Information Theory Basics

crackhopper

Sun Dec 18 15:51:29 2016

Outline

Encoding

Definition

A source code C for a random variable X is a mapping from $\mathcal{X}$, the domain of X, to $\mathcal{D}^*$, the set of finite-length strings of symbols from a D -ary alphabet. Let $C(x)$ denote the codeword corresponding to x and let $l(x)$ denote the length of C(x).

Example

random variable X, with domain $\mathcal{X} = \{\text{red}, \text{blue}\}$, choose alphabet $\mathcal{D} = 0, 1$, then we can have a source code

$$C(\text{red}) = 00, C(\text{blue}) = 11$$

and $l(\text{red}) = 2, l(\text{blue}) = 2$

Expected Encoding Length

Expected Encoding Length

$$L(C) = \sum_{x \in \mathcal{X}} p(x) l(x)$$

Example

$$\begin{aligned}
\Pr(X=1) &= \tfrac{1}{2}, \quad \text{codeword } C(1) = 0 \\
\Pr(X=2) &= \tfrac{1}{4}, \quad \text{codeword } C(2) = 10 \\
\Pr(X=3) &= \tfrac{1}{8}, \quad \text{codeword } C(3) = 110 \\
\Pr(X=4) &= \tfrac{1}{8}, \quad \text{codeword } C(4) = 111.
\end{aligned}$$

The expected length is 1.75

Uniquely Decodable

A code is called uniquely decodable if its extension is non-singular.

Nonsingular

A code is said to be nonsingular if every case of X maps into a different string in $\mathcal{D}^*$

$$x \neq x' \Rightarrow C(x) \neq C(x')$$

Extension

The extension C^* of a code C is the mapping from finite-length strings of $\mathcal{X}$ to finite-length strings of $\mathcal{D}$, defined by

$$C(x_1 x_2 ... x_n) = C(x_1)C(x_2)...C(x_n)$$

Prefix Code

Prefix Code

A code is called a *prefix* code or an *instantaneous* code if no codeword is a prefix of any other codeword.

$$
\begin{aligned}
\Pr(X=1) &= \tfrac{1}{2}, \quad \text{codeword } C(1) = 0\\
\Pr(X=2) &= \tfrac{1}{4}, \quad \text{codeword } C(2) = 10\\
\Pr(X=3) &= \tfrac{1}{8}, \quad \text{codeword } C(3) = 110\\
\Pr(X=4) &= \tfrac{1}{8}, \quad \text{codeword } C(4) = 111.
\end{aligned}
$$

For example, the binary string 01011111010 produced by the code above is parsed as 0,10,111,110,10.

Classes of Codes

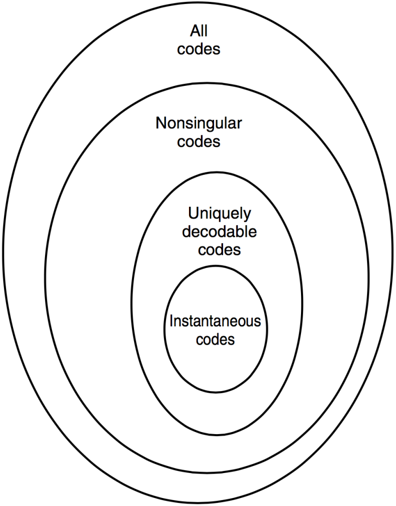

FIGURE 5.1. Classes of codes.

Classes of Codes

TABLE 5.1 Classes of Codes

X	Singular	Nonsingular, But Not Uniquely Decodable	Uniquely Decodable, But Not Instantaneous	Instantaneous
1	0	0	10	0
2	0	010	00	10
3	0	01	11	110
4	0	10	110	111

Kraft Inequality

It is clear that we cannot assign short codewords to all source symbols and still be prefix-free.

Theorem 5.2.1 (Kraft inequality)

For any instantaneous code (prefix code) over an alphabet of size D, the codeword lengths $l_1, l_2, ..., l_m$ must satisfy the inequality

$$\sum_i D^{-l_i} \leq 1$$

Intuitive Proof of Kraft Inequality

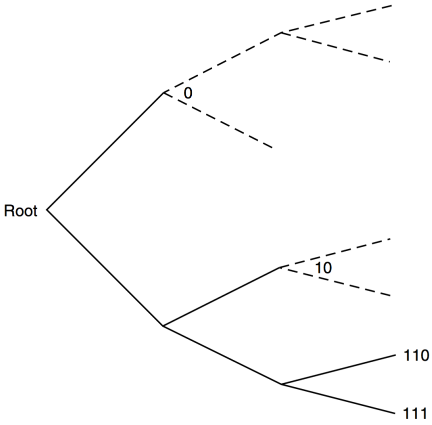

FIGURE 5.2. Code tree for the Kraft inequality.

