An introduction to Information Theory

Adrish Banerjee

Department of Electrical Engineering
Indian Institute of Technology Kanpur
Kanpur, Uttar Pradesh
India

July 25, 2016

Lecture #3A: Block to variable length coding-I: Prefix-free code

Outline of the lecture

- Coding a single random variable

Coding a single random variable Prefix-free code Kraft's inequality

Outline of the lecture

- Coding a single random variable
- Prefix-free code

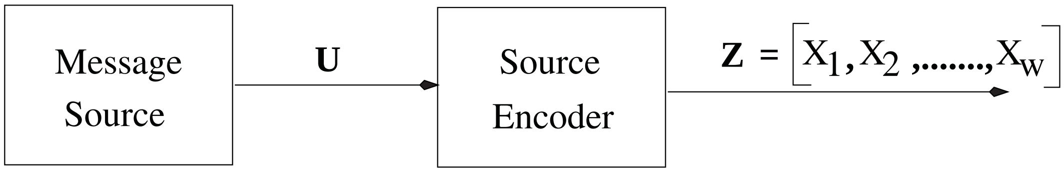
Adrish Banerjee Department of Electrical Engineering Indian Institute of Technology Kanpur Kanpur, Uttar Pradesh India
An introduction to Information Theory

Outline of the lecture

- Coding a single random variable
- Prefix-free code
- Kraft's inequality

Outline

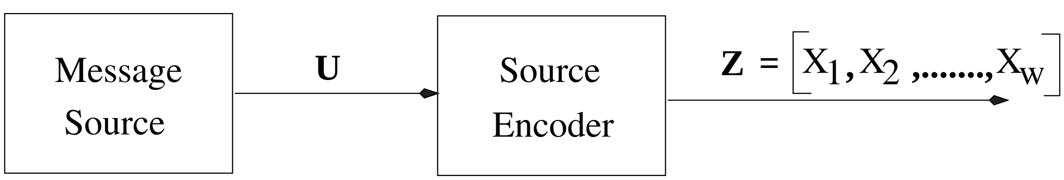
Coding a single random variable



Variable length coding scheme

- **U** is a $K-$ary random variable.

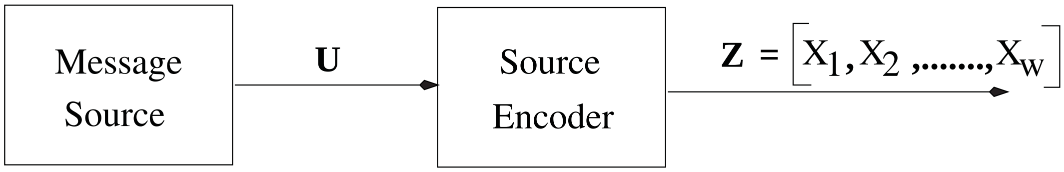
Coding a single random variable



Variable length coding scheme

- **U** is a $K-$ary random variable.
- X_i takes on values in the $D-$ary alphabet.

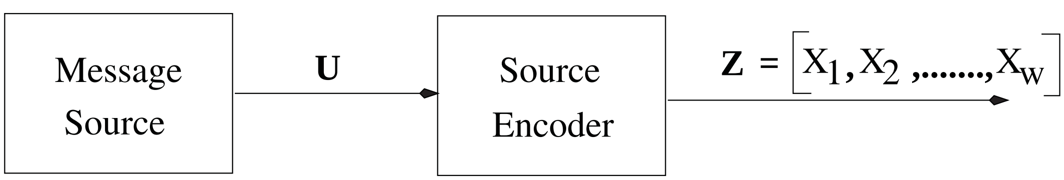
Coding a single random variable



Variable length coding scheme

- **U** is a $K-$ary random variable.
- X_i takes on values in the $D-$ary alphabet.
- **W** is a random variable, i.e. **Z** is variable length.

Coding a single random variable



Variable length coding scheme

- **U** is a $K-$ary random variable.
- X_i takes on values in the $D-$ary alphabet.
- **W** is a random variable, i.e. **Z** is variable length.
- A list $(z_1, z_2, \cdots, z_K)$ of D-ary sequences is a codeword of $U = [u_1, u_2, \cdots, u_K]$.

