

# Bayesian Learning

## Supervised Learning

Some slides were adapted/taken from various sources, including Prof. Andrew Ng's Coursera Lectures, Stanford University, Prof. Kilian Q. Weinberger's lectures on Machine Learning, Cornell University, Prof. Sudeshna Sarkar's Lecture on Machine Learning, IIT Kharagpur, Prof. Bing Liu's lecture, University of Illinois at Chicago (UIC), CS231n: Convolutional Neural Networks for Visual Recognition lectures, Stanford University and many more. We thankfully acknowledge them. Students are requested to use this material for their study only and **NOT** to distribute it.

# Bayesian Learning

- For any supervised learning problem, given a set of examples (labeled data), we have to find a suitable hypothesis function  $h$  from a hypothesis class ( $\mathcal{H}$ ) such that

$$h(\vec{x}_1) = y_1, \text{ where } (\mathbf{x}_1, y_1) \subseteq \mathcal{R}^d \times \mathcal{C} \text{ and } h \in \mathcal{H}$$

- We have different possible competing hypothesis and we have to find out the probability of the individual hypothesis given the data, so that we can find out which is the most probable or most likely hypothesis.
- In Bayesian Learning, we will see how this can be computed using the Bayes theorem.

# Bayes Theorem

$$P(h | D) = \frac{P(D | h)P(h)}{P(D)}$$

- By the law of product:  $P(hD) = P(h) P(D|h)$
- (since commutative)  $P(Dh) = P(D)P(h|D)$
- Since  $P(hD) = P(Dh)$ , we get the **Bayes rule**.

# Application of Bayes Theorem

- A person takes a lab test for gluten allergy and the result comes positive. Given the result is positive(+), we have to calculate the probability that the person has the gluten allergy and the probability that the person does not have the gluten allergy.
- We have the following data (probabilities). Using Bayes theorem, we can calculate the above probabilities.
- The test returns a correct positive result in only 98% of the cases in which the allergy is actually present, and a correct negative result in only 97% of the cases in which the allergy is not present. Furthermore, 0.8% of the entire population have this allergy.

$$P(\text{allergic}) = 0.008$$

$$P(\neg \text{allergic}) = 0.992$$

$$P(+|\text{allergic}) = 0.98$$

$$P(-|\text{allergic}) = 0.02$$

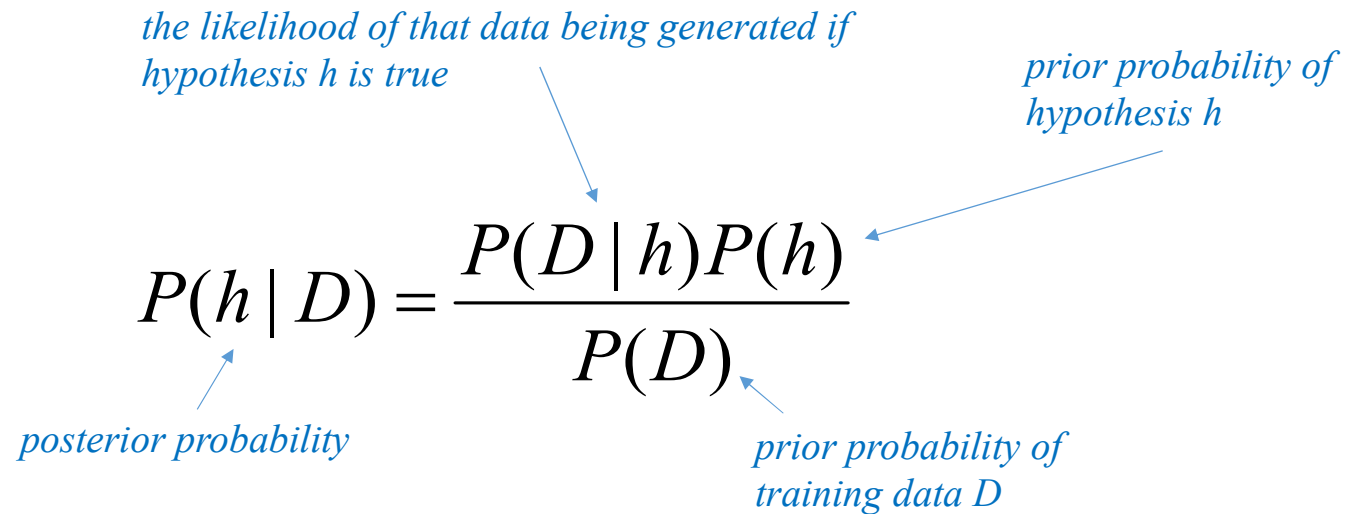
$$P(+|\neg \text{allergic}) = 0.03$$

$$P(-|\neg \text{allergic}) = 0.97$$

$$P(\text{allergic}|+) = \frac{P(+|\text{allergic})P(\text{allergic})}{P(D)} = \frac{0.98 * 0.008}{P(D)} = 0.00784 \text{ (probability that the person has the allergy)}$$

$$P(\neg \text{allergic}|+) = \frac{P(+|\neg \text{allergic})P(\neg \text{allergic})}{P(D)} = \frac{0.03 * 0.992}{P(D)} = 0.02976 \text{ (probability that the person has not the allergy)}$$

