Information-Theoretic Modeling

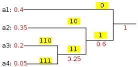
Lecture 5: Source Coding: Practice

Teemu Roos

Department of Computer Science, University of Helsinki

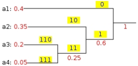
Fall 2009

1. Codes
   - Decodable Codes
   - Prefix Codes
   - Kraft-McMillan Theorem



1. Codes
   - Decodable Codes
   - Prefix Codes
   - Kraft-McMillan Theorem

2. Optimal Codes
   - Entropy Lower Bound
   - Shannon-Fano Coding



Extension Code

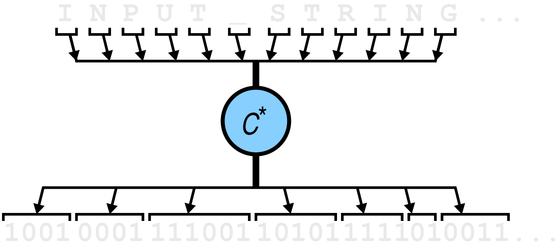
A (binary) **symbol code** $C : \mathcal{X} \to \{0,1\}^*$ is a mapping from the alphabet $\mathcal{X}$ to the set of finite binary sequences.

Extension Code

A (binary) **symbol code** $C : \mathcal{X} \to \{0,1\}^*$ is a mapping from the alphabet $\mathcal{X}$ to the set of finite binary sequences.

The **extension** of code C is the mapping $C^* : \mathcal{X}^* \to \{0,1\}^*$ obtained by concatenating the codewords $C(x_i)$ for each input symbol x_i:

$$C^*(x_1, x_2, \ldots, x_n) = C(x_1)C(x_2)\ldots C(x_n) \quad .$$

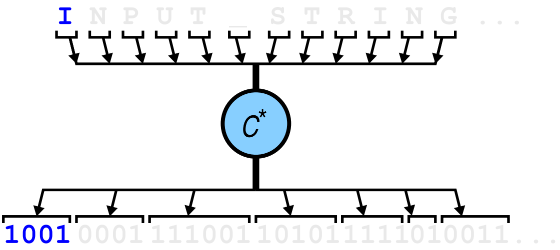


Extension Code

A (binary) **symbol code** $C : \mathcal{X} \to \{0,1\}^*$ is a mapping from the alphabet $\mathcal{X}$ to the set of finite binary sequences.

The **extension** of code C is the mapping $C^* : \mathcal{X}^* \to \{0,1\}^*$ obtained by concatenating the codewords $C(x_i)$ for each input symbol x_i:

$$C^*(x_1, x_2, \ldots, x_n) = C(x_1)C(x_2)\ldots C(x_n) \ .$$

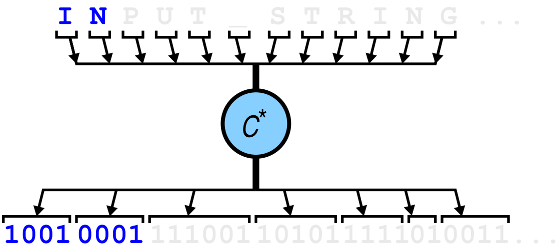


Extension Code

A (binary) **symbol code** $C : \mathcal{X} \to \{0,1\}^*$ is a mapping from the alphabet $\mathcal{X}$ to the set of finite binary sequences.

The **extension** of code C is the mapping $C^* : \mathcal{X}^* \to \{0,1\}^*$ obtained by concatenating the codewords $C(x_i)$ for each input symbol x_i:

$$C^*(x_1, x_2, \ldots, x_n) = C(x_1)C(x_2)\ldots C(x_n) \ .$$

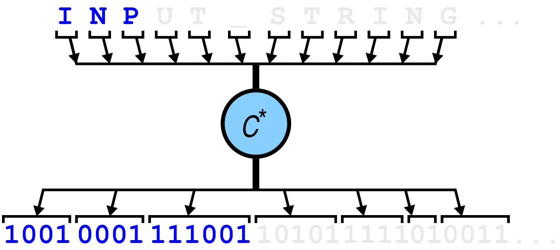


Extension Code

A (binary) **symbol code** $C : \mathcal{X} \to \{0,1\}^*$ is a mapping from the alphabet $\mathcal{X}$ to the set of finite binary sequences.

The **extension** of code C is the mapping $C^* : \mathcal{X}^* \to \{0,1\}^*$ obtained by concatenating the codewords $C(x_i)$ for each input symbol x_i:

$$C^*(x_1, x_2, \ldots, x_n) = C(x_1)C(x_2)\ldots C(x_n) \ .$$

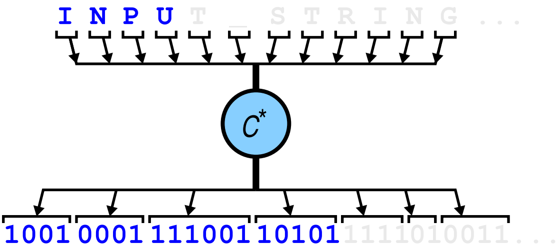


Extension Code

A (binary) **symbol code** $C : \mathcal{X} \to \{0,1\}^*$ is a mapping from the alphabet $\mathcal{X}$ to the set of finite binary sequences.

The **extension** of code C is the mapping $C^* : \mathcal{X}^* \to \{0,1\}^*$ obtained by concatenating the codewords $C(x_i)$ for each input symbol x_i:

$$C^*(x_1, x_2, \ldots, x_n) = C(x_1)C(x_2)\ldots C(x_n) \ .$$

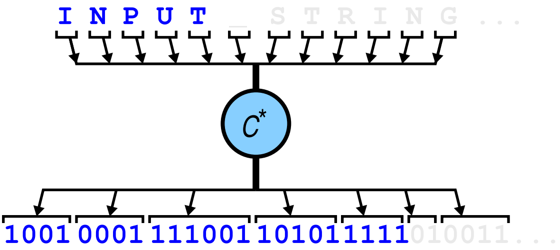


Extension Code

A (binary) **symbol code** $C : \mathcal{X} \rightarrow \{0,1\}^*$ is a mapping from the alphabet $\mathcal{X}$ to the set of finite binary sequences.

The **extension** of code C is the mapping $C^* : \mathcal{X}^* \rightarrow \{0,1\}^*$ obtained by concatenating the codewords $C(x_i)$ for each input symbol x_i:

$$C^*(x_1, x_2, \ldots, x_n) = C(x_1)C(x_2)\ldots C(x_n) \ .$$

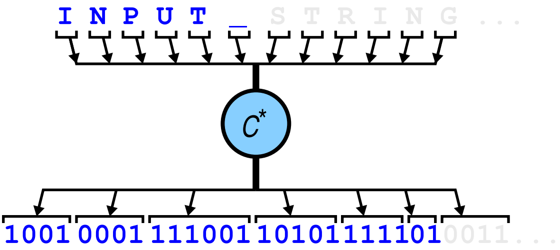


Extension Code

A (binary) **symbol code** $C : \mathcal{X} \to \{0,1\}^*$ is a mapping from the alphabet $\mathcal{X}$ to the set of finite binary sequences.

