Lecture #01 - More insights: Fitting, Ein/Eout, MLE/MAP

[Murphy 4.1, pp. 103]

- The process of estimating θ from data $\mathcal{D}$ is called model **fitting** or **training**. This is at the heart of machine learning!

- While there are many methods for producing such estimates, most boil down to an optimization problem of the form:

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}(\theta)$$

$\mathcal{L}(\theta)$: some kind of Loss/function. objective.

Definition of MLE (Murphy 4.2, pp. 103)

$$\hat{\theta}_{mle} \triangleq \arg\max_{\theta} \underbrace{p(\mathcal{D}|\theta)}_{\text{conditional likelihood}}$$

- By assuming training examples i.i.d.:

$$p(\mathcal{D}|\theta) = \prod_{n=1}^{N} p(y_n|x_n,\theta)$$

- Then, use Log-Likelihood:

$$LL(\theta) \triangleq \log p(\mathcal{D}|\theta) = \sum_{n=1}^{N} \log p(y_n|x_n,\theta)$$

$$\hat{\theta}_{mle} = \arg\max_{\theta} \sum_{n=1}^{N} \log p(y_n|x_n,\theta)$$

- Because optimization algos usually minimize, we can define Negative Log-Likelihood:

$$NLL(\theta) \triangleq -\log p(\mathcal{D}|\theta) = -\sum_{n=1}^{N} \log p(y_n|x_n,\theta)$$

$$\hat{\theta}_{mle} = \arg\min_{\theta} -\sum_{n=1}^{N} \log p(y_n|x_n,\theta)$$

Justification of MLE (Murphy 4.2, pp.104)

1 - One way to view MLE is as simple point approximation to the Bayesian posterior $p(\theta|\mathcal{D})$ using a uniform prior. (See Murphy 6.7.1)

- In particular, suppose we approximate the posterior by a delta function:

$$p(\theta|\mathcal{D}) = \delta(\theta - \hat{\theta}_{map})$$

- where $\hat{\theta}_{map}$ is the maximum a posteriori (MAP), given by:

$$\hat{\theta}_{map} = \arg\max_\theta \log p(\theta|\mathcal{D}) = \arg\max_\theta \log p(\mathcal{D}|\theta) + \log p(\theta)$$

- if we use a uniform prior, $p(\theta) \propto 1$ and the MAP estimate becomes equal to MLE:

$$\hat{\theta}_{map} = \hat{\theta}_{MLE} = \arg\max_\theta \log p(\mathcal{D}|\theta)$$

2 - Another way to justify the use of MLE is that the resulting predictive distribution $p(y|\hat{\theta}_{MLE})$ is as "close as possible" to the empirical distribution of the data $\mathcal{D}$. In the unconditional case, the empirical distribution is defined by:

$$p_{\mathcal{D}}(y) \triangleq \frac{1}{N}\sum_{n=1}^{N} \delta(y - y_n)$$

: A series of delta or "spikes" at the observed training points.

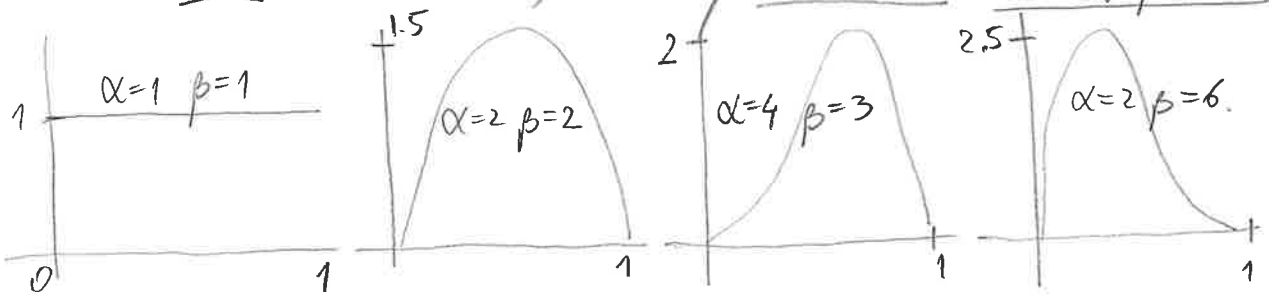
(!) - We want to create a model whose distribution $q(y) = p(y|\theta)$ is similar to $p_{\mathcal{D}}(y)$.

- See Murphy 4.2 pp.105 for more discussion on (dis)similarity and how Kullback Leibler (KL) divergence is used to show that minimizing KL divergence is equivalent to minimizing NLL!

