Bayesian Learning

Kaiqi Zhao
The University of Auckland

Slides are partially based on the materials from Mitchel’s book and Stanford’s NLP lectures

Maximum Likelihood and Least-Squared Error

Maximum Likelihood and Least-Squared Objective

- Problem: learning continuous-valued target functions (e.g. neural networks, linear regression, etc.)

- Problem setting:
  - Given a data set D containing m training examples of the form $< x_i , y_i >$
  - Let's say there exists an unknown function $f: X \rightarrow \mathbb{R}$ that describes how exactly the features maps to the target value.
  - $(\forall h \in H)[h: X \rightarrow \mathbb{R}]$, our goal is to find the best hypothesis h^* to approximate f
  - We assume the target value y_i is corrupted by random noise drawn from a Normal distribution with zero mean $y_i = f(x_i) + \epsilon$, $\epsilon \sim Normal(0, \sigma^2)$
    - This is equivalent to say y_i follows a Normal distribution with mean equals $f(x_i)$, i.e., $y_i \sim Normal(f(x_i), \sigma^2)$.

Maximum Likelihood and Least-Squared Objective

- Maximum likelihood for regression problem

$$h_{ML} = \text{argmax}_{h \in H}\, p(D|h)$$

$$= \text{argmax}_{h \in H} \prod_{i=1}^{m} p(x_i, y_i|h)$$

Assuming each instance is independent given h

- What is probability of $p(x_i, y_i|h)$?

$$p(x_i, y_i|h) = p(y_i|x_i, h)p(x_i|h) = \underbrace{p(y_i|h(x_i))}\underbrace{p(x_i|h)}$$

$$p(y_i|h(x_i)) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(y_i - h(x_i)\right)^2}{2\sigma^2}}$$

The value of x_i doesn't depend on h, so $p(x_i|h) = p(x_i)$

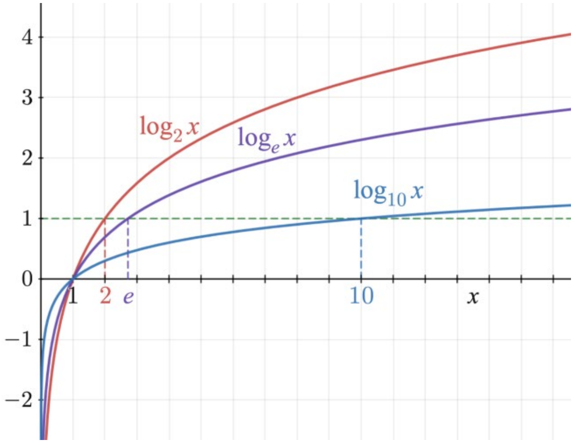
Maximum Likelihood and Least-Squared Objective

- Maximum likelihood for regression problem

$$h_{ML} = \text{argmax}_{h \in H} \prod_{i=1}^{m} p(y_i | h(x_i)) \prod_{i=1}^{m} p(x_i)$$

$$= \text{argmax}_{h \in H} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(y_i - h(x_i)\right)^2}{2\sigma^2}}$$

- How to find the best h from the above?
  - log(·) is a monotonically non-decreasing function, taking log of the likelihood does not affect the choose of the most probable hypothesis
  - We often compute log-likelihood instead of likelihood to make computation easier!



log function

Maximum Likelihood and Least-Squared Objective

- Maximum likelihood for regression problem

$$h_{ML} = \mathrm{argmax}_{h \in H} \prod_{i=1}^{m} p(y_i | h(x_i)) \prod_{i=1}^{m} p(x_i)$$

Assuming each instance is independent given *h*

$$= \mathrm{argmax}_{h \in H} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(y_i - h(x_i)\right)^2}{2\sigma^2}}$$

Substitute the normal distribution density function

$$= \mathrm{argmax}_{h \in H} \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(y_i - h(x_i)\right)^2}{2\sigma^2}}$$

Take log of the likelihood

$$= \mathrm{argmax}_{h \in H} \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(y_i - h(x_i)\right)^2}{2\sigma^2}}$$

Apply rules of log function

$$= \mathrm{argmax}_{h \in H} \sum_{i=1}^{m} -\frac{\left(y_i - h(x_i)\right)^2}{2\sigma^2} + \boxed{\log \frac{1}{\sqrt{2\pi\sigma^2}}}$$

