Information Theory with Applications, Math6397
Lecture Notes from September 30, 2014

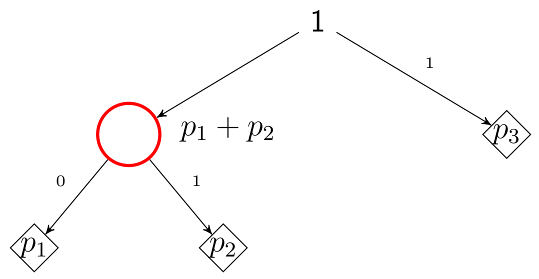
taken by Ilknur Telkes

Last Time

- Kraft inequality (sep.or) prefix code
- Shannon Fano code
- Bound for average code-word length (also for ergodic stationary sources)
- Optimal code trees
  - in binary case, no unused leaves
  - sibling property for code-words belonging to lowest probabilities

2.5.40 Definition. The code tree of a prefix code *labeled with probabilities* is a tree in which each node is labeled by the sum of the probabilities associated with its children.

This means, each node is labeled with the sum of the probabilites of all the descendant leaves. We provide a simple example to illustrate with an alphabet of 3 symbols a_1, a_2, a_3 occurring with probabilities p_1, p_2, p_3, respectively.



(a) $\Longrightarrow$ Root is sum of all probabilities of children

2.5.41 Lemma. *In a code tree of a prefix code labeled with probabilities, the average code-word length is the sum of the probabilities at all nodes (including root, but excluding leaves).*

Proof. If we exclude the root but include the probabilities on all the leaves then the sum is the same, because the root contains the sum of the probability of all the leaves, i.e. the probability one. When counting the leaves instead of the root, we note that a leaf occurring at level k of the tree appears in the sum of probabilities for each of its ancestors, that is, in the sum over all nodes it is counted once at each level from 1 to k, so a total of k times. This means, a leaf associated with probability p at level k contributes kp to the sum, so summing over all leaves gives the expected code-word length. □

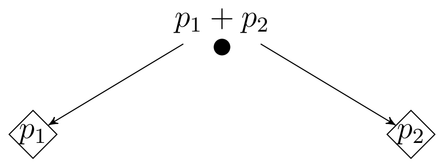
2.5.42 Example. In the tree above, p_1 appears twice so code length is 2; p_3 appears once so code length is 1.

This identity gives us a way to characterize optimal codes, i.e. ones which achieve the smallest expected length for code-words.

2.5.43 Corollary. *An optimal binary prefix code has a tree such that the sum of the probabilities of all nodes is minimal.*

We generalize the sibling property for binary trees.

2.5.44 Corollary. *Among the optimal binary code trees, there is one that has at each level a node with two children associated with the smallest probabilities among the nodes in the next level.*



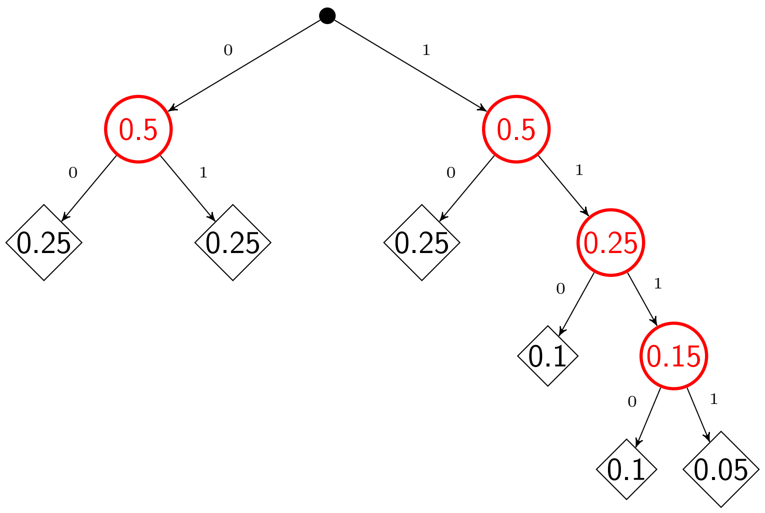
(b) Huffman's idea is to combine two leaves and their parent node into one. Then iterate this procedure to get an optimal tree.

