Source Coding

- binomial coefficient
- encoders and decoders
- source coding

Coin Flips and Typical Outcomes

- consider n (possibly biased) coin flips

$$X_i \overset{i.i.d.}{\sim} \text{bernoulli}(\theta)$$

$$\boldsymbol{x} \sim \text{bernoulli}(\theta)$$

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i) = \theta^k (1-\theta)^{n-k}$$

- example:

$$\theta = 0.9, n = 1000$$

$$\boldsymbol{x} = 0101111110101110...$$

- how many possible outcomes? 2^n
- how many outcomes with k ones? $\binom{n}{k}$

- probability of an outcome with k ones:

$$k \sim \text{binomial}(n, \theta)$$

$$\mathbb{P}(k) = \binom{n}{k} \theta^k (1-\theta)^{n-k}$$

- bounding deviations from the mean: $\widehat{\theta}_{\text{ML}} = \frac{k}{n}$

$$\mathbb{P}(|\widehat{\theta} - \theta| \geq \delta) \leq 2e^{-2\delta^2 n}$$

- with high probability, k will be close to $n\theta$

$$\mathbb{P}(|k - n\theta| \geq m) \leq 2e^{-2m^2/n}$$

Coin Flips and Typical Outcomes

- $1 dollar lottery tickets

$$\boldsymbol{x} \sim \text{bernoulli}(0.9) \qquad n = 1000$$

- lottery 1: win if you guess $\boldsymbol{x}$ exactly.
- you only have $1. which ticket do you buy?

$$\boldsymbol{x} = 1111111...$$

- you find $1000. which tickets do you buy next?

$$\boldsymbol{x} = 0111111...$$
$$\boldsymbol{x} = 1011111...$$
$$\boldsymbol{x} = 1101111...$$

- then which tickets?

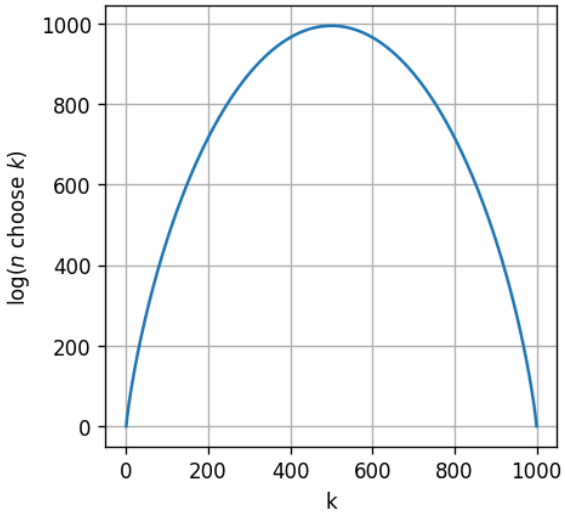
- lottery 2 - win if you guess number of 1's correctly
  - probably guess $k = 900$
  - mode of binomial distribution is $\approx n\theta$.

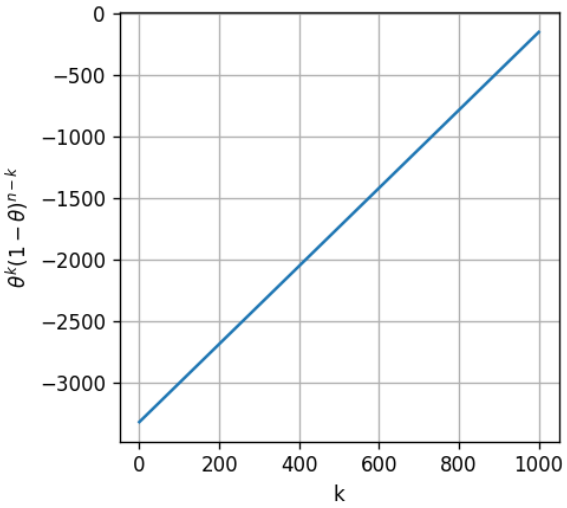
$$\mathbb{P}(k) = \binom{n}{k} \theta^k (1-\theta)^{n-k}$$

