Information and Coding Theory

MA41024/ MA60020/ MA60262

Bibhas Adhikari

Spring 2022-23, IIT Kharagpur

Lecture 3
January 10, 2023

Entropy

Entropy Suppose X is a rv distributed over $\mathcal{X} = \{a_1, \ldots, a_n\}$ such that each value $x \in \mathcal{X}$ occurs with probability $p(x)$. Then the entropy of X is

$$H(X) = \sum_{x \in \mathcal{X}} p(x) \cdot \underbrace{\log_2 \left(\frac{1}{p(x)} \right)}_{\text{surprise}}$$

Entropy

Entropy Suppose X is a rv distributed over $\mathcal{X} = \{a_1, \ldots, a_n\}$ such that each value $x \in \mathcal{X}$ occurs with probability $p(x)$. Then the entropy of X is

$$H(X) = \sum_{x \in \mathcal{X}} p(x) \cdot \underbrace{\log_2 \left(\frac{1}{p(x)} \right)}_{\text{surprise}} = - \sum_{x \in \mathcal{X}} p(x) \cdot \log_2(p(x))$$

Entropy

Entropy Suppose X is a rv distributed over $\mathcal{X} = \{a_1, \ldots, a_n\}$ such that each value $x \in \mathcal{X}$ occurs with probability $p(x)$. Then the entropy of X is

$$H(X) = \sum_{x \in \mathcal{X}} p(x) \cdot \underbrace{\log_2 \left(\frac{1}{p(x)} \right)}_{\text{surprise}} = - \sum_{x \in \mathcal{X}} p(x) \cdot \log_2(p(x))$$

Example Consider the entropy of coin toss with p as probability of head.

Entropy

Entropy Suppose X is a rv distributed over $\mathcal{X} = \{a_1, \ldots, a_n\}$ such that each value $x \in \mathcal{X}$ occurs with probability $p(x)$. Then the entropy of X is

$$H(X) = \sum_{x \in \mathcal{X}} p(x) \cdot \underbrace{\log_2 \left(\frac{1}{p(x)} \right)}_{\text{surprise}} = - \sum_{x \in \mathcal{X}} p(x) \cdot \log_2(p(x))$$

Example Consider the entropy of coin toss with p as probability of head. What happens if $p_i = 1/n$?

Entropy

Entropy Suppose X is a rv distributed over $\mathcal{X} = \{a_1, \ldots, a_n\}$ such that each value $x \in \mathcal{X}$ occurs with probability $p(x)$. Then the entropy of X is

$$H(X) = \sum_{x \in \mathcal{X}} p(x) \cdot \underbrace{\log_2 \left(\frac{1}{p(x)} \right)}_{\text{surprise}} = - \sum_{x \in \mathcal{X}} p(x) \cdot \log_2(p(x))$$

Example Consider the entropy of coin toss with p as probability of head. What happens if $p_i = 1/n$?

Proposition $0 \leq H(X) \leq \log(|\mathcal{X}|)$

Entropy

Entropy Suppose X is a rv distributed over $\mathcal{X} = \{a_1, \ldots, a_n\}$ such that each value $x \in \mathcal{X}$ occurs with probability $p(x)$. Then the entropy of X is

$$H(X) = \sum_{x \in \mathcal{X}} p(x) \cdot \underbrace{\log_2 \left(\frac{1}{p(x)} \right)}_{\text{surprise}} = - \sum_{x \in \mathcal{X}} p(x) \cdot \log_2(p(x))$$

Example Consider the entropy of coin toss with p as probability of head. What happens if $p_i = 1/n$?

Proposition $0 \leq H(X) \leq \log(|\mathcal{X}|)$

Proof Let Y be a rv which takes the value $1/p(x)$ with probability $p(x)$. Then

$$\sum_{x \in \mathcal{X}} p(x) \cdot \log \left(\frac{1}{p(x)} \right) = \mathbb{E}[\log(Y)] \leq \log(\mathbb{E}[Y])$$

Entropy

Entropy Suppose X is a rv distributed over $\mathcal{X} = \{a_1, \ldots, a_n\}$ such that each value $x \in \mathcal{X}$ occurs with probability $p(x)$. Then the entropy of X is

$$H(X) = \sum_{x \in \mathcal{X}} p(x) \cdot \underbrace{\log_2 \left(\frac{1}{p(x)} \right)}_{\text{surprise}} = - \sum_{x \in \mathcal{X}} p(x) \cdot \log_2(p(x))$$

Example Consider the entropy of coin toss with p as probability of head. What happens if $p_i = 1/n$?

