Lecture Nodes on Variable Length Coding

Georg Böcherer

www.georg-boecherer.de
Email: mail@georg-boecherer.de

2016

Contents

Outline

Logarithm

The *binary logarithm* and the *natural logarithm* are defined as

$$\log_2 x = a \;\Leftrightarrow\; 2^a = x, \quad \ln x = a \;\Leftrightarrow\; e^a = x. \tag{1}$$

Problem 1.

1. For which real numbers x is the logarithm defined?
2. Express $\log_2 x$ by the natural logarithm.
3. Use the definition of the binary logarithm to derive the following identities.

$$\log_2(xy) = \log_2 x + \log_2 y \tag{2}$$
$$x \log_2 y = \log_2(y^x) \tag{3}$$
$$-\log_2 x = \log_2 \frac{1}{x}. \tag{4}$$

Logarithm

Problem 2.

1. The derivative of the natural logarithm is $\frac{\partial \ln x}{\partial x} = \frac{1}{x}$. Use it to calculate $\frac{\partial \log_2 x}{\partial x}$. Express your result in terms of the binary logarithm.
2. Show that

$$\log_2 x \leq (x - 1) \log_2 e. \tag{5}$$

When does equality hold?

Random Variables

- Random variable X.
- Alphabet $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$.
- Distribution $P_X \colon \mathcal{X} \to \mathbf{R}$, $a \mapsto P_X(a) = \Pr\{X = a\}$ and

$$\forall a \in \mathcal{X} \colon P_X(a) \geq 0 \tag{6}$$

$$\sum_{a \in \mathcal{X}} P_X(a) = 1. \tag{7}$$

- Support: $\operatorname{supp} P_X = \{a \in \mathcal{X} \colon P_X(a) > 0\}$.

Joint Distribution

- Let X, Y be two random variables with joint distribution $P_{XY}\colon \mathcal{X} \times \mathcal{Y} \to \mathbf{R}$.
- The marginal distributions of X and Y are

$$P_X(a) = \sum_{b \in \mathcal{Y}} P_{XY}(a, b), \quad P_Y(b) = \sum_{a \in \mathcal{X}} P_{XY}(a, b). \tag{8}$$

- If $P_Y(b) > 0$, then the distribution of X conditioned on Y is

$$P_{X|Y}(a|b) = \frac{P_{XY}(a, b)}{P_Y(b)}. \tag{9}$$

Joint and conditional distributions: Problems

Problem 3.

1. Show that for each $b \in \operatorname{supp} P_Y$, $P_{X|Y}(\cdot|b)$ is a distribution.
2. Show that if $P_X(a) = 0$ then $P_{XY}(a, b) = 0$ for all $b \in \mathcal{Y}$.
3. Let $P_Y(b) = 0$. Show that the identity

$$P_{X|Y}(a|b) P_Y(b) = P_{XY}(a, b) \tag{10}$$

holds for each $a \in \mathcal{X}$ and each choice of $P_{X|Y}(\cdot|b)$.
Consequently, for $P_Y(b) = 0$, we can freely choose $P_{X|Y}(\cdot|b)$.

Expectation

- Let X be a random variable and consider the real-valued function $f\colon \mathcal{X} \to \mathbf{R}$. The *expectation* of $f(X)$ is defined as

$$\mathrm{E}[f(X)] := \sum_{a \in \operatorname{supp} P_X} P_X(a) f(a). \tag{11}$$

- If $\mathcal{X} \subset \mathbf{R}$, then $\mathrm{E}(X)$ is defined and called *the expectation of* X.

Informational Divergence

The *informational divergence* of two distributions P_X and P_Y with $\mathcal{X} = \mathcal{Y}$ is

$$\mathrm{D}(P_X \| P_Y) = \sum_{a \in \operatorname{supp} P_X} P_X(a) \log_2 \frac{P_X(a)}{P_Y(a)} = E\left[\log_2 \frac{P_X(X)}{P_Y(X)}\right] \tag{12}$$

Informational divergence: Problems

Problem 4.

1. Show that

$$0 \overset{\text{(a)}}{\leq} \mathrm{D}(P_X \| P_Y). \tag{13}$$

   Hint: Use $\log_2 x \leq (x-1)\log_2 e$.

2. When does equality hold in (a)?
3. Provide an example where $\mathrm{D}(P_X \| P_Y) \neq \mathrm{D}(P_Y \| P_X)$.

Entropy

The *entropy* of a random variable X is

$$\mathrm{H}(P_X) := \sum_{a \in \operatorname{supp} P_X} P_X(a)[-\log_2 P_X(a)] = \mathrm{E}[-\log_2 P_X(X)]. \tag{14}$$

Entropy: Problems

Problem 5.

