Lec 2.1 Machine Learning Example

1. Example of ML:

eg1: Recommdation system (youtube, netflex ..)

⇒ predict how a user will rate a movie not seen

↳ Approach 1: (No ML)

- Have an expert view movie

  ⇒ Attribute: comedy / action? actors? blockbuster?

- Interview each user
- Compute a matching score
- Problems: tedious, user preference are subjective, change overtime

↳ Approach 2: learning from data - use past ratings.

- def: r_{ij} : rating of movie j by user i

  S : set of rated user - movie pairs

  $S = \{(i,j) \mid \text{movie } j \text{ is rated by user } i\}$

- Assume each user associate with a perference vector:

  $\underline{a}_i = (a_{i1}, a_{i2}, \cdots a_{im})$

  Assume a movie attribute vector

  $\underline{b}_j = (b_{j1}, b_{j2} \cdots b_{j3})$

  both unknown

- Assume certain model of matching:

  $r_{ij} = \underline{a}_i^T \underline{b}_j$

- Train our machine: Given S, find a_i b_j.

$$\min_{\{a_i\}\{b_j\}} \sum_{(i,j)\in S} \underbrace{(r_{ij} - a_i^T b_j)^2}_{\text{error}}$$

- Prediction: for $(i, j) \notin S$, (i.e. Not in the data set)

$$\hat{r}_{ij} = a_i^T b_j$$

($\hat{r}_{ij}$: estimate)

↳ Advantage:

- Data driven
- Adapt to user preference / movie attributes

2. Example 2: Credit Approval

Determine whether to approve credit to a new customer. (Y/N)

↳ Input: $\underline{x}_n$ = (age, salary, years at job, debt)

Output: $y_n = \begin{cases} +1 : \text{approve} \\ -1 : \text{not} \end{cases}$

Given historical data (know the right answer)

$\{ (\underline{x}_1, y_1), (\underline{x}_2, y_2) \cdots (\underline{x}_n, y_n) \}$

($\underline{x}_1$: first customer; y_1: first decision)

Determine whether to approve to a new customer

↳ Model:

$\underline{w} = (w_1, w_2, \cdots w_d)$ // weight vectors

$\underline{x} = (x_1, x_2, \cdots x_d)$ (x_1: age, x_2: salary, x_d: debt)

Model: $\sum_{i=1}^{d} w_i x_i \underset{y=-1}{\overset{y=+1}{\gtrless}} \text{threshold}$ ⊗ // compare $\sum w_i x_i$ with a threshold.

But we don't know $\underline{w}$ & thres ⇒ Training

↳. **Training:**

Given historical data, find $(\underline{w}, \text{threshold})$ to minimize $\mathcal{E}$, where

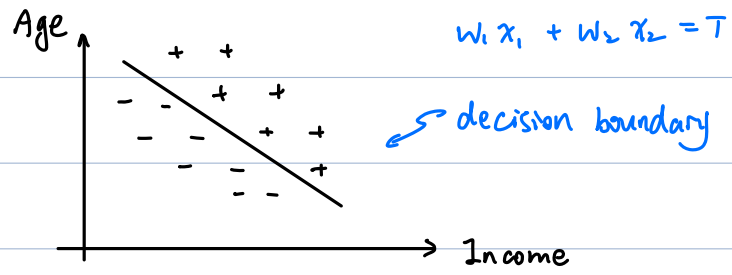
$$\mathcal{E} = \sum_{n=1}^{N} 1\{ y_n \neq \hat{y}_n(\underline{w}, \text{threshold}) \}$$

- indicator function (if the statement is true ⇒ returns 1)
  - i.e. count the times you made a mistake
- y_n: true value
- $\hat{y}_n(\underline{w}, \text{threshold})$: model prediction for $\underline{x}$

↳. **Prediction:**

Given new customer $\underline{x}$, use ⊛ to predict $\hat{y}$

e.g.