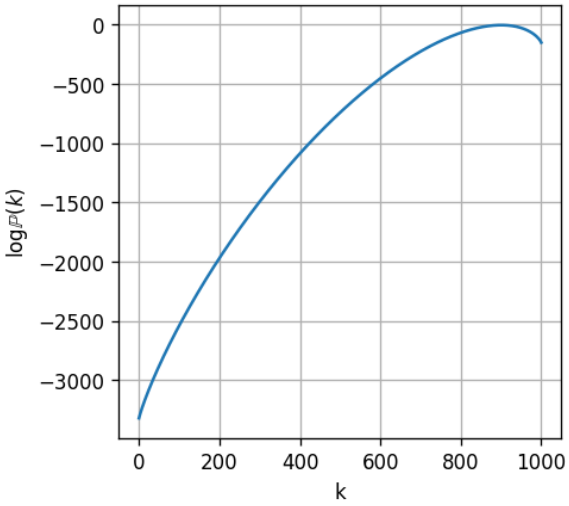
Binomial Distribution

$$\mathbb{P}(k) = \binom{n}{k} \theta^k (1-\theta)^{n-k}$$

$$\theta = 0.9, n = 1000$$

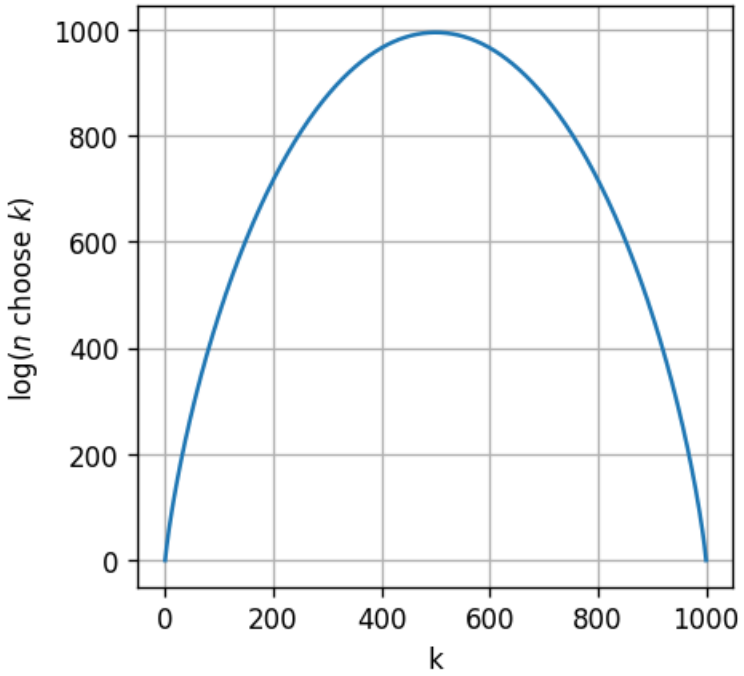






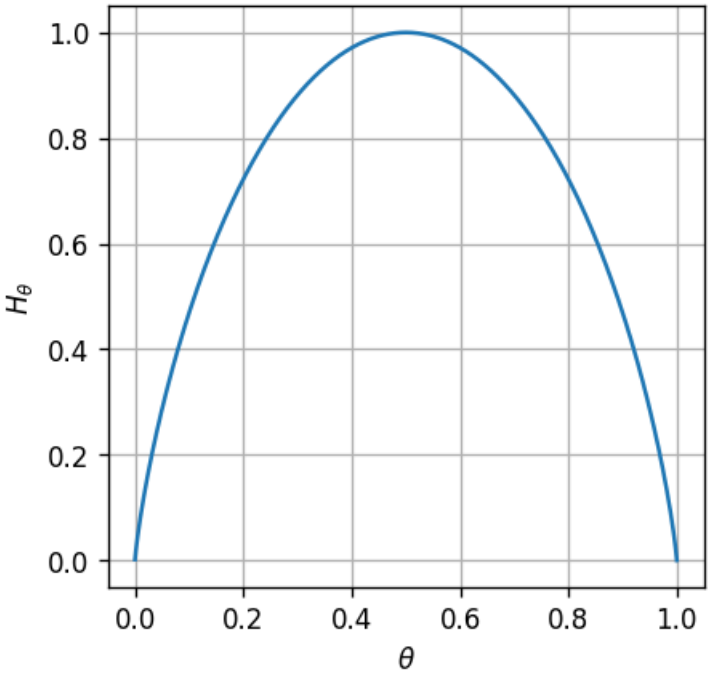
Binomial Coefficient, Binary Entropy

$$\log_2 \binom{n}{k}$$



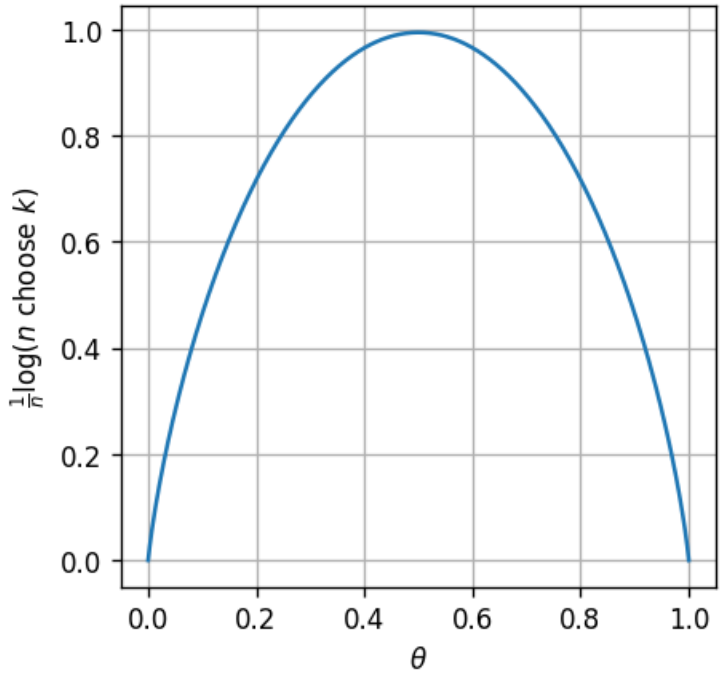
entropy of Bernoulli with bias θ

$$\theta \log_2 \tfrac{1}{\theta} + (1-\theta) \log_2 \tfrac{1}{1-\theta}$$



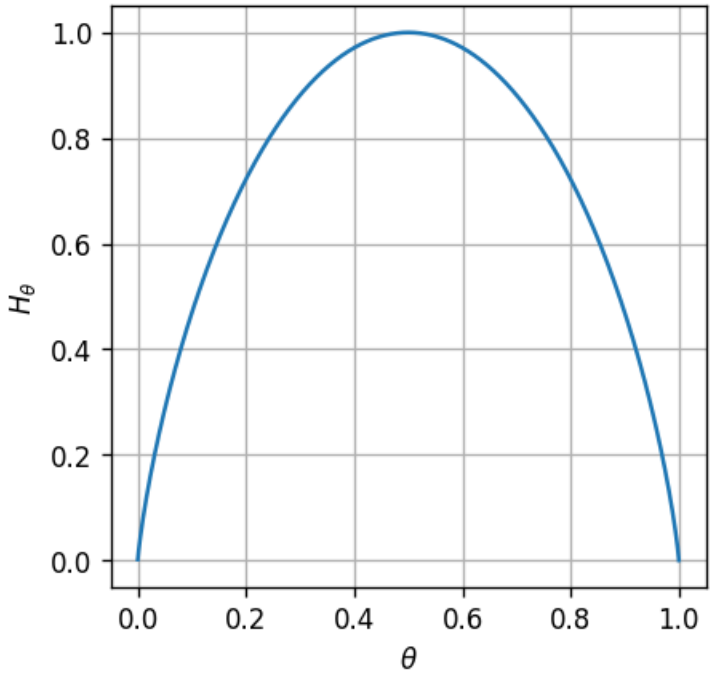
Binomial Coefficient, Binary Entropy

$\frac{1}{n}\log_2 \binom{n}{\theta n}$



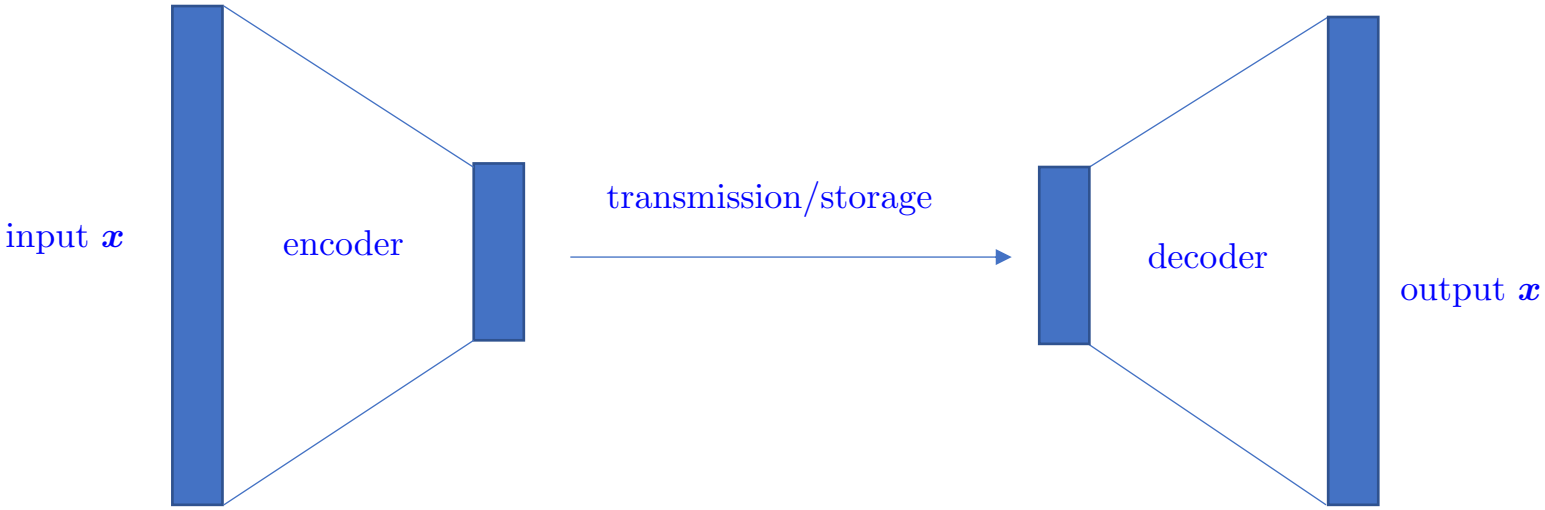
entropy of Bernoulli with bias θ