1. Let P_X be some distribution on $\mathcal{X}$ and let P_U be the uniform distribution on $\mathcal{X}$. Show that

$$\mathsf{H}(P_X) = \log_2 |\mathcal{X}| - \mathsf{D}(P_X \| P_U). \tag{15}$$

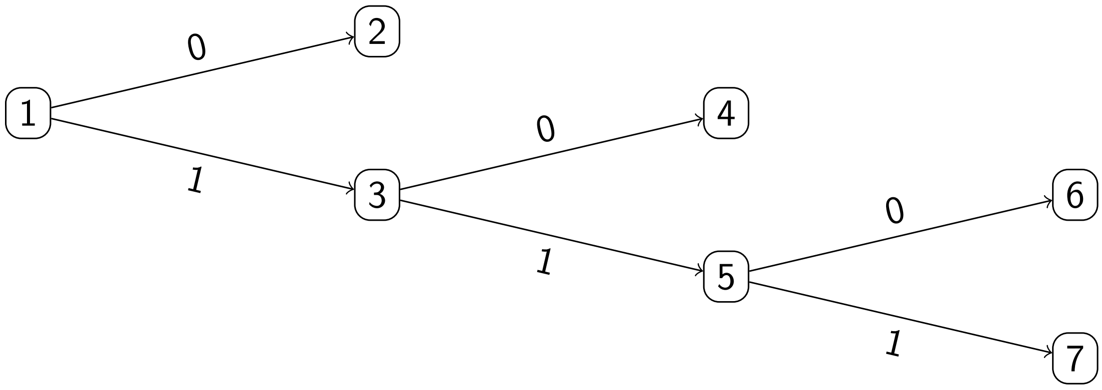
2. Show that

$$0 \overset{(a)}{\leq} \mathsf{H}(P_X) \overset{(b)}{\leq} \log_2 |\mathcal{X}|. \tag{16}$$

3. When does equality hold in (a) and when does equality hold in (b)?

Outline

Rooted Trees: Example



Rooted trees: Nodes

- A node is connected by a *directed edge* with its sucessors.
- A node without sucessors is a *leaf*.
- A node with successors is a *branching node*.
- All node except one have exactly one *predecessor*.
- The node without predecessor is the *root*.
- The *depth* of a node is the number of edges on the path from the root to the node.

Node enumeration

We use the following convention:

- The root has number 1.
- The node numbers increase with increasing depth.
- Leafs have smaller numbers than branching nodes of the same depth.

Problem 6. Suppose a binary rooted tree has n leaves. What is the number of branching nodes? What is the total number of nodes?

Paths and path length

- A sequence of edges connecting the root with a leave is called a *path*.
- The number of edges in a path is the *path length*.
- The path length is equal to the depth of the corresponding leaf.

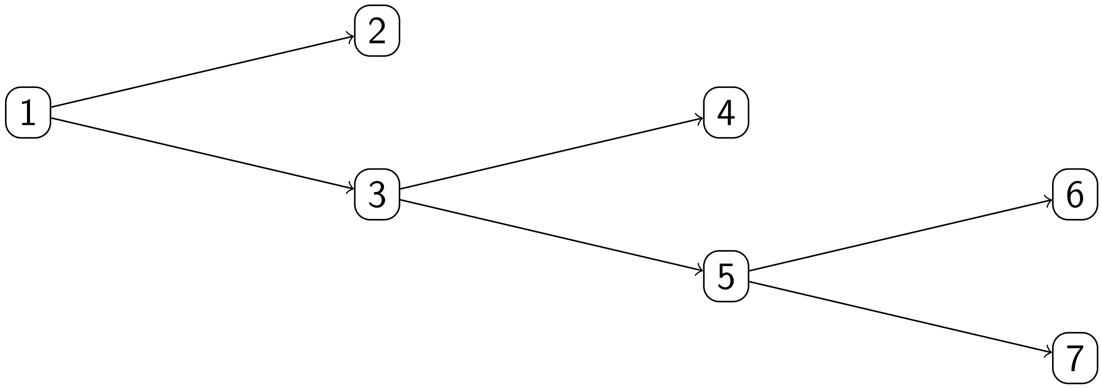
Node Classes

For a rooted tree, we define the following node classes:

- $\mathcal{N}$: all nodes.
- $\mathcal{L}$: leaves.
- $\mathcal{L}_i$: leaves reachable from node i.
- $\mathcal{B} = \mathcal{N} \setminus \mathcal{L}$: branching nodes.

