

Lecture 1: Jan 13

*Lecturer: Aarti Singh**Scribes: Akshay Krishnamurthy, Min Xu*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

1.1 About the class

This class focuses on information theory, signal processing, machine learning, and the connections between these fields. Both signal processing and machine learning are about how to extract useful information from signals/data. Classically signals in signal processing involve a temporal component and are usually transmitted over a channel while data in machine learning may be slightly more general. A more fundamental difference between signal processing/information theory and machine learning is that signals and channels are often *designed* in the former, whereas in ML, we do not have much control over the data generating distribution.

Information theory is about how much information is available in data, how much information can possibly be extracted from a signal, or how much information can be transmitted over a channel. Information theory studies two main questions:

1. How much information is contained in a signal/data?

Example 1. *Consider the data compression (source coding) problem.*

$$\text{Source} \rightarrow \text{Compressor} \rightarrow \text{Decompressor} \rightarrow \text{Receiver}.$$

What is the fewest number of bits needed to describe a source (or a message) while preserving all the information, in the sense that a receiver can reconstruct the message with low probability of error?

2. How much information can be reliably transmitted through a noisy channel?

Example 2. *Consider the data transmission (channel coding) problem.*

$$\text{Source} \rightarrow \text{Compressor} \rightarrow \text{Channel} \rightarrow \text{Decompressor} \rightarrow \text{Receiver}.$$

What is the maximum number of bits per channel use that can be reliably sent through a channel in the sense that the receiver can reconstruct the message with low probability of error? Since the channel is noisy, we will have to add redundancy into the messages. A related question is: how little redundancy do we need so that the receiver can still recover our messages?

Remark 1. *Data compression or source coding can be thought of as a noiseless version of the data transmission/channel coding problem.*

Connection to Machine Learning:

1. Source coding in ML: in ML, the source can be thought of as a model (for example $p(X_1, \dots, X_n)$) that generates data points X_1, \dots, X_n , and the least number of bits needed to encode this data reflect the complexity of the source or model. Ideas from source coding can be used to pick a descriptive model with low complexity, in line with the principle of Occam's Razor.
2. Channel coding in ML: The channel specifies a distribution $p(y|x)$ where x is the input to the channel and y is the output. For instance, we can view the output $y_i = m(x_i) + \epsilon$ in regression as the output of a noisy channel that takes $m(x_i)$ as input. Similarly, in density estimation, x can be a parameter and y is a sample generated according to $p(y|x)$.

We will return to these later in the course.

1.2 Information Content of a Random Experiment

We will define the information content of a random outcome X to be $\log_2(1/p(X))$. To see why this is a useful definition, let's turn to some examples.

1. We choose an integer from 0 – 63 uniformly at random. What is the smallest number of yes/no questions needed to identify that integer? Answer: $\log_2(64) = 6$ which can be achieved by doing binary search.

Note that all outcomes have probability $p = 1/64$, $\log_2(64) = \log_2(1/p) = 6$. This is the **Shannon Information Content**.

2. Here is an experiment that does not lead to equi-probable outcomes. An enemy ship is located somewhere in a 8×8 grid (64 possible locations). We can launch a missile that hits one location. Since the ship can be hidden in any of the 64 possible locations, we expect that we will still gain 6 bits of information when we find the ship. However, each question (firing of a missile) now may not provide the same amount of information.

The probability of hitting on first launch is $p_1(h) = 1/64$, so the Shannon Information Content of hitting on first launch is $\log_2(1/p_1(h)) = 6$. Since this was a low probability event, we gained a lot of information (in fact all the information we hoped to gain on discovering the ship). However, we will not gain the same amount of information on more probable events. For example:

The information gained from missing on the first launch is $\log_2(64/63) = 0.0227$ bits.

The information gained from missing on the first 32 launches is:

$$\begin{aligned} \sum_{i=1}^{32} \log_2(p_i(m)) &= \log_2\left(\prod_{i=1}^{32} p_i(m)\right) \\ &= \log_2\left(\frac{64}{63} \frac{63}{62} \cdots \frac{33}{32}\right) \\ &= \log_2(2) = 1 \text{ bit} \end{aligned}$$

This is intuitive, since ruling out 32 locations is equivalent to asking one question in the previous experiment.

If we hit on the next try, we will gain $\log_2(1/p_{33}(h)) = \log_2(32) = 5$ bits of information. Simple calculation will show that, regardless of how many launches we needed, we gain a total of 6 bits of information whenever we hit the ship.

3. What if the questions are allowed to have more than 2 answers?

Suppose I give you 9 balls and tell you that one ball is heavier. We have a balance and we want to find the heavier ball with the fewest number of weighings. There are now three possible outcomes of an experiment: left side heavier, right side heavier, or equal weight. The minimum number of experiments is then $\log_3(\text{number of balls}) = \log_3(9) = 2$.

