COMPSCI 650 Applied Information Theory	Feb 25, 2016

Lecture 10

Instructor: Arya Mazumdar *Scribe: Names Redacted*

1 Background Knowledge

For continuous random variables, the probability density function (PDF) is defined as:

$$P[W \leq w] = F_W(w) \tag{1}$$

$$\frac{dF_W(w)}{dx} \triangleq f_W(w), \tag{2}$$

where $F_W(w)$ is the cumulative distribution function (CDF). For continuous variables, the probability at any point of the PDF is actually 0, but the probability over an interval can be calculated by an integration on PDF.

2 Applications for Binary Hypothesis Testing

2.1 Noisy channel communication problem

In order to send 1 bit of information (either 0 or 1) over a noisy communication channel, some modulation is usually needed. One simple modulation is called Binary phase-shift keying (BPSK), which sends positive or negative electrical signal over the channel. The rule is that it will send electrical signal +1 if the actual signal is 0, and -1 if the actual signal is 1. However, there is always some noise in the communication channel. What is actually received is the sum of the deterministic signal and some random noise:

$$Y = X + W, \tag{3}$$

where Y is the received random variable, X is the actual information, and W is the noise. In many cases, we may assume that W is white Gaussian noise, i.e. $W \sim N(0, \sigma^2)$, with probability density function:

$$f_W(w) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{w^2}{2\sigma^2}}, \tag{4}$$

and the following properties:

$$E[W^2] = \sigma^2, E[W] = 0. \tag{5}$$

After receiving the random variable Y, we have to decide whether 1 was sent or 0 was sent. This is actually a typical binary hypothesis testing problem, in which the two hypotheses are:

$$\begin{aligned} H_1 &: 0 \text{ was sent (electrical signal +1)} \\ H_2 &: 1 \text{ was sent (electrical signal -1)}. \end{aligned} \tag{6}$$

Intuitively, one may easily guess that H_1 is true when a positive signal is received, and bet on H_2 when a negative signal is received, but mathematically, we are doing binary hypothesis testing choosing from the following 2 probability distributions:

$$\begin{aligned} H_1 &: P_1 = +1 + W = W_1 \sim N(1, \sigma^2) \\ H_2 &: P_2 = -1 + W = W_2 \sim N(-1, \sigma^2). \end{aligned} \tag{7}$$

According to Neyman-Pearson lemma, we may construct the test by the likelihood ratio, which in this case is:

$$g(w) = \begin{cases} 1 & \frac{f_{W_1}(w)}{f_{W_2}(w)} \geq T \\ 0 & \text{Otherwise} \end{cases}, \tag{8}$$