Coding a single random variable

- If $\mathbf{z}_i = [x_{i1}, x_{i2}, \cdots, x_{iw_i}]$ is the codeword for u_i, and w_i is the length of this codeword, then average codeword length is defined as

$$E[W] = \sum_{i=1}^{K} w_i P_U(u_i)$$

Coding a single random variable

- If $\mathbf{z}_i = [x_{i1}, x_{i2}, \cdots, x_{iw_i}]$ is the codeword for u_i, and w_i is the length of this codeword, then average codeword length is defined as

$$E[W] = \sum_{i=1}^{K} w_i P_U(u_i)$$

- Smallness of average codeword length is a measure of goodness of the code.

Outline

Uniquely decodeable codes

- A code is uniquely decodeable if every finite sequence of the code corresponds to atmost one message.

Uniquely decodeable codes

- A code is uniquely decodeable if every finite sequence of the code corresponds to atmost one message.
- One way to ensure unique decodeability is to require no codeword to be prefix of another codeword.

Coding a single random variable Prefix-free code Kraft's inequality

Uniquely decodeable codes

- A code is uniquely decodeable if every finite sequence of the code corresponds to atmost one message.
- One way to ensure unique decodeability is to require no codeword to be prefix of another codeword.
- A D-ary sequence z is a prefix of another D-ary sequence z' if z has length n and the first n digits of z' form exactly the sequence z.

Adrish Banerjee Department of Electrical Engineering Indian Institute of Technology Kanpur Kanpur, Uttar Pradesh India
An introduction to Information Theory

Uniquely decodeable codes

- A code is uniquely decodeable if every finite sequence of the code corresponds to atmost one message.
- One way to ensure unique decodeability is to require no codeword to be prefix of another codeword.
- A D-ary sequence z is a prefix of another D-ary sequence z' if z has length n and the first n digits of z' form exactly the sequence z.
- No codeword should be the prefix of another codeword for instantaneous decodeability.

Uniquely decodeable codes

- A code is uniquely decodeable if every finite sequence of the code corresponds to atmost one message.
- One way to ensure unique decodeability is to require no codeword to be prefix of another codeword.
- A D-ary sequence z is a prefix of another D-ary sequence z' if z has length n and the first n digits of z' form exactly the sequence z.
- No codeword should be the prefix of another codeword for instantaneous decodeability.
- A code for **U** that satisfies the above condition is known as prefix-free code.

Uniquely decodeable codes

- A code is uniquely decodeable if every finite sequence of the code corresponds to atmost one message.
- One way to ensure unique decodeability is to require no codeword to be prefix of another codeword.
- A D-ary sequence z is a prefix of another D-ary sequence z' if z has length n and the first n digits of z' form exactly the sequence z.
- No codeword should be the prefix of another codeword for instantaneous decodeability.
- A code for **U** that satisfies the above condition is known as prefix-free code.
- Can a non-prefix free code be uniquely decodeable?

U	Z
u_1	0
u_2	10
u_3	11

Prefix-free code

U	Z
u_1	1
u_2	00
u_3	11

Not a prefix-free code

Test for uniquely decodeability

- Let S_0 be the original set of code words. We construct a sequence of sets $S_0, S_1, S_2, \cdots$ as follows:

Test for uniquely decodeability

- Let S_0 be the original set of code words. We construct a sequence of sets $S_0, S_1, S_2, \cdots$ as follows:
- If a codeword W_i in S_0 is a prefix of another codeword $W_j = W_i A$, we place the suffix A in S_1.

Test for uniquely decodeability

- Let S_0 be the original set of code words. We construct a sequence of sets $S_0, S_1, S_2, \cdots$ as follows:
- If a codeword W_i in S_0 is a prefix of another codeword $W_j = W_i A$, we place the suffix A in S_1.
- In general to form $S_n, n > 1$, we compare S_0 and S_{n-1}. If a codeword in S_0 is a prefix of a codeword in S_{n-1} or vice-versa, we place the suffix $\in S_n$.

