Regression

Regression

- Supervised learning
- label = quantity
- $y \in Y = \mathbb{R}$
- $\ell(x, y, h) := (y - h(x))^2$
- Risk $= E[(y - h(x))^2] = MSE$

$$h_{Bayes}(x) = h_{MMSE}(x) = E[Y \mid X = x]$$

Ordinary Least Squares Regression

aka linear least squares Reg.

$\underline{x} \in X = \mathbb{R}^d$ $\qquad h(x) = w^T x + b$

$\theta = \begin{pmatrix} w \\ b \end{pmatrix} \in \mathbb{R}^{d+1}$

$$\hat{\theta}_{OLS} = (\hat{w}_{OLS}, \hat{b}_{OLS})^T = \underset{(w,b)}{\operatorname{argmin}} \sum_{j=1}^{n} (y_j - w^T x_j - b)^2$$

Solution

$$\hat{w}_{OLS} = \left(\hat{\Sigma}_x\right)^{-1} \hat{\Sigma}_{xy}$$

$$\hat{b}_{OLS} = \hat{\mu}_y - \underbrace{\hat{\Sigma}_{yx}\left(\hat{\Sigma}_x\right)^{-1}}_{w_{OLS}^T} \hat{\mu}_x$$

Where,

Empirical means and self/cross covariances

$$\underbrace{\hat{\mu}_x}_{d\times 1} = \frac{1}{n}\sum_{j=1}^{n} x_j, \quad \underbrace{\hat{\mu}_y}_{\text{scalar}} := \frac{1}{n}\sum_{j=1}^{n} y_j$$

$$(d\times d) \quad \hat{\Sigma}_x := \frac{1}{n}\sum_{j=1}^{n}(x_j - \hat{\mu}_x)(x_j - \hat{\mu}_x)^T$$

$$(1\times d) \quad \hat{\Sigma}_{yx} = \hat{\Sigma}_{xy}^T = \frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{\mu}_y)(x_j - \hat{\mu}_x)^T$$

Equivalent Expressions

$$\hat{\mu}_x = \frac{1}{n}\underbrace{[x_1 \;..\; x_n]}_{X}\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}_{1_n} = \boxed{\frac{1}{n} X 1_n}$$

$$\hat{\mu}_y = \boxed{\frac{1}{n} Y 1_n}$$

$$\hat{\Sigma}_x = \frac{1}{n}\tilde{X}\tilde{X}^T, \quad \tilde{X} = \begin{bmatrix}\vdots\\\vdots\end{bmatrix} = \underset{(d\times n)}{X}\underset{(n\times n)}{B}$$

$$\hat{\Sigma}_{xy} = \frac{1}{n} \tilde{X} \tilde{Y}^T \quad , \quad \tilde{Y} = [y_1 - \hat{\mu}_y, \ldots, y_n - \hat{\mu}_y] = Y B$$

So, $\hat{w}_{OLS} = \left(\tilde{X} \tilde{X}^T\right)^{-1} \tilde{X} \tilde{Y}^T$

$$\hat{b}_{OLS} = \frac{1}{n} Y \cdot 1_n - (\hat{w}_{OLS})^T \frac{1}{n} X 1_n$$

Discriminitive Learning

OLS as $\uparrow$ $\quad p(y \mid x, w, b) = \mathcal{N}(w^T x + b, 1)(y)$

$$= \frac{1}{\sqrt{2\pi}} e^{\frac{-(y - w^T x - b)^2}{2}}$$

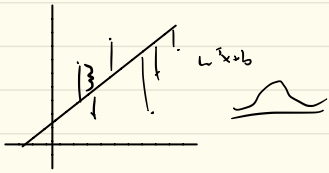
then

$$\hat{\theta}_{ML} = \begin{pmatrix} \hat{w}_{OLS} \\ \hat{b}_{OLS} \end{pmatrix} = \underset{(w,b)}{\operatorname{argmin}} \sum_{i=1}^{n} - \ln p(y_j \mid x_j, w, b)$$

$$= \underset{(w,b)}{\operatorname{argmin}} \left[\sum_{j=1}^{n} \frac{(y_j - w^T x_j - b)^T}{2} + \text{const} \right]$$

Discriminative model

Given $X = x$, $Y = w^T x + b +$ $\mathcal{N}(0,1)$ Gaussian noise



$$(\hat{w}_{OLS}, \hat{b}_{OLS}) = \underset{(w,b)}{\operatorname{argmin}} \sum_{j=1}^{n} \frac{1}{n} (y_j - w^T x_j - b)^2$$

$$= \underset{(w,b)}{\operatorname{argmin}} \; E\left[\underbrace{(Y_{emp} - w^T X_{emp} - b)}_{Z_{emp}}{}^2 \right]$$

$$(X_{emp}, Y_{emp}) \sim Unif\{(x_1, y_1), \dots, (x_n, y_n)\}$$

For any choice of w the best value of b is given by:

$$E[Y_{emp} - w^T X_{emp}]$$

$a = E[Z] \mid E[(Z-a)^2]$ best

$$= E[Y_{emp}] - w^T E[X_{emp}]$$

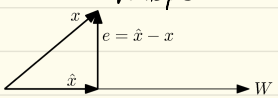
$$= \hat{\mu}_y - w^T \hat{\mu}_x$$

$$\Rightarrow \hat{w}_{OLS} = \underset{w}{\operatorname{argmin}} \left[E\left[(Y_{emp} - w^T X_{emp} - \hat{\mu}_y + w^T \hat{\mu}_x)^2 \right] \right]$$

$$= \underset{w}{\operatorname{argmin}} \; E\left[\left(\underbrace{Y_{emp} - \hat{\mu}_y}_{\tilde{Y}_{emp}} - w^T \underbrace{(X_{emp} - \hat{\mu}_x)}_{\tilde{X}_{emp}} \right)^2 \right]$$

$$= \underset{w}{\operatorname{argmin}} \; E\left[(\tilde{Y}_{emp} - w^T \tilde{X}_{emp})^2 \right]$$

by orthogonality principle: $\tilde{Y}_{emp} - \hat{w}_{OLS}^T \tilde{X}_{emp} \perp \tilde{X}_{emp}$



$$\Rightarrow E\left[(\tilde{Y}_{emp} - \hat{w}^T \tilde{X}_{emp}) \tilde{X}_{emp}^T \right] = 0$$