Huffman's algorithm:

- Combine two least probable source symbols into new symbol with the sum of their probabilities.
- Continue with the procedure until only two symbols remain.
- Recover code word by reversing the procedure. If a symbol is split attach to "0" or "1" to code word to indicate the corresponding tree branch.

2.5.45 Example.

$$A = \{1,2,3,4,5,6\}$$
$$P = \{0.25,0.25,0.25,0.1,0.1,0.05\}$$



Codebook:

{{0,0}, {0,1}, {1,0}, {1,1,0}, {1,1,1,0}, {1,1,1,1}}

Average length:

$$\mathbb{E}[\ell(x_j)] = \underbrace{1}_{\text{parent}} + \underbrace{0.5 + 0.5}_{\text{level-1}} + 0.25 + 0.15 = 2.4$$

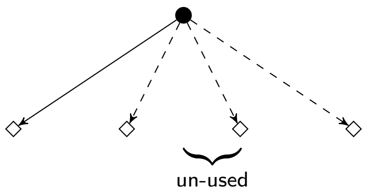
Compare with binary entropy (relevant lower bound):

$$H_2(\mathbb{P}) \approx 2.38$$

2.5.46 Remark. We can do more general form of Huffman's code for K-ary code alphabets, but then the tree can have (at most) $K-2$ un-used leaves.

2.5.47 Question. Why $K-2$?

2.5.48 Answer. If there were $K-1$ or more un-used leaves, switch leaves with siblings of longest code words. From resulting tree, we can remove one level.



From this insight, we also see that all un-used leaves can be chosen to be siblings.

3 Coding for discrete channels

3.1 The discrete memoryless channel

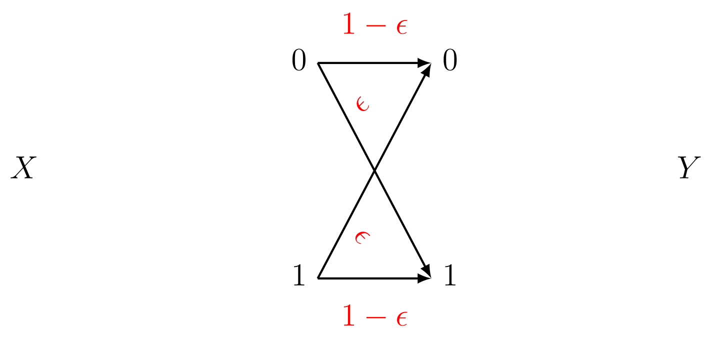
3.1.1 Definition. A discrete memoryless channel (DMC) is a random map $\gamma : \mathbb{A} \times \Omega \to \mathbb{B}$ between discrete alphabets $\mathbb{A}$ and $\mathbb{B}$. It is characterized by conditional probabilities.

$\mathbb{P}(\gamma(a) = b) = \mathbb{W}(b|a)$ for each $a \in \mathbb{A}$

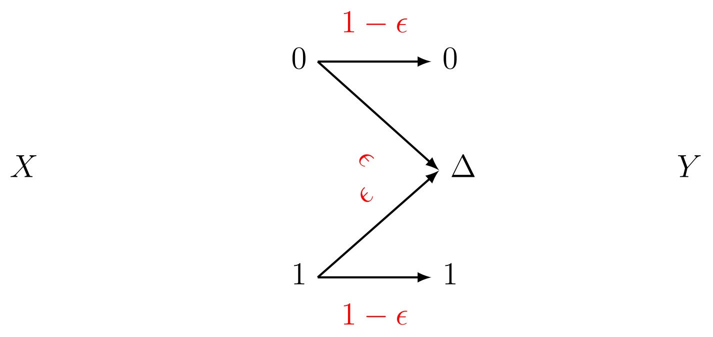
When the context is clear, we often abbreviate $Y = \gamma(X)$

3.1.2 Example.

- Binary symmetric channel with $\mathbb{A}, \mathbb{B} = \{0, 1\}$.