# Bayes Theorem



The diagram shows the Bayes Theorem equation with four blue arrows pointing to its components from descriptive text labels:

- An arrow from *the likelihood of that data being generated if hypothesis  $h$  is true* points to  $P(D | h)$ .
- An arrow from *prior probability of hypothesis  $h$*  points to  $P(h)$ .
- An arrow from *posterior probability* points to  $P(h | D)$ .
- An arrow from *prior probability of training data  $D$*  points to  $P(D)$ .

$$P(h | D) = \frac{P(D | h)P(h)}{P(D)}$$

- $P(h)$  = prior probability of hypothesis  $h$  or *class prior probability*
- $P(D)$  = prior probability of training data  $D$  or *Predictor prior probability*
- $P(h|D)$  = probability of  $h$  given  $D$ , it is called *posterior probability*
- $P(D|h)$  = probability of  $D$  given  $h$ , or *If the hypothesis is true, what is the likelihood of that data being generated*

# Bayesian Learning

- Can Bayes theorem be applied to find the suitable hypothesis?
- Maximum a Posteriori (MAP) is a probabilistic framework that finds the **most probable hypothesis** that describes the training dataset.

# Choosing Hypotheses

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

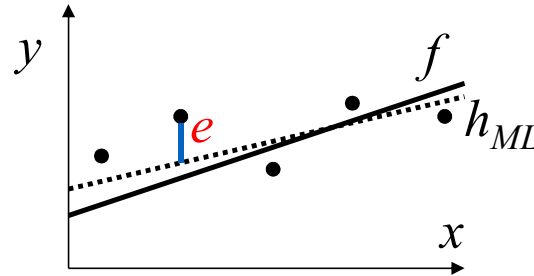
Generally want the most probable hypothesis given the training data *Maximum a posteriori* hypothesis  $h_{MAP}$ . In otherwords, the map hypothesis is given by that value of  $h$  for which probability  $h$  given data is maximized.

$$\begin{aligned} h_{MAP} &= \arg \max_{h \in H} P(h \mid D) \\ &= \arg \max_{h \in H} \frac{P(D \mid h)P(h)}{P(D)} \\ &= \arg \max_{h \in H} P(D \mid h)P(h) \end{aligned}$$

If we assume  $P(h_i) = P(h_j)$  then can further simplify, and choose the *Maximum likelihood* (ML) hypothesis

$$h_{ML} = \arg \max_{h_i \in H} P(D \mid h_i)$$

# Learning a Real Valued Function



Consider any real-valued target function  $f$

Training examples  $(x_i, d_i)$ , where  $d_i$  is noisy training value and generated by

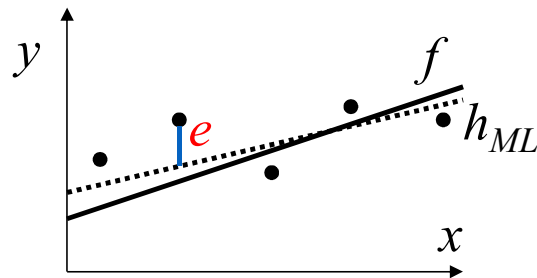
$$d_i = f(x_i) + \varepsilon_i$$

where  $\varepsilon_i$  (the noise drawn independently for each  $x_i$ ) follows Gaussian (normal) distribution with *mean* = 0 and *variance* =  $\sigma^2$  (e.g. standard deviation =  $\sigma$ )

So we can think that,  $d_i$  is coming from a normal distribution whose mean is  $f(x_i)$  and error (variance) is  $\sigma^2$ ) i.e.  $d_i \sim N(f(x_i), \sigma^2)$  [here  $\sigma^2$  is the variance of the error term  $\varepsilon_i$ ]

Now, as the noise drawn independently for individual instances  $(x_i)$





- We have to find  $h$  which estimate  $f$ . How to find it.
- We will use maximum likelihood hypothesis.

$$h_{ML} = \arg \max_{h \in \mathcal{H}} p(D|h)$$

# Learning a Real Valued Function

$$h_{ML} = \arg \max_{h \in H} p(D | h)$$

$$= \arg \max_{h \in H} \prod_{i=1}^m p(d_i | h)$$

$$= \arg \max_{h \in H} \prod_{i=1}^m \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma}\right)^2}$$