$\theta \log_2 \frac{1}{\theta} + (1-\theta)\log_2 \frac{1}{1-\theta}$

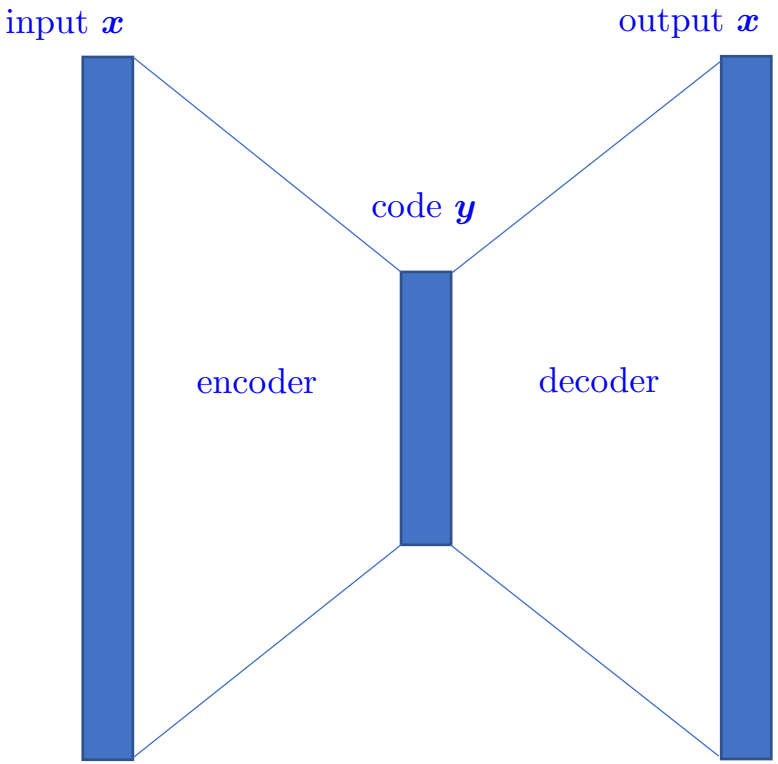


Traditional Encoders and Decoders

- common goal: make encoded representation more efficient
- encoder/decoder could be lossy
- compression: make y shorter than $\boldsymbol{x}$

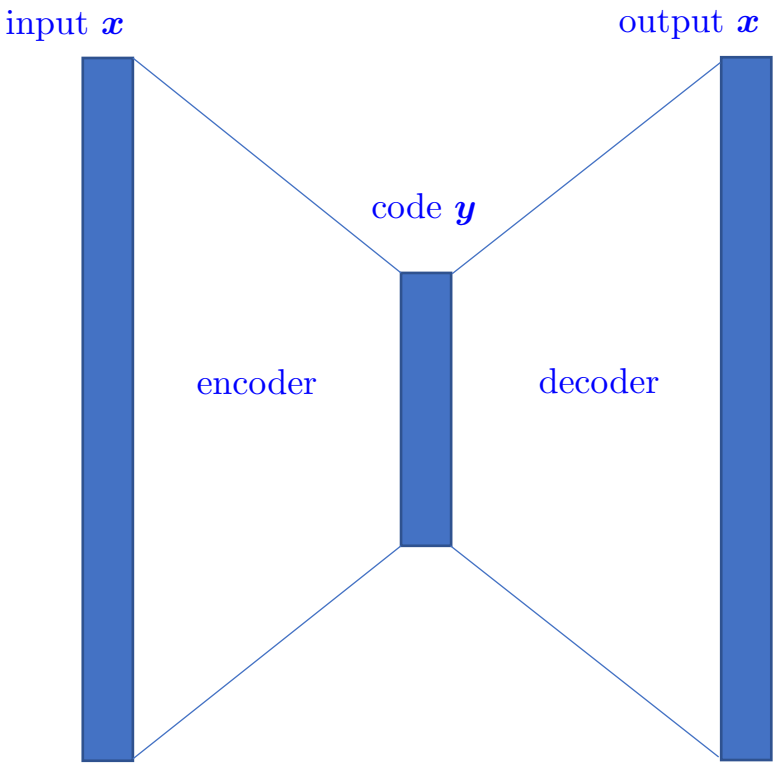


Traditional Encoders and Decoders

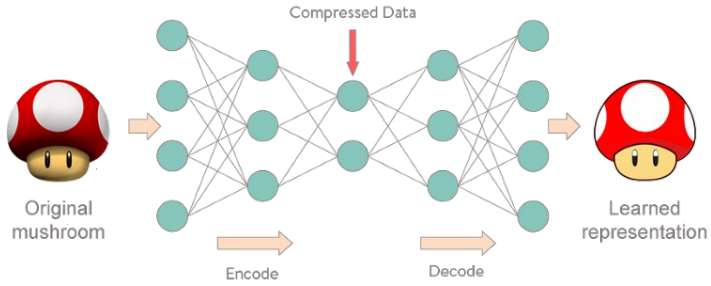


- desired property: output should equal the input
- encoder $f(\boldsymbol{x})$ and decoder $g(\boldsymbol{y})$ may just be a look-up table i.e., a *codebook*
- encoder $f(\boldsymbol{x})$ and decoder $g(\boldsymbol{y})$ designed by a person/non-ML algorithm
- encoder compresses data, decoder decompresses data

Autoencoders



- autoencoders use machine learning to learn $f(\boldsymbol{x})$ and $g(\boldsymbol{y})$
- usually $f(\boldsymbol{x})$ and $g(\boldsymbol{y})$ are artificial neural networks
- basic idea: output should equal input: $g(f(\boldsymbol{x})) \approx \boldsymbol{x}$
- autoencoders and traditional source coding usually have very different purposes



[credit: Venelin Valkov]

Source Coding

- Shannon's *source coding* theorem is fundamental to quantifying information
- applies to discrete data, i.e, discrete X
- interpretation:

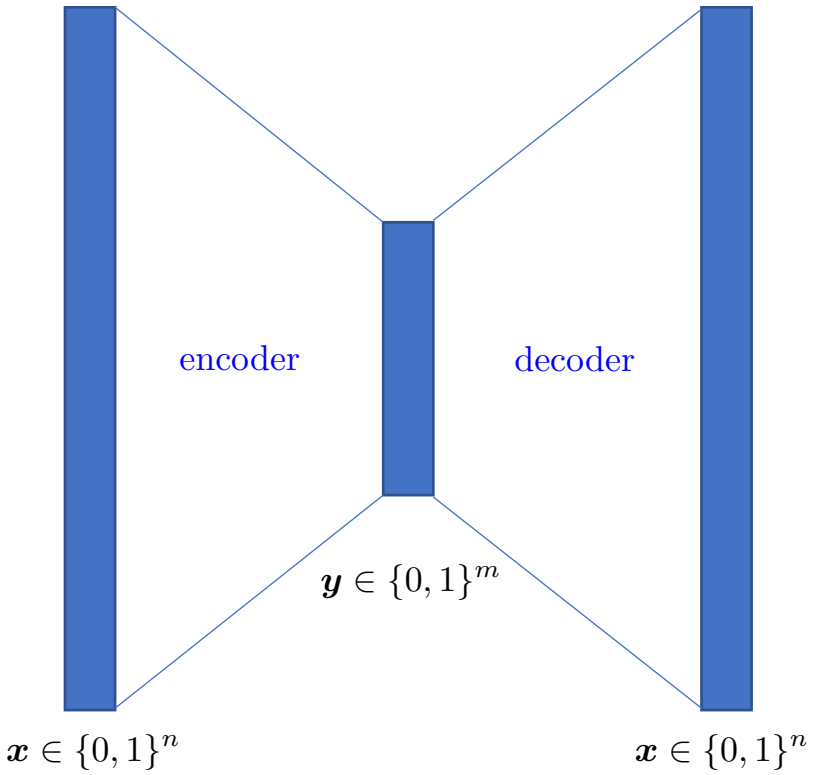
> n i.i.d. random variables each with entropy $H(X)$ can be compressed into $nH(X)$ bits with negligible risk of loss of information as n grows. Conversely, if compressed into fewer than $nH(X)$ bits, it is almost certain that information will be lost
>
> [adapted from Cover and Thomas, 2006]

- we will argue the first half of this statement

Source Coding

- *encoder* takes $\boldsymbol{x} \in \{0,1\}^n$ and maps to $\boldsymbol{y} \in \{0,1\}^m$
- *decoder* takes $\boldsymbol{y} \in \{0,1\}^m$ and maps to $\boldsymbol{x} \in \{0,1\}^n$

$$\boldsymbol{x} \sim \text{bernoulli}(0.8), n = 1000$$



- allow encoder/decoder to fail with some small probability
- if fewer than 700 ones, or more than 900, accept that encoder/decoder fail

$$\mathbb{P}(|k - n\theta| \geq m) \leq 2e^{-2m^2/n}$$

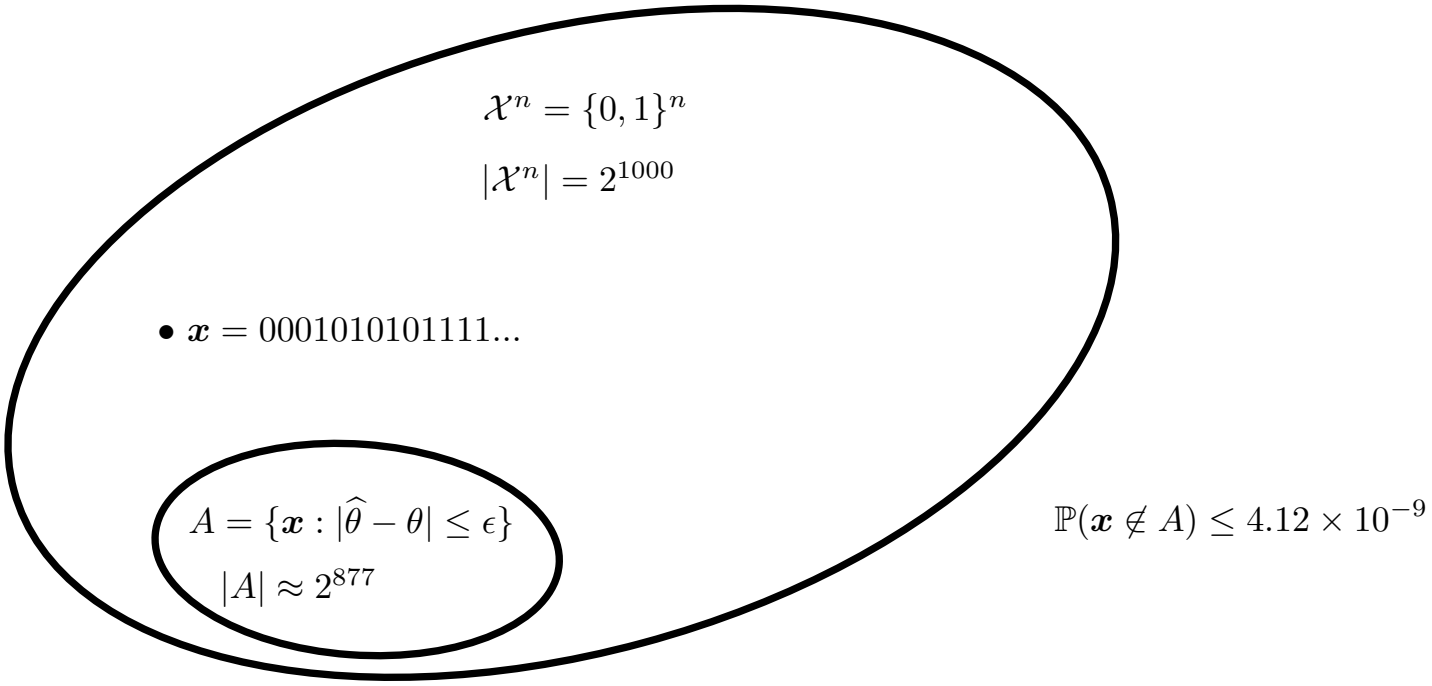
$$\mathbb{P}(|k - 800| \geq 100) \leq 2e^{-2\times 100^2/1000} \approx 4.12 \times 10^{-9}$$