Proposition $0 \leq H(X) \leq \log(|\mathcal{X}|)$

Proof Let Y be a rv which takes the value $1/p(x)$ with probability $p(x)$. Then

$$\sum_{x \in \mathcal{X}} p(x) \cdot \log \left(\frac{1}{p(x)} \right) = \mathbb{E}[\log(Y)] \leq \log(\mathbb{E}[Y]) = \log \left(\sum_{x \in \mathcal{X}} p(x) \cdot \frac{1}{p(x)} \right)$$

Entropy

Entropy Suppose X is a rv distributed over $\mathcal{X} = \{a_1, \ldots, a_n\}$ such that each value $x \in \mathcal{X}$ occurs with probability $p(x)$. Then the entropy of X is

$$H(X) = \sum_{x \in \mathcal{X}} p(x) \cdot \underbrace{\log_2 \left(\frac{1}{p(x)} \right)}_{\text{surprise}} = - \sum_{x \in \mathcal{X}} p(x) \cdot \log_2(p(x))$$

Example Consider the entropy of coin toss with p as probability of head. What happens if $p_i = 1/n$?

Proposition $0 \leq H(X) \leq \log(|\mathcal{X}|)$

Proof Let Y be a rv which takes the value $1/p(x)$ with probability $p(x)$. Then

$$\sum_{x \in \mathcal{X}} p(x) \cdot \log \left(\frac{1}{p(x)} \right) = \mathbb{E}[\log(Y)] \leq \log(\mathbb{E}[Y]) = \log \left(\sum_{x \in \mathcal{X}} p(x) \cdot \frac{1}{p(x)} \right)$$

Question When does the upper bound attend?

Entropy

Source coding - How many bits are required to describe $n = 2^k$ outcomes of an experiment?

Entropy

Source coding - How many bits are required to describe $n = 2^k$ outcomes of an experiment?

Question What is the operational meaning of entropy?

Entropy

Source coding - How many bits are required to describe $n = 2^k$ outcomes of an experiment?

Question What is the operational meaning of entropy?
Answer the rv X takes $H(X)$ bits to describe on average

Entropy

Source coding - How many bits are required to describe $n = 2^k$ outcomes of an experiment?

Question What is the operational meaning of entropy?
Answer the rv X takes $H(X)$ bits to describe on average

Code A code for a set $\mathcal{X}$ over an alphabet Σ is a map $C : \mathcal{X} \to \Sigma^*$ which maps each element of $\mathcal{X}$ to a finite string of elements of Σ.

Entropy

Source coding - How many bits are required to describe $n = 2^k$ outcomes of an experiment?

Question What is the operational meaning of entropy?
Answer the rv X takes $H(X)$ bits to describe on average

Code A code for a set $\mathcal{X}$ over an alphabet Σ is a map $C : \mathcal{X} \to \Sigma^*$ which maps each element of $\mathcal{X}$ to a finite string of elements of Σ. $C(x)$ is called the codeword of x

Entropy

Source coding - How many bits are required to describe $n = 2^k$ outcomes of an experiment?

Question What is the operational meaning of entropy?
Answer the rv X takes $H(X)$ bits to describe on average

Code A code for a set $\mathcal{X}$ over an alphabet Σ is a map $C : \mathcal{X} \to \Sigma^*$ which maps each element of $\mathcal{X}$ to a finite string of elements of Σ. $C(x)$ is called the codeword of x

Prefix-free code A code is prefix-free if for any $x, y \in \mathcal{X}$ such that $x \neq y$, $C(x)$ is not a prefix of $C(y)$ i.e. $C(y) \neq C(x) \circ \sigma$ for any $\sigma \in \Sigma^*$

Entropy

Source coding - How many bits are required to describe $n = 2^k$ outcomes of an experiment?