The **extension** of code C is the mapping $C^* : \mathcal{X}^* \to \{0,1\}^*$ obtained by concatenating the codewords $C(x_i)$ for each input symbol x_i:

$$C^*(x_1, x_2, \ldots, x_n) = C(x_1)C(x_2)\ldots C(x_n) \quad .$$

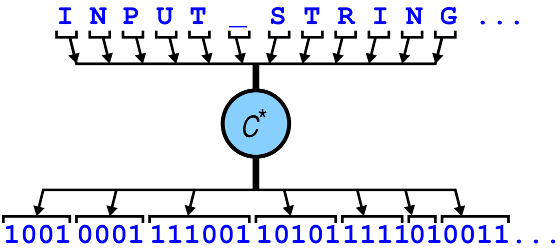


Extension Code

A (binary) **symbol code** $C : \mathcal{X} \to \{0,1\}^*$ is a mapping from the alphabet $\mathcal{X}$ to the set of finite binary sequences.

The **extension** of code C is the mapping $C^* : \mathcal{X}^* \to \{0,1\}^*$ obtained by concatenating the codewords $C(x_i)$ for each input symbol x_i:

$$C^*(x_1, x_2, \ldots, x_n) = C(x_1)C(x_2)\ldots C(x_n) \ .$$



Decodable Codes

Decodable Code

Code C is (uniquely) **decodable** iff its extension C^* is a one-to-one mapping, i.e., iff

$$(x_1, \ldots, x_n) \neq (y_1, \ldots, y_n) \;\Rightarrow\; C^*(x_1, \ldots, x_n) \neq C^*(y_1, \ldots, y_n) \;.$$

Decodable Codes

Decodable Code

Code C is (uniquely) **decodable** iff its extension C^* is a one-to-one mapping, i.e., iff

$$(x_1, \ldots, x_n) \neq (y_1, \ldots, y_n) \;\Rightarrow\; C^*(x_1, \ldots, x_n) \neq C^*(y_1, \ldots, y_n) \;.$$

X A code with codewords $\{0, 1, 10, 11\}$ is *not* uniquely decodable: What does 10 mean?

Decodable Codes

Decodable Code

Code C is (uniquely) **decodable** iff its extension C^* is a one-to-one mapping, i.e., iff

$$(x_1, \ldots, x_n) \neq (y_1, \ldots, y_n) \ \Rightarrow \ C^*(x_1, \ldots, x_n) \neq C^*(y_1, \ldots, y_n) \ .$$

- X A code with codewords $\{0, 1, 10, 11\}$ is *not* uniquely decodable: What does 10 mean?
- $\surd$ A code with codewords $\{00, 01, 10, 11\}$ *is* uniquely decodable: Each pair of bits can be decoded individually.

Decodable Codes

Decodable Code

Code C is (uniquely) **decodable** iff its extension C^* is a one-to-one mapping, i.e., iff

$$(x_1, \ldots, x_n) \neq (y_1, \ldots, y_n) \Rightarrow C^*(x_1, \ldots, x_n) \neq C^*(y_1, \ldots, y_n) \ .$$

- ✗ A code with codewords $\{0, 1, 10, 11\}$ is *not* uniquely decodable: What does 10 mean?
- √ A code with codewords $\{00, 01, 10, 11\}$ *is* uniquely decodable: Each pair of bits can be decoded individually.
- √ A code with codewords $\{0, 01, 011, 0111\}$ is also uniquely decodable: What does 0011 mean?

Prefix Codes

An important subset of decodable codes is the set of **prefix(-free) codes**.

Prefix Code

A code $C : \mathcal{X} \to \{0, 1\}^*$ is called a **prefix code** iff no codeword is a prefix of another.

It is easily seen that all prefix codes are uniquely decodable: each symbol can be decoded as soon as its codeword is read. Therefore, prefix codes are also called *instantaneous* codes.

Prefix Codes

An important subset of decodable codes is the set of **prefix(-free) codes**.

Prefix Code

A code $C : \mathcal{X} \to \{0,1\}^*$ is called a **prefix code** iff no codeword is a prefix of another.

It is easily seen that all prefix codes are uniquely decodable: each symbol can be decoded as soon as its codeword is read. Therefore, prefix codes are also called *instantaneous* codes.

✗ A code with codewords $\{0, 01, 011, 0111\}$ is uniquely decodable *but not prefix-free*: e.g., 0 is a prefix of 01.

Prefix Codes

An important subset of decodable codes is the set of **prefix(-free) codes**.

Prefix Code

A code $C : \mathcal{X} \rightarrow \{0,1\}^*$ is called a **prefix code** iff no codeword is a prefix of another.

It is easily seen that all prefix codes are uniquely decodable: each symbol can be decoded as soon as its codeword is read. Therefore, prefix codes are also called *instantaneous* codes.

- X A code with codewords $\{0, 01, 011, 0111\}$ is uniquely decodable *but not prefix-free*: e.g., 0 is a prefix of 01.
- $\surd$ A code with codewords $\{0, 10, 110, 111\}$ *is* prefix-free.

Kraft Inequality

The codeword lengths of a prefix codes satisfy the following important property.

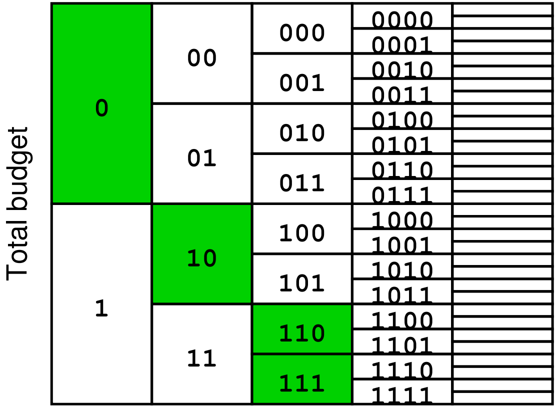
Kraft Inequality

The codeword lengths $\ell_1, \ldots, \ell_m$ of any (binary) prefix code satisfy

$$\sum_{i=1}^{m} 2^{-\ell_i} \leq 1 \ .$$

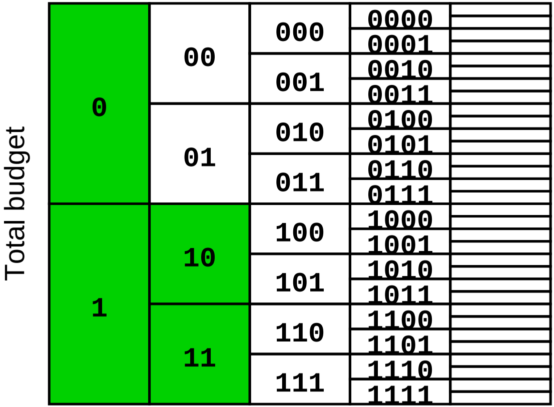
Conversely, given a set of codeword lengths that satisfy this inequality, there is a prefix code with these codeword lengths.

