables $X^N := (X_1, X_2, \cdots, X_N)$, with N sufficiently large, the outcome $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is almost certain to belong to a subset of $\mathcal{X}^N$ having only $2^{NH(X)}$ members, each having probability 'close to' $2^{-NH(X)}$.

The difference between the smallest δ-sufficient subset and the typical set

Consider coin flip problem again. The typical sequences in this case are the sequence in which the proportion of 0's is close to 0.9. However, this does not includes the sequence of all 0's, which is the most likely single sequence. The smallest δ-sufficient subset includes all the most probable sequences and therefore includes the sequence of all 0's.


$\log_2 P(x)$
$-NH(X)$
$T_{N\beta}$
1111111111110...11111110111
0000100000010...00001000010
0100000001000...00010000000
0001000000000...00000000000
0000000000000...00000000000

Why do we introduce the typical set?

The best choice of subset for block compression is (by definition) S_δ, not a typical set. So why did we bother introducing the typical set?

Why do we introduce the typical set?

The best choice of subset for block compression is (by definition) S_δ, not a typical set. So why did we bother introducing the typical set?

The answer is, we can count the typical set.

Outline

Theorem (Weak law of large numbers)

Let $X_1, \cdots, X_n$ be N independent random variables, having common mean μ and common variance σ^2. Then

$$P((\frac{1}{N}\sum_{i=1}^{N} X_i - \mu)^2 \geq \alpha) \leq \sigma^2/\alpha N.$$

We again define the typical set with parameters N and β thus:

$$T_{N\beta} := \{\mathbf{x} \in \mathcal{X}^N : [\frac{1}{N} \log_2 \frac{1}{P(\mathbf{x})} - H]^2 < \beta^2\}.$$

For all $\mathbf{x} \in T_{N\beta}$, the probability of $\mathbf{x}$ satisfies

$$2^{-N(H+\beta)} < P(\mathbf{x}) < 2^{-N(H-\beta)}.$$

So from the weak law of large numbers, we have that

$$P(\mathbf{x} \in T_{N\beta}) \geq 1 - \frac{\sigma^2}{\beta^2 N}.$$

Step 1. $\frac{1}{N}H_\delta(X^N) < H + \epsilon$ when N is large enough.

- The set $T_{N\beta}$ is not the best subset for compression. So the size of $T_{N\beta}$ gives an upper bound on H_δ.
- We shall show how small $h_\delta(X^N)$ must be by calculating how big $T_{N\beta}$ could possibly be.
- The smallest possible probability that a member of $T_{N\beta}$ can have is $2^{-N(H+\beta)}$, and the total probability contained by $T_{N\beta}$ can't be any bigger than 1.
- So $|T_{N\beta}|2^{-N(H+\beta)} < 1$, that is, the size of the typical set is bounded by
$$|T_{N\beta}| < 2^{N(H+\beta)}.$$
- If we set $\beta = \epsilon$ and N_0 such that $\frac{\sigma^2}{\epsilon^2 N_0} \leq \delta$, then $P(T_{N\beta}) \geq 1 - \delta$, and the set $T_{N\beta}$ becomes a witness to the fact that $H_\delta(X^N) \leq \log_2 |T_{N\beta}| < N(H + \epsilon)$.