$$\Rightarrow \quad E\left[\left(\tilde{Y}_{exp} \tilde{X}_{exp}^T \right) \right] = \hat{W}_{OLS}^T \, E\left[\tilde{X}_{exp} \tilde{X}_{exp}^T \right]$$

$$\hat{\Sigma}_{xy} = \left(\hat{W}_{OLS} \right)^T \left(\hat{\Sigma}_x \right)$$

$$\Rightarrow \quad \hat{\Sigma}_{xy} = \left(\hat{\Sigma}_x \right) \hat{W}_{OLS}$$

$$\Rightarrow \quad \hat{W}_{OLS} = \left(\hat{\Sigma}$$

Regression Solution 1 $\qquad y \in Y = \mathbb{R}$

$$l(x, y, h) = (y - h(x))^2$$

OLS $\qquad h(x) = w^T x + b$

$$\left(\hat{w}_{OLS}, \hat{b}_{OLS} \right) = \underset{(w,b)}{\arg} \sum_{i=1}^{n} \left(y_j - w^T x_j - b \right)^2$$

$$\hat{w}_{OLS} = \left(\hat{\Sigma}_x \right)^{-1} \hat{\Sigma}_{xy}$$

$$\hat{b}_{OLS} = \hat{\mu}_y - \left(\hat{w}_{OLS} \right)^T \hat{\mu}_x$$

$$\hat{W}_{OLS} = \left(\tilde{X} \tilde{X}^T \right)^{-1} \tilde{X} Y^T$$

$$\hat{b}_{OLS} = \left(\frac{1}{n} Y 1_n \right) - \left(\hat{W}_{OLS} \right)^T \cdot \frac{1}{n} X 1_n$$

$X_{ext} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}$

More compactly,

$$\hat{\theta}_{OLS} = \underbrace{\begin{pmatrix} \hat{w}_{OLS} \\ \hat{b}_{OLS} \end{pmatrix}}_{(d+1)\times 1} = \underbrace{(X_{ext}}_{(d+1)\times n} \underbrace{X_{ext}^T)^{-1}}_{(d+1)\times n} \underbrace{X_{ext}}_{(d+1)\times n} \underbrace{Y^T}_{n\times 1}$$

$x_j \to \begin{pmatrix} x_j \\ 1 \end{pmatrix}$ then $X \to X_{ext} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}$ 2×18

$Y = (y_1 \cdots y_n)$ 1×18

$X = (x_1 \cdots x_n)$ 1×18

$$\sum_{j=1}^{n} (y_j - w^T x_j - b)^2 =$$

$$\left\| \begin{pmatrix} y_1 - w^T x_1 - b \\ y_2 - w^T x_2 - b \\ \vdots \\ y_n - w^T x_n - b \end{pmatrix} \right\|_2^2$$

$$= \left\| \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} - \begin{pmatrix} -x_1^T- \\ \vdots \\ -x_n^T- \end{pmatrix} w - b \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \right\|_2^2$$

$\underline{X}$ is size $d \times n$ 1×50

$\underline{Y}$ same size

$$\hat{\theta}_{OLS} = \underbrace{\begin{pmatrix} \hat{w}_{OLS} \\ \hat{b}_{OLS} \end{pmatrix}}_{(d+1)\times 1} = \left(\underbrace{\underline{X}_{ext}}_{(d+1)\times n} \underline{X}_{ext}^{T} \right)^{-1} \underline{X}_{ext} \underbrace{\underline{Y}^{T}}_{n\times 1}$$

where $\underline{X}_{ext} = \begin{bmatrix} x_1, x_2, \ldots, x_n \\ 1, 1, \ldots, 1 \end{bmatrix}$

$$= \| Y^T - X^T w - b 1_n \|_2^2$$

for any value of w, best b is given by the O.P. as

$$(Y^T - X^T w - b 1_n) \perp 1_n$$

$$(Y - w^T X - b 1_n^T) 1_n = 0$$

$$\Rightarrow (Y 1_n - w^T X 1_n) = b(1_n^T 1_n) = nb$$

$\Rightarrow$ best value of b for any given w is

$$b = \frac{1}{n} Y 1_n - w^T \left(\frac{1}{n} X 1_n\right)$$

$$= \hat{\mu}_y - w^T \hat{\mu}_x$$

Substituting this b into cost we get

$$\| Y^T - X^T w - \hat{\mu}_y 1_n + \left(\frac{1}{n}\right) 1_n 1_n^T X^T w \|_2^2$$

$$= \| (Y^T - \hat{\mu}_y 1_n) - (X^T - \frac{1}{n} 1_n 1_n^T X^T) w \|_2^2$$

$$= \| \tilde{Y}^T - \tilde{X}^T w \|_2^2$$

By orthog O.P., best w (i.e. $\hat{w}_{OLS}$) is given by

$(\tilde{Y} - \tilde{X}^T \hat{w}_{OLS}) \perp$ all cols of $\tilde{X}^T$

$\Rightarrow \tilde{X}(\tilde{Y}^T - \tilde{X}^T \hat{w}_{OLS}) = 0$

$\Rightarrow \hat{w}_{OLS} = (\tilde{X}\tilde{X}^T)^{-1} \tilde{X} Y^T$

$w^T x_j + b = (w^T, b) \begin{pmatrix} x_j \\ 1 \end{pmatrix}$

$= \theta^T x_j^{ext}$ (extended x_j)

$\Rightarrow$ Cost function $= \| Y^T - X_{ext}^T \theta \|_2^2$

By O.P., $(Y^T - X_{ext}^T \hat{\theta}_{OLS}) \perp$ all columns of X_{ext}^T

$\Rightarrow (X_{ext}^T)^T (Y^T - X_{ext}^T \hat{\theta}_{OLS}) = 0$

$\Rightarrow X_{ext} Y^T - X_{ext} X_{ext}^T \hat{\theta}_{OLS} = 0$

$\Rightarrow \hat{\theta}_{OLS} = (X_{ext} X_{ext}^T)^{-1} X_{ext} Y^T$

C,

so, note that whenever you see an optimization problem which looks like this:

$$\underset{w}{\operatorname{argmin}} \sum_{j=1}^{n} \left(\tilde{y}_j - w^T \tilde{x}_j \right)^2$$

$$\hat{w}_{opt} = \left(\tilde{\underline{X}} \tilde{\underline{X}}^T \right)^{-1} \tilde{\underline{X}} \tilde{\underline{Y}}^T$$

where

$$\tilde{\underline{X}} = \left(\tilde{x}_1 \ \cdots \ \tilde{x}_n \right)$$

$$\tilde{\underline{Y}} = \left(\tilde{y}_1 \ \cdots \ \tilde{y}_n \right)$$

Regularization : Penalize Complexity, introduce additional information in order to solve ill-posed problems or to prevent overfitting