Lec 2.2 Supervised Learning

1. Formal setup:

↳ **Input:** Data point $\underline{x} = (x_1, x_2, \cdots x_d) \in \mathbb{R}^d$

↑ One data point. May have more

e.g. 1 (user i, movie j), $d=2$

e.g. 2 customer attributes $\underline{x}$ $d=4$

↳ **Output:** Label $y \in \mathbb{R}$

Depending on y, we have two types:

- Classification: discrete values

  e.g #2: $y \in \{+1, -1\}$

- Regression: continuous values

  e.g. #1: $y = r_{ij}$

↳ **Unknown:**

Target function $f: \mathbb{R}^d \to \mathbb{R}$

$$y = f(\underline{x})$$

↳ **Task:**

given training data $\mathcal{D} = \{(\underline{x}_1, y_1), (\underline{x}_2, y_2), \cdots (\underline{x}_n, y_n)\}$ (data points / examples)

produce a function $g: \mathbb{R}^d \to \mathbb{R}$ s.t. $y = g(\underline{x})$

to make prediction on new inputs

↳ **Learning Model:**

- Hypothesis set: $\mathcal{H} = \{h_1, h_2, \cdots h_n\}$

  each are a function that maps $\mathbb{R}^d$ to $\mathbb{R}$. $h_i: \mathbb{R}^d \to \mathbb{R}$

each being a candidate function $y = h_i(\underline{x})$

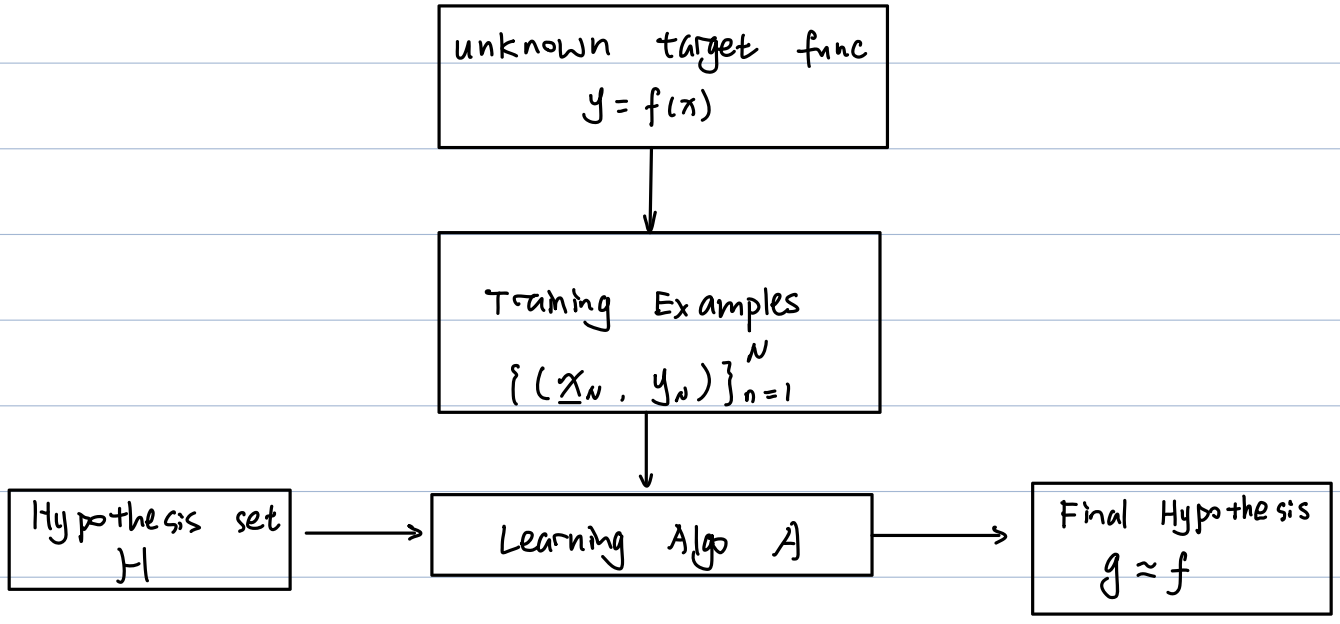
e.g. # 1. $(i, j) \xrightarrow{\underline{a}_i^T \underline{b}_j} \gamma_{ij}$

e.g # 2. $\underline{x} \xrightarrow{\text{sign}(\underline{w}^T \underline{x} - \text{threshold})} \pm 1$

↳. Learning function:

select a $g \in \mathcal{H}$ using the training set

↳. Producure graph:



prediction / testing:

$y = g(\underline{x})$