This term is irrelevant to h

Maximum Likelihood and Least-Squared Objective

- Maximum likelihood for regression problem

$$h_{ML} = \text{argmax}_{h \in H} - \sum_{i=1}^{m} \frac{\left(y_i - h(x_i)\right)^2}{2\sigma^2}$$

- Maximizing the above equation is equivalent to minimizing the following

$$h_{ML} = \text{argmin}_{h \in H} \sum_{i=1}^{m} \frac{\left(y_i - h(x_i)\right)^2}{2\sigma^2}$$

⇒ the h_{ML} is one that minimizes the sum of the squared errors

Maximum Likelihood and Least-Squared Objective

$$h_{ML} = \text{argmin}_{h \in H} \sum_{i=1}^{m} \left(y_i - h(x_i)\right)^2$$

- Why is it reasonable to choose the Normal distribution to characterize noise?
  - Good approximation of many types of noise in physical systems
  - Central Limit Theorem shows that the sum of a sufficiently large number of independent, identically distributed random variables itself obeys a Normal distribution
- Only noise in the target value is considered, not in the attributes describing the instances themselves

Minimum Description Length

Minimum Description Length Principle

- Occam's razor: choose the shortest explanation for the observed data
- Here, we consider a Bayesian perspective on this issue and a closely related principle
- Minimum Description Length (MDL) Principle
  - Motivated by interpreting the definition of h_{MAP} in the light of information theory concepts

$$\begin{aligned} h_{MAP} &= \text{argmax}_{h \in H} P(D|h)P(h) \\ &= \text{argmax}_{h \in H} \log_2 P(D|h) + \log_2 P(h) \\ &= argmin_{h \in H} - \log_2 P(D|h) - \log_2 P(h) \end{aligned}$$

Minimum Description Length Principle

- Introduction to a basic result of information theory
  - Consider the problem of designing a code C to transmit messages drawn at random
  - Probability of encountering message i is p_i
  - Interested in the most compact code C
  - Shannon and Weaver (1949) showed that the optimal code assigns $-\log_2 p_i$ bits to encode message i
  - $L_C(i) \approx$ description length of message i with respect to C

Minimum Description Length Principle

$$h_{MAP} = \mathrm{argmin}_{h\in H} - \log_2 P(D|h) - \log_2 P(h)$$

- By information theory
  - $L_{C_H}(h) = -\log_2 P(h)$, where C_H is the optimal code for hypothesis space H
  - $L_{C_{D|h}}(D|h) = -\log_2 P(D|h)$, where $C_{D|h}$ is the optimal code for describing data D assuming that both the sender and receiver know hypothesis h

⇒ Minimum description length principle

$$h_{MAP} = \mathrm{argmin}_{h\in H} L_{C_H}(h) + L_{C_{D|h}}(D|h)$$

Minimum Description Length Principle

- To apply this principle in practice, specific encodings or representations appropriate for the given learning task must be chosen

- **Application to decision tree learning**
  - C_H might have some obvious encoding, in which the description length grows with **the number of nodes** and with **the number of edges**
  - Choice of $C_{D|h}$?
    - For simplicity, assume both sender and receiver know the m examples $< x_1, \ldots, x_m >$
    - Under this assumption, what message do we need to transmit?
      - If the hypothesis can correctly predict the class of an example, nothing is needed to transmit
      - Otherwise, the example and the correct class label needs to transmit to the sender

$$(\ \log_2 m \text{ bits} \quad + \quad \log_2 k \text{ bits}\ ) \times \ \#\text{missclassification}$$

Minimum Description Length Principle

- C_H - the number of nodes and the number of edges Model complexity
- $C_{D|h}$ - $(\log_2 m \text{ bits} + \log_2 k \text{ bits}) \times \#\text{missclassification}$ Errors created

- MDL principle provides a way for trading off hypothesis complexity for the number of errors committed by the hypothesis
  - The shorter C_H for hypothesis, the more likely we make mistakes, and hence $C_{D|h}$ might be longer

- One way of dealing with the issue of overfitting

Minimum Description Length Principle

- C_H - the number of nodes and the number of edges Model complexity
- $C_{D|h}$ - $(\log_2 m \text{ bits} + \log_2 k \text{ bits}) \times \text{\#missclassification}$ Errors created