Ridge Regression (l_2 - regularized regression)

$$(\hat{w}_{ridge}, \hat{b}_{ridge}) := \underset{(w,b)}{\operatorname{argmin}} \left[\underbrace{\sum_{j=1}^{n} (y_j - w^T x_j - b)^T}_{N.L.L.} + \underbrace{\lambda \|w\|^2}_{-\ln \pi(w)} \right]$$

Prior

$y_j = w^T x_j + b + \text{noise}$, noise $\sim N(0,1)$

$w \sim N(0, \frac{1}{2\lambda})$

This is equivalent to Bayesian Discriminative Learning with

$$p(y \mid w, b, x) = N(w^T x + b, \tfrac{1}{2})$$

$$\pi(w) \sim N(0, \tfrac{1}{2\lambda})$$

Solution : 1) for any w, best b is given by

$$(\hat{\mu}_y - w^T \hat{\mu}_x)$$

2) $$\tilde{w}_{ridge} = (\tilde{X}\tilde{X}^T + \lambda I_\chi)^{-1} \tilde{X}\tilde{Y}^T$$

1) for any w, best b choice is the one which minimizes the first (likelihood) term which is the same as in OLS

$$\Rightarrow \boxed{b_{best}(w) = \hat{\mu}_y - w^T \hat{\mu}_x}$$

the cost function (in terms of only w) becomes

$$\sum_{j=1}^{n} (\tilde{y}_j - w^T \tilde{x}_j)^2 + \lambda \|w\|_2^2$$

$$\sum_{j=1}^{n} (\tilde{y}_j - x_j^T w)^2 + \lambda \sum_{k=1}^{d} w_k^2$$

$$(0 - w^T(\sqrt{\lambda} e_k))^2$$

$e_k = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \leftarrow k^{th}$ position

$$(w_1 \dots w_k \dots w_d) \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} = w_k$$

$$(\tilde{y}_1 \ \tilde{y}_2 \ \dots \ \tilde{y}_n \underbrace{0 \ 0 \ \dots \ 0}_{d \text{ zeros}})$$

$$(\tilde{x}_1 \ \tilde{x}_2 \ \dots \ \tilde{x}_n \ \sqrt{\lambda} e_1 \ \sqrt{\lambda} e_2 \ \dots \ \sqrt{\lambda} e_d)$$

$$\tilde{X} \rightarrow (\tilde{X}, \ \sqrt{\lambda} I_d) \rightarrow \tilde{X}_{new}$$

$$Y \rightarrow (\tilde{y}, \ \underbrace{0}_{1 \times d}) \rightarrow \tilde{Y}_{new}$$

$$\Rightarrow \tilde{w}_{ridge} = (\tilde{X}_{new} \tilde{X}_{new}^T)^{-1} \tilde{X}_{new} \tilde{Y}_{new}^T$$

$$= \left((\tilde{X} \ \sqrt{\lambda} I_n) \begin{pmatrix} \tilde{X}^T \\ \sqrt{\lambda} I_n \end{pmatrix} \right)^{-1} (\tilde{X} \ \sqrt{\lambda} I_n) \begin{pmatrix} \tilde{Y}^T \\ 0^T \end{pmatrix}$$

$$= \boxed{(\tilde{X} \tilde{X}^T + \lambda I_d)^{-1} \tilde{X} \tilde{Y}^T}$$

$$h_{\text{ridge}}(x) = (\hat{w}_{\text{ridge}})^T x + (\hat{b}_{\text{ridge}}) \qquad \boxed{\lambda > 0}$$

$$= (\hat{w}_{\text{ridge}})^T (x - \hat{\mu}_x) + \hat{\mu}_y$$

$$= \tilde{Y} \tilde{X}^T (\lambda I_d + \tilde{X} \tilde{X}^T)^{-1} (x - \hat{\mu}_x) + \hat{\mu}_y$$

$$= \tilde{Y} \tilde{X}^T (\lambda I_d + \tilde{X} \tilde{X}^T)^{-1} (x - \tfrac{1}{n} X 1_n) + \tfrac{1}{n} Y 1_n$$

$$= \tilde{Y} (\lambda I_n + \tilde{X}^T \tilde{X})^{-1} \tilde{X}^T (x - \tfrac{1}{n} X 1_n) + \tfrac{1}{n} Y 1_n$$

$$\tilde{X} = X \underbrace{(I - \tfrac{1}{n} 1_n 1_n^T)}_{\text{"centering" matrix}} = XB$$

Kernalization of Ridge Regression

Key matrix identity

$$\underbrace{X^T}_{n \times d} (\lambda I_d + \underbrace{G}_{d \times n} \underbrace{X^T}_{n \times d})^{-1}$$

$$= (\lambda I_d + \underbrace{X^T}_{n \times d} \underbrace{G}_{d \times n})^{-1} X^T$$

$$h_{ridge}(x) = \bar{Y}\left(\lambda I_n + B^T(X^T X)B\right)^{-1} B^T\left(X^T x - \frac{1}{n} X^T X 1_n\right) + \frac{1}{n} Y 1_n$$

- $(X^T X)$: Gram matrix, i,j th element $= x_i^T x_j$ → replace by $K_{n \times n}$
- $X^T x$: i th element $= (x_i^T x)$ → replace by $k(x_i, x)$
- $X^T X$: replace by K
- Y: replace by K

$$K(i,j) = K(x_i, x_j)$$

· Weighted LS regression

$$(\hat{w}_{weighted}, \hat{b}_{weighted}) = \underset{(w,b)}{\operatorname{argmin}} \left[\sum_{j=1}^{n} \gamma_j \left(y_j - w^T x_j - b\right)^2 + \lambda \|w\|_2^2 \right]$$

non-neg. weights $\gamma_1, \ldots, \gamma_n$

$$\hat{b}_{weighted} = \hat{\mu}_y^{weighted} - \hat{w}^T_{weighted} \hat{\mu}_{x\,weighted}$$

