Most efficient binary encoding of a message

Soumyadeep Paul

November 2, 2023

CMI Student Seminar

Outline

Source Coding

Optimality

Entropy and source coding theorem

Source Coding

Definitions

Ensemble - X

An ensemble X is a random variable taking values in $\mathcal{A}_X = (a_1, a_2, \cdots, a_q)$, called the alphabet of X and having the probabilities $p_i = \mathbb{P}(X = a_i)$.

A source $\mathcal{S}$ is a sequence of i.i.d. ensembles.

Source code

A source code C for an ensemble X is a mapping from $\mathcal{A}_X$ to $\mathcal{D}^*$, where $\mathcal{D}$ is a D-ary alphabet.
$C(x)$ denotes the codeword corresponding to x and $l(x)$ denotes the length of $C(x)$.
We call C a D-ary code.

We will generally take $\mathcal{D} = 0, 1$.
Efficiency of a source code C will be measured by its expected length $\mathbb{E}[l(C)] = \sum_{x \in \mathcal{A}_X} \mathbb{P}(X = x) l(x)$.

Source code extension

Extension

The extention C^* of a source code C is the from finite length strings in $\mathcal{A}_X$ to finite length strings in $\mathcal{D}$, defined by

$$C(x_1 x_2 \cdots x_n) = C(x_1)C(x_2)\cdots C(x_n),$$

where rhs denotes concatenation.

Uniquely decodable codes

Uniquely decodable codes

A code C for which its extension is injective.

Example

$a_1 \to 0, a_2 \to 01, a_3 \to 011$

$a_1 \to 00, a_2 \to 01, a_3 \to 11$

Uniquely decodable codes

Uniquely decodable codes

A code C for which its extension is injective.

Example

$a_1 \to 0, a_2 \to 01, a_3 \to 011$

$a_1 \to 00, a_2 \to 01, a_3 \to 11$

Uniquely decodable codes are hard to work with:

$$a_1 \to 0, a_2 \to 01, a_3 \to 11$$

Uniquely decodable codes

Uniquely decodable codes

A code C for which its extension is injective.

Example

$a_1 \to 0, a_2 \to 01, a_3 \to 011$

$a_1 \to 00, a_2 \to 01, a_3 \to 11$

Uniquely decodable codes are hard to work with:

$$a_1 \to 0, a_2 \to 01, a_3 \to 11$$

The above code is uniquely decodable but if we get a stream $0111\cdots$ we can't be sure how to decode it until we get to the end of the stream.

Instantaneous codes

We can instead work with a more well behaved subset of uniquely decodable codes:

Prefix-free codes

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

Instantaneous codes

We can instead work with a more well behaved subset of uniquely decodable codes:

Prefix-free codes

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

An instantaneous code can be decoded without reference to future code- words since the end of a codeword is immediately recognizable.

Instantaneous codes

We can instead work with a more well behaved subset of uniquely decodable codes:

Prefix-free codes

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

An instantaneous code can be decoded without reference to future code- words since the end of a codeword is immediately recognizable.
We will show later that we don't lose any performance if we constrain our source code to be a prefix-free code.

Kraft's inequality

Kraft's inequality

There is an instantaneous r-ary code C with word-lengths $l_1 \leq l_2 \leq \cdots \leq l_q$ if and only if

$$\sum_{i=1}^{q} r^{-l_i} \leq 1.$$

Kraft's inequality: Proof

We will use the following tree for both directions of the proof:

We will use the following tree for both directions of the proof:

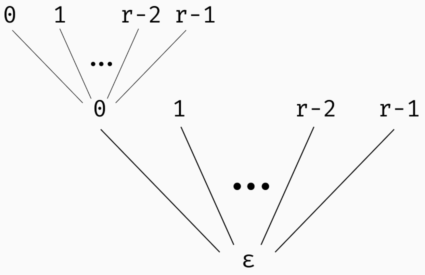

Figure 1: Tree for all codewords.

Kraft's inequality: Proof

Given an r-ary prefix free code. We consider the tree till a height $l_q = l$.

Kraft's inequality: Proof

Given an r-ary prefix free code. We consider the tree till a height $l_q = l$.Now, if we take the subtrees of all the codewords, they can't have any same leaf since that would contradict the prefix free condition.

Kraft's inequality: Proof

Given an r-ary prefix free code. We consider the tree till a height $l_q = l$.Now, if we take the subtrees of all the codewords, they can't have any same leaf since that would contradict the prefix free condition. The subtree of a word of length k has r^{l-k} leaves.