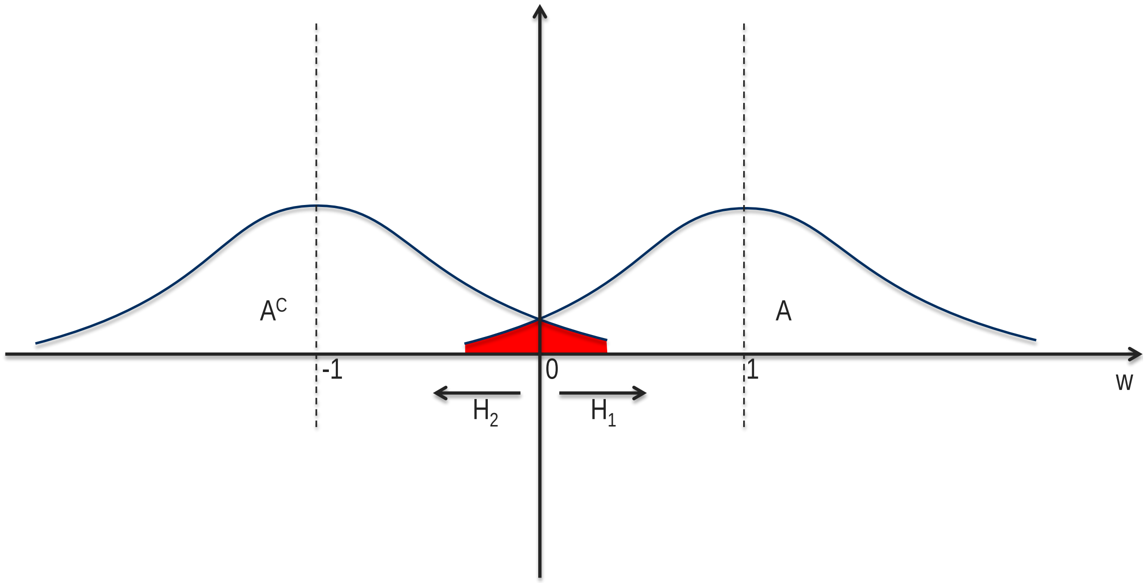

Figure 1: Probability of errors (red shaded area) when threshold T=1

where

$$\frac{f_{W_1}(w)}{f_{W_2}(w)} = \frac{e^{-\frac{(W-1)^2}{2\sigma^2}}}{e^{-\frac{(W+1)^2}{2\sigma^2}}} = e^{\frac{1}{2\sigma^2}[(w+1)^2-(w-1)^2]} = e^{\frac{2w}{\sigma^2}}. \tag{9}$$

And the log likelihood ratio (LLR) is $\frac{2w}{\sigma^2}$. If we set the threshold T to be 1, then $\frac{2w}{\sigma^2} = 0$ or $w = 0$, which perfectly matches our intuition about the decision boundary. If we view this from the Bayes Test point of view, we set $T = 1 = \frac{P(H_2)}{P(H_1)}$ means that we intuitively assume that $P(H_1)$ and $P(H_2)$ are the same. On the other hand, if T is not set as 1, the threshold would depend on σ and T.

2.2 Probability error

There are two types of probability errors that may occur: (1) signal 1 was sent but we output 0; (2) signal 0 was sent but we output 1. As shown in the red shaded area of Figure 1, when the threshold $T = 1$, i.e. $w = 0$, the two types of probability errors can be calculated as:

$$P_1(A^C) = \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(w-1)^2}{2\sigma^2}} dw \tag{10}$$

$$P_2(A) = \int_{0}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(w+1)^2}{2\sigma^2}} dw. \tag{11}$$

We can see from symmetry that $P_1(A^C) = P_2(A)$ when the threshold is $w = 0$ ($T = 1$). Moreover, according to Neyman-Pearson test, we cannot simultaneously optimize both types of probability errors. Instead, we can fix one, and try to minimize the other. In other words, looking at Figure 1, we can move the threshold to the left or right to reduce one of the two shaded areas, but we cannot reduce the two of them at the same time.

2.3 Binary symmetric channel

Suppose we are sending 1 bit of information (0 or 1) over a binary symmetric channel (BSC) with probability of error $0 < p < \frac{1}{2}$, as shown in Figure 2. What strategy should we take to reduce the

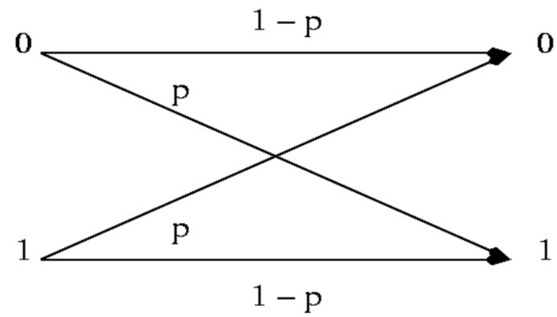

Figure 2: Binary symmetric channel

probability of error? One simple way is sending the bit multiple times, and takes the majority as output.