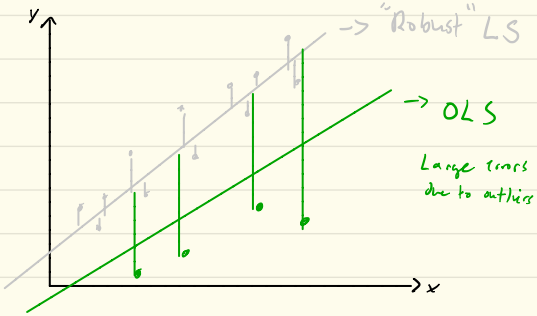
$$\hat{\mu}_{y\,weighted} = \frac{\left(\sum_{j=1}^{n} \gamma_j y_j\right)}{\left(\sum_{j=1}^{n} \gamma_j\right)} \qquad \hat{\mu}_{x\,weighted} = \frac{\sum_{j=1}^{n} \gamma_j x_j}{\left(\sum_{j=1}^{n} \gamma_j\right)}$$

$$\hat{w}_{weighted} = \left(\tilde{X}^{weighted} \cdot \Gamma \cdot \left(\tilde{X}^{weighted}\right)^T + \lambda I_d\right)^{-1} \tilde{X}_{weighted} \cdot \Gamma \cdot \left(\tilde{Y}_{weighted}\right)^T$$

$$\Gamma = \begin{pmatrix} \gamma_1 & & 0 \\ & \ddots & \\ 0 & & \gamma_n \end{pmatrix}, \quad \tilde{x}_j^{weighted} = x_j - \hat{\mu}_x^{weighted}$$

$$\tilde{y}_j^{weighted} = y_j - \hat{\mu}_y^{weighted}$$

Robust Least Squares



$$\hat{\theta}_{OLS} = (\hat{w}_{OLS}, \hat{b}_{OLS})^T = \underset{\theta}{\operatorname{argmin}} \sum_{j=1}^{n} \underbrace{(y_j - \theta^T x_j)^2}_{r_j}$$

$\rho(r) = r^2$ — jth residual / prediction error

$$= \underset{\theta}{\operatorname{argmin}} \sum_{j=1}^{n} \rho(r_j)$$

$$r_j = (y_j - \theta^T x_j)$$

$$\hat{\theta}_{Robust} := \underset{\theta}{\operatorname{argmin}} \left[\sum_{j=1}^{n} \rho(y_j - \theta^T x_j) + \lambda \|w\|^2 \right]$$

$\rho(r)$:

1) real, non-negative $\rho(r) \geq 0$

2) symmetric : $\rho(r) = \rho(-r)$

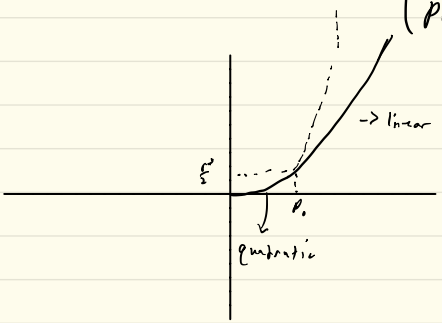
3) non-decreasing in $|r|$

if $|r| \leq |r'| \Rightarrow \rho(r) \leq \rho(r')$

4) differentiable, $\rho'(r)$ exists

5) $\rho(0) = 0$

Huber cost $\rightarrow$ $\rho(r) = \begin{cases} \frac{r^2}{2} & |r| \le \rho_0 \\ \rho_0 |r| - \frac{\rho_0^2}{2} & |r| \ge \rho_0 \end{cases}$



$\rho(r) \rightarrow$ - Huber
- quadratic
etc...

Solution idea (if $\rho()$ convex $\rightarrow$ can use gradient descent)

$$\nabla_\theta = 0$$

$$\nabla_\theta \left[\sum_{j=1}^{n} \rho(y_j - \theta^T x_j) + \lambda \|w\|^2 \right] = 0$$

$$\Rightarrow \sum_{j=1}^{n} \rho'(\underbrace{y_j - \theta^T x_j}_{r_j})(-x_j) + \lambda \begin{pmatrix} 2w \\ 0 \end{pmatrix} = 0$$

Define Influence function $\psi(r) := \rho'(r) = \frac{\partial}{\partial r} \rho(r)$

weight function $\gamma(r) := \frac{\psi(r)}{r}$

since ρ is non-decreasing in $|r|$

$\Rightarrow \gamma(r) \ge 0 \quad \forall r$, then let $\gamma_j := \gamma(y_j - y^T x_j)$

$$\sum_{j=1}^{n} \frac{\rho'(r_j)}{r_j} \quad r_j(-x_j) + \lambda \begin{pmatrix} 2w \\ 0 \end{pmatrix} = 0$$

$$= \sum_{j=1}^{n} \psi(r_j) \cdot r_j(-x_j) + \lambda = 0$$

$$\sum_{j=1}^{n} r_j (y_j - w^T x_j)(-x_j) + \lambda \nabla_\theta (\|w\|^2) = 0$$

$$\vdots$$

Iteratively Re-weighted LS (IRLS)

① Initialization: select $\hat{\theta}^{(0)}$, e.g. $\hat{\theta}_{OLS}$

② Iterations: for $t = 1, \ldots, d\theta$

$\forall j$ calculate: $r_j^{(t-1)} = y - (\hat{\theta}^{(t-1)})^T x_j$ (residuals)

(wts) $\gamma_j^{(t-1)} = \dfrac{\psi(r_j^{(t-1)})}{r_j^{(t-1)}}$

Solve for new θ using weighted LS

$$\hat{\theta}^{(t)} = (X^T W^{(t-1)} X + \lambda I)^{-1} X^T W^{(t-1)} y$$

Regression

- OLS $\underset{w,b}{\operatorname{argmin}} \sum_{j=1}^{n} (y_j - w^T x_j - b)^2$
- Ridge $\underset{w,b}{\operatorname{argmin}} \left[\sum_{j=1}^{n} (y_j - w^T x_j - b)^2 + \lambda \|w\|_2^2 \right]$

$$\hat{w}_{ridge} = (\tilde{X}\tilde{X}^T + \lambda I_d)^{-1} \tilde{X}\tilde{Y}^T$$

$$\hat{b}_{ridge} = \hat{\mu}_y - (\hat{w}_{ridge})^T \hat{\mu}_x$$

- Kernel ridge regression
- Weighted least squares:

$$\underset{w,b}{\operatorname{argmin}} \left[\sum_{j=1}^{n} \gamma_j (y_j - w^T x_j - b)^2 + \lambda \|w\|_2^2 \right]$$