Kraft's inequality: Proof

Given an r-ary prefix free code. We consider the tree till a height $l_q = l$.Now, if we take the subtrees of all the codewords, they can't have any same leaf since that would contradict the prefix free condition. The subtree of a word of length k has r^{l-k} leaves.Therefore,

$$\sum_{i=1}^{q} r^{l-l_i} \leq r^l,$$

from which Kraft's inequality follows.

Existence of a code:
We take the tree till $l_q = l$ levels. We take any word w_1 of length l_1 and remove the subtree of that word.

Existence of a code:

We take the tree till $l_q = l$ levels. We take any word w_1 of length l_1 and remove the subtree of that word.

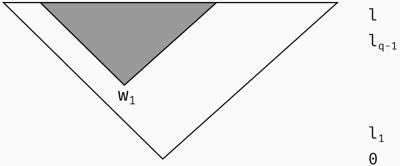

Figure 2: Subtree of w_1 deleted

Kraft's inequality: Proof

Now, if $q > 1$,

$$r^{l-l_1} < r^l \sum_{i=1}^{q} r^{-l_i} \leq r^l.$$

So there exists a leaf which is not deleted.

Kraft's inequality: Proof

Now, if $q > 1$,

$$r^{l-l_1} < r^l \sum_{i=1}^{q} r^{-l_i} \leq r^l.$$

So there exists a leaf which is not deleted. We take the first l_2 letters of this leaf to make w_2 and delete w_2's subtree. Clearly w_2 is not in the subtree of w_1. Continuing this way for $k < q$ many steps we would delete $\sum_{i=1}^{k} r^{l-l_i}$ many leaves, but then we will have

$$\sum_{i=1}^{k} r^{l-l_i} < r^l \sum_{i=1}^{q} r^{-l_i} \leq r^l,$$

so we would have at least 1 leaf not deleted and can add another word. $\square$

Macmillan's inequality

There is an uniquely decodable r-ary code C with word-lengths $l_1 \leq l_2 \leq \cdots \leq l_q$ if and only if

$$\sum_{i=1}^{q} r^{-l_i} \leq 1.$$

Macmillans's inequality Proof:

Since we know a prefix-free code exists for lengths satisfying the inequality and prefix free codes are uniquely decodable, one half of the proof is already done.

Macmillans's inequality Proof:

Since we know a prefix-free code exists for lengths satisfying the inequality and prefix free codes are uniquely decodable, one half of the proof is already done. For the other direction, Let

$$K = \sum_{i=1}^{q} r^{-l_i}$$

$$l = \max_i(l_i), m = \min_i(l_i)$$

Macmillans's inequality Proof:

We have, for any n

$$K^n = \left(\sum_{i=1}^{q} r^{-l_i}\right)^n = \sum_{j=mn}^{ln} \frac{N_{j,n}}{r^j}$$

$N_{j,n}$ must be the number of no. of ways we can take n codewords $w_{i_1}, w_{i_2}, \cdots, w_{i_n}$ of C of total length j. Each such sequence determines $t = w_{i_1} w_{i_2} \cdots w_{i_n}$. Since C is uniquely decodable each t arises from at most 1 such sequence. We must then have $N_{j,n} \leq r^j$.

Macmillans's inequality Proof:

$$K^n = \sum_{j=mn}^{ln} \frac{N_{j,n}}{r^j} \leq (l-m)n+1$$

Now, if $K > 1$, lhs grows exponentially and rhs grows linearly in n. There for some n this inequality will be contradicted. We must then have $K \leq 1$, which proves Macmillan's inequality. $\square$

Macmillans's inequality Proof:

$$K^n = \sum_{j=mn}^{ln} \frac{N_{j,n}}{r^j} \leq (l-m)n + 1$$

Now, if $K > 1$, lhs grows exponentially and rhs grows linearly in n. There for some n this inequality will be contradicted. We must then have $K \leq 1$, which proves Macmillan's inequality. $\square$

Takeaway

There is an instantaneous r-ary code with word-lengths $l_1, l_2, \cdots, l_q$ if and only if there is a uniquely decodable r-ary code with these word-lengths.

Optimality

Optimal codes

Given r and a probability distribution p_i for a source $\mathcal{S}$ we want to find instantaneous r-ary codes C minimising $L(C)$. Such codes are called optimal codes.

Optimal codes exist

Each source $\mathcal{S}$ has an optimal r-ary code for each integer $r \geq 2$.

Proof: We renumber the alphabets $a_1, a_2, \cdots a_q$ so that $P_i > 0$ for $i \leq k$, and $p_i = 0$ for $i > k$. Let $p = \min(p_1, p_2, \cdots, p_k)$. There exists an instantaneous r-ary code C, for $\mathcal{S}$: take $l_1 = l_2 = \cdots = l_q = l$ for some l such that $r^l \geq q$.

