Machine Learning: Foundations		Spring semester, 1996/7
Lecture 10	**Information Theory**	May 14, 1997
Lecturer: Yishay Mansour		*Scribe: Mark Last*

10.1 Introduction

10.1.1 Data Transmission

The Theory of Information was developed by Claude Shannon in 1948. The problem is to transmit as much information as possible (from a source A to a destination B) by using minimum number of bits. A special problem arises when a transmission medium (called the "channel") is a noisy one (i.e., the received bits may differ from the sent bits).

Entropy of a discrete random variable X is defined by Shannon as:
$H(X) = -\sum_x p(x)\log p(x)$

The number of bits required to transmit M distinct messages is $\log_2 M$, if every message has the same probability. Let us calculate the average number of bits required to transmit vectors of length n (bits) where k bits are 'one':
Total number of possible messages is: $M = \binom{n}{k}$.
We also know that: $\log M = log\binom{n}{k}$.
Using the Sterling formula, we get:
$\binom{n}{k} \approx \frac{(\frac{n}{e})^n}{(\frac{k}{e})^k(\frac{n-k}{e})^{n-k}} =$
$(\frac{n}{k})^k(\frac{n}{n-k})^{n-k} =$
$2^{k\log\frac{n}{k}+(n-k)\log\frac{n}{n-k}} = 2^{nH(\frac{k}{n})}$
i.e., the average code length (in bits) will be $nH(\frac{k}{n})$.

Typical sequence is defined as a sequence having a number of one's between $(p-\epsilon)n$ and $(p+\epsilon)n$.

10.1.2 Kraft Inequality

Given that every message m_i has a probability p_i ($\sum p_i = 1$), our objective is to find the minimum average code length.
If every message has the same probability ($p_i = \frac{1}{M}$, where M is the total number of messages), the code word will be the serial number of a message and the code length will be

$\log M$ bits.

Definition: A code is *prefix-free* if no code word is a prefix of another code word. At this lesson we will assume that all codes are prefix-free.
Kraft Inequality: There is a prefix-free code with lengths $l_1 ... l_m$ iff $\sum_i 2^{-l_i} \leq 1$.
Proof:
Necessary Condition: Given lengths of code words $l_1 ... l_m$, we can produce random code words by choosing randomly strings of l_i bits. A code word of length l_i has a probability of 2^{-l_i}, because the code is prefix-free and the events are mutually exclusive. This implies: $\sum_i 2^{-l_i} \leq 1$.
Sufficient Condition: We can build a prefix-free code by the following algorithm (called the Huffman Code):
Step 1 - Sort the symbols in the ascending order of probability.
Step 2 - Create two terminal nodes for the least likely unassigned symbols and create a common node for them.
Step 3 - Create a new node for two unassigned symbols with least probability.
Step 4 - If there are unassigned symbols left, go to Step 3.
Step 5 - If there no unassigned symbols left, stop.

10.1.3 Optimal Codelength

Notation: $P_i = Prob[x = i]$
Fano-Shannon Code: $l_i = [\log \frac{1}{p_i}]^+$
The code satisfies the Kraft Inequality: $\sum 2^{-l_i} \leq \sum 2^{-\log \frac{1}{p_i}} = \sum p_i = 1$
The average length of the optimal code: $H(x) \leq \sum l_i p_i = \sum [\log \frac{1}{p_i}] p_i < H(x) + 1$

10.1.4 Lower Bound of Codelength

Let us show that the entropy $H(x)$ is a lower bound of code length for *any* code. We assume that the code words have lengths $l_1 ... l_m$. A new variable q_i is defined by $q_i = 2^{-l_i} (l_i = \log \frac{1}{q_i})$.
Then the average code length is $\sum p_i \log \frac{1}{q_i}$.
By Kraft inequality: $\sum 2^{-l_i} = \sum q_i \leq 1$.
We will assume that q_i is a probability measure ($\sum q_i = 1$).
Information Divergence: $KL(P\|Q) = \sum p_i \log \frac{1}{q_i} - \sum p_i \log \frac{1}{p_i} = \sum p_i \log \frac{p_i}{q_i}$
$-KL(P\|Q) = \sum p_i \log \frac{q_i}{p_i} \leq \log \sum \frac{p_i q_i}{p_i} = \log \sum q_i = 0$ (by Jensen's Inequality)
which implies that $KL(P\|Q) \geq 0$ (if $P = Q$ then $KL(P\|Q) = 0$)
Hence, $\sum p_i \log \frac{1}{q_i} - H(X) \geq 0$, which means that the entropy is the lower bound of the

average code length (we cannot do better than the entropy in data compression).
Interpretation of Information Divergence: the excess code length as a result of assuming the distribution Q while the real distribution is P.