Node Classes: Example

- $\mathcal{N} = \{1, 2, 3, 5, 6, 7\}$
- $\mathcal{L} = \{2, 4, 6, 7\}$
- $\mathcal{L}_3 = \{4, 6, 7\}$
- $\mathcal{B} = \mathcal{N} \setminus \mathcal{L} = \{1, 3, 5\}$



Leaf Distribution

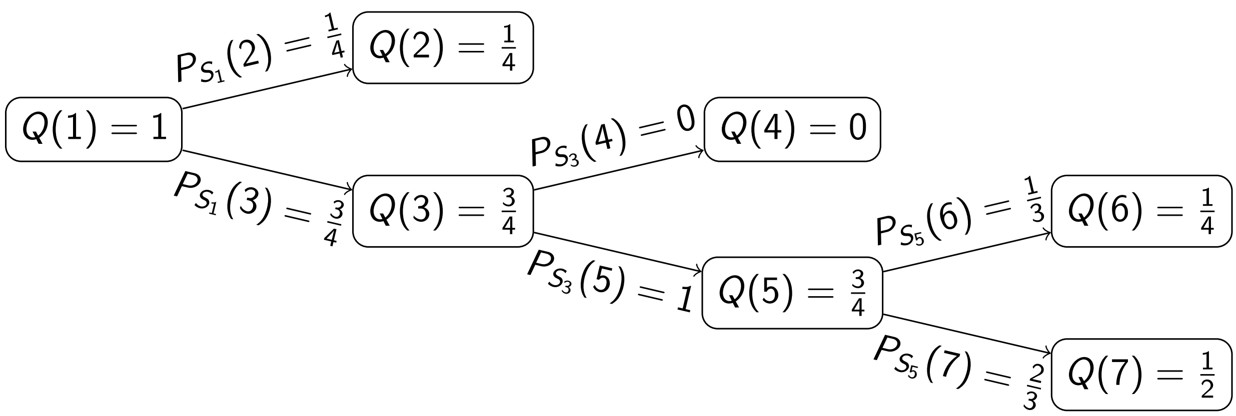
Problem 7. Let L be a random variable with alphabet $\mathcal{L}$ and distribution Q.

1. What is the probability that a path that ends in L passes through node i? We denote this probability by $Q(i)$.
2. Let t be the minimal leaf depth and $s \leq t$. Show that Q defines a distribution on the nodes of depth s.
3. Let $\mathcal{S}_i$ be the successors of i. Suppose a path to L crosses i. What is the probability that it crosses $a \in \mathcal{S}_i$? We denote this branching distribution by P_{S_i} the corresponding random number by S_i.

Example

Leaf distribution: $Q(2) = \frac{1}{4},\ Q(4) = 0,\ Q(6) = \frac{1}{4},\ Q(7) = \frac{1}{2}$.

- $Q(3) = \sum_{i \in \mathcal{L}_3} Q(i) = 0 + \frac{1}{4} + \frac{1}{2} = \frac{3}{4}$
- $\mathcal{S}_1 = \{2, 3\}$
- $P_1(3) = \frac{Q(2)}{Q(1)}$



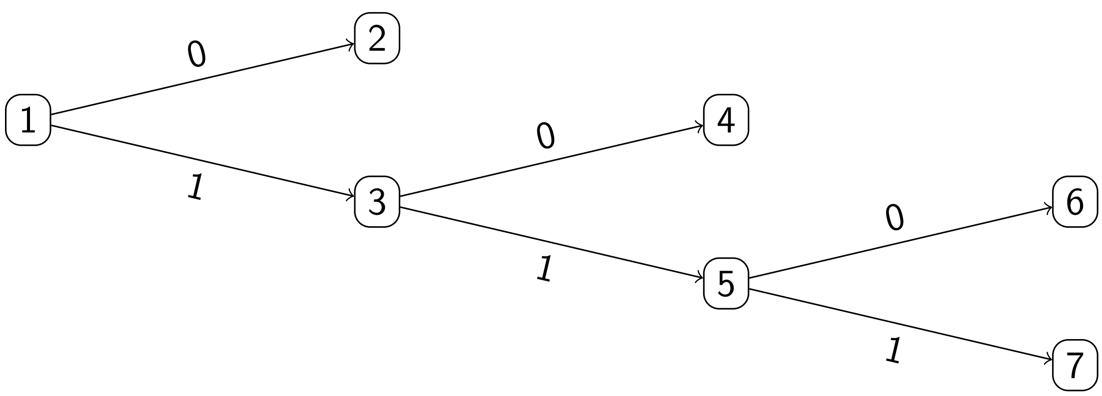
Edge Labels

A tree with edge labels in $\mathcal{X}$ is defined as follows:

- Each node has $|\mathcal{X}|$ sucessors.
- We label the edges emerging from a branching node by the letters in $\mathcal{X}$.
- We define $x(i)$ as the label of the edge that ends in node i.
- The labels of paths through the tree form the set $\mathcal{W}$ of words with letters in $\mathcal{X}$.

Edge labels: Example

Consider the binary labels $\mathcal{X} = \{0, 1\}$.