Since  $d_i$  is coming from a normal distribution

Constant

Maximize natural log of this instead ...

$$h_{ML} = \arg \max_{h \in H} \sum_{i=1}^m \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2} \left( \frac{d_i - h(x_i)}{\sigma} \right)^2$$

$$= \arg \max_{h \in H} -\frac{1}{2} \left( \frac{d_i - h(x_i)}{\sigma} \right)^2$$

$$= \arg \max_{h \in H} -(d_i - h(x_i))^2$$

$$= \arg \min_{h \in H} (d_i - h(x_i))^2$$

This is the least square criteria which we have used in linear regression. For this hypothesis function, the cost is the function which minimize the sum of square distance.

This is the Bayesian explanation for linear least square fitting.

# Bayes Optimal Classifier

- As of now, we have observed:
  - Bayes Theorem provides a principled way for calculating conditional probabilities, called **a posterior probability**.
  - Maximum a Posteriori (MAP) is a probabilistic framework that finds the **most probable hypothesis** that describes the training dataset.
- **Bayes Optimal Classifier** is a probabilistic model that finds the **most probable prediction** using the training data and space of hypotheses to make a prediction for a new data instance.

# Difference between MAP and Bayes Optimal Classifier

- MAP and MLE frameworks answer the following question:
  - *What is the most probable hypothesis given the training data?*
- Bayes optimal classifier answers
  - *What is the most probable classification of the new instance given the training data?*
- This is different from the MAP framework that seeks the most probable hypothesis (model). Instead, we are interested in making a specific prediction.
  - *In general, the most probable classification of the new instance is obtained by combining the predictions of all hypotheses, weighted by their posterior probabilities.*

## Example

- Let  $h_1$ ,  $h_2$  and  $h_3$  are three candidate hypothesis ( $h_1, h_2, h_3 \in \mathcal{H}$ )
- Let  $p(h_1|D) = 0.4$ ,  $p(h_2|D) = 0.3$  and  $p(h_3|D) = 0.3$
- So,  $h_1$  is MAP hypothesis because it has maximum posterior probability
- Now consider a new data  $x$ , so that  $h_1(x) = +$ ,  $h_2(x) = -$  and  $h_3(x) = -$
- Question is what is **most probable classification** of  $x$ ?
- The most probable classification of  $x$  will be **negative** as

$$p(h_2|D) + p(h_3|D) > p(h_1|D)$$

# Bayes Optimal Classifier

Bayes optimal classification

$$\arg \max_{v_j \in V} \sum_{h_i \in H} P(v_j | h_i) P(h_i | D)$$

V = set of all possible classes

Example:

$$P(h_1|D)=.4, P(-|h_1)=0, P(+|h_1)=1$$

$$P(h_2|D)=.3, P(-|h_2)=1, P(+|h_2)=0$$

$$P(h_3|D)=.3, P(-|h_3)=1, P(+|h_3)=0$$

therefore

$$\sum_{h_i \in H} P(+ | h_i) P(h_i | D) = .4$$

and

$$\sum_{h_i \in H} P(- | h_i) P(h_i | D) = .6$$

$$\arg \max_{v_j \in V} \sum_{h_i \in H} P(v_j | h_i) P(h_i | D) = -$$

# Gibbs Classifier

- Bayes optimal classifier provides best result, but can be expensive (intractable) if there are many hypotheses in the hypothesis class (e.g. hypothesis space is big).
- Some approximation is used: **Gibbs Sampling**: Sample hypothesis from hypothesis class ( $\mathcal{H}$ )

Gibbs algorithm:

1. Choose one hypothesis at random, according to  $P(h|D)$
  2. Use this to classify new instance
- A probability is associated with each of the hypothesis function, and there is a probability distribution over the hypothesis space.
  - Based on the our training data i.e. our evidence, we get a posterior probability distribution over the hypothesis space.
  - In the Bayes optimal classifier, each of the hypothesis is applied on the test instances [e.g.  $P(+|h_1)$ ] weighted according to their posterior probabilities [e.g.  $P(h_1|D)$ ].

# Gibbs Classifier

- Surprising fact: errors for Gibbs algorithm is quite **bounded**
- Assume target concepts are drawn at random from  $\mathcal{H}$  according to priors on  $\mathcal{H}$ , then:

$$E[\text{error}_{\text{Gibbs}}] \leq 2E[\text{error}_{\text{BayesOptimal}}]$$

- Suppose correct, uniform prior distribution over  $\mathcal{H}$ , then
- Pick any hypothesis from  $V$ , with uniform probability
- Its expected **error no worse than twice Bayes optimal**



to continue...