10.1.5 Entropy Function

If we assume that the variables x and y are independent, we can calculate their joint entropy as follows:
$H(x,y) = -\sum_{x,y} p(x,y)\log p(x,y) = -\sum_{x,y} p(x)p(y)[\log p(x) + \log p(y)] =$
(since x and y are independent)
$-\sum_x p(x)\log p(x)\sum_y p(y) - \sum_y p(y)\log p(y)\sum_x p(x) = H(x) + H(y)$
(since $\sum_y p(y) = \sum_x p(x) = 1$)
Conclusion: The optimal coding of two independent sources is to code each one separately.
Source Coding Theorem: Let $Y = x_1 ... x_n$ be a sequence of independent identically distributed (iid) random variables x_i, then $H(Y) = nH(X)$. The minimum expected length of code words for Y (per symbol) satisfies:
$H(X) \leq \frac{1}{n}E[length(Y)] \leq H(X) + \frac{1}{n}$

10.1.6 Mutual Information

Definition: $I(X;Y) = H(X) - H(X/Y) = H(Y) - H(Y/X)$
Interpretation: The reduction in the uncertainty of X due to the knowledge of Y (and vice versa).
$H(Y/X) = \sum_x P(X = x)H(Y/X = x) = E_{x,y}[-\log(Y/X)]$
Chain Rule: $H(X,Y) = H(X) + H(Y/X)$ (can be proved by Bayes Law)

10.1.7 Data Processing Inequality

If $X \to Y \to Z$ then $I(X;Y) \geq I(X;Z)$
Proof: $I(X;Y,Z) = I(X;Z) + I(X;Y/Z) = I(X;Y) + I(X;Z/Y)$
$I(X;Z/Y) = 0, I(X;Y/Z) \geq 0 \Rightarrow I(X;Y) \geq I(X;Z)$

10.2 Model Selection based on the Information Theory

10.2.1 Minimum Description Length

Objective: to transmit a sequence of n random binary variables $(x_1...x_n)$ from source A to destination B.
The trivial approach is to transmit n bits. Another approach is that A and B agree on a hypotheses class H. Then the transmission of a binary sequence will include two stages:

1. A sends to B a hypothesis $h \in H$.

2. A sends to B a list of corrections (sequence bits misclassified by the hypothesis h).

Average description length of a sequence (in transmitted bits) will be: $\|h\| + nH(\frac{k}{n})$
where k is the number of errors in a sequence.
This means that we are interested to minimize the sum of the description size of hypothesis h ($\sqrt{\frac{d}{n}}$) and the description size of the error ($\hat{\epsilon}$) . The longer the description length of the hypothesis, the smaller the percentage of errors and, consequently, the description length of errors will be shorter.
Average description length per sequence bit: $\frac{d}{n} + H(\hat{\epsilon})$
where d is the description length of the hypothesis (in transmitted bits).
Note: The MDL approach does not require independence between sequence bits x_i.

10.2.2 Decision Tree Coding

We denote each internal node by (I, x_i) and each leaf by $(L, 1/0)$. The nodes are sorted in the BFS order: $(I, x_1)(I, x_2)(L, 1)(L, 0)(L, 1)$.
The description size of a single node is:

1. Internal node: $1 + \log m$
   where m is the number of possible coordinates.

2. Leaf node: $1 + 1 = 2$

The description size of a tree having $2k + 1$ nodes (k internal nodes and $k + 1$ leaf nodes)is:

$$k(\log m + 1) + 2(k + 1)$$

Error Coding

A code is selected for k errors out of m. The description size of errors is: $\log \frac{m}{k} = kH(\frac{k}{m})$
If k is small compared to m, the description size will be $k \log m$ (i.e., the list of misclassified places). The error coding will be done separately for every leaf.

Coding of Real Values

Example: Age ≥ 30
The problem: coding of real numbers requires an infinite number of bits.
A solution: use the m values of the sample for coding (requires $\log m$ bits). The coding will be in terms of an interval between two values in a sample (rather than *any* two real values). If the variance of the sample is assumed to be known, the description size can sometimes be reduced to $\log \sqrt{m}$ bits.

10.2.3 Decision Tree Building

Splitting a node means changing the coding. We have seen at Lecture 7 the following splitting criterion:

$$H(q) vs. \tau H(p) + (1 - \tau) H(r)$$

At next lesson we will discuss a method of decreasing the number of leaves after the tree building is completed. This process is called *prunning*.

10.2.4 MDL vs. Bayes

The*MDL method* selects a coding for the hypotheses class H.
The *Bayes method* selects prior probabilities of H.
The two methods are equivalent, since: $l_i = \log \frac{1}{p_i}$.
The reason: The code length is proportional to the a-priori probability.