- now we only need to encode vectors that have between 700 to 900 ones

$$\sum_{k=700}^{900} \binom{1000}{k} \approx 10^{264} \approx 2^{877}$$

$$\sum_{k=1}^{1000} \binom{1000}{k} = 2^{1000}$$

Source Coding



- enumerate all 2^{877} vectors that have between 700 and 900 ones

$$\boldsymbol{x} \in \{0,1\}^{1000} \longrightarrow \boldsymbol{y} \in \{0,1\}^{877} \longrightarrow \boldsymbol{x} \in \{0,1\}^{1000}$$

- compression ratio of $877/1000 = 0.877$

Source Coding

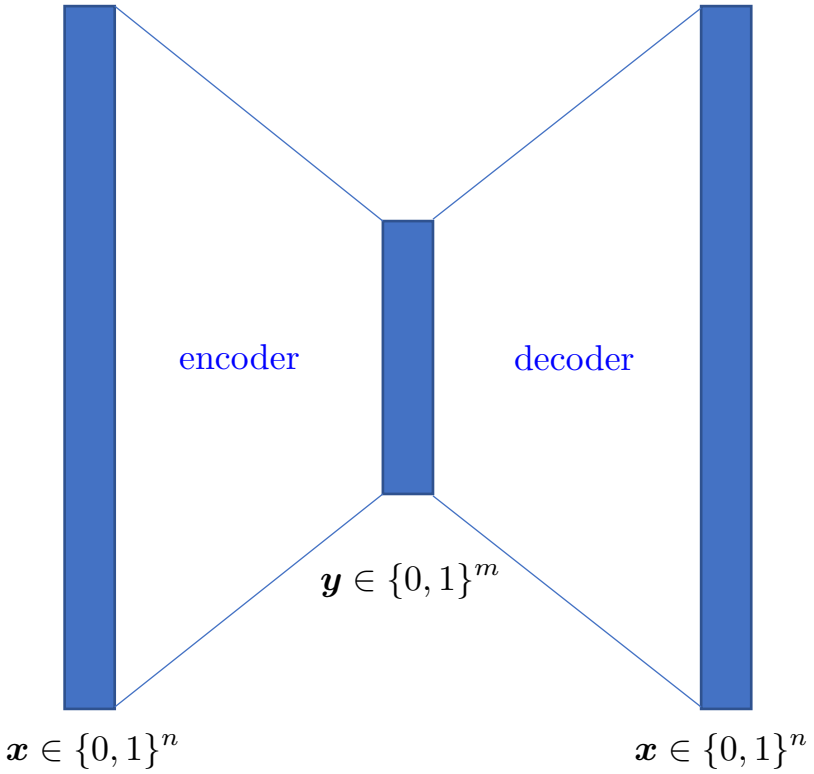
- what if n grows?
- example: $n = 100,000$

- if fewer than $79,000$ ones, or more than $81,000$, accept that encoder/decoder fail

$$\mathbb{P}(|k - 80,000| \geq 1000) \leq 2e^{-2\times 1000^2/100,000} \approx 4.12 \times 10^{-9}$$

- now we only need to encode vectors that have between $79,000$ to $81,000$

$$\sum_{k=79,000}^{81,000} \binom{100,000}{k} \leq 2,000 \times \binom{100,000}{79,000} \approx 2^{74,163}$$



- enumerate all 2^{74k} vectors

$$\boldsymbol{x} \in \{0,1\}^{100k} \longrightarrow \boldsymbol{y} \in \{0,1\}^{74k} \longrightarrow \boldsymbol{x} \in \{0,1\}^{100k}$$

- compression ratio of $74,163/100,000 = 0.7416$
- not so coincidentally, $H_2(0.8) = 0.722$ bits