Test for uniquely decodeability

- Let S_0 be the original set of code words. We construct a sequence of sets $S_0, S_1, S_2, \cdots$ as follows:
- If a codeword W_i in S_0 is a prefix of another codeword $W_j = W_i A$, we place the suffix A in S_1.
- In general to form $S_n, n > 1$, we compare S_0 and S_{n-1}. If a codeword in S_0 is a prefix of a codeword in S_{n-1} or vice-versa, we place the suffix $\in S_n$.
- A code is uniquely decodeable if and only if none of the sets $S_1, S_2, \cdots$ contains a code word that is a member of S_0.

Test for uniquely decodeability

S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7
a	d	eb	de	b	ad	d	eb
c	bb	cde			bcde		
ad							
abb							
bad							
deb							
bbcde							

Prefix-free code

- A $D-$ary tree, is a finite rooted tree such that D branches stem outward from each node.

Prefix-free code

- A $D-$ary tree, is a finite rooted tree such that D branches stem outward from each node.
- The full $D-$ary tree of length N is the $D-$ary tree with D^N leaves each at depth N from the root.

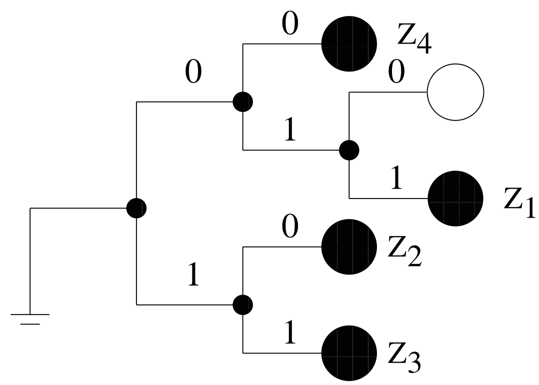
Prefix-free code

- A $D-$ary tree, is a finite rooted tree such that D branches stem outward from each node.
- The full $D-$ary tree of length N is the $D-$ary tree with D^N leaves each at depth N from the root.
- D-ary prefix-free code can be identified with a set of leaves in a D-ary tree.

Prefix-free code

- A $D-$ary tree, is a finite rooted tree such that D branches stem outward from each node.
- The full $D-$ary tree of length N is the $D-$ary tree with D^N leaves each at depth N from the root.
- D-ary prefix-free code can be identified with a set of leaves in a D-ary tree.
- Ex. Draw binary tree for the following code $z_1 = [011], z_2 = [10], z_3 = [11],$ and $z_4 = [00]$

Prefix-free code



Binary tree for prefix-tree code

Outline

Kraft's inequality

- There exists a D-ary prefix-free code whose codeword lengths are the positive integers $w_1, w_2, \cdots, w_K$ if and only if

$$\sum_{i=1}^{K} D^{-w_i} \leq 1$$

Sketch of Proof:

Coding a single random variable Prefix-free code Kraft's inequality

Kraft's inequality

- There exists a D-ary prefix-free code whose codeword lengths are the positive integers $w_1, w_2, \cdots, w_K$ if and only if

$$\sum_{i=1}^{K} D^{-w_i} \leq 1$$

Sketch of Proof:

- In the full D-ary tree of length N, D^{N-w} leaves stem from each node at depth w where $w < N$.

Kraft's inequality

- There exists a D-ary prefix-free code whose codeword lengths are the positive integers $w_1, w_2, \cdots, w_K$ if and only if

$$\sum_{i=1}^{K} D^{-w_i} \leq 1$$

Sketch of Proof:

- In the full D-ary tree of length N, D^{N-w} leaves stem from each node at depth w where $w < N$.
- Suppose there exist a D-ary prefix-free code, construct a tree for the code by pruning the full D-ary tree of length $N = \max_i w_i$ at all vertices corresponding to codewords.

Kraft's inequality

- Due to prefix-free condition, if at depth w_i, we delete D^{N-w_i} leaves, none of these leaves could have been previously deleted.