- Robust least squares:

$$\underset{w,b}{\operatorname{argmin}} \sum_{j=1}^{n} \rho(y_j - w^T x_j - b) + \lambda \|w\|^2$$

$\rho(\cdot) \rightarrow$ non-neg., symmetric, non-decreasing $[0, \infty)$, differentiable, $\rho(0) = 0$

$$\nabla_\theta (\text{cost}) = 0$$

$$\Rightarrow \sum_{j=1}^{n} \rho'(y_j - x^T x_j - b) \begin{pmatrix} -x_j \\ -1 \end{pmatrix} + \lambda \begin{pmatrix} 2w \\ 0 \end{pmatrix} = 0$$

Influence function

$$\psi(r) = \rho'(r)$$

Weight function $\quad \gamma(r) = \frac{\psi(r)}{r} \geq 0$ (since ρ is non-decreasing on $[0, \infty)$ & symmetric)

$$\Rightarrow \nabla_\theta (\text{cost}) = 0$$

$$\Rightarrow \sum_{j=1}^{n} \underbrace{\frac{\rho'(r_j)}{r_j}}_{\gamma(r_j)} (y_j - w^T x_j - b) \nabla_\theta (y_j - w^T x_j - b) + \lambda \nabla_\theta (\|w\|^2)$$

$$\Rightarrow \sum_{j=1}^{n} \gamma_j (y_j - w^T x_j - b) \cdot (\nabla_\theta (y_j - w^T x_j - b) + \nabla_\theta (\|w\|^2) = 0$$

If γ_j did not depend on θ

$$\nabla_\theta \left[\sum_{j=1}^{n} \gamma_j (y_j - w^T x_j - b)^2 + \lambda \|w\|_2^2 \right] = 0$$

Sparse Linear Regression (LASSO)

- Assume data is "centered" $x_j \to \tilde{x}_j = x_j - \hat{\mu}_x$

$y_j \to \tilde{y}_j = y_j - \hat{\mu}_y$

Consider OLS

$$\hat{w}_{OLS} = \underset{w,b}{\operatorname{argmin}} \sum_{j=1}^{n} (\tilde{y}_j - w^T \tilde{x}_j)^2$$

Decision rule: $h_{OLS}(x) = (\hat{w}_{OLS})^T x + \hat{b}_{OLS}$

$$w^T \tilde{x}_j = \underset{1\times d}{(w_1\ w_2\ \cdots\ w_d)} \underset{d\times 1}{\begin{pmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_d \end{pmatrix}}$$

If $d \gg 1$

$\Rightarrow$ many features to predict $\tilde{y}_j$

But only some may be significant

- would like to use only a "few" relevant features to predict label.

$\Rightarrow$ Want solutions w that are SPARSE meaning only a few large non-zero values in w (many components of w are zero)

Idea: introduce sparsity-promoting prior for w

$$\hat{w}_{LASSO} = \underset{w}{\operatorname{argmin}} \left[\sum_{j=1}^{n} (\tilde{y}_j - w^T \tilde{x}_j)^2 + \lambda \|w\|_1 \right]$$

(strictly convex + convex = strictly convex)

$$\|w\|_1 = |w_1| + |w_2| + \ldots + |w_d|$$

This is equivalent to Bayesian Discriminative Learning with

$$\Pi(w) = \prod_{i=1}^{d} \left(\frac{\lambda}{2}\right) e^{-\lambda |w_i|} \quad \text{(i.i.d. Laplacian components)}$$

→ this produces sparse w

→ larger the λ, more sparse is the solution

→ **Objective Function** is strictly convex

⇒ unique global minimum

→ no closed form solution

→ objective function is not differentiable everywhere

⇒ can't use gradient descent

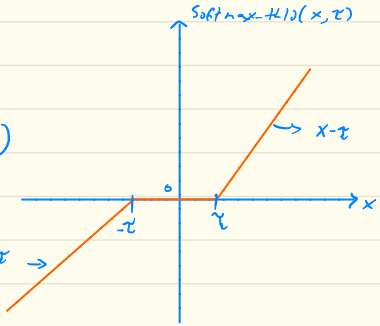
Recall (from assignment #2)

Soft-thresholding $\underset{y}{\operatorname{argmin}} \left[(x-y)^2 + 2\tau|y| \right]$

$= soft\_thld(x;\tau)$

$= sign(x)(|x| - \tau)$

$= sign(x)\max(0, |x| - \tau)$



LASSO: least absolute selection & shrinkage operator

(Tibshirani - 1996)

Intuition behind properties of solution

$$\text{Cost}(w) = \sum_{j=1}^{n} (\tilde{y}_j - w^T \tilde{x}_j)^2 + \lambda \sum_{i=1}^{d} |w_i|$$

$$\| \tilde{Y}^T - \tilde{X}^T w \|_2^2 + \lambda \|w\|_1$$

$$(\tilde{Y}\tilde{Y}^T) + w^T(\tilde{X}\tilde{X}^T)w - 2\underbrace{(\tilde{Y}\tilde{X}^T)}_{z^T} w + \lambda \|w\|_1$$

Suppose (for simplicity of explanation) that $\tilde{X}\tilde{X}^T = I$

then

$$\hat{w}_{LASSO} = \underset{w}{\text{argmin}} \left[\|w\|^2 - z^T w + \lambda \|w\|_1 \right]$$

$$= \underset{w}{\text{argmin}} \sum_{i=1}^{d} \left[w_i^2 - 2 z_i w_i + z_i^2 + \lambda |w_i| \right]$$

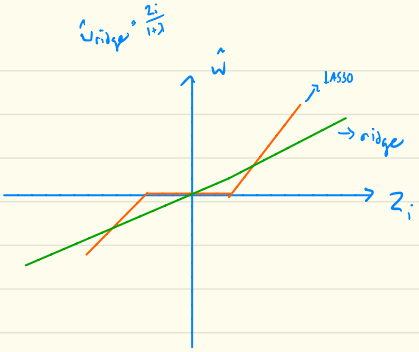
$$= \underset{w}{\text{argmin}} \sum_{i=1}^{d} \left[(w_i - z_i)^2 + \lambda |w_i| \right]$$

$$\hat{w}_i^{LASSO} = \text{soft\_thld}\left(z_i, \frac{\lambda}{2}\right)$$

For ridge regression,

$$\hat{w}_{ridge} = \underset{w}{\text{argmin}} \left[\sum_{i=1}^{d} (z_i - w_i)^2 + \lambda w_i^2 \right]$$

