Section B

Monday, January 23, 2023 9:00 AM

ML as Function Approximation

Problem Setting

- Set of possible inputs, $\mathcal{X}$ (all possible feature vectors)
- Set of possible outputs, $\mathcal{Y}$ (all possible labels)
- Unknown target function, $c^*: \mathcal{X} \rightarrow \mathcal{Y}$
- Set of candidate hypotheses

$$\mathcal{H} = \{h \mid h: \mathcal{X} \rightarrow \mathcal{Y}\}$$

Aside: Functions Types

$$f(x_1, x_2, x_3) = (x_1 x_2)^2 + x_3 + 7$$

$$f: \mathbb{R}^3 \rightarrow \mathbb{R}$$

Learner is given:

- Training examples $D_{train} = \{(\vec{x}^{(1)}, y^{(1)}), (\vec{x}^{(2)}, y^{(2)}), \ldots, (\vec{x}^{(N)}, y^{(N)})\}$ of unknown target function $y^{(i)} = c^*(\vec{x}^{(i)})$
- N = # of training examples, M = # of features $= |\vec{x}^{(i)}|$

Learner produces:

- Hypothesis $h \in \mathcal{H}$ that "best" approximates c^*

training

To Evaluate:

- Loss function $\ell: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ measures how "bad" predictions $\hat{y} = h(\vec{x})$ are compared to $c^*(\vec{x})$
- Another dataset $D_{test} = \{(\vec{x}^{(1)}, y^{(1)}), \ldots, (\vec{x}^{(N')}, y^{(N')})\}$
- Evaluate the average loss of $h(x)$ on D_{test}

testing

Algorithms for Classification

Alg 1. Majority Vote Algorithm

```
def train(D):
    store v = majority_vote(D)
            = the class y∈𝒴 that appears most often in D
def h(x⃗):
    return v
def predict(D_test)
    for (x⃗^(i), y^(i)) ∈ D_test:
        ŷ^(i) = h(x⃗^(i))
```

reuse this for any classifier today

for $(\vec{x}^{(i)}, y^{(i)}) \in D_{test}$:
    $\hat{y}^{(i)} = h(\vec{x}^{(i)})$

any classifier today

$\boxed{\text{Alg. 2}}$ Memorizer

```
def train(D):
    store dataset D

def h(x⃗):
    if ∃ x⃗⁽ⁱ⁾ ∈ D s.t. x⃗⁽ⁱ⁾ = x⃗ :
        return y⁽ⁱ⁾
    else
        return y ∈ 𝒴 randomly
```

$\boxed{\text{Alg. 3}}$ Decision Stump

def train(D):

1. pick an attribute, m
2. divide dataset D on x_m

$$D^{(0)} = \{ (\vec{x}^{(i)}, y^{(i)}) \in D \mid x_m = 0 \}$$

$$D^{(1)} = \{ (\vec{x}^{(i)}, y^{(i)}) \in D \mid x_m = 1 \}$$

3. two votes

$$v^{(0)} = \text{majority\_vote}(D^{(0)})$$

$$v^{(1)} = \text{majority\_vote}(D^{(1)})$$

def $h(\vec{x})$:

if $x_m = 0$: return $v^{(0)}$

if $x_m = 1$: return $v^{(1)}$

$\boxed{\text{Alg. 4}}$ Decision Tree