Branching Distribution

- A label distribution P_X can be used to define a branching distribution:

$$j \in \mathcal{B}, i \in \mathcal{S}_j \colon P_{S_j}(i) = P_X[x(i)]. \tag{17}$$

- P_X also defines a distribution on the words defined by the tree, namely

$$P_X^{\mathcal{W}}(a) = P_X(a_1) \cdots P_X(a_{\ell(a)}), \quad a \in \mathcal{W}. \tag{18}$$

LANSIT[1]

- Let f be a real-valued function on the nodes $\mathcal{N}$.
- For each node $i \in \mathcal{N} \setminus 1$, define $\Delta f(i) := f(i) - f(\text{predecessor of } i)$.
- Let S_i be a random variable with alphabet $\mathcal{S}_i$ and distribution P_{S_i}.

Proposition 1 (LANSIT)

$$\mathrm{E}[f(L)] - f(1) = \sum_{j \in \mathcal{B}} Q(j)\, \mathrm{E}[\Delta f(S_j)] \tag{19}$$

[1] Leaf-Average Node-Sum Interchange Theorem [1].

LANSIT: Proof

- Consider a tree with nodes $\mathcal{N}$.
- Let $\mathcal{S}_j \subseteq \mathcal{L}$ be a set of leaves with common predecessor j.

$$\sum_{i\in\mathcal{S}_j} Q(i)f(i) = \sum_{i\in\mathcal{S}_j} Q(j)P_{S_j}(i)\Big[f(i) - f(j) + f(j)\Big] \tag{20}$$

$$= Q(j)f(j)\underbrace{\Big[\sum_{i\in\mathcal{S}_j} P_{S_j}(i)\Big]}_{=1} + Q(j)\sum_{i\in\mathcal{S}_j} P_{S_j}(i)\Delta f(i) \tag{21}$$

$$= Q(j)f(j) + Q(j)\,\mathrm{E}[\Delta f(S_j)] \tag{22}$$

- $\mathcal{N} \leftarrow \mathcal{N} \setminus \mathcal{S}_j$ is a new tree with fewer leaves. The node probabilities are still defined via Q.
- Repeat the procedure until j has become the root node 1. Then $Q(j=1)=1$ and $Q(j=1)f(j=1)=f(1)$.

□

LANSIT: Problems[2]

Problem 8. Use the LANSIT to show the following identities.

1. **Path Length Lemma.** Function $\ell(i) :=$ node depth of i.

$$\mathrm{E}[\ell(L)] = \sum_{i\in\mathcal{B}} Q(i). \tag{23}$$

2. **Leaf Entropy Lemma.** Function $f(i) = -\log_2 Q(i)$.

$$\mathrm{H}(P_L) = \sum_{i\in\mathcal{B}} Q(i)\,\mathrm{H}(P_{S_i}). \tag{24}$$

3. **Leaf Divergence Lemma.** Let Q' be another node distribution with corresponding leaf distribution $P_{L'}$. Function $f(i) = \log_2 \frac{Q(i)}{Q'(i)}$. Then

$$\mathrm{D}(P_L \| P_{L'}) = \sum_{i\in\mathcal{B}} Q(i)\,\mathrm{D}(P_{S_i}\|P_{S_i'}). \tag{25}$$

[2] See [2].

LANSIT: Aufgaben

Problem 9.

1. Verify the lemmas for Path Length, Leaf Entropy, and Leaf Divergence by calculating for example trees separately the left-hand and the right-hand sides of the identities.

Complete Trees

- A binary tree is *complete*, if each node has either 2 or no successors.
- A tree with edge labels $\mathcal{X}$ is complete if each node has either $|\mathcal{X}|$ or no successors.

Permissible Path Lengths

Let $\ell_1, \ell_2, \ldots, \ell_n$ be path lengths. We want to develop a test, by which we can check, whether or not a complete binary tree with these path lengths exists. Let $\ell_{\max} = \max_i \ell_i$. Consider a complete tree, where all paths have length $\ell_{\max}$.

- The tree has $2^{\ell_{\max}}$ nodes with depth $\ell_{\max}$.
- A node with depth $\ell \leq \ell_{\max}$ has $2^{\ell_{\max}-\ell}$ successors with depth $\ell_{\max}$.
- We have

$$\sum_{i=1}^{n} 2^{-\ell_i} = 2^{-\ell_{\max}} \underbrace{\sum_{i=1}^{n} 2^{\ell_{\max}-\ell_i}}_{(\star)} \tag{26}$$

The sum $(\star)$ is equal to $2^{\ell_{\max}}$, if the ℓ_i are path lengths of a complete tree.

Kraft-Inequality

We now have the following test. Let $\ell_1, \ell_2, \ldots, \ell_n$ be positive integers.

- $\sum_{i=1}^{n} 2^{-\ell_i} = 1 \Rightarrow$ a complete binary tree exists with path lengths ℓ_i.
- $\sum_{i=1}^{n} 2^{-\ell_i} < 1 \Rightarrow$ an incomplete binary exists with path lengths ℓ_i.
- $\sum_{i=1}^{n} 2^{-\ell_i} > 1 \Rightarrow$ neither a complete nor an incomplete binary tree exists with path lengths ℓ_i.

