Math/Comp 562 Lecture 1

2022/01/10

Lecture 1.

Formal Model. Batch Supervised Machine Learning

$\mathcal{X}$ Domain $x \in \mathcal{X}$

$\mathcal{Y}$ Labels $y \in \mathcal{Y}$ $\mathcal{Y} = \{0,1\}$ or $\mathcal{Y} = \{-1,1\}$ Class

$\mathcal{Y} = \mathbb{R}$ regression

S^n training data

$S = S^n = \{(x_1, y_1), \dots, (x_n, y_n)\}$

$f : \mathcal{X} \to \mathcal{Y}$ hypothesis / predictor / prediction rule / classifier

P data generation model / statistical data distribution (unknown) / arbitrary

Noisy labels

$P = P(x,y)$ so on x, $P_y(y) = P(y|x)$

Det. labels

$y = f^*(x)$ and $P = P_x(x)$.

Defn Losses $\ell(y,z): Y \times Y \to \mathbb{R}^+$

ℓ is a loss if $\ell(y,z) \geq 0$

with $\ell(y,z)=0$ iff $y=z$

Ex Class:

Given $y, y' \in Y$

$$\ell_{0-1}(y,y') = \begin{cases} 0 & \text{if } y=y' \\ 1 & \text{otherwise} \end{cases}$$

Ex Regression

$$\ell_2(y,z) = (y-z)^2$$

Def Case: Global / generalization loss/error (risk)

of function $f: X \to Y$

$$R(f) = R_{\ell,\rho}(f) = \mathbb{E}[\ell(f(x), f^*(x)]$$

Ex $R(f) = \mathbb{E}\, \ell_{0-1}(f(x), f^*(x)) = \mathbb{P}(f(x) \neq f^*(x))$

noisy label case

Defn 2.1 Expected Risk / loss.

Given $\ell : Y \times Y \to \mathbb{R}^+$ loss fn

p prob dist on $X \times Y$

$f : X \to Y$

the expected risk/loss of f is

$$R(f) = R_{p,\ell}(f) = \mathbb{E}[\ell(y, f(x))]$$

$$= \int_{X \times Y} \ell(y, f(x)) \, dp(x, y)$$

Defn 2.2 Empirical Risk

Given $\ell : Y \times Y \to \mathbb{R}^+$ loss fn

data $S = S^n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$

$f : X \to Y$

$$\hat{R}(f) = \hat{R}_{S^n,\ell}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(x_i))$$

Random RHS

when f is nonrandom

$R(f)$ is number

when S^n = random iid

then $f = A(S^n)$ is a random funct.

$\Rightarrow R(f)$ random variable

and

$\hat{R}(f)$ r.v.

Dealing with noisy labels:

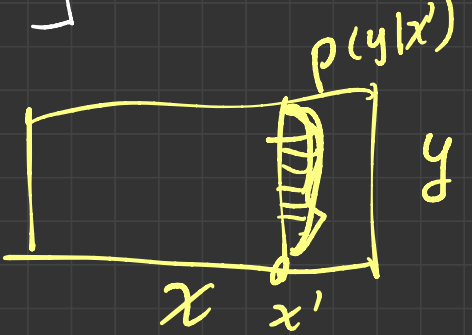
Given x' $\quad y \sim p(z|x')$ $\quad p_{x,y}(x,y)$

Define conditional risk for any $z \in \mathcal{Y}$

$$r(z|x') = \mathbb{E}[\ell(y,z)\,|\,x = x'] = \int \ell(y,z)\, dp(y|x')$$

Ex $\quad x'$: $y=+1$ $p=0.8$, $y=-1$ $p=0.2$ $\quad r(1|x') = 0.2$, $r(-1|x') = 0.8$ $\quad \ell = \ell_{0-1}$

$$\mathcal{R}(f) = \mathop{\mathbb{E}}_{x,y \sim p_{x,y}}[\ell(y, f(x))] = \mathop{\mathbb{E}}_{x' \sim p_x}[r(f(x')|x')]$$



Defn $f^*: \mathcal{X} \to \mathcal{Y}$ is a Bayes predictor / fn

if $f^*(x') \in \underset{z \in \mathcal{Y}}{\operatorname{argmin}} \mathbb{E}[\ell(y,z) \mid x = x']$

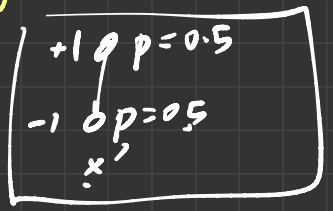
$= \underset{z \in \mathcal{Y}}{\operatorname{argmin}} \; r(z|x')$

prop 2.1

All Bayes predictors have risk equal to

$$R^* = \mathop{\mathbb{E}}_{x' \sim p_x} \left[\inf_{z \in \mathcal{Y}} r(z|x') \right]$$

Note: can have more than one $f^*(x')$ = +1 or −1



Defn Excess Risk

The excess risk of f is $R(f) - R^*$ (always ≥ 0).