Kraft Inequality



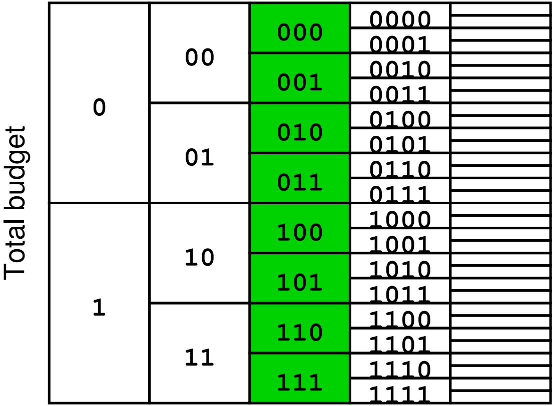
$\surd$ Codewords $\{0, 10, 110, 111\}$

Kraft Inequality



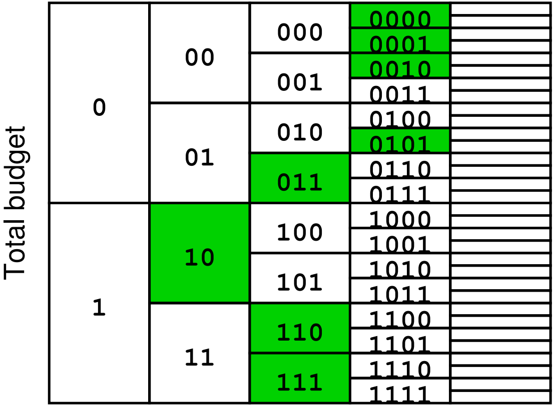
✗ Kraft inequality violated. $\Rightarrow$ Not decodable.

Kraft Inequality



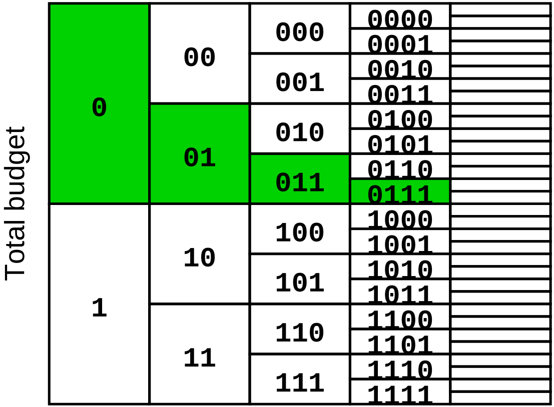
√ Fixed-length code

Kraft Inequality



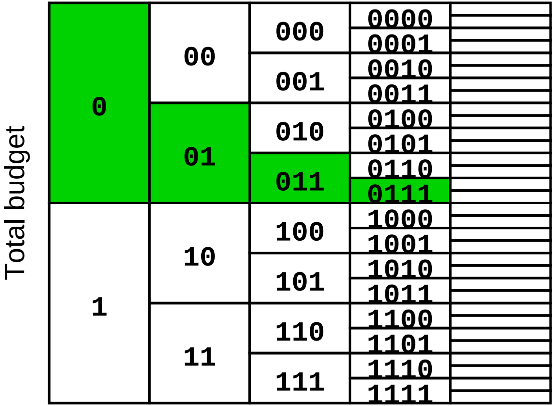
√ Decodable & prefix-free

Kraft Inequality



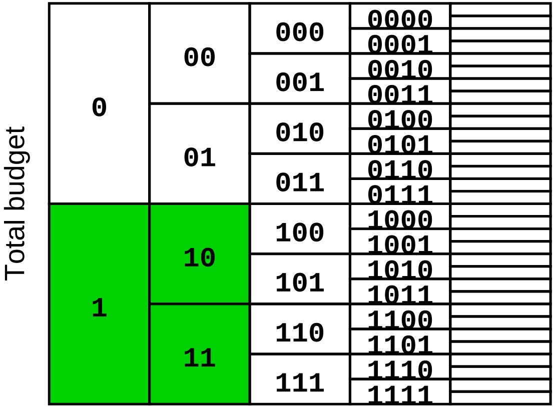
Kraft? Decodable? Prefix-free?

Kraft Inequality



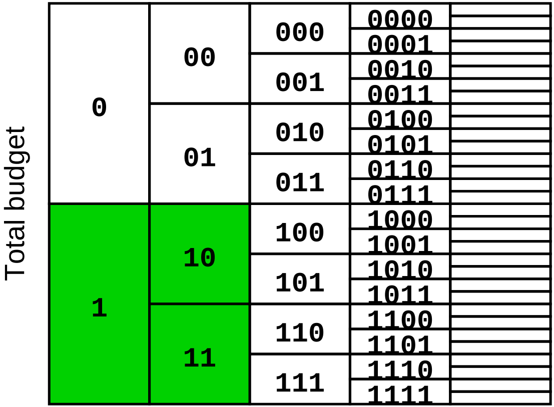
Kraft? √ Decodable? √ Prefix-free? X

Kraft Inequality



Kraft? Decodable? Prefix-free?

Kraft Inequality



Kraft? √ Decodable? X Prefix-free? X

Kraft Inequality

Question: What if the inequality is satisfied strictly, i.e., the sum of the terms in the sum equals *less* than one:

$$\sum_{i=1}^{m} 2^{-\ell_i} < 1 \ .$$

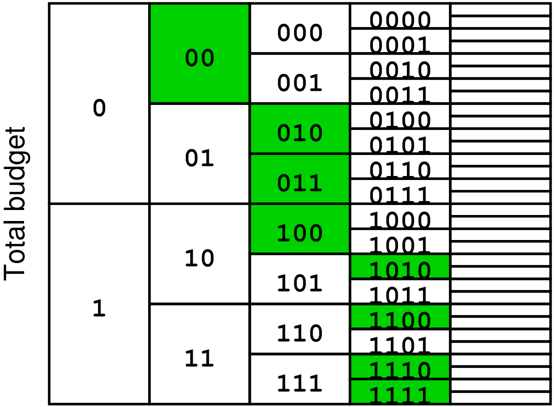
Kraft Inequality

Question: What if the inequality is satisfied strictly, i.e., the sum of the terms in the sum equals *less* than one:

$$\sum_{i=1}^{m} 2^{-\ell_i} < 1 \ .$$

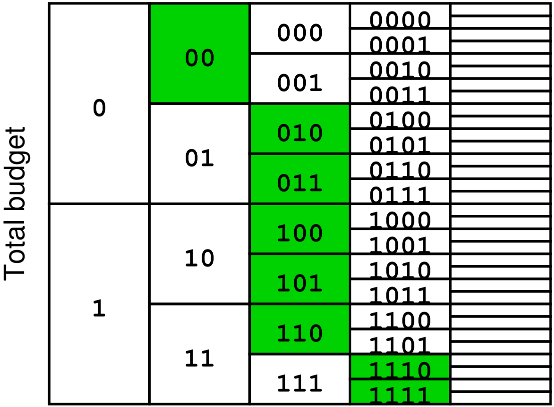
Then it is possible to make the codewords shorter and still have a decodable (prefix) code.

Kraft Inequality



Not all of budget used. $\Rightarrow$ Some codewords can be made shorter.

Kraft Inequality



"Kraft tight" / complete code.

