Information Theory and Communication

Ritwik Banerjee

rbanerjee@cs.stonybrook.edu
Information Theory and Artificial Intelligence

Signal and Data

- In AI, and in particular in machine learning (ML), the focus is often on gleaning useful information from real-world phenomena . . .
- in the form of **signal** or **data** (e.g., speech, video, images, text).
  - Some literature draws a clear distinction between signal and data, while others say that they are largely overlapping concepts.
  - E.g., in electrical engineering, signals can be different from data in two ways: (i) signals have a temporal aspect, and (ii) are usually transmitted through a medium known as a **channel**.
Two fundamental questions in Information Theory

▶ How much information is contained in the signal (or data)?

Source $\rightarrow$ Compressor $\rightarrow$ Decompressor $\rightarrow$ Receiver

This is the data compression (a.k.a, the **source coding**) problem.

▶ What is the fewest number of bits needed to describe the output of a source while preserving all the information so that a receiver can reconstruct the message from the bits with arbitrarily low error probability?

▶ How much information can be reliably transmitted through a given channel?

Source $\rightarrow$ Encoder $\rightarrow$ Channel $\rightarrow$ Decoder $\rightarrow$ Receiver

This is the data transmission (a.k.a the **channel coding**) problem.

▶ What is the maximum number of bits per channel use that can be reliably sent through a noisy channel so that a receiver can reconstruct the message with arbitrarily low error probability?
Connection to AI and ML

- Machine learning algorithms can be divided into two main groups:
  1. **Predictive models** where the algorithm tries to predict the value of one target variable using the values of the other variables.
     This may or may not be about predicting future events.
  2. **Descriptive models** are used where we need to summarize the data and find new insights. In these models, there is no target variable.
     E.g., dividing the data into homogeneous **clusters** where similar data points are grouped together, and (hopefully) far away from dissimilar data points.

Source Coding in Machine Learning

- In ML, the “source” is a model that generates data points $X_1, \ldots, X_n$. Usually, the model is an approximation of the joint probability mass function $p(x_1, \ldots, x_n)$.
- The least number of bits needed to encode these data reflect the **complexity** of the source.
- Source coding can be used to pick a descriptive model with the least complexity.
connection to AI and ML

- Machine learning algorithms can be divided into two main groups:
  1. **Predictive models** where the algorithm tries to predict the value of one target variable using the values of the other variables.
     This may or may not be about predicting future events.
  2. **Descriptive models** are used where we need to summarize the data and find new insights. In these models, there is no target variable.
     E.g., dividing the data into homogeneous clusters where similar data points are grouped together, and (hopefully) far away from dissimilar data points.

**Channing Coding in Machine Learning**

- The channel specifies a distribution $p(y|x)$ where $x$ and $y$ are the input and output, respectively.
- E.g., 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.
Maximum Likelihood Estimation: Basic Idea

- We observe various phenomena/events. What we observe are facts.
- These facts have some underlying processes generating them.
- The processes themselves are hidden and unobserved.
- But if we somehow figure out what the process is, then we can answer the “why” behind the observed events.

Key Question

Given an observed event $e$, and several “candidate” processes $P_1, P_2, \ldots P_k$, what is the likelihood that $P_i \ (1 \leq i \leq k)$ generated the event $e$?

- The best explanation we have is the $i$ for which the likelihood of $P_i$ generating $e$ is the maximum among all $1 \leq i \leq k$.
- Maximum likelihood estimation (MLE) is a function that extracts this highest likelihood.
Maximum Likelihood Estimation: An Example

- Estimate the probability $\pi$ of getting a head upon tossing a coin:
  - We toss the coin 10 times, and observe $HHTHHHTTHH$.
  - If we had not actually collected the data, the probability of observing this sequence is a function of the hidden parameter $\pi$
    \[ Pr(\text{data}|\text{parameter}) = Pr(HHTHHHTTHH|\pi) = \pi^7(1 - \pi)^3 \]
  - But the data for our observed sample is fixed, and so is the parameter $\pi$. However, $\pi$ is a fixed unknown.
  - So we let $\pi$ vary between 0 and 1 and see how the probability of the observed data changes with different values of $\pi$.
  - That is, treat the probability of the observed data as a function of $\pi$. This function is called the likelihood function:
    \[ \mathcal{L}(|HHTHHHTTHH) = \pi^7(1 - \pi)^3 \]

- **NOTE:** The probability function is a function of the data with the value of the parameter fixed, while the likelihood function is a function of the parameter with the data fixed.
Maximum Likelihood Estimation: An Example

Figure: The complete graph of the likelihood function $L$. The MLE is given by $\hat{\pi} = 0.7$. 
In general, for \( n \) independent tosses of a coin that produce \( x \) heads, 
\[
\mathcal{L}(\pi) = Pr(\text{observed data} \mid \pi) = \pi^x (1 - \pi)^{n-x}
\]

The value of \( \pi \) that maximizes \( \mathcal{L}(\pi) \) also maximizes its logarithm, which is easier to compute:
\[
\log \mathcal{L}(\pi) = x \log \pi + (n - x) \log(1 - \pi)
\]

Thus, in MLE, we often directly speak of the log-likelihood.

In this coin toss scenario, differentiation the log-likelihood function w.r.t. \( \pi \) produces
\[
\frac{\partial}{\partial \pi} \log \mathcal{L}(\pi) = \frac{x}{\pi} - \frac{n - x}{1 - \pi},
\]
which is zero when \( \pi = x/n \).

That is, the maximum likelihood estimate is \( \hat{\pi} = x/n \).
Connection to Maximum Likelihood Estimation (MLE)

- Suppose $X = (X_1, ..., X_n)$ are data generated from a distribution $p$.
- In MLE, we want to find a distribution $q$ in some family of distributions $Q$ such that the likelihood $\mathcal{L}(q)$ is maximized. This is often done by minimizing the negative logarithm:

$$\hat{q} = \arg\max_{q \in Q} \mathcal{L}(q) = \arg\min_{q \in Q} ( - \log \mathcal{L}(q) )$$

- The negative log $-\log q_X$ is called a **loss function**.
- The expected value of this loss is the **risk**:

$$\text{Risk}(q) = E \left( \log \frac{1}{q(x)} \right)$$

- But note that

$$E \left( \log \frac{1}{q(x)} \right) = E \left( \log \frac{p(x)}{q(x)} \right) + E \left( \log \frac{1}{p(x)} \right) = D(p||q) + \text{Risk}(p)$$
Connection to AI and ML

Connection to Maximum Likelihood Estimation (MLE)

- The goal is to find a distribution $q$ that minimizes the risk.
- Since we know that KL divergence (i.e., relative entropy) is always non-negative, the risk is minimized when $p = q$.
- That is, the minimum risk is given by

$$\hat{R} = \text{Risk}(p) = H(p)$$

- The uncertainty associated with the original distribution that generated the data is the minimum risk.
- The additional risk of a different distribution $q \neq p$, that is, $\text{Risk}(q) - \hat{R}$, is the divergence of $p$ from $q$. 