For non-binary labels $\mathcal{X}$ with $|\mathcal{X}| = D > 2$, we test $\sum_{i=1}^{n} D^{-\ell_i}$.

Outline

Discrete Memoryless Source

- A *discrete memoryless source* $\boxed{P_X}$ generates random variables $X_1X_2X_3\cdots$, which are stochastically independent and identically distributed according to P_X.
- Let $n > 0$. We denote $X^n := X_1X_2\cdots X_n$. We have

$$\Pr(X^n = a^n) = P_X(a_1)P_X(a_2)\cdots P_X(a_n) \tag{27}$$

for all $a^n \in \mathcal{X}^n = \mathcal{X} \times \cdots \times \mathcal{X}$.

LANSIT: Problems

Problem 10. Let X^n and Y^n be random vectors. Use the LANSIT to show the following chain rules.

1. Entropy chain rule:

$$H(P_{X^n}) = \sum_{i=1}^{n} H(P_{X_i|X^{i-1}}|P_{X^{i-1}}) \tag{28}$$

2. Informational divergence chain rule:

$$D(P_{X^n}\|P_{Y^n}) = \sum_{i=1}^{n} D(P_{X_i|X^{i-1}}\|P_{Y_i|X^{i-1}}|P_{X^{i-1}}) \tag{29}$$

Distrbution Matcher

A distribution matcher transforms an input sequence into an output sequence:

$$\boxed{P_X} \to X_1X_2\cdots \to \boxed{\text{Distribution Matcher}} \to Y_1Y_2\cdots$$

The output is a sequence of letters in $\mathcal{Z}$, and the frequency by which the letters occur in the output sequence should resemble a target distribution P_Z.

Dictionary and Codebook

- Dicionary:
  - The input letter alphabet is $\mathcal{X}$
  - The path labels of a complete tree with labels in $\mathcal{X}$ form a *dictionary* $\mathcal{W}$.
  - Example: $\mathcal{X} = \{a, b, c\}$, $\mathcal{W} = \{a, b, ca, cb, cc\}$.
- Codebook:
  - The output letter alphabet is $\mathcal{Z}$
  - The path labels of a complete tree with labels in $\mathcal{Z}$ form a *codebook*.
  - Example: $\mathcal{Z} = \{0, 1\}$, $\mathcal{C} = \{0, 100, 101, 110, 111\}$. $\mathcal{C}$ is a binary codebook.

Parsing the Input

- We parse the input by a dictionary $\mathcal{W}$ with letters in $\mathcal{X}$.
- This generates words W with distribution $P_X^{\mathcal{W}}$ given by

$$P_X^{\mathcal{W}}(w) = P_X(w_1)P_X(w_2)\cdots P_X(w_{\ell(w)}), \quad \text{für jedes } w \in \mathcal{W}. \tag{30}$$

Problem 11. Using the LANSIT, show that

$$\mathrm{H}(P_X^{\mathcal{W}}) = \mathrm{E}[\ell(W)]\,\mathrm{H}(P_X) \tag{31}$$

Output of Codewords

We choose as output codewords in $\mathcal{C}$ with letters in $\mathcal{Z}$. The DM maps the parsed words to codewords by an *injective* function $f: \mathcal{W} \to \mathcal{C}$. Let $Y = f(W)$ denote the codeword at the DM output.

- The expected codeword length is $\mathrm{E}[\ell(Y)]$.
- The codeword target distribution is

$$P_Z^{\mathcal{C}}(y) = P_Z(y_1)P_Z(y_2)\dots P_Z(y_{\ell(y)}), \quad \text{for all } y \in \mathcal{C}. \tag{32}$$

- The actual distribution of Y is

$$P_Y(y) = \begin{cases} P_X^{\mathcal{W}}[f^{-1}(y)] & \text{if } \exists w\colon f(w) = y, \\ 0 & \text{otherwise.} \end{cases} \tag{33}$$

Variable length code: Problem

Problem 12. Let $P_Z(0) = P_Z(1) = \frac{1}{2}$ be the target distribution and let $\mathcal{C} = \{0, 10, 11\}$ be the codebook. Suppose the actual distribution is $P_Y(0) = P_Y(10) = P_Y(11) = \frac{1}{3}$.

1. Calculate the target codeword distribution.
2. Calculate the expected codeword length.

Rate

- The DM *Rate* R is given by

$$\frac{\text{average amount of information}}{\text{average output length}} \tag{34}$$

that is

$$R := \frac{\mathrm{H}(P_X^{\mathcal{W}})]}{\mathrm{E}[\ell(Y)]}. \tag{35}$$

The function $f\colon \mathcal{W} \to \mathcal{C}$

We index the dictionary $\mathcal{W} = \{w_1, w_2, \dots, w_n\}$ and the codebook $\mathcal{C} = \{c_1, c_2, \dots, c_m\}$ with $m \geq n$ so that

$$P_X^{\mathcal{W}}(w_1) \geq P_X^{\mathcal{W}}(w_2) \geq \cdots \geq P_X^{\mathcal{W}}(w_n), \tag{36}$$

$$P_Z^{\mathcal{C}}(c_1) \geq P_Z^{\mathcal{C}}(c_2) \geq \cdots \geq P_Z^{\mathcal{W}}(c_m). \tag{37}$$

We then define f by $f\colon w_i \mapsto c_i$, that is, we map words of smaller probability to codewords of smaller target probability.