Kraft–McMillan Theorem

The Kraft inequality restricts the codeword lengths of prefix codes. Could we do much better if we would only require decodability?

Kraft–McMillan Theorem

The Kraft inequality restricts the codeword lengths of prefix codes. Could we do much better if we would only require decodability?

In fact it can be shown that we do not lose anything at all!

Kraft–McMillan Theorem

The Kraft inequality restricts the codeword lengths of prefix codes. Could we do much better if we would only require decodability?

In fact it can be shown that we do not lose anything at all!

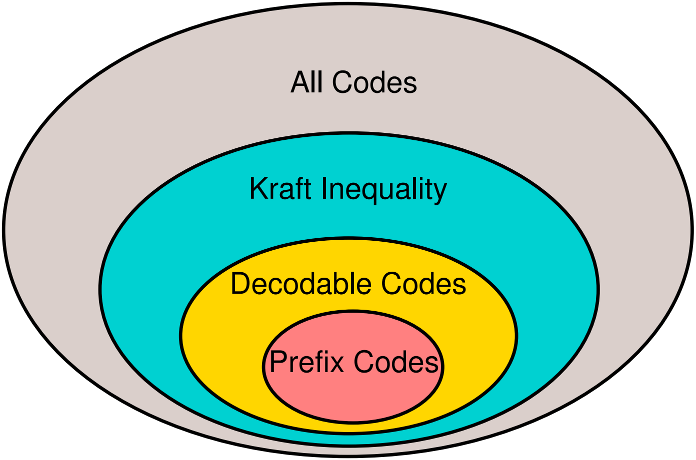
Kraft-McMillan Theorem

The codeword lengths $\ell_1, \ldots, \ell_m$ of any **uniquely decodable** (binary) code satisfy

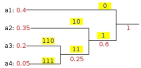
$$\sum_{i=1}^{m} 2^{-\ell_i} \leq 1 \ .$$

Conversely, given a set of codeword lengths that satisfy this inequality, there is a uniquely decodable (prefix) code with these codeword lengths.

Kraft-McMillan Theorem & Codes



1. Codes
   - Decodable Codes
   - Prefix Codes
   - Kraft-McMillan Theorem
2. Optimal Codes
   - Entropy Lower Bound
   - Shannon-Fano Coding



Codelengths and Probabilities

Let $\ell_1, \ldots, \ell_m$ be the codeword lengths of a uniquely decodable code $C : \mathcal{X} \to \{0,1\}^*$. By the Kraft-McMillan theorem we have

$$c = \sum_{i=1}^{m} 2^{-\ell_i} \leq 1 \enspace .$$

Codelengths and Probabilities

Let $\ell_1, \ldots, \ell_m$ be the codeword lengths of a uniquely decodable code $C : \mathcal{X} \to \{0,1\}^*$. By the Kraft-McMillan theorem we have

$$c = \sum_{i=1}^{m} 2^{-\ell_i} \leq 1 \ .$$

Define a probability mass function $p : \mathcal{X} \to [0,1]$ as follows:

$$p_i = \frac{2^{-\ell_i}}{c}$$

where c is given above.

Codelengths and Probabilities

Let $\ell_1, \ldots, \ell_m$ be the codeword lengths of a uniquely decodable code $C : \mathcal{X} \to \{0,1\}^*$. By the Kraft-McMillan theorem we have

$$c = \sum_{i=1}^{m} 2^{-\ell_i} \leq 1 \ .$$

Define a probability mass function $p : \mathcal{X} \to [0,1]$ as follows:

$$p_i = \frac{2^{-\ell_i}}{c} \quad \Leftrightarrow \quad \ell_i = \log_2 \frac{1}{c p_i} \ ,$$

where c is given above.

Codelengths and Probabilities

Let $\ell_1, \ldots, \ell_m$ be the codeword lengths of a uniquely decodable code $C : \mathcal{X} \to \{0,1\}^*$. By the Kraft-McMillan theorem we have

$$c = \sum_{i=1}^{m} 2^{-\ell_i} \leq 1 \ .$$

Define a probability mass function $p : \mathcal{X} \to [0,1]$ as follows:

$$p_i = \frac{2^{-\ell_i}}{c} \quad \Leftrightarrow \quad \ell_i = \log_2 \frac{1}{c p_i} \ ,$$

where c is given above.

Function p is indeed a pmf:

1. Non-negative: $p(x) \geq 0$ for all $x \in \mathcal{X}$.

Codelengths and Probabilities

Let $\ell_1, \ldots, \ell_m$ be the codeword lengths of a uniquely decodable code $C : \mathcal{X} \to \{0,1\}^*$. By the Kraft-McMillan theorem we have

$$c = \sum_{i=1}^{m} 2^{-\ell_i} \leq 1 \ .$$

Define a probability mass function $p : \mathcal{X} \to [0,1]$ as follows:

$$p_i = \frac{2^{-\ell_i}}{c} \quad \Leftrightarrow \quad \ell_i = \log_2 \frac{1}{c p_i} \ ,$$

where c is given above.

Function p is indeed a pmf:

1. Non-negative: $p(x) \geq 0$ for all $x \in \mathcal{X}$.
2. Sums to one: $\sum_{x \in \mathcal{X}} p(x) = \sum_{i=1}^{m} \frac{1}{c} 2^{-\ell_i} = \frac{c}{c} = 1 \ .$

Codelengths and Probabilities

Assuming that the code is "Kraft tight", $c = 1$, then under the pmf p corresponding to the codeword lengths $\ell_1, \ldots, \ell_m$, the expected codeword length is

$$E[\ell(X)] = \sum_{i=1}^{m} 2^{-\ell_i} \, \ell_i$$

Codelengths and Probabilities

Assuming that the code is "Kraft tight", $c = 1$, then under the pmf p corresponding to the codeword lengths $\ell_1, \ldots, \ell_m$, the expected codeword length is

$$E[\ell(X)] = \sum_{i=1}^{m} 2^{-\ell_i} \, \ell_i$$
$$= \sum_{i=1}^{m} p_i \, \log_2 \frac{1}{p_i}$$

Codelengths and Probabilities

Assuming that the code is "Kraft tight", $c = 1$, then under the pmf p corresponding to the codeword lengths $\ell_1, \ldots, \ell_m$, the expected codeword length is

$$\begin{aligned} E[\ell(X)] &= \sum_{i=1}^{m} 2^{-\ell_i}\,\ell_i \\ &= \sum_{i=1}^{m} p_i \log_2 \frac{1}{p_i} = H(X) \ . \end{aligned}$$

Codelengths and Probabilities

Assuming that the code is "Kraft tight", $c = 1$, then under the pmf p corresponding to the codeword lengths $\ell_1, \ldots, \ell_m$, the expected codeword length is

$$E[\ell(X)] = \sum_{i=1}^{m} 2^{-\ell_i}\,\ell_i = \sum_{i=1}^{m} p_i \log_2 \frac{1}{p_i} = H(X) \ .$$

This is the best we can hope for:

The expected codelength of any uniquely decodable code is at least the entropy:

$$E[\ell(X)] \geq H(X) \ .$$

Entropy Lower Bound

$$E[\ell(X)] \geq H(X) \quad .$$

Entropy Lower Bound

$$E[\ell(X)] \geq H(X) \quad .$$

Proof.