Ufenn Description: $\begin{cases} n_H = k \text{ Heads} \\ n_T = n-k \text{ Tails} \end{cases}$ in n flips experiment.

3 The Bayesian Way

- in the context of the coin-flips experiment: If you are a Bayesian, (you will embrace uncertainty and quantify it) by assuming that your prior belief about coin fairness can be described with a distribution $P(\theta)$.
- in particular the Beta distribution, is a very convenient class of priors:



- Bayesian framework uses Data to update its prior distribution over the parameters. Using Bays Rule, we obtain a posterior over parameters:

$$P(\theta|\mathcal{D}) = \frac{P(\mathcal{D}|\theta)\cdot P(\theta)}{P(\mathcal{D})} \propto P(\mathcal{D}|\theta)P(\theta)$$

- Beta distribution:

Beta(α,β): $P(\theta \mid \alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \cdot \theta^{\alpha-1} \cdot (1-\theta)^{\beta-1}$

where Γ is gamma function $\Gamma(n) = (n-1)!$ $n>0$

α,β hyperparameters $\alpha>1$, $\beta>1$ control shape of prior: $\begin{cases} \text{their sum controls "peakiness"} \\ \text{their ratio controls left-right bias.} \end{cases}$

- Beta is convenient because:

$$P(\theta|\mathcal{D}) \propto P(\mathcal{D}|\theta)\cdot P(\theta) \propto \theta^{n_H+\alpha-1}\cdot(1-\theta)^{n_T+\beta-1}$$

Hence: $P(\theta|\mathcal{D}) = \text{Beta}(n_H+\alpha, n_T+\beta)$

This property of fit between a model and its prior is called conjugecy $\equiv$ means the posterior is of the same distributional family as the prior!

- Now, we have a posterior, but what if we want a single number (an estimate)?

A: Most common answer is the **maximum a posteriori** (MAP), (because is it often computationally the easiest) **estimate**:

$$\hat{\theta}_{MAP} = \arg\max_{\theta} P(\theta|\mathcal{D}) = \arg\max_{\theta} \left(\log P(\mathcal{D}|\theta) + \log P(\theta) \right)$$

which for the Beta (α, β) prior, the **MAP estimate** is:

$$\hat{\theta}_{MAP} = \frac{n_H + \alpha - 1}{\underbrace{n_H + n_T}_{=n} + \alpha + \beta - 2} \quad \begin{matrix} n_H = k \\ n_H + n_T = n \end{matrix} \quad \frac{k + \alpha - 1}{n + \alpha + \beta - 2}$$

Note that as $n = n_H + n_T \to \infty$, the prior's effect (thru α, β) **vanishes** and we recover MLE; which is what we want: If the evidence of enough data, the prior should not matter.

$$\hat{\theta}_{MLE} = \frac{n_H}{n_H + n_T} = \frac{k}{n}$$

This vanishing of prior effect will be seen in many scenarios!

- **MAP estimate** is only the simplest Bayesian approach to parameter estimation

- it sets our model parameter to the **mode** of the posterior distribution $P(\theta|\mathcal{D})$. There is much more information in the posterior than is expressed by the mode; for instance, the posterior mean and variance of the parameter!
- why the mean? Suppose, want to predict the probability of the next flip coming up Heads; given that the seen data $\mathcal{D}$ is exactly the posterior mean:

$$P(X_{n+1} = H|\mathcal{D}) = \int_{\theta} P(X_{n+1} = H, \theta|\mathcal{D})\, d\theta = \int_{\theta} P(X_{n+1} = H|\theta) \cdot P(\theta|\mathcal{D})\, d\theta = \int_{\theta} \theta \cdot P(\theta|\mathcal{D})\, d\theta$$

→

or:

The probability that the next flip gives Heads, given the previous outcomes $\mathcal{D}$ ≡ The probability that the next flip gives Heads, and that θ is some value, summed (integrated) over all possible values of θ.

we then break up:

$$P(H, \theta|\mathcal{D}) = P(H|\theta, \mathcal{D}) \cdot P(\theta|\mathcal{D}) = P(H|\theta) \cdot P(\theta|\mathcal{D})$$

noting that: once you know θ, then $\mathcal{D}$ does not tell you anything more about the probability of Heads!

and finally that: $P(H|\theta) = \theta$ since θ is the probability of seeing Heads!

Errors (Abu-Mostafa Book) "In-Sample" "Out-of-Sample"

Note: The whole purpose of learning the unknown f is to be able to predict the value of f on points we have not seen before.

- In the context of picking marbles from a bin w/ Red and Green marbles.

