<u>Book</u>: • Hastie, Tibshirani + Friedman

• Material from Jordan's course notes

• Papers, etc. + Semisupervised
+ Bishop

<u>Themes</u>: • Classification

• Regression

• Inference

<u>Regression</u>: • predict some value
y from vars x

• Choose some loss function

$$L(y, f(x))$$

and obtain f by minimizing

$$E[L(y, f(x))]$$

with respect to
$P(x, y)$

Important case: • Square error loss

$$L(Y, f(x)) = (y - f(x))^2$$

hence

$$\min E\left((y - f(x))^2\right)$$ EPE = expected prediction error

$$= E_X\left[E_{Y|X}\left[(y - f(x))^2\right]\right]$$

and so we can minimize

$$E_{Y|X}\left[(y - f(x))^2\right]$$

$$= \int (y - f(x))^2 P(y|x)\, dy$$

at each x.

but the solution is

$$f(x) = E(Y \mid X = x)$$

$$= \int y\, P(Y \mid X = x)\, dy.$$

Example cases: **linear regression**

$$f(x) = \underline{x}^T \underline{\beta}$$

we usually ensure that one column of x is a 1 to simplify notation

$$EPE = E\left[\left(Y - X^T \beta\right)^2\right]$$

$$= E[y^2] - 2\beta' E[XY] + \beta' E[XX']\beta$$

so

$$\beta = E[XX']^{-1} E[XY]$$

~~alternativ~~
what we're accustomed to is

$$Y = \begin{bmatrix} y_1 \\ \vdots \end{bmatrix} ; \quad X = \begin{bmatrix} x_1' \\ \vdots \\ \vdots \end{bmatrix}$$

$$\min \left\| Y - X\beta \right\|^2 \qquad \beta = (X'X)^{-1}(X'Y)$$

$\longrightarrow$ this replaces $E_{X,Y}$ with $\sum_{data}$

k - nearest neighbours

$$f(x) = \frac{1}{K} \sum_{\substack{i \in K \text{ examples} \\ \text{closest to } x}} y_i$$

now this approximates $E(Y \mid X = x)$ by assuming that x changes "slowly" and then replacing an expect with a sum.

Notice:

for most $P(X, Y)$ as $N, k \to \infty$ and $\frac{k}{N} \to 0$

$$f(x) \to E(Y / X = x).$$

How does this relate to classification?

- regression where Y is a categorical var
- for now, $Y \in \{0, 1\}$

Zero-One loss

$$L(Y, f(x)) = \begin{cases} 1 & \text{don't agree} \\ 0 & \text{otherwise} \end{cases}$$

$$EPE = E_X\left[E_{Y|X}[L(Y, f(x))]\right]$$

$$= E_X\left[L(1, f(x)) P(1|X=x) + L(0, f(x)) P(0|X=x) \right]$$

→ we can minimize pointwise.

by choosing the ~~class~~ $f(x)$ such that

$\cancel{L(class}$, $L(1, f(x)) P(1|X=x) + L(0, f) P(0|X=x)$

is min.

for 0-1 loss, we get

$$\begin{cases} 1 & \text{if } P(1/x) < P(0/x) \\ 0 & \phantom{\text{if } P(1/x)} > \\ \text{(doesn't matter)} & \phantom{\text{if } P(1/x)} = \end{cases}$$

We could do this with K-NN easily

- but there are problems in high dimensions

The curse of Dimension

1) Where is volume in HD?
   - on the <u>skin</u>

2) What % of data is nearby?
   - very little

e.g $\longrightarrow$

unit cube, p dim, uniform data.

- we want fraction r of data.

- this is r of volume
$\Rightarrow r^{1/p}$ of edge length

10 dim, to capture 10% of data

we need edge length of 0.8

- hardly local

Bias and Variance:

- consider $y = f(x) = e^{-\|x\|^2}$

(deterministic fn)

- we will draw a bunch of (x,y) samples T and approx fn using 1-NN

- All error is due to choice of samples.

- consider $\hat{y}_0$ = 1-NN est of $y(0)$

and we can write

$$MSE(x_0) = E_T\left[\left(f(x_0) - \hat{y}_0\right)^2\right]$$

$$= f^2(x_0) - 2E_T[y_0]\,f(x_0) + E_T[y_0^2]$$

$$= f^2(x_0) - 2E_T[y_0]f(x_0) + \left(E_T[y_0]\right)^2$$
$$+ \left(E_T[y_0]\right)^2 - 2\left(E_T[y_0]\right)^2 + E_T[y_0^2]$$

$$= \left(f(x_0) - E_T[y_0]\right)^2 \longleftarrow \text{BIAS}^2$$

$$+ E_T\left[\left(y_0 - E_T[y_0]\right)^2\right] \longleftarrow \text{VARIANCE}$$

- this sort of decomposition is universal.
- In our example, as dimension goes up, $E_T[y_0]$ goes down fast (why?) so bias goes up fast

~~linear m~~ Stabilizing linear regression

- wish to reduce prediction error
  - by ditching variables
  - disadvantages:
    - search
    - variance could go up.

- penalizing large coeffs (in the hope they shrink to zero)

$$\text{ridge}\ (\beta) = \left\{ \| Y - \beta_0 1 - X\beta \|^2 + \lambda \beta'\beta \right\}$$

- we dont want to penalize the constant because that would result in fits that aren't covariant under translation in y — doesnt make sense.

- for this discussion, center y s

$$\left(\text{ie. } \sum_i y_i = 0 \right) \Rightarrow \beta_0 = 0$$

then

$$(Y - X^{\bullet}\beta)'(Y - X^{\bullet}\beta) + \lambda \beta'\beta$$

or

$$\beta^{ridge} = \underbrace{\left(X^{T\bullet}X^{\bullet} + \lambda I\right)^{-1}}_{\text{deals with possible rank problems.}} X^{T}Y.$$

<u>Notice that</u>

$$\min \; (Y - X^{\bullet}\beta)'(Y - X^{\bullet}\beta) + \lambda \beta'\beta$$

$$\lambda \geq 0$$

$$\equiv$$

$$\min \; (Y - X^{\bullet}\beta)^{T}(Y - X^{\bullet}\beta)$$

$$\text{st} \quad \beta'\beta \leq s$$

Notice that:

- ridge regression isn't covar under scaling of inputs (why?)

- qualitative arg:

  - β can be poorly determined in presence of correlated vars

Ridge regression + SVD

write $X = U D V^T$

diagonal

U: $N \times p$

V: $p \times p$

orthogonal

$$\beta^{LS} = (X^T X)^{-1} X^T y$$

so
$$X\beta^{LS} = X(X^T X)^{-1} X^T y$$
$$= U \underbrace{U^T y}$$

these are coords of y in U basis.

$$X\beta^{ridge} = X(X^TX+\lambda I)^{-1}X^Ty$$

$$= UD(D^2+\lambda I)^{-1}DU^Ty$$

$$= \sum_j u_j \left[\frac{d_j^2}{d_j^2+\lambda}\right] u_j^T y$$

col of u

- we shrink coords by $\dfrac{d_j^2}{d_j^2+\lambda}$

- shrinkage is most pronounced for dirns where d_j^2 is small

$\Rightarrow$ components of x with <u>low</u> variance.

$\Rightarrow$ poor estimates of gradient of y in this dirn.

Lasso :

$$\beta^{lasso} = \underset{\min}{\arg} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2$$

s.t.

$$\sum_j |\beta_j| \leq t$$

- can no longer use linear alg; this is a quadratic programming problem
- sufficiently small t forces some $\beta_j = 0$