$$E[\ell(X)] - H(X) = \sum_{x \in \mathcal{X}} p(x)\,\ell(x) - \sum_{x \in \mathcal{X}} p(x)\,\log_2 \frac{1}{p(x)}$$

Entropy Lower Bound

$$E[\ell(X)] \geq H(X) \quad .$$

Proof.

$$\begin{aligned} E[\ell(X)] - H(X) &= \sum_{x \in \mathcal{X}} p(x)\,\ell(x) - \sum_{x \in \mathcal{X}} p(x)\,\log_2 \frac{1}{p(x)} \\ &= \sum_{x \in \mathcal{X}} p(x)\,\log_2 \frac{1}{2^{-\ell_x}} - \sum_{x \in \mathcal{X}} p(x)\,\log_2 \frac{1}{p(x)} \end{aligned}$$

Entropy Lower Bound

$$E[\ell(X)] \geq H(X) \ .$$

Proof.

$$\begin{aligned} E[\ell(X)] - H(X) &= \sum_{x \in \mathcal{X}} p(x)\,\ell(x) - \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{1}{p(x)} \\ &= \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{1}{2^{-\ell_x}} - \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{1}{p(x)} \\ &= \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{2^{-\ell_x}} \end{aligned}$$

Entropy Lower Bound

$$E[\ell(X)] \geq H(X) \quad .$$

Proof.

$$\begin{aligned}
E[\ell(X)] - H(X) &= \sum_{x \in \mathcal{X}} p(x)\,\ell(x) - \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{1}{p(x)} \\
&= \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{1}{2^{-\ell_x}} - \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{1}{p(x)} \\
&= \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{2^{-\ell_x}} \\
&= \sum_{x \in \mathcal{X}} p(x) \left[\log_2 \frac{p(x)}{q(x)} + \log_2 \frac{1}{c} \right] \qquad \boxed{q(x) = \frac{2^{-\ell(x)}}{c}}
\end{aligned}$$

Entropy Lower Bound

$$E[\ell(X)] \geq H(X) \ .$$

Proof.

$$\begin{aligned}
E[\ell(X)] - H(X) &= \sum_{x \in \mathcal{X}} p(x)\, \ell(x) - \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{1}{p(x)} \\
&= \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{1}{2^{-\ell_x}} - \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{1}{p(x)} \\
&= \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{2^{-\ell_x}} \\
&= \sum_{x \in \mathcal{X}} p(x) \left[\log_2 \frac{p(x)}{q(x)} + \log_2 \frac{1}{c} \right] \qquad \boxed{q(x) = \frac{2^{-\ell(x)}}{c}} \\
&= D(p \,\|\, q) + \log_2 \frac{1}{c} \geq 0 \ .
\end{aligned}$$

Entropy Lower Bound

So what have we learned?

Entropy Lower Bound

So what have we learned? For decodable symbols codes:

1. $E[\ell(X)] - H(X) = D(p \parallel q) + \log_2 \frac{1}{c}$, where $q(x) = \dfrac{2^{-\ell(x)}}{c}$.

Entropy Lower Bound

So what have we learned? For decodable symbols codes:

1. $E[\ell(X)] - H(X) = D(p \parallel q) + \log_2 \frac{1}{c}$, where $q(x) = \dfrac{2^{-\ell(x)}}{c}$.
2. $E[\ell(X)] \geq H(X)$.

Entropy Lower Bound

So what have we learned? For decodable symbols codes:

1. $E[\ell(X)] - H(X) = D(p \parallel q) + \log_2 \frac{1}{c}$, where $q(x) = \dfrac{2^{-\ell(x)}}{c}$.
2. $E[\ell(X)] \geq H(X)$.
3. If $\ell(x) = \log_2 \frac{1}{p(x)}$, then $E[\ell(X)] = H(X)$. **Optimal!**

Entropy Lower Bound

So what have we learned? For decodable symbols codes:

1. $E[\ell(X)] - H(X) = D(p \parallel q) + \log_2 \frac{1}{c}$, where $q(x) = \dfrac{2^{-\ell(x)}}{c}$.
2. $E[\ell(X)] \geq H(X)$.
3. If $\ell(x) = \log_2 \frac{1}{p(x)}$, then $E[\ell(X)] = H(X)$. **Optimal!**

Note also that for a sequence $X_1, \ldots, X_n$ the expected codelength becomes

$$E[\ell(X_1, \ldots, X_n)] = E\left[\sum_{i=1}^{n} \ell(X_i)\right]$$

Entropy Lower Bound

So what have we learned? For decodable symbols codes:

1. $E[\ell(X)] - H(X) = D(p \parallel q) + \log_2 \frac{1}{c}$, where $q(x) = \dfrac{2^{-\ell(x)}}{c}$.
2. $E[\ell(X)] \geq H(X)$.
3. If $\ell(x) = \log_2 \frac{1}{p(x)}$, then $E[\ell(X)] = H(X)$. **Optimal!**

Note also that for a sequence $X_1, \ldots, X_n$ the expected codelength becomes

$$E[\ell(X_1, \ldots, X_n)] = E\left[\sum_{i=1}^{n} \ell(X_i)\right] = \sum_{i=1}^{n} E[\ell(X_i)]$$

Entropy Lower Bound

So what have we learned? For decodable symbols codes:

1. $E[\ell(X)] - H(X) = D(p \parallel q) + \log_2 \frac{1}{c}$, where $q(x) = \dfrac{2^{-\ell(x)}}{c}$.
2. $E[\ell(X)] \geq H(X)$.
3. If $\ell(x) = \log_2 \frac{1}{p(x)}$, then $E[\ell(X)] = H(X)$. **Optimal!**

Note also that for a sequence $X_1, \ldots, X_n$ the expected codelength becomes

$$E[\ell(X_1, \ldots, X_n)] = E\left[\sum_{i=1}^{n} \ell(X_i)\right] = \sum_{i=1}^{n} E[\ell(X_i)] = nH(X) \ .$$

Entropy Lower Bound

So what have we learned? For decodable symbols codes:

1. $E[\ell(X)] - H(X) = D(p \parallel q) + \log_2 \frac{1}{c}$, where $q(x) = \dfrac{2^{-\ell(x)}}{c}$.
2. $E[\ell(X)] \geq H(X)$.
3. If $\ell(x) = \log_2 \frac{1}{p(x)}$, then $E[\ell(X)] = H(X)$. **Optimal!**

Note also that for a sequence $X_1, \ldots, X_n$ the expected codelength becomes

$$E[\ell(X_1, \ldots, X_n)] = E\left[\sum_{i=1}^{n} \ell(X_i)\right] = \sum_{i=1}^{n} E[\ell(X_i)] = nH(X) \ .$$

! By Shannon's Noiseless Channel Coding Theorem, this is optimal among all codes, **not only symbol codes.**

Entropy Lower Bound

So what have we learned? For decodable symbols codes:

1. $E[\ell(X)] - H(X) = D(p \parallel q) + \log_2 \frac{1}{c}$, where $q(x) = \dfrac{2^{-\ell(x)}}{c}$.
2. $E[\ell(X)] \geq H(X)$.
3. If $\ell(x) = \log_2 \frac{1}{p(x)}$, then $E[\ell(X)] = H(X)$. **Optimal!**

Note also that for a sequence $X_1, \ldots, X_n$ the expected codelength becomes

$$E[\ell(X_1, \ldots, X_n)] = E\left[\sum_{i=1}^{n} \ell(X_i)\right] = \sum_{i=1}^{n} E[\ell(X_i)] = nH(X) \ .$$

! By Shannon's Noiseless Channel Coding Theorem, this is optimal among all codes, **not only symbol codes.** Fine print: only if X_i i.i.d.!