10.3 Sufficient Statistics

Let us assume, there is a class of distribution functions with parameter θ: $f_\theta \in F$. We take a sample of size n ($X = x_1 ... x_n$) from a population distributed by f_θ. $T(X)$ is function of the sample (e.g., an average).

$$\theta \to X \to T(X)$$
$$I(\theta, T(X)) \leq I(\theta, X) \text{ (by Data Processing Inequality)}$$

In case of equality, no information is lost.
Definition: $T(X)$ is a *sufficient statistics* on θ if it contains all information of X on θ. Formally, this means that θ and X are independent given $T(X)$ $[I(\theta, T(X)) = I(\theta, X)]$. This is equivalent to: $\theta \to T(X) \to X$

10.3.1 Examples

Example 1 - Unfair Coin

x_i is a result of tossing an unfair coin with $\theta = Prob[x_i = 1]$. A sufficient statistics is : $T(x_1 ... x_n) = \frac{\sum x_i}{n}$ (an average). We can show that given $T(X)$, all sequences with the same number of 1's have equal probability, independent of θ:
$Pr[(X_1 ... X_n) = (x_1 ... x_n) / \sum X_i = k] = \frac{1}{\binom{n}{k}}$ (from symmetry).

Example 2 - Normal Distribution

$\varphi_{\mu,\sigma} = \frac{1}{\sqrt{2\pi}\sigma} \exp -(\frac{(x-\mu)^2}{2\sigma^2})$
$T(X_1 ... X_n) = (\frac{\sum x_i}{n}, \sqrt{\frac{x_i^2}{n}})$
Let us look at all sequences of x_i where $\sum x_i = L$ and $\sum x_i^2 = K$:
$Pr[X_1 ... X_n] = \prod \varphi_{\mu,\sigma}(x_i) = (\frac{1}{2\pi})^{\frac{n}{2}} \exp - \sum (\frac{x-\mu}{\sigma})^2 = (\frac{1}{2\pi})^{\frac{n}{2}} \exp - \frac{K^2 - 2\mu L + \mu^2}{\sigma^2}$
Conclusion: all the sequences with the same statistics (K and L) have the same density of probability distribution.

Example 3 - Uniform Distribution

$T(X) = (\min x_i \max x_i)$

10.4 Communication Channel

A *communication channel* is defined as a system consisting of an input alphabet A, an output alphabet B and a conditional probability matrix $P(B/A)$.
In machine learning we are interested in φ_θ. In other words, we would like to obtain as much

information as possible on the input θ, given the output vector b.
The *"information" channel capacity* is defined as $C = \max_{P(\theta)} I(\theta, b)$

A communication system:
Source $\rightarrow$ Encoder $\rightarrow$ Channel $\rightarrow$ Decoder $\rightarrow$ Destination

Explanation: The source sends M different binary sequences of equal length L ($2^L = M$).The encoder provides a code word of length n for each sequence ($X = x_1, ..., x_n$). Let $Y = y_1, ..., y_n$ be the output sequence from the channel corresponding to a code word input. The function of the decoder is to use Y to predict which of the M binary sequences entered the encoder.
The *rate* of an (M, n) code is defined as: $R = \frac{\log M}{n}$.
Interpretation: R in bits is the number of binary digits entering the encoder per transmitted channel digit. We require that the probability of error goes to zero, when the code word length n goes to infinity.

10.4.1 The Channel Coding Theorem

All rates below capacity C are achievable with maximum probability of error $\rightarrow 0$. Conversely, any code with error probability $\rightarrow 0$ must have $R \leq C$.

Lower Bound: We are interested to reveal, with high probability, an input sequence a from an output sequence b, i.e. there should not be an output sequence produced by two distinct input sequences a_1, a_2. One can think of a circle, with a centre at a, containing all code words that are close to a. The radius of the circle is $nH(B/A)$.
However, if we look at the entire set, the total number of possible (typical) B sequences is $\approx 2^{nH(B)}$, since B includes only $H(B)$ bits of "real information". Therefore:

$$\frac{2^{nH(B)}}{2^{nH(B/A)}} = 2^{nI(B;A)} \leq 2^{nC}$$

One can build a random code which is close to this.

10.5 Typical Sequences

Definition:

$$A_\epsilon^n = x_1, ..., x_n : 2^{-n(H(X)+\epsilon)} \leq p(x_1, ..., x_n) \leq 2^{-n(H(X)-\epsilon)}$$

Statements

1. if $A_\epsilon^n \in (x_1, ..., x_n)$ then $H(X) - \epsilon \leq \frac{1}{n} \log p(x_1, ..., x_n) \leq H(X) + \epsilon$
2. For n sufficiently large: $Pr(A_\epsilon^n) > 1 - \epsilon$
3. $(1-\epsilon)2^{n(H(X)-\epsilon)} \leq \|A_\epsilon^n\| \leq 2^{n(H(X)+\epsilon)}$

Conceptually, all elements of the typical set are nearly equiprobable.