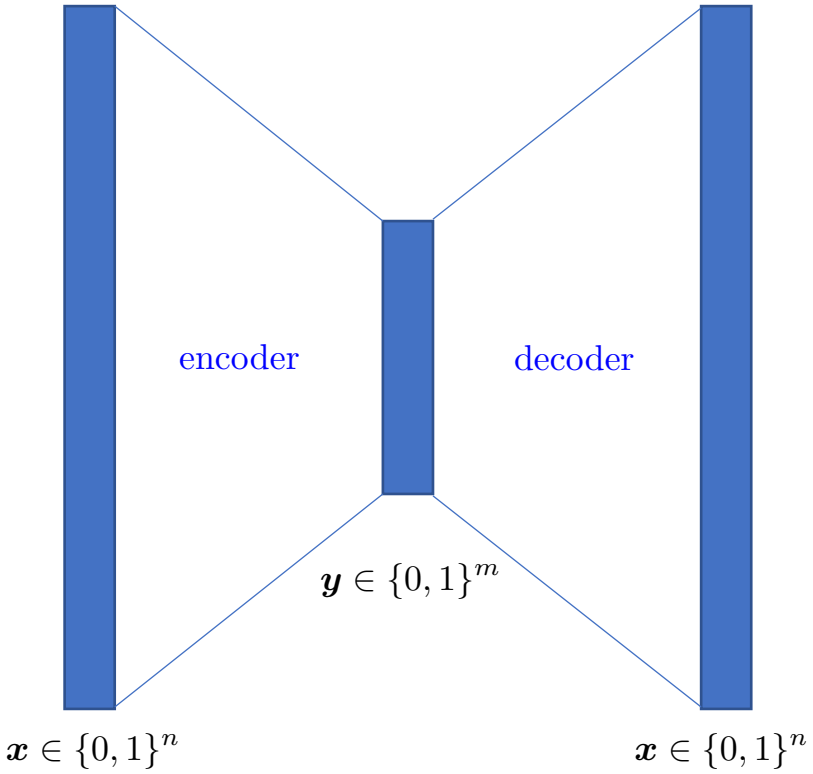
Source Coding

- what if n grows?
- setting $m = c\sqrt{n}$ fixes probability of error

$$\mathbb{P}(|k - n\theta| \geq m) \leq 2e^{-2m^2/n}$$
$$= 2e^{-2c^2}$$

- only need to encode vectors
that have between $n\theta - m$ and $n\theta + m$ ones.

$$\sum_{k=\theta n - m}^{\theta n + m} \binom{n}{k} \approx 2m \binom{n}{\theta n} \approx 2^{\log_2(2m)} \times 2^{nH(X)}$$



- compression ratio $\approx H(X)$
- we need to show that $\binom{n}{\theta n} \approx 2^{nH(X)}$

Source Coding

- approximating $\binom{n}{k}$

$$\log \binom{n}{k} = \log \frac{n!}{k!(n-k)!}$$

$$\approx \log \frac{n^n}{k^k (n-k)^{n-k}}$$

$$= (n-k) \log \frac{n}{n-k} + k \log \frac{n}{k}$$

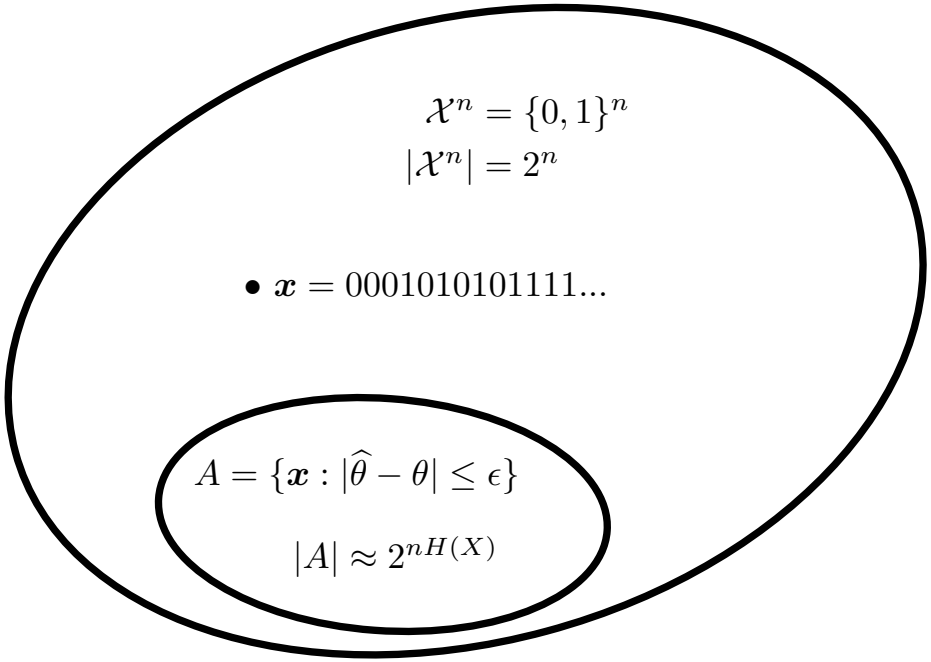
Log Factorial Approximation $\ln n! \approx n \ln n$

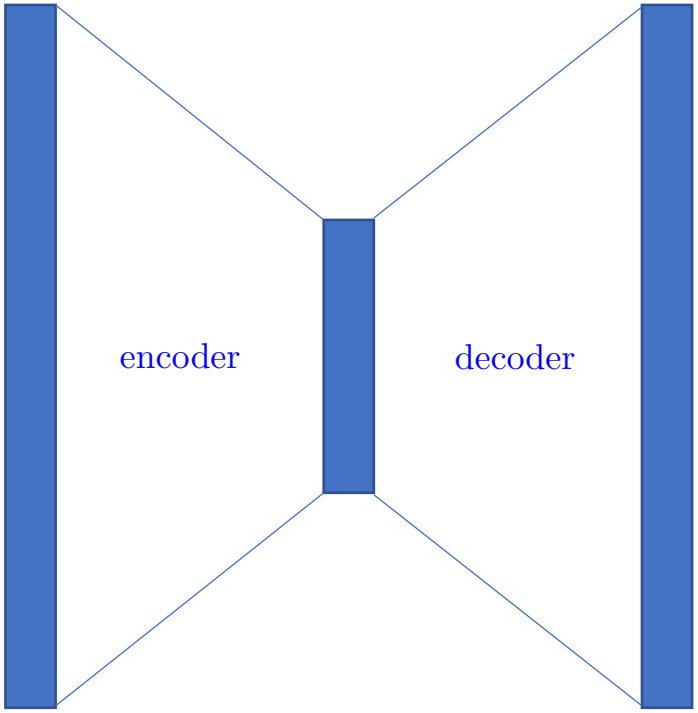
$$\ln n! = \ln(1 \cdot 2 \cdots (n-1)n) = \sum_{k=1}^{n} \ln k \leq n \ln n$$

$$\ln n! = \sum_{k=1}^{n} \ln k = \sum_{k=1}^{n} \int_{k-1}^{k} \ln k \, dt \geq \sum_{k=1}^{n} \int_{k-1}^{k} \ln t \, dt = n \ln n - n$$

$$\log \binom{n}{\theta n} \approx n(1-\theta) \log \frac{1}{1-\theta} + n\theta \log \frac{1}{\theta} \quad = nH(X)$$