Codelengths and Probabilities

The only problem with the $\ell(x) = \log_2 \frac{1}{p(x)}$ codeword choice is the requirement that codeword lengths must be **integers** (try to think about a codeword with length 0.123, for instance), while the so obtained ℓ is not in general an integer.

Codelengths and Probabilities

The only problem with the $\ell(x) = \log_2 \frac{1}{p(x)}$ codeword choice is the requirement that codeword lengths must be **integers** (try to think about a codeword with length 0.123, for instance), while the so obtained ℓ is not in general an integer.

The simplest solution is to round upwards:

Shannon's Code

Given a pmf, the **Shannon code** has the codeword lengths

$$\ell(x) = \left\lceil \log_2 \frac{1}{p(x)} \right\rceil \quad \text{for all } x \in \mathcal{X}.$$

Alice in Wonderland

Shannon's code: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$	$\ell(X)$
a	0.0644	3.9	4
b	0.0108	6.5	7
c	0.0178	5.8	6
d	0.0359	4.7	5
e	0.0991	3.3	4
f	0.0147	6.0	7
g	0.0184	5.7	6
h	0.0535	4.2	5
i	0.0551	4.1	5
j	0.0011	9.8	10
k	0.0083	6.8	7
l	0.0343	4.8	5
	$\vdots$		
y	0.0165	5.9	6
z	0.0005	10.7	11
	0.2111	2.2	3

$$H(X) = 4.03$$

Shannon's code: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$	$\ell(X)$
a	0.0644	3.9	4
b	0.0108	6.5	7
c	0.0178	5.8	6
d	0.0359	4.7	5
e	0.0991	3.3	4
f	0.0147	6.0	7
g	0.0184	5.7	6
h	0.0535	4.2	5
i	0.0551	4.1	5
j	0.0011	9.8	10
k	0.0083	6.8	7
l	0.0343	4.8	5
	$\vdots$		
y	0.0165	5.9	6
z	0.0005	10.7	11
	0.2111	2.2	3

$H(X) = 4.03$

Shannon (1948):

Shannon's code: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$	$\ell(X)$
a	0.0644	3.9	4
b	0.0108	6.5	7
c	0.0178	5.8	6
d	0.0359	4.7	5
e	0.0991	3.3	4
f	0.0147	6.0	7
g	0.0184	5.7	6
h	0.0535	4.2	5
i	0.0551	4.1	5
j	0.0011	9.8	10
k	0.0083	6.8	7
l	0.0343	4.8	5
	$\vdots$		
y	0.0165	5.9	6
z	0.0005	10.7	11
	0.2111	2.2	3

$H(X) = 4.03$

Shannon (1948):

1. Sort by probability.

Shannon's code: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$	$\ell(X)$
	0.2111	2.2	3
e	0.0991	3.3	4
t	0.0781	3.6	4
a	0.0644	3.9	4
o	0.0598	4.0	5
i	0.0551	4.1	5
h	0.0535	4.2	5
n	0.0516	4.2	5
s	0.0475	4.3	5
r	0.0401	4.6	5
d	0.0359	4.7	5
l	0.0343	4.8	5
	$\vdots$		
x	0.0011	9.8	10
j	0.0011	9.8	10
z	0.0005	10.7	11

$$H(X) = 4.03$$

Shannon (1948):

1. Sort by probability.

Shannon's code: Example

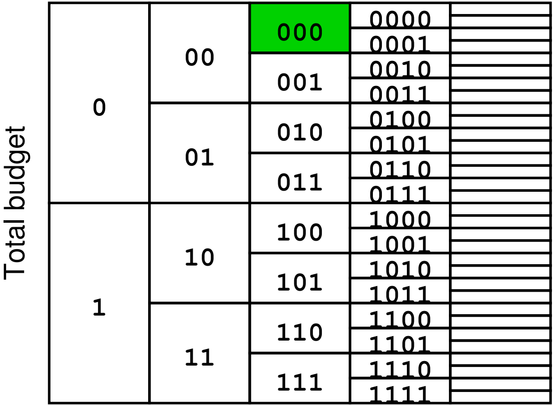
X	$p(X)$	$\log_2 \frac{1}{p(X)}$	$\ell(X)$
	0.2111	2.2	3
e	0.0991	3.3	4
t	0.0781	3.6	4
a	0.0644	3.9	4
o	0.0598	4.0	5
i	0.0551	4.1	5
h	0.0535	4.2	5
n	0.0516	4.2	5
s	0.0475	4.3	5
r	0.0401	4.6	5
d	0.0359	4.7	5
l	0.0343	4.8	5
	⋮		
x	0.0011	9.8	10
j	0.0011	9.8	10
z	0.0005	10.7	11

$H(X) = 4.03$

Shannon (1948):

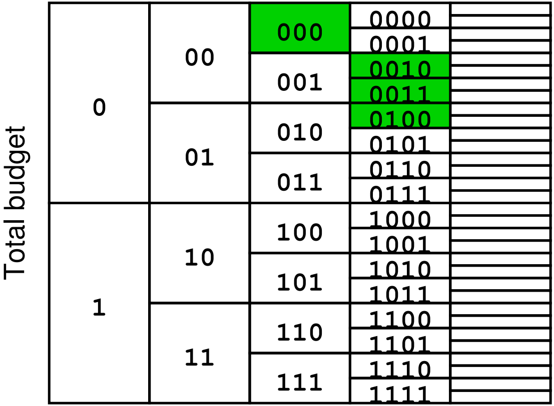
1. Sort by probability.
2. Choose codewords in order, avoiding prefixes. ("Kraft table"!)