The strategy is to split the balls into three groups of three and weigh one group against another. If these two are equal, you split the last group into three and repeat. If the first group is heavier, you split it and repeat and vice versa for the second group.

Note that to gain a lot of information, you want to design experiments so that each outcome is equally probable.

Suppose now that the odd ball is either heavier or lighter, but I don't tell you which. How many weighings do we need? There are 18 possible outcomes so information theory tells us $\log_3(18)$ experiments would be necessary, and this number is between 2 and 3. Note that we may need more; information theoretic bounds are often not achievable.

1.3 Information Content of Random Variables

A random variable is an assignment of probability to outcomes of a random experiment. We then define the information content of a random variable as the average Shannon Information Content of the outcomes. We can also think of it as a measure of uncertainty about the random variable.

Definition 2. The *entropy* of a random variable X with probability distribution $p(x)$ is:

$$H(X) = \sum_{x \in \mathcal{X}} p(x) \log_2(1/p(x)) = -\mathbb{E}_{X \sim p}[\log_2 p(X)], \quad (1.1)$$

where \mathcal{X} is the set of all possible values of X . We often write $H(X) = H(p)$ since entropy is a property of the distribution.

Here are some examples:

- If $X \sim \text{Uniform}(\mathcal{X})$, then $H(X) = \sum_{x \in \mathcal{X}} \frac{1}{|\mathcal{X}|} \log_2(|\mathcal{X}|) = \log_2(|\mathcal{X}|)$.
- If $X \sim \text{Bernoulli}(p)$ then $H(X) = H(p) = -p \log_2 p - (1-p) \log_2(1-p)$

Below are several definitions that will appear throughout the course.

Definition 3. The *joint entropy* of two random variables X, Y with joint distribution $p(x, y)$ is:

$$H(X, Y) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x, y) \log_2 \left(\frac{1}{p(x, y)} \right) \quad (1.2)$$

Definition 4. The *conditional entropy* of Y conditioned on X is the average uncertainty about Y after observing X . Formally:

$$H(Y|X) = \sum_{x \in \mathcal{X}} p(x) H(Y|X=x) = \sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log_2 \left(\frac{1}{p(y|x)} \right) \quad (1.3)$$

Definition 5. Given two distributions p, q for a random variable X , the **relative entropy** between p and q is:

$$D(p||q) = \mathbb{E}_{X \sim p}[\log 1/q(X)] - \mathbb{E}_{X \sim p}[\log 1/p(X)] = \mathbb{E}_p[\log p/q] = \sum_x p(x) \log \frac{p(x)}{q(x)} \quad (1.4)$$

The relative entropy is also known as the **Information divergence** or the **Kullback-Leibler (KL) divergence**.

The relative entropy is the cost incurred if we used distribution q to encode X when the true underlying distribution is p .

Definition 6. Let X, Y be two random variables. The **Mutual Information** between X and Y is the KL-divergence between the joint distribution and the product of the marginals. Formally:

$$I(X; Y) = D(p(x, y) || p(x)p(y)), \quad (1.5)$$

where $p(x, y)$ is the joint distribution of X, Y and $p(x), p(y)$ are the corresponding marginal distributions. Thus, $p(x)p(y)$ denotes the joint distribution that would result if X, Y were independent.

The mutual information quantifies how much dependence there is between two random variables. If $X \perp Y$ then $p(x, y) = p(x)p(y)$ and $I(X; Y) = 0$.

1.4 Connection to Maximum Likelihood Estimation

Suppose $X = (X_1, \dots, X_n)$ are generated from a distribution p (for example $X_i \sim p$ i.i.d.). In maximum likelihood estimation we want to find a distribution q from some family \mathcal{Q} such that the likelihood of the data is maximized.

$$\max_{q \in \mathcal{Q}} q(X) = \min_{q \in \mathcal{Q}} -\log q(X)$$

In machine learning, we often define a loss function. In this case, the loss function is the negative log loss: $\text{loss}(q, X) = -\log q(X)$. The expected value of this loss function is the risk: $\text{Risk}(q) = \mathbb{E}_p[\log 1/q(x)]$. We want to find a distribution q that minimizes the risk. However, notice that minimizing the risk with respect to a distribution q is exactly minimizing the relative entropy between p and q . This is because:

$$\text{Risk}(q) = \mathbb{E}_p[\log 1/q(x)] = \mathbb{E}_p \left[\log \frac{p(x)}{q(x)} \right] + \mathbb{E}_p \left[\log \frac{1}{p(x)} \right] = D(p||q) + \text{Risk}(p)$$

Because we know that relative entropy is always non-negative, we know here that the risk is minimized by setting q equal to p . Thus the minimum risk $R^* = \text{Risk}(p) = H(p)$, the entropy of distribution p . The excess risk, $\text{Risk}(q) - R^*$ is precisely the relative entropy between p and q .