Question What is the operational meaning of entropy?
Answer the rv X takes $H(X)$ bits to describe on average

Code A code for a set $\mathcal{X}$ over an alphabet Σ is a map $C : \mathcal{X} \to \Sigma^*$ which maps each element of $\mathcal{X}$ to a finite string of elements of Σ. $C(x)$ is called the codeword of x

Prefix-free code A code is prefix-free if for any $x, y \in \mathcal{X}$ such that $x \neq y$, $C(x)$ is not a prefix of $C(y)$ i.e. $C(y) \neq C(x) \circ \sigma$ for any $\sigma \in \Sigma^*$

Example $\Sigma = \{0, 1\}$. Let $\mathcal{X} = \{a, b, c, d\}$ with $p(a) = 1/2$, $p(b) = 1/4$, $p(c) = 1/8$ and $p(d) = 1/8$. How do we design a code for $\mathcal{X}$ such that expected length of the code is minimized?

Entropy

Source coding

Question What is the advantage of have a prefix-free code?

Entropy

Source coding

Question What is the advantage of have a prefix-free code?

Question Does a prefix-free code always exist for a given source, pmf and alphabet ?

Entropy

Source coding

Question What is the advantage of have a prefix-free code?

Question Does a prefix-free code always exist for a given source, pmf and alphabet ? If exists, how do we decide the length of the codewords?

Entropy

Source coding

Question What is the advantage of have a prefix-free code?

Question Does a prefix-free code always exist for a given source, pmf and alphabet ? If exists, how do we decide the length of the codewords?

Proposition (Kraft's inequality) Let $|\mathcal{X}| = n$. Then there exists a prefix-free code for $\mathcal{X}$ over $\Sigma = \{0, 1\}$ with codeword lengths $l_1, \ldots, l_n$ if and only if

$$\sum_{i=1}^{n} \frac{1}{2^{l_i}} \leq 1.$$

Entropy

Source coding

Question What is the advantage of have a prefix-free code?

Question Does a prefix-free code always exist for a given source, pmf and alphabet ? If exists, how do we decide the length of the codewords?

Proposition (Kraft's inequality) Let $|\mathcal{X}| = n$. Then there exists a prefix-free code for $\mathcal{X}$ over $\Sigma = \{0, 1\}$ with codeword lengths $l_1, \ldots, l_n$ if and only if

$$\sum_{i=1}^{n} \frac{1}{2^{l_i}} \leq 1.$$

For any alphabet Σ, replace 2^{l_i} by $|\Sigma|^{l_i}$.

Entropy

Proof of Kraft's inequality 'if' part first: let $l_1, \ldots, l_n$ satisfy $\sum_i 2^{-l_i} \leq 1$. Then we construct a prefix-free code C with these codeword lengths. WLOG $l_1 \leq l_2 \leq \ldots \leq l_n = l^*$.

Entropy

Proof of Kraft's inequality 'if' part first: let $l_1, \ldots, l_n$ satisfy $\sum_i 2^{-l_i} \leq 1$. Then we construct a prefix-free code C with these codeword lengths. WLOG $l_1 \leq l_2 \leq \ldots \leq l_n = l^*$.

- ▷ we can think of all binary strings of length at most l^* as a complete binary tree

Entropy

Proof of Kraft's inequality 'if' part first: let $l_1, \ldots, l_n$ satisfy $\sum_i 2^{-l_i} \leq 1$. Then we construct a prefix-free code C with these codeword lengths. WLOG $l_1 \leq l_2 \leq \ldots \leq l_n = l^*$.

- ▷ we can think of all binary strings of length at most l^* as a complete binary tree
- ▷ the root corresponds to the empty string and each node at depth d corresponds to a string of length d

Entropy

Proof of Kraft's inequality 'if' part first: let $l_1, \ldots, l_n$ satisfy $\sum_i 2^{-l_i} \leq 1$. Then we construct a prefix-free code C with these codeword lengths. WLOG $l_1 \leq l_2 \leq \ldots \leq l_n = l^*$.

- ▷ we can think of all binary strings of length at most l^* as a complete binary tree
- ▷ the root corresponds to the empty string and each node at depth d corresponds to a string of length d
- ▷ for a node with string s, the left and right children correspond to the strings $s0$ and $s1$

Entropy

Proof of Kraft's inequality 'if' part first: let $l_1, \ldots, l_n$ satisfy $\sum_i 2^{-l_i} \leq 1$. Then we construct a prefix-free code C with these codeword lengths. WLOG $l_1 \leq l_2 \leq \ldots \leq l_n = l^*$.