- MDL principle provides a way for trading off hypothesis complexity for the number of errors committed by the hypothesis
  - The shorter C_H for hypothesis, the more likely we make mistakes, and hence $C_{D|h}$ might be longer

- One way of dealing with the issue of overfitting

Example of MDL – Decision Tree Pruning

Table 1 – An example classification dataset

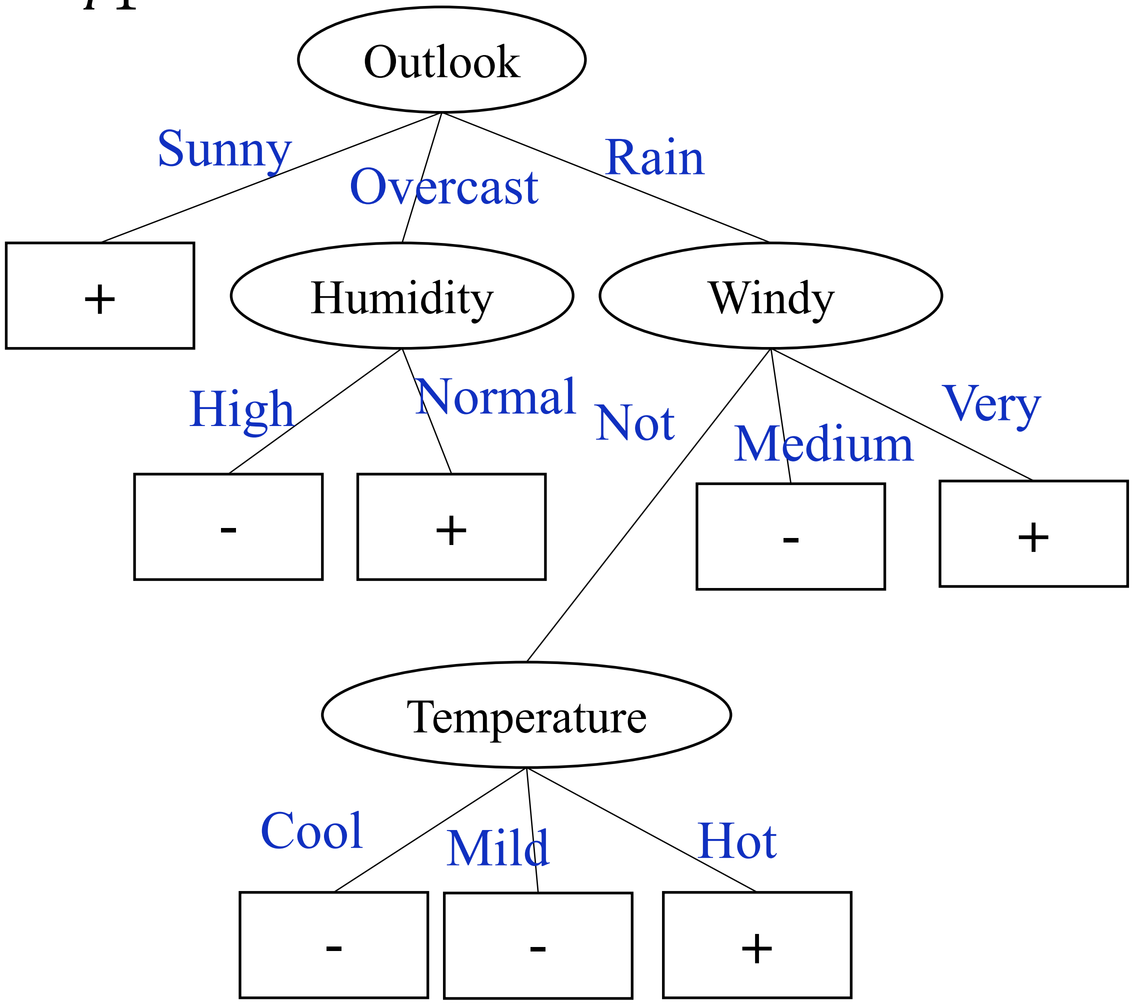
Outlook	Temperature	Humidity	Windy	Class
Overcast	Hot	High	Not	-
Sunny	Mild	Normal	Very	+
…	…	…	…	…
Rain	Hot	High	Medium	-
…	…	…	…	…

We have 32 instances, and the decision tree on the right hand side achieves 100% accuracy

Let's say we encode the tree with each row denoting a split. We can use 2 bits to encode the attribute and 1 bit to record a leaf node, e.g.