Outline

Design Objective

For a given DMS $\boxed{P_X}$, we want to maximize the rate

$$R = \frac{\mathrm{H}(P_X^{\mathcal{W}})]}{\mathrm{E}[\ell(Y)]} \tag{38}$$

for a binary output alphabet.

Informational Divergence

We choose as target distribution the uniform distribution P_U on $\mathcal{Z} = \{0, 1\}$ and we evaluate the informational divergence.

$$\mathrm{D}(P_X^{\mathcal{W}} \| P_U^{\mathcal{C}}) \overset{(a)}{=} \sum_{j \in \mathcal{B}} Q(j)\, \mathrm{D}(P_{S_j} \| P_U) \tag{39}$$

$$= \sum_{j \in \mathcal{B}} Q(j) \left[\sum_{a \in \mathcal{S}_j} P_{S_j}(a) \log_2 \frac{P_{S_j}(a)}{\frac{1}{2}} \right] \tag{40}$$

$$= \sum_{j \in \mathcal{B}} Q(j) \left[1 - \mathrm{H}(P_{S_j}) \right] \tag{41}$$

$$\overset{(b)}{=} \mathrm{E}[\ell(Y)] - \mathrm{H}(P_X^{\mathcal{W}}). \tag{42}$$

Equality (a) follows by the Leaf Divergence Lemma and (b) by the Path Length Lemma and the Leaf Entropy Lemma. $Q(j)$ are node probabilities for the codebook tree $\mathcal{C}$ with leaf distribution $P_X^{\mathcal{W}}$.

Limits

- Rate:

$$\frac{\mathrm{H}(P_X^{\mathcal{W}})}{\mathrm{E}[\ell(Y)]} = 1 - \frac{\mathrm{D}(P_X^{\mathcal{W}} \| P_U^{\mathcal{C}})}{\mathrm{E}[\ell(Y)]} \leq 1. \tag{43}$$

- Expected codeword length:

$$\mathrm{E}[\ell(Y)] = \mathrm{H}(P_X^{\mathcal{W}}) + \mathrm{D}(P_X^{\mathcal{W}} \| P_U^{\mathcal{C}}) \geq \mathrm{H}(P_X^{\mathcal{W}}). \tag{44}$$

We can either maximize the rate or minimize the minimize the informational divergence per output bit, over all dicionaries $\mathcal{W}$ and all codes $\mathcal{C}$. *No efficient algorithm is known!*
Note: we achieve the maximum rate, if $\mathrm{D}(P_X^{\mathcal{W}} \| P_U^{\mathcal{C}}) = 0$, the uniform target distribution P_U that we chose before indeed maximizes the rate!

Huffman Coding

For Huffman Coding, we fix the dicionary $\mathcal{W} = \mathcal{X}$. The limits are now

- Rate:

$$\frac{\mathrm{H}(P_X)}{\mathrm{E}[\ell(Y)]} = 1 - \frac{\mathrm{D}(P_X \| P_U^{\mathcal{C}})}{\mathrm{E}[\ell(Y)]} \leq 1. \tag{45}$$

- Expected codeword length:

$$\mathrm{E}[\ell(Y)] = \mathrm{H}(P_X) + \mathrm{D}(P_X \| P_U^{\mathcal{C}}) \geq \mathrm{H}(P_X). \tag{46}$$

To maximize the rate, we can now either minimize the expected codeword length or the informational divergence.

Huffman Coding[3]

- The remaining problem: Choose $\mathcal{C}$, so that the expected output length

$$\mathrm{E}[\ell(Y)] = \sum_{x \in \mathcal{X}} P_X(x)\ell[f(x)] \tag{47}$$

is minimized.

[3] See [3].

Huffman Coding: Problem

For notational simplicity, we denote the probabilities by $p_1, p_2, \dots, p_n$ and the expected output lengths by $\ell_1, \ell_2, \dots, \ell_n$.
Problem 13. Show the following properties of an optimal code.

1. If $p_i < p_j$ then $\ell_i \geq \ell_j$.
2. An optimal codebook is complete.
3. Suppose $p_1 \geq p_2 \geq \cdots \geq p_{n-1} \geq p_n$. Then there exists an optimal codebook with

$$\ell_n = \ell_{n-1} = \max_i \ell_i, \tag{48}$$

that is, the leaves with the lengths ℓ_n, ℓ_{n-1} are siblings with a common predecessor.