- ▷ we can think of all binary strings of length at most l^* as a complete binary tree
- ▷ the root corresponds to the empty string and each node at depth d corresponds to a string of length d
- ▷ for a node with string s, the left and right children correspond to the strings $s0$ and $s1$
- ▷ then the tree has 2^{l^*} leaves corresponding to all strings in $\{0,1\}^{l^*}$

Entropy

Proof of Kraft's inequality 'if' part first: let $l_1, \ldots, l_n$ satisfy $\sum_i 2^{-l_i} \leq 1$. Then we construct a prefix-free code C with these codeword lengths. WLOG $l_1 \leq l_2 \leq \ldots \leq l_n = l^*$.

- ▷ we can think of all binary strings of length at most l^* as a complete binary tree
- ▷ the root corresponds to the empty string and each node at depth d corresponds to a string of length d
- ▷ for a node with string s, the left and right children correspond to the strings $s0$ and $s1$
- ▷ then the tree has 2^{l^*} leaves corresponding to all strings in $\{0,1\}^{l^*}$

now we want to construct the code such that codewords correspond to nodes of the tree. any guess which nodes should we choose?

Entropy

Proof of Kraft's inequality 'if' part first: let $l_1, \ldots, l_n$ satisfy $\sum_i 2^{-l_i} \leq 1$. Then we construct a prefix-free code C with these codeword lengths. WLOG $l_1 \leq l_2 \leq \ldots \leq l_n = l^*$.

- ▷ we can think of all binary strings of length at most l^* as a complete binary tree
- ▷ the root corresponds to the empty string and each node at depth d corresponds to a string of length d
- ▷ for a node with string s, the left and right children correspond to the strings $s0$ and $s1$
- ▷ then the tree has 2^{l^*} leaves corresponding to all strings in $\{0,1\}^{l^*}$

now we want to construct the code such that codewords correspond to nodes of the tree. any guess which nodes should we choose? when we select a node the the subtree originated from that node should be deleted to guarantee prefix-free codewords

Entropy

- First choose an arbitrary node v_1 at depth l_1 as a codeword of length l_1 and delete the subtree below it

Entropy

- First choose an arbitrary node v_1 at depth l_1 as a codeword of length l_1 and delete the subtree below it
- this deletes $1/2^{l_1}$ fraction of the leaves

Entropy

- First choose an arbitrary node v_1 at depth l_1 as a codeword of length l_1 and delete the subtree below it
- this deletes $1/2^{l_1}$ fraction of the leaves
- since there are still more leaves left in the tree, there exists a node v_2 at depth l_2

Entropy

- First choose an arbitrary node v_1 at depth l_1 as a codeword of length l_1 and delete the subtree below it
- this deletes $1/2^{l_1}$ fraction of the leaves
- since there are still more leaves left in the tree, there exists a node v_2 at depth l_2
- note that v_1 cannot be a prefix of v_2 since v_2 does not lie in the subtree below v_1
- continue the above till l_n

Entropy

- ▷ First choose an arbitrary node v_1 at depth l_1 as a codeword of length l_1 and delete the subtree below it
- ▷ this deletes $1/2^{l_1}$ fraction of the leaves
- ▷ since there are still more leaves left in the tree, there exists a node v_2 at depth l_2
- ▷ note that v_1 cannot be a prefix of v_2 since v_2 does not lie in the subtree below v_1
- ▷ continue the above till l_n

For 'only if' part, simply reverse the above proof.

Entropy

Source coding

A probabilistic interpretation of the proof of Kraft's inequality:

- ▷ Consider an experiment that can generate l^* random bits

Entropy

Source coding

A probabilistic interpretation of the proof of Kraft's inequality:

- Consider an experiment that can generate l^* random bits
- for $x \in \mathcal{X}$, let E_x denote the event that the first $|C(x)|$ bits we generate are equal to $C(x)$

Entropy

Source coding

A probabilistic interpretation of the proof of Kraft's inequality:

- Consider an experiment that can generate l^* random bits
- for $x \in \mathcal{X}$, let E_x denote the event that the first $|C(x)|$ bits we generate are equal to $C(x)$
- since C is a prefix-free code, E_x and E_y are mutually exclusive for $x \neq y$. Moreover, the probability that E_x happens is exactly $1/2^{|C(x)|}$