Solution $\frac{\partial}{\partial w}$ (ith term) $= 0$, $\quad -2(z_i - w_i) + 2\lambda w_i = 0$



Best Subset Selection

$$\hat{w}_{\text{best-subset}} = \underset{w:\ \text{not more } k \text{ nonzero components in } w}{\operatorname{argmin}} \left[\sum_{j=1}^{n} (\tilde{y}_j - w^T \tilde{x}_j)^2 \right]$$

(If $\tilde{X}\tilde{X}^T = I$)

$$= \underset{\substack{w \\ \text{supp}(w) \le k}}{\operatorname{argmin}} \sum_{i=1}^{d} (z_i - w_i)^2$$

$$\hat{w}_{\text{best-subset}}(i) = \begin{cases} z_i & \text{if } \text{rank}(z_i) \le k \\ 0 & \text{otherwise} \end{cases}$$

$$\text{rank}(z_i) = \begin{cases} = 1 & \text{if } |z_i| = \text{largest among } |z_1| \dots |z_d| \\ = 2 & \text{if 2nd largest value} \\ = 3 \dots \end{cases}$$

$\rightarrow$ Hard Thresholding Rule

LASSO Solution (general case)

$$\underset{w}{\arg\min} \left[\sum_{j=1}^{n} \left(\tilde{y}_j - w^T \tilde{x}_j \right)^2 + \lambda \|w\|_1 \right]$$

Method 1 : $|w_i| = w_i^+ + w_i^-$

$$|t| = t^+ + t^-$$
$$t = t^+ + t^-$$

$$t^+ = \begin{cases} t & \text{if } t > 0 \\ 0 & \text{o/w} \end{cases} \qquad t^- = \begin{cases} t & \text{if } t < 0 \\ 0 & \text{else} \end{cases}$$

e.g. $(2)^+ = 2$
$(2)^- = 0$

then with these "SLACK" variables w_i^+, w_i^-, cost-fn becomes a Quadratic Program

Cyclic Coordinate Descent

Idea: $w = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix}$

Initialize $\underline{w} = w_0$

for $t = 1$ (iterate this many times)

  for $i = 1, \ldots, d$ (cycle through all coordinates)

  - fix all values of w except w_i
  - minimize with respect to just w_i

  end for

Formal algorithm

Initialize $w = w^{(0)}$ say $\hat{w}_{ridge}$

$$w = \begin{pmatrix} w_1 \\ \vdots \\ w_k \\ \vdots \\ w_d \end{pmatrix}$$

For $t = 1, \ldots,$ until convergence / stop

For $k = 1, \ldots, d$ (cycle through each coordinate)

$$w_k = \underset{z}{\operatorname{argmin}}\ \text{Cost}(w_1, \ldots, w_{k-1}, z, w_{k+1}, \ldots, w_d) \quad \text{(} k^{th} \text{ coordinate)}$$

$$a_k = 2 \sum_{j=1}^{n} (\tilde{x}_{kj})^2$$

$$c_k = 2 \left[\sum_{j=1}^{n} \tilde{x}_{kj} \left(\tilde{y}_j - w^T \tilde{x}_j + w_k \tilde{x}_{kj} \right) \right]$$

($\tilde{x}_j$: vector; $w_k \tilde{x}_{kj}$: change)

$$w_k = \text{soft\_thld}\left(\frac{c_k}{a_k}, \frac{\lambda}{a_k} \right)$$

endfor

endfor

initialize $w = \hat{w}_{ridge} = (\tilde{X}\tilde{X} + \lambda I)^{-1} \tilde{X} \tilde{y}^T$

Set max # iterations t_{max}

For $t = 1, \ldots, t_{max}$ or convergence

For $k = 1, \ldots, d$

ABOVE

end

end

---

* quiz next tuesday

Cyclic coordinate Descent

$$Cost(w) = \underbrace{f(w)}_{\text{strictly convex, differentiable}} + \underbrace{g(w)}_{\text{convex}}$$

Initialize $w = \underline{w}^{(0)}$ say $\hat{w}_{ridge}$

For $t = 1, \ldots,$ until convergence / stop

For $k = 1, \ldots, d$ (cycle through each coordinate)

Keep all but k fixed so that is only a function of z, solve minimization problem for w_k

$$w_k = \underset{z}{\arg\min}\ Cost(w_1, \ldots, w_{k-1}, \underbrace{z}_{k^{th} \text{ coordinate}}, w_{k+1}, \ldots, w_d)$$

end for

end for

Stopping criteria

1) Value of $Cost(\cdot)$ does not change "much" from one cycle to another

2) Values of $\underline{w}$ don't change "much"

Solution to minimization problem on previous page

$$\tilde{x}_j = \begin{pmatrix} \tilde{x}_{k_j} \\ \vdots \\ \tilde{x}_{d_j} \end{pmatrix} = x_j - \tilde{\mu}_x$$

$$\text{Cost}(w) = \sum_{j=1}^{n} \left[\left(\tilde{y}_j - w^T \tilde{x}_j \right)^2 + \lambda |w_1| \right]$$

$$= \left[\sum_{j=1}^{n} \left(\left(\tilde{y}_j - \sum_{i, i \neq k} w_i x_{i_j} \right) - w_k \tilde{x}_{k_j} \right)^2 + \lambda |w_k| + \lambda \sum_{i, i \neq k} |w_i| \right]$$

$$= \sum_{j=1}^{n} \left(\tilde{y}_j - \sum_{i, i \neq k} w_i \tilde{x}_{i_j} \right)^2 + \sum_{i=1}^{n} w_k^2 \left(\tilde{x}_{k_j} \right)^2 - 2 w_k \sum_{j=1}^{n} \tilde{x}_{k_j} \left(\tilde{y}_j - \sum_{i, i \neq k} w_i \tilde{x}_{i_j} \right) + \lambda |w_k| + \lambda \sum_{i \neq k} |w_i|$$

$$\text{Cost}(w) = (\text{terms that don't depend on } w_k) + w_k^2 \left(\sum_{j=1}^{n} \left(x_{k_j} \right)^2 \right) - 2 w_k \left(\sum_{j=1}^{n} \tilde{x}_{k_j} \left(\tilde{y}_j - \sum_{i, i \neq k} w_i \tilde{x}_{i_j} \right) \right) + \lambda |w_k|$$

$$= \begin{pmatrix} ? \\ w_k \end{pmatrix} + w_k^2 \frac{a_k}{2} + \lambda |w_k| - 2 w_k \left(\frac{c_k}{2} \right)$$