Huffman Coding: the algorithm

Suppose the path lengths are optimal and fulfill (48). Let L be a random variable on the leaves. The lengths $\ell_1, \ell_2, \dots, \ell_{n-2}, \ell_{n-1} - 1$ are path lengths of a new tree with the predecessor of the leaves with lengths ℓ_n, ℓ_{n-1} as new leaf. The new leaf has probability $p_n + p_{n-1}$. The new tree has $n-1$ leaves. Let L' be a random variable on the leaves of the new tree. Because of the Path Length Lemma, we have

$$\mathrm{E}[\ell(L)] = \mathrm{E}[\ell(L')] + p_n + p_{n-1}. \tag{49}$$

Because the path lengths of the tree with n leaves is optimal, the path lengths of the new tree with $n-1$ leaves must also be optimal, that is, it must minimize $\mathrm{E}[\ell(L')]$. We then therefore construct the optimal tree, by recursively connecting the leaves of smallest probability to a common predecessor.

Outline

Problem Statement

- At the input, we have independent and uniformly distributed bits $\boxed{P_U}$.
- The output letters $a \in \mathcal{Z}$ have different lengths $v(a)$. The values $v(a)$ are positive real number, but not necessarily integers.
- We want to transmit at maximum rate.
- The length function v defines a *discrete noiseless channel*.

Target distribution

Define the target distribution P_Z as

$$P_Z(a) = 2^{-Cv(a)}, \qquad \text{for each } a \in \mathcal{Z} \tag{50}$$

where C is chosen such that

$$\sum_{a \in \mathcal{Z}} 2^{-Cv(a)} = 1. \tag{51}$$

Zielverteilung

We develop the informational divergence:

$$\begin{aligned} \mathrm{D}(P_Y \| P_Z) &= \sum_{a \in \mathcal{Z}} P_Y(a) \log_2 \frac{P_Y(a)}{2^{-Cv(a)}} && (52) \\ &= C \sum_{a \in \mathcal{Z}} P_Y(a) v(a) - \mathrm{H}(P_Y) && (53) \\ &= C\,\mathrm{E}[v(Y)] - \mathrm{H}(P_Y). && (54) \end{aligned}$$

Cosequently, we have

$$R = \frac{\mathrm{H}(P_Y)}{\mathrm{E}[v(Y)]} = C - \frac{\mathrm{D}(P_Y \| P_Z)}{\mathrm{E}[v(Y)]}. \tag{55}$$

Thus, C is the maximum rate (also called *capacity* of the noiseless channel v) and it is reached, if $P_Y = P_Z$. This shows that the P_Z chosen by us is indeed optimal.

Distribution Matching

We develop the informational divergence.

$$\mathrm{D}(P_U^{\mathcal{W}} \| P_Z^{\mathcal{C}}) \overset{(a)}{=} \sum_{i \in \mathcal{B}} Q(i)\, \mathrm{D}(P_{S_i} \| P_Z) \tag{56}$$

$$\overset{(b)}{=} \sum_{i \in \mathcal{B}} Q(i) \{C\, \mathrm{E}[\Delta v(S_i)] - \mathrm{H}(P_{S_i})\} \tag{57}$$

$$\overset{(c)}{=} C\, \mathrm{E}[v(Y)] - \mathrm{H}(P_U^{\mathcal{W}}). \tag{58}$$

Equality (a) follows by the Leaf Divergence Lemma, (b) follows by (54), and (c) follows by the LANSIT and the Leaf Entropy Lemma. Thus, we have

$$R = \frac{\mathrm{H}(P_U^{\mathcal{W}})}{\mathrm{E}[v(Y)]} = C - \frac{\mathrm{D}(P_U^{\mathcal{W}} \| P_Z^{\mathcal{C}})}{\mathrm{E}[v(Y)]}. \tag{59}$$

Maximizing the rate and equivalently, minimizing the informational divergence per expected codeword length over the dictionary $\mathcal{W}$ *and* the codebook $\mathcal{C}$ is difficult and no efficient algorithm is known.

Fixed Codebook

We fix the codebook. The remaining problem is to minimize

$$\frac{\mathrm{D}(P_U^{\mathcal{W}} \| P_Z)}{\mathrm{E}[v(Y)]} \tag{60}$$

over the dictionary $\mathcal{W}$.