SAMPLE

N independent marbles.

ν = fraction of red marbles (or k in other notes)

μ = probability of Red marbles = is unknown.

→ What does the fraction ν tell us about μ ??

- To quantify relationship between ν and μ, we use simple bound called the **Hoeffding inequality**; which states that for any sample size N:

$$(*) \quad P[|\nu - \mu| > \varepsilon] \leq 2e^{-2\varepsilon^2 N} \quad \text{for } (\forall)\, \varepsilon > 0$$

probability of an event, with respect to the random sample we pick.

$(\forall)$ positive value we choose.

→ It says: As the sample size N grows, it becomes exponentially unlikely that ν will deviate from μ by more than our "tolerance" ε.

(See [Abu-Mostafa, Ch. 1, pp. 19])

If we choose ε to be very small, in order to make ν a good approximation of μ, we need a larger sample size N to make RHS of $(*)$ small.

- **Relation of Bin model to Learning problem**: (Abu-Mostafa book, pp. 20-21)

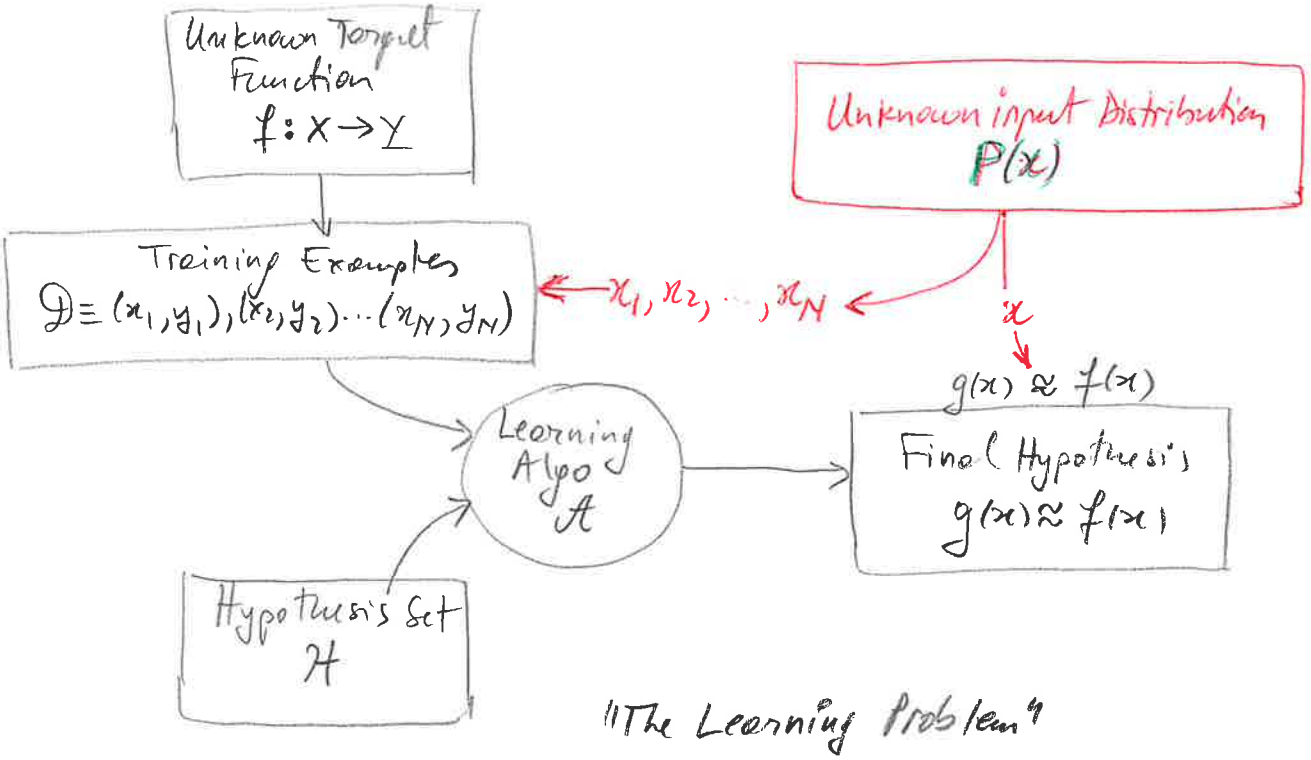
Unknown is μ — unknown is entire function $f: X \to Y$

Connection: - Take any hypothesis $h \in \mathcal{H}$ and compare to f on any point $x \in X$

- if $h(x) = f(x)$ color point x Green
- if $h(x) \neq f(x)$ color point x Red
- if we pick x at random according to some probability

distribution P over the input X we know that x will be red w/ some probability, call it μ, and green w/ probability $1-\mu$.