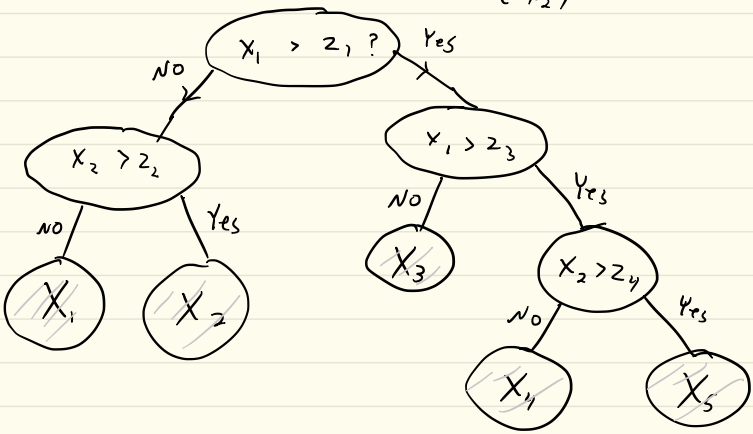
Binary Decision Trees for Classification & Regression

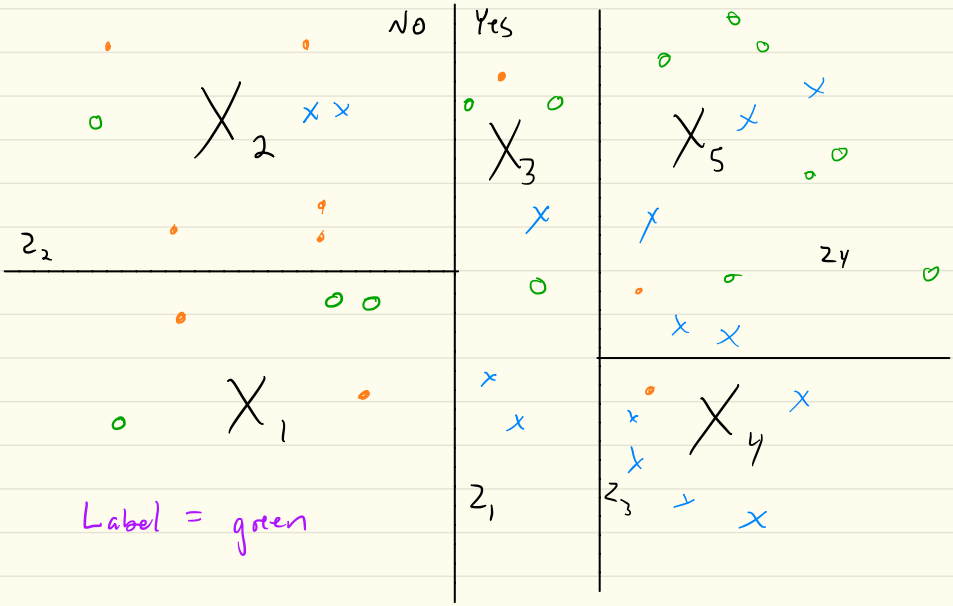
- Elements of Statistical Learning - Hastie, Tibshirani, Friedman
  pgs. 305-317, ch. 9.2

Idea

1) partition feature space X into non-overlapping regions
2) via a sequence of SIMPLE Yes/no questions
3) Estimate the label in each region as:
   - for classification: most abundant label among all training features in regions
   - for regression: simple average of labels of training features in region

Example $X = \mathbb{R}^2$, $\underline{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$





T = decision tree = tree structure + all node questions

node questions

$$x_i > z_i \overset{?}{>} 0$$

specified by "i" (which feature to use) & z (threshold)

1) $n(\text{node})$ = # training features in node

2) $\text{leaf}_T(x)$ = leaf node of T containing x

Classification

3) $n_y(\text{node})$ = # class-y training features in node

4) $\hat{\rho}(y \mid \text{node}) = \frac{n_y(\text{node})}{n(\text{node})}$

Decision Rules

Classification: $h(x)$ = highest in-leaf prob.

regression: μ_x in leaf

Learning Decision Trees

- overfitting problem — can make training error = 0 if leaf nodes contain exactly one (or no) feature vecs

- Need constraints on tree

  * max # leaves
  * min leaf size (# of examples in leaf)
  * degree of "purity / homogeneity" of training labels in leaves

- Finding "best" tree is computationally intractable

⇒ Grow-then-prune strategy

1) Greedily grow a large tree T_0
2) Then prune T_0 to optimize some cost

Regression for Classification

Say 2 classes '0', '1'

$$h_{Bayes}(x) = h_{MAP}(x) = \underset{y \in ('0','1')}{\operatorname{argmax}} \; p(y|x)$$

$$r(x) = p(y=1 \mid X=x) \quad \ldots \quad (1 - p(y=0 \mid X=x))$$

$$r(x) > \frac{1}{2} \Rightarrow h_{Bayes}(x) = 1$$

$$h_{Bayes}(x) = 1\left(r(x) > \frac{1}{2}\right)$$

$$p(y=1 \mid X=x)$$

If '0' = 0 (real value)
'1' = 1 (real value)

$p(y=1 \mid X=x) = E[Y \mid X=x]$ = MMSE estimate of numerical Y given $X=x$

Back to decision trees

- Exhaustive search → Computationally intractable

- Grow-then-prune
  - ↳ Greedily grow a large tree T_0
  - ↳ Prune T_0 to optimize some cost which measures tree complexity

Top-down growing of T_0
- Initialize $T_0 = X$ (contains all examples)
- Find best split for each leaf node
- Keep best split if acceptable
- Continue until no more acceptable splits possible for any leaf where depth < max depth