Shannon's code: Example



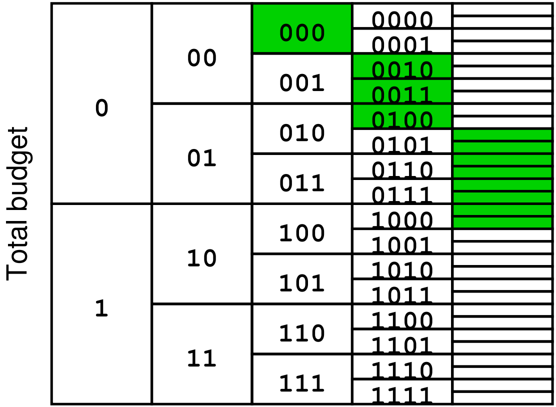
Codeword lengths $(\color{red}{3}, 4, 4, 4, 5, 5, 5, 5, \ldots, 10, 10, 11)$

Shannon's code: Example



Codeword lengths $(3, 4, 4, 4, 5, 5, 5, 5, \ldots, 10, 10, 11)$

Shannon's code: Example



Codeword lengths $(3, 4, 4, 4, 5, 5, 5, 5, \ldots, 10, 10, 11)$

Shannon's code: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$	$\ell(X)$	$C(X)$
	0.2111	2.2	3	000
e	0.0991	3.3	4	0010
t	0.0781	3.6	4	0011
a	0.0644	3.9	4	0100
o	0.0598	4.0	5	01010
i	0.0551	4.1	5	01011
h	0.0535	4.2	5	01100
n	0.0516	4.2	5	01101
s	0.0475	4.3	5	01110
r	0.0401	4.6	5	01111
d	0.0359	4.7	5	10000
l	0.0343	4.8	5	10001
	⋮			
x	0.0011	9.8	10	1010111101
j	0.0011	9.8	10	1010111110
z	0.0005	10.7	11	10101111110

Shannon's code: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$	$\ell(X)$	$C(X)$
	0.2111	2.2	3	000
e	0.0991	3.3	4	0010
t	0.0781	3.6	4	0011
a	0.0644	3.9	4	0100
o	0.0598	4.0	5	01010
i	0.0551	4.1	5	01011
h	0.0535	4.2	5	01100
n	0.0516	4.2	5	01101
s	0.0475	4.3	5	01110
r	0.0401	4.6	5	01111
d	0.0359	4.7	5	10000
l	0.0343	4.8	5	10001
	$\vdots$			
x	0.0011	9.8	10	1010111101
j	0.0011	9.8	10	1010111110
z	0.0005	10.7	11	10101111110

$$H(X) = 4.03$$
$$E[\ell(X)] = 4.60$$
$$E[\ell(X)] - H(X) = 0.57$$

Shannon's code

The expected codeword length of Shannon's code is

$$E\left[\ell(X)\right] = E\left[\left\lceil \log_2 \frac{1}{p(X)} \right\rceil\right]$$
$$\leq E\left[\log_2 \frac{1}{p(X)} + 1\right] = H(X) + 1 \quad .$$

Shannon's code

The expected codeword length of Shannon's code is

$$E\left[\ell(X)\right] = E\left[\left\lceil \log_2 \frac{1}{p(X)} \right\rceil\right]$$
$$\leq E\left[\log_2 \frac{1}{p(X)} + 1\right] = H(X) + 1 \ .$$

In the Alice example we had

$$E[\ell(X)] - H(X) = 4.60 - 4.03 = 0.57 \leq 1 \ .$$

Fano: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$
a	0.0644	3.9
b	0.0108	6.5
c	0.0178	5.8
d	0.0359	4.7
e	0.0991	3.3
f	0.0147	6.0
g	0.0184	5.7
h	0.0535	4.2
i	0.0551	4.1
j	0.0011	9.8
k	0.0083	6.8
l	0.0343	4.8
	$\vdots$	
y	0.0165	5.9
z	0.0005	10.7
	0.2111	2.2

Fano: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$
a	0.0644	3.9
b	0.0108	6.5
c	0.0178	5.8
d	0.0359	4.7
e	0.0991	3.3
f	0.0147	6.0
g	0.0184	5.7
h	0.0535	4.2
i	0.0551	4.1
j	0.0011	9.8
k	0.0083	6.8
l	0.0343	4.8
	$\vdots$	
y	0.0165	5.9
z	0.0005	10.7
	0.2111	2.2

(Shannon-)Fano code:

Fano: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$
a	0.0644	3.9
b	0.0108	6.5
c	0.0178	5.8
d	0.0359	4.7
e	0.0991	3.3
f	0.0147	6.0
g	0.0184	5.7
h	0.0535	4.2
i	0.0551	4.1
j	0.0011	9.8
k	0.0083	6.8
l	0.0343	4.8
	⋮	
y	0.0165	5.9
z	0.0005	10.7
	0.2111	2.2

(Shannon-)Fano code:

1. Sort by probability.

Fano: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$
	0.2111	2.2
e	0.0991	3.3
t	0.0781	3.6
a	0.0644	3.9
o	0.0598	4.0
i	0.0551	4.1
h	0.0535	4.2
n	0.0516	4.2
s	0.0475	4.3
r	0.0401	4.6
d	0.0359	4.7
l	0.0343	4.8
	$\vdots$	
x	0.0011	9.8
j	0.0011	9.8
z	0.0005	10.7

(Shannon-)Fano code:

1. Sort by probability.

Fano: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$
	0.2111	2.2
e	0.0991	3.3
t	0.0781	3.6
a	0.0644	3.9
o	0.0598	4.0
i	0.0551	4.1
h	0.0535	4.2
n	0.0516	4.2
s	0.0475	4.3
r	0.0401	4.6
d	0.0359	4.7
l	0.0343	4.8
	⋮	
x	0.0011	9.8
j	0.0011	9.8
z	0.0005	10.7

(Shannon-)Fano code:

1. Sort by probability.
2. Divide in two equally probable parts (as equal as possible)

Fano: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$
	0.2111	2.2
e	0.0991	3.3
t	0.0781	3.6
a	0.0644	3.9
o	0.0598	4.0
i	0.0551	4.1
h	0.0535	4.2
n	0.0516	4.2
s	0.0475	4.3
r	0.0401	4.6
d	0.0359	4.7
l	0.0343	4.8
	$\vdots$	
x	0.0011	9.8
j	0.0011	9.8
z	0.0005	10.7

(Shannon-)Fano code:

1. Sort by probability.
2. Divide in two equally probable parts (as equal as possible)
3. Add '0' to the codewords in the first part, '1' to the others.

Fano: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$
	0.2111	2.2
e	0.0991	3.3
t	0.0781	3.6
a	0.0644	3.9
o	0.0598	4.0
i	0.0551	4.1
h	0.0535	4.2
n	0.0516	4.2
s	0.0475	4.3
r	0.0401	4.6
d	0.0359	4.7
l	0.0343	4.8
	⋮	
x	0.0011	9.8
j	0.0011	9.8
z	0.0005	10.7

(Shannon-)Fano code:

1. Sort by probability.
2. Divide in two equally probable parts (as equal as possible)
3. Add '0' to the codewords in the first part, '1' to the others.
4. Repeat recursively for both parts.