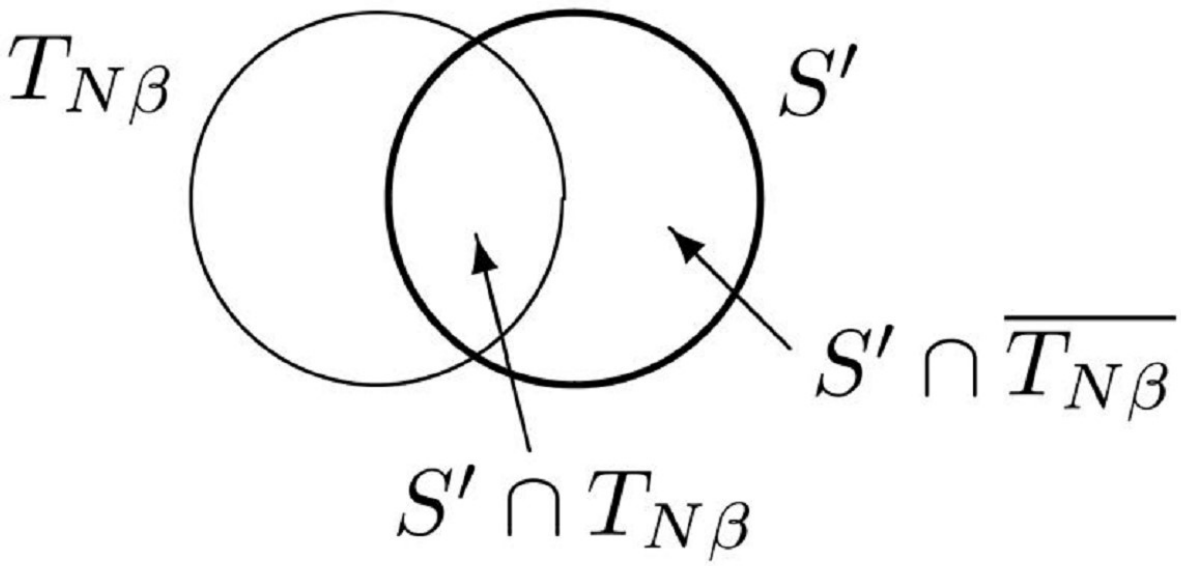
Step 2. $\frac{1}{N} H_\delta(X^N) > H - \epsilon$ when N is large enough.

Imagine that someone claims this is not so, which means that for any N, the smallest δ-sufficient S_δ is smaller then the above inequality would allow. We can make use of our typical set to show that they must be mistaken.

Step 2. $\frac{1}{N} H_\delta(X^N) > H - \epsilon$ when N is large enough.

Imagine that someone claims this is not so, which means that for any N, the smallest δ-sufficient S_δ is smaller then the above inequality would allow. We can make use of our typical set to show that they must be mistaken.

Remember that we are free to set β to any value we choose. We will set $\beta = \epsilon/2$, so that our task is to prove that a subset S' having $|S'| \leq 2^{N(H-2\beta)}$ and achieving $P(\mathbf{x} \in S') \geq 1 - \delta$ cannot exist (for N greater than an N_0 that we will specify).



So, let us consider the probability of falling in this rival smaller subset S'. The probability of the subset S' is

$$P(\mathbf{x} \in S') = P(\mathbf{x} \in S' \cap T_{N\beta}) + P(\mathbf{x} \in S' \cap \overline{T_{N\beta}}),$$

where $\overline{T_{N\beta}}$ denotes the complement $\{\mathbf{x} \notin T_{N\beta}\}$.