- Regardless of value of μ, the space X now behaves like the Bin with red and green marbles!
- Training examples play the role of a sample from Bin.
- if inputs $\mathcal{D} = \{x_1, x_2, \cdots, x_N\}$ are picked independently according to P, we will get a random sample of Red, $h(x) \neq f(x)$; and Green, $h(x) = f(x)$ points!
  - Red: w/ probability μ
  - Green: w/ probability $(1-\mu)$
- The color of each point is known to us because both $h(x_n)$, $f(x_n)$ are known for $n \in \{1, 2, \cdots, N\}$
  - $h(x_n)$: our hypothesis, known to us, so we can evaluate it
  - $f(x_n) = y_n$ is given to us for all points in dataset $\mathcal{D}$.
- Learning problem is now reduced to a Bin problem!
  - Under assumption inputs in $\mathcal{D}$ are picked independently according to some distribution P over X.
  - This probabilistic component is added to the figure below:



"The Learning Problem"

- With this equivalence, the Hoeffding Inequality can be applied to the learning problem; allowing us to make a prediction outside $\mathcal{D}$.
  - using ν to predict μ tells us something about f, although it does not tell us what f is.
  - what μ tells us is the error rate h makes in approximating f!
  - if ν happens to be close to zero, we can predict that h will approximate f well over the entire input space!
  - we have no control over ν, since ν is based on a particular hypothesis h.
  - in real learning, we explore an entire hypothesis set $\mathcal{H}$, looking for a particular $h \in \mathcal{H}$ that has a small error rate.
  - NOTE: if we have only one h to begin with, we are not really learning! but rather "verifying" whether that particular hypothesis h is good or bad!
  - To have learning, we need to have multiple hypotheses.

- Now, we can introduce definitions first:

Definitions

- **In-sample error**, the error rate within the sample (corresponds to ν in the Bin model) is:

$E_{in}(h) =$ (fraction of $\mathcal{D}$ where f and h disagree)

$$E_{in}(h) = \frac{1}{N}\sum_{n=1}^{N} [\![h(x_n) \neq f(x_n)]\!]$$

← makes explicit dependence of E_{in} on particular h that we are considering!

where $[\![\text{statement}]\!] = \begin{cases} 1, & \text{if statement} = \text{True} \\ 0, & \text{if statement} = \text{false} \end{cases}$

- **Out-of sample error** corresponds to μ in the Bin model.

$$E_{out}(h) = P[h(x) \neq f(x)]$$

↑ probability is based on the distribution P over X, which is used to sample the data points x.

- Now, consider an entire hypothesis set $\mathcal{H}$ w/ a finite number of hypotheses:
$$\mathcal{H} = \{h_1, h_2, \cdots, h_M\}$$
- we can construct a Bin equivalent by having M Bins, one for each $h_m \in \mathcal{H}$
- The Hoeffding inequality:
$$P[|E_{in}(h) - E_{out}(h)| > \varepsilon] \leq 2e^{-2\varepsilon^2 N}, \text{ for } (\forall)\ \varepsilon > 0$$
where hypothesis h is fixed before you generate dataset; and the probability P is w/ respect to random dataset $\mathcal{D}$.
- With multiple hypotheses in $\mathcal{H}$, the learning algorithm picks final hypothesis g based on $\mathcal{D}$; i.e., after generating the dataset. Hence, we want to make statement:

  "$P[|E_{in}(g) - E_{out}(g)| > \varepsilon]$ is small" for final hypothesis g.

  but because g is selected after, we cannot just plug it into Hoeffding inequality!
- See [Abu-Mostafa book, pp 23-27] for how to achieve that! It arrives at this:
$$P[|E_{in}(g) - E_{out}(g)| > \varepsilon] \leq 2 \cdot M \cdot e^{-2\varepsilon^2 N}$$
↑ bound is a factor of M looser; meaningful if M is finite.

- Now, we can talk about:

Feasibility of Learning is split into two questions:

(1) Can we make sure that $E_{out}(g)$ is close enough to $E_{in}(g)$?

(2) Can we make sure $E_{in}(g)$ is small enough?

If Learning is successful, then g should approximate f well, which means $E_{out}(g) \approx 0$!
Probabilistic analysis gives us $E_{out}(g) \approx E_{in}(g)$; so, we still need to make $E_{in}(g) \approx 0$ to conclude that $E_{out}(g) \approx 0$.

Note: See [Abu-Mostafa book] for details!