Entropy

Source coding

A probabilistic interpretation of the proof of Kraft's inequality:

- Consider an experiment that can generate l^* random bits
- for $x \in \mathcal{X}$, let E_x denote the event that the first $|C(x)|$ bits we generate are equal to $C(x)$
- since C is a prefix-free code, E_x and E_y are mutually exclusive for $x \neq y$. Moreover, the probability that E_x happens is exactly $1/2^{|C(x)|}$
- Thus

$$1 \geq \sum_{x \in \mathcal{X}} \mathbb{P}[E_x] = \sum_{x \in \mathcal{X}} \frac{1}{2^{|C(x)|}} = \sum_{i=1}^{n} \frac{1}{2^{l_i}}$$

Entropy

Source coding

Proposition Let X be a random variable taking values in $\mathcal{X}$, and let $C : \mathcal{X} \to \{0, 1\}$. Then the expected number of bits used by C to communicate the value of X is at least $H(X)$.

Entropy

Source coding

Proposition Let X be a random variable taking values in $\mathcal{X}$, and let $C : \mathcal{X} \to \{0, 1\}$. Then the expected number of bits used by C to communicate the value of X is at least $H(X)$.

Proof the expected number of bits is $\sum_{x \in \mathcal{X}} p(x) \cdot |C(x)|$. Then

$$\begin{aligned} H(X) - \sum_{x \in \mathcal{X}} p(x) \cdot |C(x)| &= \sum_{x \in \mathcal{X}} p(x) \cdot \left(\log \left(\frac{1}{p(x)} \right) - |C(x)| \right) \\ &= \sum_{x \in \mathcal{X}} p(x) \cdot \log \left(\frac{1}{p(x) \cdot 2^{|C(x)|}} \right) \end{aligned} \tag{1}$$

Entropy

Source coding

Proposition Let X be a random variable taking values in $\mathcal{X}$, and let $C : \mathcal{X} \to \{0, 1\}$. Then the expected number of bits used by C to communicate the value of X is at least $H(X)$.

Proof the expected number of bits is $\sum_{x \in \mathcal{X}} p(x) \cdot |C(x)|$. Then

$$\begin{aligned} H(X) - \sum_{x \in \mathcal{X}} p(x) \cdot |C(x)| &= \sum_{x \in \mathcal{X}} p(x) \cdot \left(\log \left(\frac{1}{p(x)} \right) - |C(x)| \right) \\ &= \sum_{x \in \mathcal{X}} p(x) \cdot \log \left(\frac{1}{p(x) \cdot 2^{|C(x)|}} \right) \end{aligned} \tag{1}$$

Now let Y be the rv which takes the value $\frac{1}{p(x) \cdot 2^{|C(x)|}}$ with probability $p(x)$.

Entropy

Source coding

Proposition Let X be a random variable taking values in $\mathcal{X}$, and let $C : \mathcal{X} \to \{0, 1\}$. Then the expected number of bits used by C to communicate the value of X is at least $H(X)$.

Proof the expected number of bits is $\sum_{x \in \mathcal{X}} p(x) \cdot |C(x)|$. Then

$$\begin{aligned} H(X) - \sum_{x \in \mathcal{X}} p(x) \cdot |C(x)| &= \sum_{x \in \mathcal{X}} p(x) \cdot \left(\log \left(\frac{1}{p(x)} \right) - |C(x)| \right) \\ &= \sum_{x \in \mathcal{X}} p(x) \cdot \log \left(\frac{1}{p(x) \cdot 2^{|C(x)|}} \right) \end{aligned} \tag{1}$$

Now let Y be the rv which takes the value $\frac{1}{p(x) \cdot 2^{|C(x)|}}$ with probability $p(x)$. Then

$$\mathbb{E}[\log(Y)] \leq \log(\mathbb{E}[Y]) = \log \left(\sum_{x \in \mathcal{X}} p(x) \cdot \frac{1}{p(x) \cdot 2^{|C(x)|}} \right) = \log \left(\sum_{x \in \mathcal{X}} \frac{1}{2^{|C(x)|}} \right)$$

Entropy

Question revisited - $\Sigma = \{0, 1\}$. Let $\mathcal{X} = \{a, b, c, d\}$ with $p(a) = 1/2$, $p(b) = 1/4$, $p(c) = 1/8$ and $p(d) = 1/8$. How do we design a code for $\mathcal{X}$ such that expected length of the code is minimized?