Finding best split for a node

- Split ≡ choosing feature # i, threshold τ

$$\text{left\_child}(i, \tau) = \{ x_j \in \text{node} : x_{ij} \leq \tau \}$$

$$\text{right\_child}(i, \tau) = \{ x_j \in \text{node} : x_{ij} > \tau \}$$

$$\text{Cost\_of\_split}(i, \tau) = \frac{n(\text{leftchild})}{n} \cdot \text{cost}(\text{left\_child}) + \frac{n(\text{right\_child})}{n} \cdot \text{cost}(\text{right\_child})$$

Best split:

$$(i_{best}, \tau_{best}) = \underset{(i,\tau)}{\text{argmin}} \ \text{Cost\_of\_split}(i, \tau)$$

Node cost functions

Regression

$$MSE(node) = \frac{1}{n(node)} \sum_{j : x_j \in node} \left(y_j - \hat{\mu}_{y(node)}\right)^2$$

Classification

1) misclassification rate

$$1 - \max_{y \in \{1, \dots, m\}} \hat{p}(y | node)$$

2) Entropy $= \sum_{y=1}^{m} \hat{p}(y|node) \log_2 \left(\frac{1}{\hat{p}(y|node)}\right)$

3) Gini-index $\sum_{y=1}^{m} \hat{p}(y|node)\left(1 - \hat{p}(y|node)\right)$

Can show that

Misclassification rate $\leq$ Gini-index $\leq$ Entropy

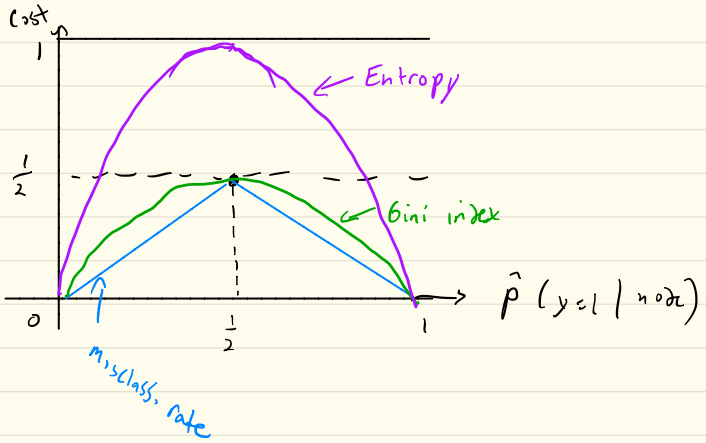
Misclass. rate $\leq 1 - \frac{1}{m}$

Gini-index $\leq 1 - \frac{1}{m}$

Entropy $\leq \log_2 m$

For $m=2$

$$\hat{p}(y=1 \mid \text{node}) = 1 - \hat{p}(y=0 \mid \text{node})$$

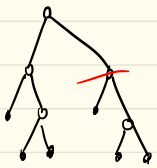


- All 3 costs are measure of "impurity"
- Gini & Entropy are differentiable everywhere
- Gini & Entropy are more sensitive to impurity than misclassification rate

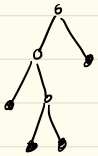
Pruning

- sub tree T of T_0 obtained by pruning T_0

e.g. T_0 =



pruned: T =



Best pruned subtree

$$T_{best} = \underset{T = \text{subtree of } T_0}{\operatorname{argmin}} = \left[\sum_{node \in leaf(T)} n(node)\, Cost(node) + \lambda\, (\#\ leaves\ in\ T) \right]$$

"complexity of T"

There are efficient ways to solve above optimization problem.

Decision Trees

Pros

+ Interpretability / simplicity
+ Can handle quantitative & categorical features together
+ "automatic" feature selection
+ Can handle MISSING features
+ insensitive to monotonic transformations of quantitative features
+ Somewhat insensitive to outliers
+ scalable to large datasets

Cons

− Instability / high variance
  ↳ small changes in dataset can cause large changes in T_{best}
− For regression: the estimated function is non-smooth

Can fix first problem with Random Forests (MS Kinect V1)

Random Forests

- Train multiple decision trees
- Each trained on a random subset of D
- Each trained on a random subset of features

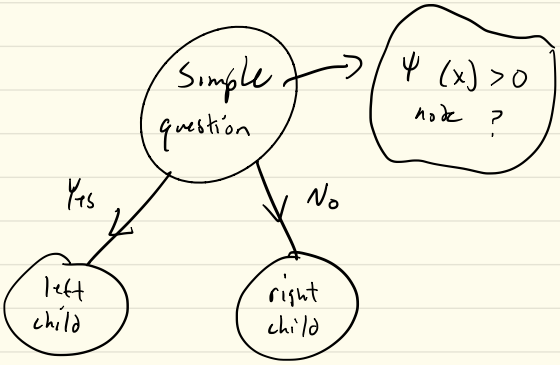
Final decision

- Classif: weighted majority vote (weighted by # samples in the leaf node containing x_{test})
- Regression: Wtd average of $\hat{u}_y(\text{leaf}_{T_k}(x))$

Bagging

- Ensemble decision rules
- used for variance reduction

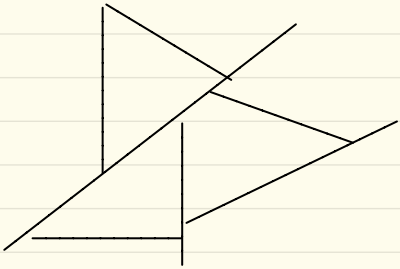
General decision trees



Binary space-partitioning tree (BSP) tree

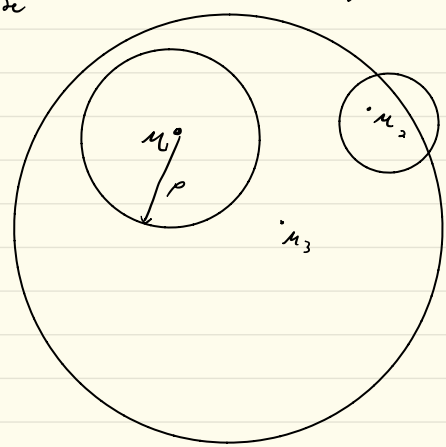
$\psi_{node}(x) = a T_x + b > 0$?

Partitions = convex polyhedral cells



Spherical trees

$\psi_{node}(x)$ $\|x - u\| - \rho \gtrless 0$



Asymtotic Consistency

diam (node) = vol (region ≡ node)

If as $n \to \infty$

1) diam (leaf) $\to 0$

2) n (leaf) $\to \infty$

then Risk $\longrightarrow$ Bayes Risk