

# AI\_12\_20

## Supervised learning

---

- Learning problem
  - 給定一個模型的假設空間  $H$
  - 找到假設  $h$ ，可以使得獲得的結果約等於真實的答案
  - 這樣就可以找到真正的函數，在預測的函數為基底

## Consistency vs. Simplicity 一致性 vs. 簡單性

---

Consistency :  $h(x_i) \cong y_i$

Simplicity

## Measuring Consistency using Loss

---

得到一個假設，做一個好的預測，且他們有一致性，可以有對應的答案

- **Measure mistakes**: loss function  $L(x, y)$ 
  - + Absolute-value loss
  - + Squared-error loss
  - + 0/1 loss
  - + Log loss and other....

Empirical loss:

$$EmpLoss_{L,E}(h) = \frac{1}{|E|} \sum_{(x,y) \in E} L(y, h(x))$$

## The Bayes Classifier

---

給定一個sample，希望能預測到的一個答案，事後機率

找出一個假設公式是最大化的機率

$$h(x)^* = \operatorname{argmax} P(Y = y | X = x) = \operatorname{argmax} \frac{P(x|y)P(y)}{P(x)} = \operatorname{argmax} P(x|y)P(y)$$

## Simplic

---

- **Ease of use**
  - 更簡單的假設，則需要存儲的參數更少
  - 更簡單的假設更容易估計

- **Generalization 泛化:**

可以普遍的猜出位在訓練資料裡面的舉例，而若是跟訓練資料裡的資料不一樣卻能猜得出來，代表泛化的很好。但若是每個都能猜出一個答案，連離群值都放進去了，將模型做得太複雜，有可能會是 overfit ( 過度擬合 )，不是真正要的結果

- **How to achieve simplicity?**
  - 限制參數空間，可以以防過度擬合
  - 在資料上面萃取較少的特徵
  - 正則化 ( 對複雜性進行懲罰 )

$$h^* = \operatorname{argmin} [EmpLoss_{L,E}(h) + \lambda Complexity(h)]$$

"+" 後面的就是懲罰項

前面雖然越小越好，但後面複雜參數會慢慢提高，所以會選擇相對相加較少的

# Model Selection : Bias vs. Variance 均差 vs. 方差

---

越簡單 <-> 越複雜  
H <-> L Bias  
L <-> H Variance

## Feature Engineering

---

特徵的選擇 ( 重要 )

## Training a Model

---

- Hyperparameters 超參數:  
很多學習的演算法有很多選擇，基於很多條件下，在訓練前可以調整泛化程度、學習深度、最大的選擇、特徵的選擇、懲罰制度、學習率....，  
需要調整參數去影響結果

## Hyperparameters Tuning/Model Selection

---

將自己的檔案分割，訓練資料切成訓練資料與測試資料，改變一些參數調整，然後選出一個最好的選擇，然後再拿去沒看過的資料集做測試