Cases

Binary Class: $f^*(x') = \underset{z \in \{0,1\}}{\arg\max} \underbrace{P(y=z \mid x=x')}_{\eta(x)}$

$$R^* = \mathbb{E}[\min(\eta(x), 1-\eta(x))]$$

Regression: $\mathcal{Y} = \mathbb{R} \quad \ell(y,z) = (y-z)^2$

$$f^*(x') = \mathbb{E}[y \mid x = x']$$

2.3.2 Empirical Risk Minimization (ERM)

choose f to minimize $\hat{R}(f)$ empirical Risk/Loss

even though really care about $\hat{R}(f)$ generalization Loss / Exp-Risk

Defn Hyp Class

$$\mathcal{H} = \{ f_\theta : \mathcal{X} \to \mathcal{Y} \mid \theta \in \Theta \}$$

Ex Feature $\varphi : \mathcal{X} \to \mathbb{R}^d$ $\varphi(x) = (\varphi_1(x), \dots, \varphi_d(x))^T$

$f_\theta(x) = \theta^T \varphi(x)$, $\theta \in \mathbb{R}^d = \Theta$

Do: $\min_{f \in \mathcal{H}} \hat{R}(f) \iff \min_{\theta \in \Theta} \hat{R}(f_\theta)$

functional opt — parametric opt

Pros
- can optimize, using GD / SGD
- can be applied in high dim

Cons
- harder opt in non-convex case (NN)

⟹ need a good feature vector
- by experts
- learned by nn (no theory)

- need to control overfitting
- in classification case, need to deal with optimization on real values
→ round to class. (solved using score based losses)

2.3.1 Compare to "Local Averaging" methods

instead of learning f_θ, interpolate (average) directly from data.

EX KNN, decision trees,

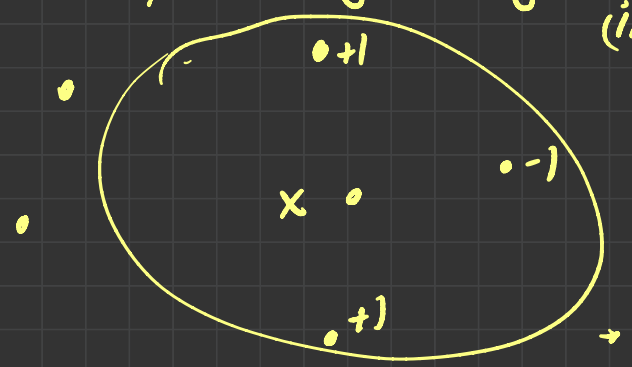
K-nearest neighbors:

Given $S^n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$

$X = \mathbb{R}^d \quad Y = \{0, 1\}$

Define: $f(x)$ by: (i) find k nearest nbrs of x

(ii) $f(x)$ = majority vote of labels of nbrs.



$f(x) = \text{maj}\{+1, +1, -1\}$

$= +1$

pros:
- no opt. / train
- easy to implement
- works well (often) in low dim

Cons:
- slow at query time: need to search
- bad in high dimensions because of curse of dim.
- the distance fn is crucial
- need to tune K hyperparam. can underfit / overfit with wrong K

Risk decomposition

Defn Given: • ℓ
• $\mathcal{H} = \{f_\theta \mid \theta \in \Theta\}$
• $S^n = \{(x_i, y_i)\}_{i=1}^n$

Way to get rid of randomness coming from $f_{\hat{\theta}}$. Still random.

Defn Approximation error: $E_A(\mathcal{H}) = \inf_{\theta \in \Theta} \{R(f_\theta) - R^*\}$ ← deterministic

how well can approx f^* using $\mathcal{H}$.

Estimation Error $E_E(f_{\hat{\theta}}, \mathcal{H}) = R(f_{\hat{\theta}}) - \inf_{\theta \in \Theta} R(f_\theta)$ ← random

Then Excess Risk $= ER(f_{\hat{\theta}}) - R(f_{\hat{\theta}}) - R^*$

$= E_E + E_A$

Uniform Deviation : $U(\mathcal{H}) = \sup_{\theta \in \Theta} |R(f_\theta) - \hat{R}(f_\theta)|$ random by S^n

by defn: $R(f_{\hat{\theta}}) - \hat{R}(f_{\hat{\theta}}) \leq U(\mathcal{H})$

$$E_\varepsilon(f_{\hat{\theta}}, \mathcal{H}) = R(f_{\hat{\theta}}) - \inf_{\theta \in \Theta} R(f_\theta)$$

$$= \{R(f_{\hat{\theta}}) - \hat{R}(f_{\hat{\theta}})\} + \underbrace{\{\hat{R}(f_{\hat{\theta}}) - \hat{R}(f_{\theta^*})\}}_{E_{opt}(f_{\hat{\theta}}, \mathcal{H})} + \{\hat{R}(f_{\theta^*}) - R(f_{\theta^*})\}$$

$$\leq 2\,U(\mathcal{H}) + E_{opt}$$

Thus

$$\text{Excess Risk} = E_A + E_\varepsilon$$

$$\leq E_A + E_{opt} + 2\,U(\mathcal{H})$$

(E_A: fit, E_{opt}: opt, $2\,U(\mathcal{H})$: capacity to overfit)

Diff NN/ML:

NN: control fit & opt, not cap. ML: cap opt not E_A.