Fano: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$	$\ell(X)$	$C(X)$
	0.2111	2.2	2	0
e	0.0991	3.3	4	0
t	0.0781	3.6	4	0
a	0.0644	3.9	4	0
o	0.0598	4.0	4	0
i	0.0551	4.1	4	1
h	0.0535	4.2	4	1
n	0.0516	4.2	4	1
s	0.0475	4.3	5	1
r	0.0401	4.6	5	1
d	0.0359	4.7	5	1
l	0.0343	4.8	5	1
	$\vdots$			
x	0.0011	9.8	10	1
j	0.0011	9.8	10	1
z	0.0005	10.7	10	1

Fano: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$	$\ell(X)$	$C(X)$
	0.2111	2.2	2	00
e	0.0991	3.3	4	01
t	0.0781	3.6	4	01
a	0.0644	3.9	4	01
o	0.0598	4.0	4	01
i	0.0551	4.1	4	10
h	0.0535	4.2	4	10
n	0.0516	4.2	4	10
s	0.0475	4.3	5	10
r	0.0401	4.6	5	10
d	0.0359	4.7	5	11
l	0.0343	4.8	5	11
	⋮			
x	0.0011	9.8	10	11
j	0.0011	9.8	10	11
z	0.0005	10.7	10	11

Fano: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$	$\ell(X)$	$C(X)$
	0.2111	2.2	2	00
e	0.0991	3.3	4	010
t	0.0781	3.6	4	010
a	0.0644	3.9	4	011
o	0.0598	4.0	4	011
i	0.0551	4.1	4	100
h	0.0535	4.2	4	100
n	0.0516	4.2	4	101
s	0.0475	4.3	5	101
r	0.0401	4.6	5	101
d	0.0359	4.7	5	110
l	0.0343	4.8	5	110
	⋮			
x	0.0011	9.8	10	111
j	0.0011	9.8	10	111
z	0.0005	10.7	10	111

Fano: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$	$\ell(X)$	$C(X)$
	0.2111	2.2	2	00
e	0.0991	3.3	4	0100
t	0.0781	3.6	4	0101
a	0.0644	3.9	4	0110
o	0.0598	4.0	4	0111
i	0.0551	4.1	4	1000
h	0.0535	4.2	4	1001
n	0.0516	4.2	4	1010
s	0.0475	4.3	5	1011
r	0.0401	4.6	5	1011
d	0.0359	4.7	5	1100
l	0.0343	4.8	5	1100
	$\vdots$			
x	0.0011	9.8	10	1111
j	0.0011	9.8	10	1111
z	0.0005	10.7	10	1111

Fano: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$	$\ell(X)$	$C(X)$
	0.2111	2.2	2	00
e	0.0991	3.3	4	0100
t	0.0781	3.6	4	0101
a	0.0644	3.9	4	0110
o	0.0598	4.0	4	0111
i	0.0551	4.1	4	1000
h	0.0535	4.2	4	1001
n	0.0516	4.2	4	1010
s	0.0475	4.3	5	10110
r	0.0401	4.6	5	10111
d	0.0359	4.7	5	11000
l	0.0343	4.8	5	11001
	⋮			
x	0.0011	9.8	10	11111
j	0.0011	9.8	10	11111
z	0.0005	10.7	10	11111

Fano: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$	$\ell(X)$	$C(X)$
	0.2111	2.2	2	00
e	0.0991	3.3	4	0100
t	0.0781	3.6	4	0101
a	0.0644	3.9	4	0110
o	0.0598	4.0	4	0111
i	0.0551	4.1	4	1000
h	0.0535	4.2	4	1001
n	0.0516	4.2	4	1010
s	0.0475	4.3	5	10110
r	0.0401	4.6	5	10111
d	0.0359	4.7	5	11000
l	0.0343	4.8	5	11001
	$\vdots$			
x	0.0011	9.8	10	111111
j	0.0011	9.8	10	111111
z	0.0005	10.7	10	111111

Fano: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$	$\ell(X)$	$C(X)$
	0.2111	2.2	2	00
e	0.0991	3.3	4	0100
t	0.0781	3.6	4	0101
a	0.0644	3.9	4	0110
o	0.0598	4.0	4	0111
i	0.0551	4.1	4	1000
h	0.0535	4.2	4	1001
n	0.0516	4.2	4	1010
s	0.0475	4.3	5	10110
r	0.0401	4.6	5	10111
d	0.0359	4.7	5	11000
l	0.0343	4.8	5	11001
	⋮			
x	0.0011	9.8	10	1111111101
j	0.0011	9.8	10	1111111110
z	0.0005	10.7	10	1111111111

Fano: Example

X	$p(X)$	$\log_2 \frac{1}{p(X)}$	$\ell(X)$	$C(X)$
	0.2111	2.2	2	00
e	0.0991	3.3	4	0100
t	0.0781	3.6	4	0101
a	0.0644	3.9	4	0110
o	0.0598	4.0	4	0111
i	0.0551	4.1	4	1000
h	0.0535	4.2	4	1001
n	0.0516	4.2	4	1010
s	0.0475	4.3	5	10110
r	0.0401	4.6	5	10111
d	0.0359	4.7	5	11000
l	0.0343	4.8	5	11001
	$\vdots$			
x	0.0011	9.8	10	1111111101
j	0.0011	9.8	10	1111111110
z	0.0005	10.7	10	1111111111

$$H(X) = 4.03$$
$$E[\ell(X)] = 4.07$$
$$E[\ell(X)] - H(X) = 0.04$$

Shannon-Fano Code

The expected codeword length of the Shannon-Fano code is

$$\begin{aligned} E\left[\ell(X)\right] &\leq E\left[\left\lceil \log_2 \frac{1}{p(X)} \right\rceil\right] \\ &\leq E\left[\log_2 \frac{1}{p(X)} + 1\right] = H(X) + 1 \ . \end{aligned}$$

Shannon-Fano Code

The expected codeword length of the Shannon-Fano code is

$$E\left[\ell(X)\right] \leq E\left[\left\lceil \log_2 \frac{1}{p(X)} \right\rceil\right]$$
$$\leq E\left[\log_2 \frac{1}{p(X)} + 1\right] = H(X) + 1 \ .$$

In the Alice example we had

$$E[\ell(X)] - H(X) = 4.06 - 4.03 = 0.04 \leq 1 \ .$$

Shannon-Fano Code

The expected codeword length of the Shannon-Fano code is

$$E\left[\ell(X)\right] \leq E\left[\left\lceil \log_2 \frac{1}{p(X)} \right\rceil\right] \leq E\left[\log_2 \frac{1}{p(X)} + 1\right] = H(X) + 1 \ .$$

In the Alice example we had

$$E[\ell(X)] - H(X) = 4.06 - 4.03 = 0.04 \leq 1 \ .$$

Is this optimal?

Shannon-Fano Code

The expected codeword length of the Shannon-Fano code is

$$E\left[\ell(X)\right] \leq E\left[\left\lceil \log_2 \frac{1}{p(X)} \right\rceil\right] \leq E\left[\log_2 \frac{1}{p(X)} + 1\right] = H(X) + 1 \ .$$

In the Alice example we had

$$E[\ell(X)] - H(X) = 4.06 - 4.03 = 0.04 \leq 1 \ .$$

Is this optimal? Not necessarily — Huffman!

What's next?

Friday:

- Daniel Schmidt's guest lecture on the **Minimum Message Length (MML) principle**.

What's next?

Friday:

- Daniel Schmidt's guest lecture on the **Minimum Message Length (MML) principle**.

Next week:

- the optimal prefix code: Huffman,

What's next?

Friday:

- Daniel Schmidt's guest lecture on the **Minimum Message Length (MML) principle**.

Next week:

- the optimal prefix code: Huffman,
- even better: Rissanen's arithmetic coding.