Source Coding





n i.i.d. random variables each with entropy $H(X)$ can be compressed into $nH(X)$ bits with negligible risk of loss of information as n grows. Conversely, if compressed into fewer than $nH(X)$ bits, it is almost certain that information will be lost

[adapted from Cover and Thomas, 2006]

Asymptotic Equipartition Theorem

- we studied *typical* sequences $\boldsymbol{x}$ that satisfied:

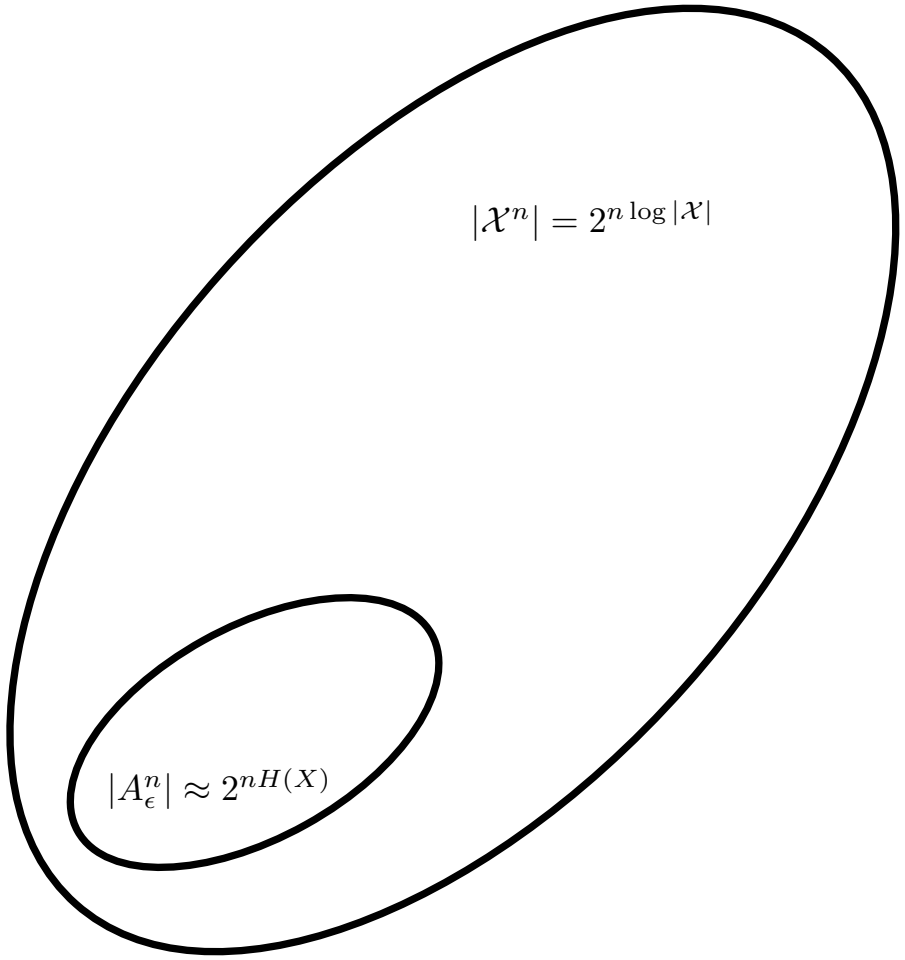
$$\mathbb{P}(|\widehat{\theta} - \theta| \geq \delta) \leq 2e^{-2\delta^2 n}$$

- another approach is to study the *entropy typical* $\boldsymbol{x}$:

$$A_\epsilon^n = \left\{ \boldsymbol{x} : \left| \frac{1}{n} \log_2 \left(\frac{1}{p(\boldsymbol{x})} \right) - H(X) \right| \leq \epsilon \right\}$$

for large n,

1. $\mathbb{P}(A_\epsilon^n) \geq 1 - \epsilon$
2. $|A_\epsilon^n| \leq 2^{n(H(X)+\epsilon)}$
3. $|A_\epsilon^n| \geq (1-\epsilon)2^{n(H(X)-\epsilon)}$



- we will show these properties in the activity

Differential Entropy

- definition - entropy:

$$H(X) = E\left[\log_2\left(\frac{1}{p(x)}\right)\right]$$

- discrete $x \in \mathcal{X}$:

$$H(X) = \sum_x p(x)\log_2\left(\frac{1}{p(x)}\right)$$

- amount of uncertainty in a random variable

- *differential entropy* is the equivalent definition of entropy, for continuous random variables

- continuous $x \in \mathbb{R}$:

$$H(X) = \int_x p(x)\log_2\left(\frac{1}{p(x)}\right)dx$$

- example: $X \sim \mathcal{N}(\mu, \sigma^2)$

$$H(X) = \frac{1}{2}\log_2(2\pi e\sigma^2) \text{ bits}$$