Optimal Codes

Optimal Codes prefix code with the minimum expected length.

Optimization Target:

$$L = \sum_i p_i l_i$$

With constraint,

$$\sum D^{-l_i} \leq 1$$

By Langrange multiplier:

$$J = \sum p_i l_i + \lambda(\sum D^{-l_i})$$

Optimal Codes

Differentiating the target with respect to l_i

$$\frac{\partial J}{\partial l_i} = p_i - \lambda D^{-l_i} \ln D$$

Setting the derivative to zero, we obtain

$$D^{-l_i} = \frac{p_i}{\lambda \ln D}$$

Substituting this in the constraint, we find $\lambda = 1/\ln D$, and hence

$$p_i = D^{-l_i}$$

Yielding the optimial code lenth, $l^* = -\log_D p_i$

$$L^* = \sum p_i l_i^* = -\sum p_i \log_D p_i = H_D(X)$$

Entropy

The entropy $H(x)$ of a discrete random variable is defined by

$$H(X) = -\sum p(x) \log p(x)$$

Let $X \in \{a, b, c, d\}$ with

$$p(X = a) = \frac{1}{2}, p(X = b) = \frac{1}{4}, p(X = c) = \frac{1}{8}, p(X = d) = \frac{1}{8}$$

Then we can get $H(X) = 1.75$.

This means if we try to encoding the X by binary codes, by using the optimal coding, the average length is around $H(X)$

Entropy

The entropy of a random variable

- is a measure of the uncertainty of the random variable;
- is a measure of the amount of information required on the average to describe the random variable.
- the minimum length of bits we need to encode the variable.

Conditional Entropy

If $(X, Y) \sim p(x, y)$, then the conditional entropy $H(Y|X)$ is defined as

$$\begin{aligned} H(Y|X) &= E_X \left[H(Y|X = x) \right] \\ &= E_{X,Y} \left[\log p(Y|X) \right] \end{aligned}$$

This means the expectation length of optimal codes for Y, when we already know about X.

Conditional Entropy

Chain Rule

$$H(X,Y) = H(X) + H(Y|X)$$

Note that :

- $H(Y|X) \neq H(X|Y)$.
- $H(X) - H(X|Y) = H(Y) - H(Y|X)$.

Relative Entropy

Definition

$$D(p||q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}$$
$$= E_p \left[\log\left(\frac{p(X)}{q(X)}\right) \right]$$

MEANING

- a measure of the distance between two distributions.
- a measure of the inefficiency of assuming that the distribution is q when the true distribution is p.

Relative Entropy

- a measure of the inefficiency of assuming that the distribution is q when the true distribution is p.

For example, if we knew the true distribution p of the random variable, we could construct a code with average description length $H(p)$. If, instead, we used the code for a distribution q, we would need $H(p) + D(p||q)$ bits on the average to describe the random variable.

Cross Entropy

Cross Entropy

$$\begin{aligned} H(p,q) &= H(p) + D(p||q) \\ &= E_{p(x)}\left[-\log(q(x))\right] \end{aligned}$$

The average description length when we use q to encode a distribution p.

When $H(p,q)$ is much larger than $H(q)$, it means q cannot encode p effiectively.

Mutual Information

Definition

$$
\begin{aligned}
I(X;Y) &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \\
&= D(p(x,y)||p(x)p(y))
\end{aligned}
$$

This means what we loss if we use the $p(x)p(y)$ to encode the joint distribution.
We also have

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = I(Y;X)$$

and the mutual information is also called information gain, which means how much lenght of code we would save when we know one of the random variable.

Information Theory Basics

1. Encoding
2. Entropy
3. Generative adversarial

Loss Function

$$
\begin{aligned}
&\min_G \max_D V(D, G) \\
= \;& E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \\
= \;& -CE(1_{x \sim p_{data}(x)}, D(x)) - CE(1_{z \sim p_{G(z)}}, 1 - D(G(z)))
\end{aligned}
$$

max D min CE + CE, means $D(x)$ should match $1_{x \sim p_{data}(x)}$, and $1 - D(G(z))$ should match $1_{z \sim p_{G(z)}}$. discriminate more precisely.

min G we already have D, means we maximize $CE(1_{z \sim p_{G(z)}}, 1 - D(G(z)))$. that means $1 - D(G(z))$ should not match $1_{z \sim p_{G(z)}}$, the generator try to confuse the discriminator. In this way, also means, $G(z)$ is trying to approximate x.