Entropy

Question revisited - $\Sigma = \{0, 1\}$. Let $\mathcal{X} = \{a, b, c, d\}$ with $p(a) = 1/2$, $p(b) = 1/4$, $p(c) = 1/8$ and $p(d) = 1/8$. How do we design a code for $\mathcal{X}$ such that expected length of the code is minimized?

Answer $a = 0$, $b = 10$, $c = 110$, $d = 111$

Entropy

Question revisited - $\Sigma = \{0, 1\}$. Let $\mathcal{X} = \{a, b, c, d\}$ with $p(a) = 1/2$, $p(b) = 1/4$, $p(c) = 1/8$ and $p(d) = 1/8$. How do we design a code for $\mathcal{X}$ such that expected length of the code is minimized?

Answer $a = 0$, $b = 10$, $c = 110$, $d = 111$

The Shannon code A prefix-free code for a rv X with at most $H(X) + 1$ bits on average can be constructed, known as Shannon code.

Entropy

Question revisited - $\Sigma = \{0, 1\}$. Let $\mathcal{X} = \{a, b, c, d\}$ with $p(a) = 1/2$, $p(b) = 1/4$, $p(c) = 1/8$ and $p(d) = 1/8$. How do we design a code for $\mathcal{X}$ such that expected length of the code is minimized?

Answer $a = 0$, $b = 10$, $c = 110$, $d = 111$

The Shannon code A prefix-free code for a rv X with at most $H(X) + 1$ bits on average can be constructed, known as Shannon code.
For an element $x \in \mathcal{X}$, which occurs with probability $p(x)$, use a codeword of length $\lceil \log(1/p(x)) \rceil$.

Entropy

Question revisited - $\Sigma = \{0, 1\}$. Let $\mathcal{X} = \{a, b, c, d\}$ with $p(a) = 1/2$, $p(b) = 1/4$, $p(c) = 1/8$ and $p(d) = 1/8$. How do we design a code for $\mathcal{X}$ such that expected length of the code is minimized?

Answer $a = 0$, $b = 10$, $c = 110$, $d = 111$

The Shannon code A prefix-free code for a rv X with at most $H(X) + 1$ bits on average can be constructed, known as Shannon code.
For an element $x \in \mathcal{X}$, which occurs with probability $p(x)$, use a codeword of length $\lceil \log(1/p(x)) \rceil$. By Kraft's inequality, such a prefix-free code since

$$\sum_{x \in \mathcal{X}} \frac{1}{2^{|C(x)|}} = \sum_{x \in \mathcal{X}} \frac{1}{2^{\lceil \log(1/p(x)) \rceil}} \leq \sum_{x \in \mathcal{X}} \frac{1}{2^{\log(1/p(x))}} = \sum_{x \in \mathcal{X}} p(x) = 1$$

Entropy

Question revisited - $\Sigma = \{0, 1\}$. Let $\mathcal{X} = \{a, b, c, d\}$ with $p(a) = 1/2$, $p(b) = 1/4$, $p(c) = 1/8$ and $p(d) = 1/8$. How do we design a code for $\mathcal{X}$ such that expected length of the code is minimized?

Answer $a = 0$, $b = 10$, $c = 110$, $d = 111$

The Shannon code A prefix-free code for a rv X with at most $H(X) + 1$ bits on average can be constructed, known as Shannon code.
For an element $x \in \mathcal{X}$, which occurs with probability $p(x)$, use a codeword of length $\lceil \log(1/p(x)) \rceil$. By Kraft's inequality, such a prefix-free code since

$$\sum_{x \in \mathcal{X}} \frac{1}{2^{|C(x)|}} = \sum_{x \in \mathcal{X}} \frac{1}{2^{\lceil \log(1/p(x)) \rceil}} \leq \sum_{x \in \mathcal{X}} \frac{1}{2^{\log(1/p(x))}} = \sum_{x \in \mathcal{X}} p(x) = 1$$

the expected number of bits used is

$$\sum_{x \in \mathcal{X}} p(x) \cdot \lceil \log(1/p(x)) \rceil \leq \sum_{x \in \mathcal{X}} p(x) \cdot (\log(1/p(x)) + 1) = H(X) + 1.$$