We will prove there exists finitely many values $L(D)$ such that $L(D) \leq L(C)$.
Let D be an instantaneous code such that $L(D) \leq L(C)$. We must have $l_i \leq \frac{L(C)}{p} \forall i = 1, \cdots, k$.
Otherwise we must have

$$L(D) = p_i l_i + \cdots + p_q l_q \geq p_i l_i > L(C).$$

The choices of w_i for $i > k$ does not afftect $L(D)$ and there are finite possibilities of w_i for $i \leq k$, those with length at most $L(C)/p$, therefore there are finitely many possibilities of $L(D) \leq L(C)$.$\square$

How to find optimal codes

Main idea

We should give smaller codewords to highly probable alphabets and longer codewords to ones with lower probability.

How to find optimal codes

Main idea

We should give smaller codewords to highly probable alphabets and longer codewords to ones with lower probability.

We concentrate on the binary case. For an ensemble X, we renumber $a_1, a_2, \cdots a_q$ so that $p_1 \geq p_2 \geq \cdots \geq p_q$. We make an ensemble X' with alphabets $a_1, a_2, \cdots, a_{q-2}, a'$ where $a' = (a_{q-1} \vee a_q)$ with probability $p' = p_{q-1} + p_q$.
Now, if we have a binary code C' for X' we can create a binary code C for X by appending 0 and 1 to the ends of the codeword for a' to get codewords for a_{q-1} and a_q.

How to find optimal codes

If C' is prefix-free, C is as well.
We can follow the same procedure amalgamating the two least likely symbols of X' to get X''.
After $q-1$ times we will have $X^{(q-1)}$ which will have 1 symbol with probaility 1.
To this 1 symbol we assign the empty codeword ϵ, therefore $C^{q-1} = \epsilon$.
Now by adding the digits as stated before we get a code C for X. This is the **huffman code** for X.

Optimality of huffman code

We call two words siblings if they are of the form $x0$ and $x1$ for some x.

Lemma

Every ensemble X has an optimal binary code D in which two of the longest code-words are siblings.

Proof : Among all optimal codes choose D which minimises $\sum_i l_i$.
Let dt be a maximal codeword in D, where $t \in \{0, 1\}$. Assume $d\bar{t}$ is not in D.
The only word in D with prefix d is dt since it is a maximal codeword and $d\bar{t} \notin D$. Then we can get a prefix-free code D' with dt replaced by d which would decrease the sum of lengths contradicting our assumption. $\square$

Huffman code optimality

If C is the huffman code for X then it is optimal.

Proof: We will induct on q. Trivially true for $q = 1$.
We assume the statement is true for $q - 1$.
Let C' be the code we obtain in the construction. We have

$$\begin{aligned} L(C) - L(C') &= (p_q)(l+1) + (p_{q-1})(l+1) - (p_q + p_{q-1})l \\ &= p_q + p_{q-1} = p'. \end{aligned} \tag{1}$$

Let D be the optimal code which has 2 longest words which are siblings. Let those words be $C(a_u), C(a_v)$.
Now, we can assume $u, v = q-1, q$. If not we can construct another code D^* where we replace $C(a_u)$ with $C(a_{q-1})$ and $C(a_v)$ with $C(a_q)$.

Optimality of huffman code

We then have

$$\begin{aligned}
L(D) - L(D^*) &= (p_v l(a_v) + p_q l(a_q)) + (p_u l(a_u) + p_{q-1} l(a_{q-1]}) \\
&\quad - (p_v l(a_q) + p_q l(a_v)) + (p_u l(a_{q-1}) + p_{q-1} l(a_u)) \\
&= (p_v - p_q)(l(a_v) - l(a_q)) \\
&\quad + (p_u - p_{q-1})(l(a_u) - l(a_{q-1})) \\
&\geq 0.
\end{aligned} \tag{2}$$

But $L(D)$ is optimal so $L(D^*)$ must also be optimal and we can work with D^* instead.

Optimality of huffman code

We now create a code D' with the reverse procedure of huffman code.
As shown in equation (1)

$$L(D) - L(D') = p' = L(C) - L(C').$$

By induction hypothesis, C' is optimal and therefore $L(D) - L(C) = L(D') - L(C') \geq 0$. Since D is optimal, C is optimal as well. $\square$

Entropy and source coding theorem

Entropy

Entropy

For an ensemble X with probabilities p_i its entropy is

$$H_r(X) = \sum_i -p_i \log_r(p_i).$$

If $p_i = 0$ for some i we take $-p_i \log(p_i) = 0$, since the limit tends to 0 is 0.

