Recitation 1: Review of Probability, Huffman Code

Lizhong Zheng, September 5, 2013

Review of Probability

- Distinguish: outcome vs. event
  - Example: roll a die
    - outcomes: elements of $\{1, 2, 3, 4, 5, 6\}$
    - events: $\{1\}, \{4\}, \{2, 4, 6\}, \{5, 6\}, \phi, \{1, 2, 3, 4, 5, 6\}$ ...
  - One outcome for each random experiment/realization, multiple events can happen.
  - Each event has a notion of probability.
- Conditioning: conditioned on event A = know that the outcome lies in A

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

  Picture.

- Independence: conditioning on one does not change the probability of the other; product of prob.

$$P(A \cap B) = P(A) \cdot P(B) \quad \Leftrightarrow \quad P(B|A) = P(B)$$

- Total Probability Theorem:
  - Separate into cases
  - Union non-overlapping events $\longleftrightarrow$ sum of probabilities
  - Case A and A^c:

$$\begin{aligned} P(B) &= P(B \cap A) + P(B \cap A^c) \\ &= P(A) \cdot P(B|A) + P(A^c) \cdot P(B|A^c) \end{aligned}$$

  - $A_1, A_2, \ldots, A_N$ form a "cover"

$$\begin{aligned} P(B) = P(A_1) \cdot P(B|A_1) + P(A_2) \cdot P(B|A_2) + \\ \ldots + P(A_N) \cdot P(B|A_N) \end{aligned}$$

- Random Variable: give a name to the random outcomes $X : \Omega \mapsto \mathcal{X}$
  - Example: roll of a die: $\omega \in \Omega = \{1, 2, 3, 4, 5, 6\}$
    * In class $X(\omega) = 1/\omega$, $\mathcal{X} = \{1, 1/2, 1/3, 1/4, 1/5, 1/6\}$
    * $Y(\omega) = \lceil \omega/2 \rceil$, $\mathcal{Y} = \{1, 2, 3\}$
      Picture
  - Induce probability, eg. $P(Y = 2) = P(X = 1/3) + P(X = 1/4)$.
  - Functions of random variables are still random variables
  - Random variable taking a value (a set of values) $\{X = x\}$ is an event.

- Expectation

$$\mathbb{E}[X] = \sum_{x \in \mathcal{X}} P(X = x) \cdot x$$

Meaning from Law of Large Numbers

- Entropy and Surprise

$$\begin{aligned} I(E) &= \log_2 \left(\frac{1}{P(E)} \right) \\ H(X) &= \sum_x P(X = x) \cdot I(\{X = x\}) \\ &= \sum_x P(X = x) \cdot \log_2 \left(\frac{1}{P(X = x)} \right) \end{aligned}$$

Picture: binary entropy

  - Only depends on the distribution (continuous);
  - Always non-negative;
  - Can have non-integer number of bits; (Distinguish format from information content)
  - A particular (rare) outcome can carry very large amount of surprise,
  - On average, entropy is limited by the alphabet: one binary digit can carry no more than 1 bit of information.
  - Goal of source coding: to represent source with binary digits. In order to save the number of digits used, we need each digit to carry close to 1 bit of information. .

Huffman Code

– Several things to remember

(1) A binary code can be represented by a binary tree, each node corresponds to a binary string, and can be used as a codeword, to represent a particular source symbol;

(2) A good code (instantaneously decodable) has codewords as leaves of the tree, so when we read to the end of the word, we know we have reached the end of the word;

eg. Moorse code is not such a "good" code, it needs the extra "PAUSE" symbol;

(3) A good code needs to be a full tree, an empty leaf means some codeword can be shortened;

(4) Outcome with smaller probability should have longer codewords

– Huffman Algorithm: use (3) and (4) above iteratively to construct a code

  * The two least probable outcomes must be "siblings" (codeword differ only on the last digit);
  * This reduces the problem to have one less possible outcome;

Example: [1/3, 1/2, 1/12 ,1/12]

A few extra comments

* Not unique due to ties
* For each outcome $\{X = x\}$, ideally the codeword length matches $I(\{X = x\})$.
* Since $I(\{X = x\})$ is not necessarily an integer, we code over many symbols together to amortize the extras.