Equivalent Problem

Suppose we would know the minimimum δ, that is

$$\frac{\mathrm{D}(P_U^{\mathcal{W}} \| P_Z)}{\mathrm{E}[v(Y)]} \geq \delta \tag{61}$$

with equality, if $\mathcal{W}$ is optimal. Equivalent are

$$\begin{aligned}
\mathrm{D}(P_U^{\mathcal{W}} \| P_Z) &\geq \delta\, \mathrm{E}[v(Y)] && (62)\\
\Leftrightarrow \mathrm{D}(P_U^{\mathcal{W}} \| P_Z) - \delta\, \mathrm{E}[v(Y)] &\geq 0 && (63)\\
\Leftrightarrow \sum_{a \in \mathcal{W}} P_U^{\mathcal{W}}(a) \left[\log_2 \frac{P_U^{\mathcal{W}}(a)}{P_Z(a)} - \delta v(a)\right] &\geq 0 && (64)\\
\Leftrightarrow \sum_{a \in \mathcal{W}} P_U^{\mathcal{W}}(a) \log_2 \frac{P_U^{\mathcal{W}}(a)}{P_Z(a) 2^{\delta v(a)}} &\geq 0 && (65)\\
\Leftrightarrow \mathrm{D}(P_U^{\mathcal{W}} \| P_Z \circ 2^{\delta v}) &\geq 0. && (66)
\end{aligned}$$

Geometric Huffman Coding[4]

By (66), we know that we must minimize $\mathrm{D}(P_U^{\mathcal{W}} \| T)$ for $T = P_Z \circ 2^{\delta v}$. T is a non-negative function on $\mathcal{Z}$, but not necessarily a distribution. *Geometric Huffman Coding* calculates the optimal dictionary $\mathcal{W}$. The algorithm is similar to Huffman Coding.

- Let $T(a) \geq T(b)$ be the smallest function values. We distinguish to cases.
  1. $T(a) \geq 4T(b)$. We simply remove b.
  2. $T(a) < 4T(b)$. We connect a and b in a common predecessor e. We assign $T(e) = 2\sqrt{T(a)T(b)}$.

We repeat this procedure until we are left with one node only, which is the root of the constructed tree. The constructed tree is the optimal dictionary $\mathcal{W}$.

[4] Proof of optimality is given in [4] and [5, Section 3.2.3]. See also [6].

Finding δ[5]

The following algorithm finds δ and the optimal dictionary $\mathcal{W}$.
Normalized Geometric Huffman Coding

$\hat{\mathcal{W}} \leftarrow \underset{\mathcal{W}}{\operatorname{argmin}} \operatorname{D}(P_U^{\mathcal{W}} \| P_Z)$
repeat
  $\hat{\delta} \leftarrow \frac{\operatorname{D}(P_U^{\hat{\mathcal{W}}} \| P_Z)}{\operatorname{E}[v(Y)]}$
  $\hat{\mathcal{W}} \leftarrow \underset{\mathcal{W}}{\operatorname{argmin}} \operatorname{D}(P_U^{\mathcal{W}} \| P_Z \circ 2^{\delta v})$
until $\hat{\delta} = \frac{\operatorname{D}(P_U^{\hat{\mathcal{W}}} \| P_Z)}{\operatorname{E}[v(Y)]}$
$\delta \leftarrow \hat{\delta}, \mathcal{W} \leftarrow \hat{\mathcal{W}}$

[5] The proof of optimality is provided in [5, Section 4.1.1].

Outline

Further Reading

- Data compression with fixed code: Tunstall Coding [7],[8, Section 2.3.4].
- Distribution matching with fixed dictionary [9].

Outline

References I

[1] R. A. Rueppel and J. L. Massey, "Leaf-average node-sum interchanges in rooted trees with applications," in *Communications and Cryptography: Two sides of One Tapestry*, R. E. Blahut, D. J. Costello Jr., U. Maurer, and T. Mittelholzer, Eds. Kluwer Academic Publishers, 1994.

[2] G. Böcherer and R. A. Amjad, "Informational divergence and entropy rate on rooted trees with probabilities," in *IEEE Int. Symp. Inf. Theory (ISIT)*, 2014. [Online]. Available: http://arxiv.org/abs/1310.2882

[3] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. IRE*, vol. 40, no. 9, pp. 1098–1101, Sep. 1952.

[4] G. Böcherer and R. Mathar, "Matching dyadic distributions to channels," in *Proc. Data Compression Conf.*, 2011, pp. 23–32.

References II

[5] G. Böcherer, "Capacity-achieving probabilistic shaping for noisy and noiseless channels," Ph.D. dissertation, RWTH Aachen University, 2012. [Online]. Available: http://www.georg-boecherer.de/capacityAchievingShaping.pdf

[6] ——, "Geometric Huffman coding," http://www.georg-boecherer.de/ghc, Dec. 2010.

[7] B. Tunstall, "Synthesis of noiseless compression codes," Ph.D. dissertation, 1967.

[8] J. L. Massey, "Applied digital information theory I," lecture notes, ETH Zurich. [Online]. Available: http://www.isiweb.ee.ethz.ch/archive/massey_scr/adit1.pdf

[9] R. A. Amjad and G. Böcherer, "Fixed-to-variable length distribution matching," in *IEEE Int. Symp. Inf. Theory (ISIT)*, 2013. [Online]. Available: http://arxiv.org/abs/1302.0019