Lemma

If $\{x\}_i$ and $\{y\}_i$ are probability distributions, then

$$\sum_{i=1}^{q} x_i \log_r \left(\frac{1}{x_i}\right) \leq \sum_{i=1}^{q} x_i \log_r \left(\frac{1}{y_i}\right)$$

Equality holds iff $x_i = y_i$.

Proof:

$$\begin{aligned} \text{LHS -RHS} &= \sum_{i=1}^{q} x_i \log_r \left(\frac{y_i}{x_i}\right) = \frac{1}{\ln r} \sum_{i=1}^{q} x_i \ln \left(\frac{y_i}{x_i}\right) \\ &\leq \frac{1}{\ln r} \sum_{i=1}^{q} x_i \left(\frac{y_i}{x_i} - 1\right) = 0.\square \end{aligned} \tag{3}$$

Entropy

Lemma

If X and Y are independent ensembles with probabilites p_i and q_j then their joint entropy $H_r(X, Y) = H_r(X) + H_r(Y)$.

Proof: Since X and Y are independent $\mathbb{P}(X = a_i, Y = b_j) = p_i q_j$.

$$\begin{aligned}
H_r(X, Y) &= -\sum_i \sum_j p_i q_j \log_r(p_i q_j) \\
&= -\sum_i p_i \log_r(p_i)(\sum_j q_j) + -\sum_j q_j \log_r(q_j)(\sum_i p_i) \quad (4) \\
&= H_r(X) + H_r(Y). \square
\end{aligned}$$

Entropy

Corrolary

If $X_1, X_2, \cdots, X_n$ are i.i.d. random ensembles then their joint entropy $H_r(X_1, X_2, \cdots, X_n) = nH_r(X)$.

Entropy and average word length

Theorem

If C is an uniquely decodable r-ary code for an ensemble X then $L(C) \geq H_r(X)$.

Proof: Let $K = \sum_{i=1}^{q} r^{-l_i}$. We apply our lemma with $x_i = p_i$ and $y_i = \frac{r^{-l_i}}{K}$.

$$\begin{aligned} H_r(X) &\leq \sum_{i=1}^{q} p_i \log_r \left(r^{l_i} K \right) \\ &= L(C) + \log_r K \\ &\leq L(C). \square \end{aligned} \tag{5}$$

Entropy and average word length

Corrolary

The expected length is minimized and is equal to H(X) only if the codelengths are equal to

$$l_i = \log_r \left(\frac{1}{p_i} \right).$$

Proof: If the expected length is equal to $H_r(X)$, equality must hold in the lemma and $K = 1$.

For the other direction, we have $\sum_i r^{-l_i} \leq 1$, thus an uniquely decodable code exists and its average length is $H_r(X)$.

Slightly non optimal code

The optimum length is not always possible. What happens if we set the codelengths to integers slightly larger than the optimum lengths.

$$l_i = \left\lceil \log_r \left(\frac{1}{p_i} \right) \right\rceil$$

We have,

$$\sum_i r^{-l_i} = \sum_i r^{-\left\lceil \log_r \left(\frac{1}{p_i} \right) \right\rceil} \leq \sum_i r^{-\log_r \left(\frac{1}{p_i} \right)} \leq 1.$$

So, a code C with these lengths exist. This code is called a Shannon-Fano code. And we have,

$$L(C) = \sum_i p_i \left\lceil \log_r \left(\frac{1}{p_i} \right) \right\rceil < \sum_i p_i \log_r \left(\frac{1}{p_i} \right) + p_i = H_r(X) + 1.$$

Source coding theorem

Optimal codes bound

for an ensemble x an r-ary optimal code c must satisfy

$$H_r(x) \leq L(c) < H_r(x) + 1.$$

Block coding

We can do much better if instead of encoding each alphabet we try encoding strings of n alphabets.

This leads to a code for an ensemble X^n. And we know an optimal code with L_n exists such that

$$H_r(X^n) \leq L_n < H_r(X^n) + 1$$

Since we know the source is i.i.d.'s, $H_r(X^n) = nH_r(X)$. Therefore,

$$H_r(X) \leq \frac{L_n}{n} < H_r(X) + \frac{1}{n}$$

Source coding theorem

By encoding X^n with n sufficiently large one can find uniquely decodable r-ary encodings of X with its average length arbitrarily close to $H_r(X)$.

References

1. Information and Coding Theory - Gareth Jones, Mary Jones
2. Information Theory, Inference, and Learning Algorithms - David J.C. MacKay

Thank You!