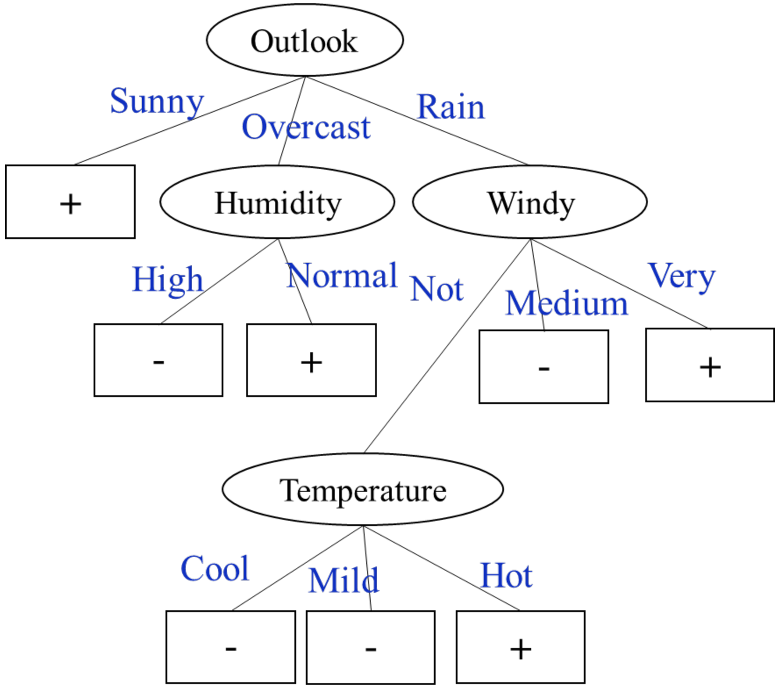
- Outlook: +, Humidity, Windy
- Humidity: -, +
- …

$T1$



Example of MDL – Decision Tree Pruning

$T1$

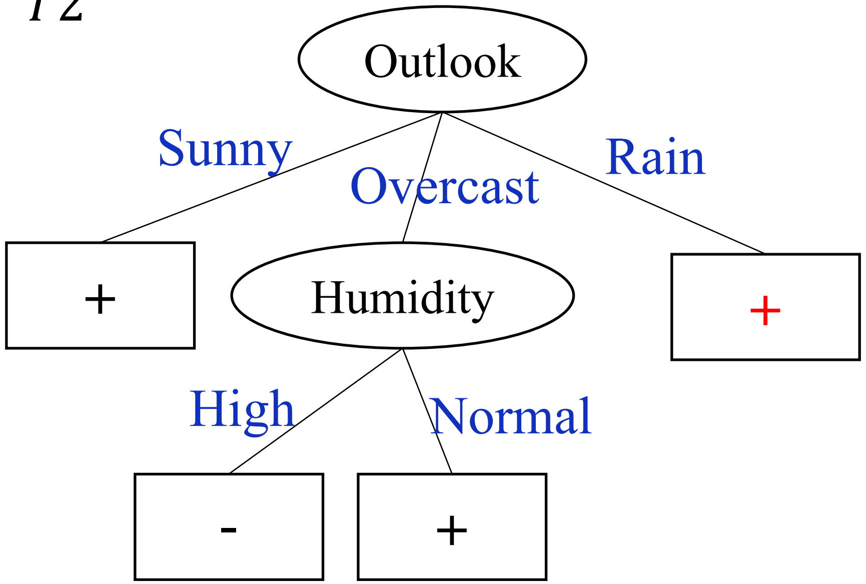


$$L_{C_H}(h) = \#\text{leaf} + 2\#\text{non-leaf} = 8 + 6 = 14$$

$$L_{C_{D|h}}(D|h) = 0$$

$$L_{C_H}(h) + L_{C_{D|h}}(D|h) = 14 \text{ bits}$$

$T2$



Let $T2$ misclassifies one of the instances in Table 1 (in red)

$$L_{C_H}(h) = \#\text{leaf} + 2\#\text{non-leaf} = 4 + 2 = 6$$

$$L_{C_{D|h}}(D|h) = \log_2 32 + \log_2 2 = 6$$

$$L_{C_H}(h) + L_{C_{D|h}}(D|h) = 12 \text{ bits}$$