Kraft's inequality

- Due to prefix-free condition, if at depth w_i, we delete D^{N-w_i} leaves, none of these leaves could have been previously deleted.
- Since there are only D^N leaves that can be deleted, we have

$$D^{N-w_1} + D^{N-w_2} + \cdots + D^{N-w_K} \leq D^N$$

Kraft's inequality

- Due to prefix-free condition, if at depth w_i, we delete D^{N-w_i} leaves, none of these leaves could have been previously deleted.
- Since there are only D^N leaves that can be deleted, we have

$$D^{N-w_1} + D^{N-w_2} + \cdots + D^{N-w_K} \leq D^N$$

- Dividing the above equation by D^N we get the necessary condition for prefix-free code.

Kraft's inequality

- Sketch of converse proof: Suppose $w_1, \cdots, w_K$ are positive integers such that Kraft inequality is satisfied. Consider an ordered list of length $w_1 \leq w_2 \leq \cdots \leq w_K$. Consider the following algorithm

Coding a single random variable Prefix-free code Kraft's inequality

Kraft's inequality

- Sketch of converse proof: Suppose $w_1, \cdots, w_K$ are positive integers such that Kraft inequality is satisfied. Consider an ordered list of length $w_1 \leq w_2 \leq \cdots \leq w_K$. Consider the following algorithm
- (1) $i \leftarrow 1$

Adrish Banerjee Department of Electrical Engineering Indian Institute of Technology Kanpur Kanpur, Uttar Pradesh India
An introduction to Information Theory

Kraft's inequality

- Sketch of converse proof: Suppose $w_1, \cdots, w_K$ are positive integers such that Kraft inequality is satisfied. Consider an ordered list of length $w_1 \leq w_2 \leq \cdots \leq w_K$. Consider the following algorithm
- (1) $i \leftarrow 1$
- (2) Choose z_i as any surviving node or leaf at depth w_i, and prune the tree. Stop if there is no such surviving node or leaf.

Kraft's inequality

- Sketch of converse proof: Suppose $w_1, \cdots, w_K$ are positive integers such that Kraft inequality is satisfied. Consider an ordered list of length $w_1 \leq w_2 \leq \cdots \leq w_K$. Consider the following algorithm
- (1) $i \leftarrow 1$
- (2) Choose z_i as any surviving node or leaf at depth w_i, and prune the tree. Stop if there is no such surviving node or leaf.
- (3) $i = K$ stop , otherwise $i \leftarrow i + 1$

Kraft's inequality

- If we are able to choose z_K in step (2), we are able to construct prefix-free code. Suppose $z_1, z_2, \cdots, z_{i-1}$ have been chosen, the number of surviving leaves at depth N not stemming from any codeword is

$$D^N - (D^{N-w_1} + D^{N-w_2} + \cdots + D^{N-w_{i-1}}) = D^N \left(1 - \sum_{j=1}^{i-1} D^{-w_j}\right)$$

Coding a single random variable Prefix-free code Kraft's inequality

Kraft's inequality

- If we are able to choose z_K in step (2), we are able to construct prefix-free code. Suppose $z_1, z_2, \cdots, z_{i-1}$ have been chosen, the number of surviving leaves at depth N not stemming from any codeword is

$$D^N - (D^{N-w_1} + D^{N-w_2} + \cdots + D^{N-w_{i-1}}) = D^N \left(1 - \sum_{j=1}^{i-1} D^{-w_j}\right)$$

- Therefore the number of surviving leaves at depth N is greater than zero. If there is a surviving leaf at depth N, there must be some unused nodes at depth $w_i < N$ and no already chosen codeword can stem outward from such a surviving node. Thus the surviving node can be chosen as z_i.

Adrish Banerjee Department of Electrical Engineering Indian Institute of Technology Kanpur Kanpur, Uttar Pradesh India
An introduction to Information Theory

Prefix-free code

- Construct a ternary prefix-free code with lengths $w_1 = 1, w_2 = 2, w_3 = 2, w_4 = 3, w_5 = 4$.

Prefix-free code

- Construct a ternary prefix-free code with lengths $w_1 = 1, w_2 = 2, w_3 = 2, w_4 = 3, w_5 = 4$.
- Since $\sum_{i=1}^{5} 3^{-w_i} = 1/3 + 1/9 + 1/9 + 1/27 + 1/81 = 49/81 < 1$, a prefix-free code exists.