- Now we have that

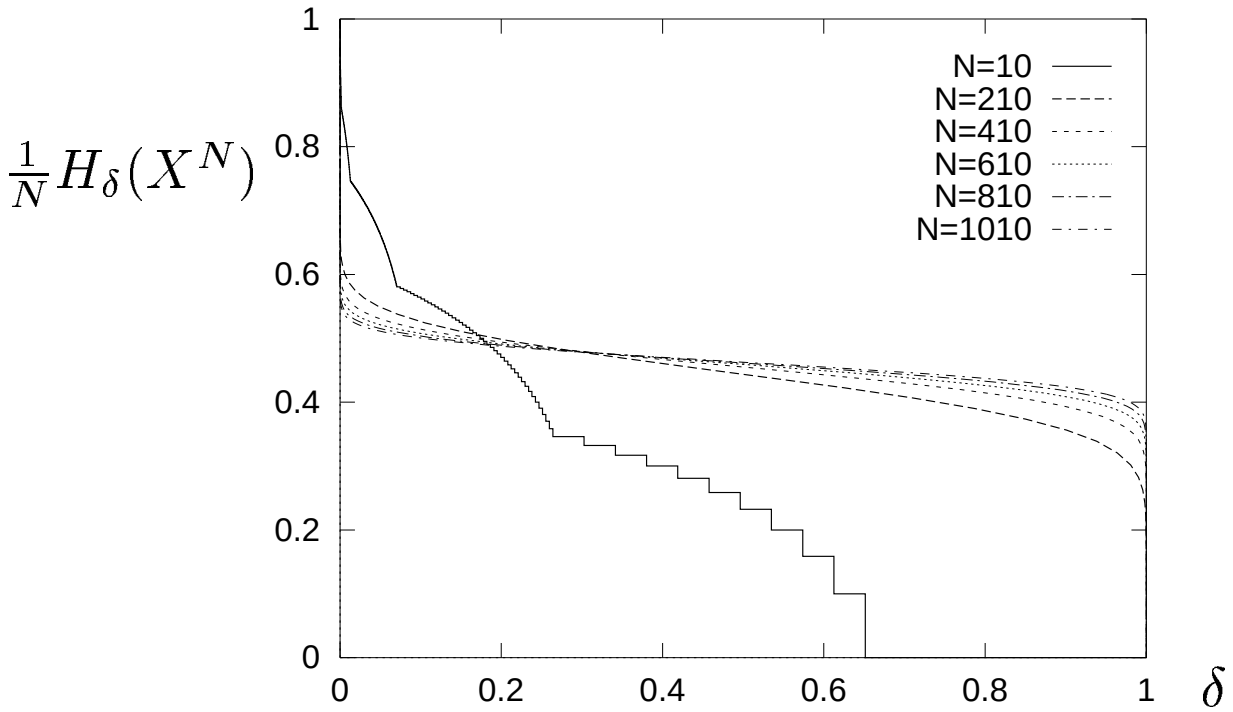
$$P(\mathbf{x} \in S') = P(\mathbf{x} \in S' \cap T_{N\beta}) + P(\mathbf{x} \in S' \cap \overline{T_{N\beta}}).$$

- The maximum value of the first term is found if $S' \cap T_{N\beta}$ contains $2^{N(H-2\beta)}$ outcomes all with the maximum probability, $2^{-N(H-\beta)}$.
- The maximum value of the first term can have is $P(\mathbf{x} \notin T_{N\beta})$. So:

$$P(\mathbf{x} \in S') \leq 2^{N(H-2\beta)} 2^{-N(H-\beta)} + \frac{\sigma^2}{\beta^2 N} = 2^{-N\beta} + \frac{\sigma^2}{\beta^2 N}.$$

- We can now set $\beta = \epsilon/2$ and N_0 such that $P(\mathbf{x} \in S') < 1 - \delta$, which shows that S' cannot satisfy the definition of a sufficient subset S_δ.
- Thus any subset S' with size $|S'| \leq 2^{N(H-\epsilon)}$ has probability less than $1 - \delta$, so by the definition of H_δ, $H_\delta > N(H - \epsilon)$.

Thus for large enough N, the function $\frac{1}{N}H_\delta(X^N)$ is essentially a constant function of δ.



Remarks

The source coding theorem has two parts, $\frac{1}{N}H_\delta(X^N) < H + \epsilon$, and $\frac{1}{N}H_\delta(X^N) > H - \epsilon$. Both results are interesting.

- The first part tells us that even if the probability of error δ is extremely small, the number of bits per symbol $\frac{1}{N}H_\delta(X^N)$ needed to specify a long N-symbol string **x** with vanishingly small error probability does not have to exceed $H + \epsilon$ bits. We need to have only a tiny tolerance for error, and the number of bits required drops significantly from $H_0(X)$ to $H + \epsilon$.
- What happens if we are yet more tolerant to compression errors? The proof of the second part tells us that if we are using the typical set to code, even δ is very close to 1, so that errors are made most of the time, the average number of bits per symbol needed to specify **x** must still be at least $H - \epsilon$ bits.
- These two extreme tells us that regardless of our specify **x** is H bits; no more or no less.

Remarks

- In we use variable-length compression, we can archive the same compression rate while it is not lossy. Check Theorem 3.2.1 in the textbook.
- The compression scheme described in the proof is impractical. From the next lecture, we shall discuss practical compression algorithms.