For example, if we send the bit by 3 times, we are going to have the following mapping:

$$\begin{aligned}000 &\longrightarrow 0\\ 001 &\longrightarrow 0\\ 010 &\longrightarrow 0\\ 011 &\longrightarrow 1\\ 100 &\longrightarrow 0\\ 101 &\longrightarrow 1\\ 110 &\longrightarrow 1\\ 111 &\longrightarrow 1\end{aligned}$$

The erroneous reception happens only when two or three bits are received in error, so the new probability of error is:

$$P_e^{(3)} = p^3 + 3p^2(1-p). \tag{12}$$

Is this better than p? Let's subtract P_3 by p:

$$\begin{aligned}P_e^{(3)} - p &= p^3 + 3p^2(1-p) - p\\ &= p(p^2 + 3p(1-p) - 1)\\ &= p(p-1)(1-2p)\\ &= 2p(p-1)(\frac{1}{2} - p).\end{aligned} \tag{13}$$

So $P_e^{(3)} - p < 0$ when $0 < p < \frac{1}{2}$, which means that we have reduced probability of error by sending the bit by 3 times.

Furthermore, we can actually improve more if we send the bit more times. In fact, when we repeat the bit by n times and output with majority decoding, the probability of error would be

$$P_e^{(n)} = \sum_{i=\frac{n+1}{2}}^{n} \binom{n}{i} p^i (1-p)^{n-i}. \tag{14}$$

As shown in Lecture 6, this probability is

$$P_e^{(n)} = \sum_{i=\frac{n+1}{2}}^{n} \binom{n}{i} p^i (1-p)^{n-i} \approx 2^{-nD(\frac{1}{2}\|p)}. \tag{15}$$

Because $D(\frac{1}{2} \parallel p) > 0$, when $n \to \infty$, this probability of error goes to 0 exponentially fast. However, this is not a free lunch, because the reduction in the probability of error came at the cost of reducing the rate of transmission. Indeed, when we send the bit n times, the rate of information transmission is $\frac{1}{n}$, which also goes to 0 as $n \to \infty$.

2.4 Shannon's theory

In 1948, Shannon showed that there exists a more clever way of adding redundancy to messages, such that the probability of error $P_e^{(n)}$ goes to zero, but the rate of information $R^{(n)}$ goes to a finite positive number called capacity of the communication channel. However, Shannon doesn't tell us how the encoding should be designed. Nowadays, researchers who study error correction code still tries to design coding algorithms to reach the full capacity of the communication channel.

2.5 Hamming code

The first encoding method was introduced by Hamming in 1949. The "Hamming code" encodes 4 bits of information into 7 bits by adding three parity bits, so the rate of transmission is $\frac{4}{7}$, but the probability of error P_e is better than the repetition of 7 times for certain values of p that is small enough. We will discuss Hamming code later in this course.

3 Multi-armed bandit

The multi-armed bandit problem is a problem in which a gambler at a row of slot machines (sometimes known as "one-armed bandits") has to decide which machines to play, how many times to play each machine and in which order to play them. The gamblers may pick a strategy at the beginning, and update their choice based on what they have observed while gambling.

Now suppose there are T steps, N choices of actions in each step, and the gambler's actual actions in all steps are called a_1, a_2, ..., a_T. We define:

$$l_t(i) = \text{loss of action i at time t} \tag{16}$$

$$\text{Regret} = \sum_{t=1}^{T} l_t(a_t) - \min_{i \in \{1,\ldots,N\}} \sum_{t=1}^{T} l_t(i). \tag{17}$$

The gamblers never know which action is optimal at the beginning, but they may update their choice while they are gambling, i.e. the reinforcement learning process. Now we are going to show that, no matter what types of strategy the gambler is following, the regret is always lower-bounded by

$$\text{Regret} \geq O(\sqrt{NT}). \tag{18}$$

Moreover, there exists an algorithm that provides the regret to the order of $O(\sqrt{NT \log N})$, which is not far from optimal.

3.1 Proof of regret bound

Suppose we have two coins, one of which is biased:

Coin 1: P[head] $= \frac{1}{2}$, P[tail] $= \frac{1}{2}$

Coin 2: P[head] $= \frac{1}{2} - \epsilon$, P[tail] $= \frac{1}{2} + \epsilon$.

Now if we choose one of the two coins, and keep tossing it for m times, how can we tell which one is biased, with probability = 0.99? Note that the probability of getting it right without tossing is $\frac{1}{2}$. Now we claim that in order to improve the probability to 0.99, we need to toss the coin $m = \Omega(\frac{1}{\epsilon^2})$ times.

...(to be continued in the next class)