- Binary erasure channel, $\mathbb{A} = \{0, 1\}, \mathbb{B} = \{0, 1, \Delta\}$



3.1.3 Remark. Given a source $\{X_j\}_{j=1}^{\infty}$ with a discrete alphabet $\mathbb{A}$ and a discrete memoryless channel $\gamma : \mathbb{A} \times \Omega \to \mathbb{B}$, then

$$\begin{aligned}\mathbb{P}((Y_1, Y_2, ..., Y_k) = (b_1, b_2, ..., b_k)) &= \sum_{x_1, x_2, ..., x_k} \mathbb{P}((X_1, X_2, ..., X_k) = (x_1, x_2, ..., x_k)) \\ &= \mathbb{W}(b_1, b_2, ..., b_k | x_1, x_2, ..., x_k)\end{aligned}$$

where

$$\mathbb{W}(b_1, b_2, ..., b_k | x_1, x_2, ..., x_k) = \mathbb{V}(b_1|x_1)\mathbb{V}(b_2|x_2)...\mathbb{V}(b_k|x_k),$$

given by product of individual conditional probabilities.

We want to use memory to design a good encoder/decoder pair.

$$\{X_j\}_{j=1}^{k} \xrightarrow{\phi_n} \{X'_j\}_{j=1}^{n} \xrightarrow[channel]{\gamma} \{Y'_j\}_{j=1}^{n} \xrightarrow[reconstruct]{\Psi_n} \{Y_j\}_{j=1}^{k}$$

Idea: $n > k$ helps restore lost information.

3.1.4 Definition. An (n, m) fixed length transmission code for an input alphabet $\mathbb{A}$ and an output alphabet $\mathbb{B}$ is given by $(\mathscr{S}, \phi_n, \Psi_n)$ where $\mathscr{S}$ is the source alphabet of size m:
$|\mathscr{S}|$ = m ,
$\phi_n : \mathscr{S} \to \mathbb{A}^n$,
$\psi_n : \mathbb{B}^n \to \mathscr{S}$.

3.1.5 Definition. The average error probability for an (n, m) fixed-length transmission code and a discrete memoryless channel $\gamma : \mathbb{A} \times \Omega \to \mathbb{B}$, with conditional probability $\mathbb{W}(b|a)_a \in \mathbb{A}$ is

$$P_e = \frac{1}{m} \sum_{x \in \mathscr{S}} \sum_{b \in \mathbb{B}^n \;\; \Psi_n(b) \neq x} \mathbb{W}(b|\phi_n(x))$$

Note: All symbols in the source alphabet $\mathscr{S}$ are assumed to have equal probability. This can be motivated by the assumption that before transmission, the source has been compressed, so all symbols appear with approximately equal probabilitites.

3.1.6 Question. Which input sequences should we choose when encoding?

3.1.7 Answer. A good choice would use most occurring/jointly typical input-output pairs. Otherwise, we would make an effort to encode and decode rare events.

3.1.8 Definition. The set F_δ^n is jointly typical sequences for an input $\{X_j\}_{j=1}^n$ with alphabet $\mathbb{A}$ and a channel $\gamma : \mathbb{A} \times \Omega \to \mathbb{B}$, $Y_n = \gamma(x_j) \in \mathbb{B}$ is defined by

$$\begin{aligned} F_\delta^n = \{(x, y) \in \mathbb{A}^n \times \mathbb{B}^n : &|\frac{1}{n} H(X_1, X_2, ..., X_n) + \frac{1}{n} \ln \mathbb{P}_{X_1,X_2,...,X_n}(x_1, x_2, ..., x_n)| < \delta, \\ &|\frac{1}{n} H(Y_1, Y_2, ..., Y_n) + \frac{1}{n} \ln \mathbb{P}_{Y_1,Y_2,...,Y_n}(y_1, y_2, ..., y_n)| < \delta, \\ &|\frac{1}{n} H(X_1, X_2, ..., X_n, Y_1, Y_2, ..., Y_n) + \frac{1}{n} \ln \mathbb{P}_{X_1,X_2,...,X_n,Y_1,Y_2,...,Y_n}(x_1, x_2, ..., x_n, y_1, y_2, ..., y_n)| < \delta\} . \end{aligned}$$