SUPERVISED → REGRESSION / → CLASSIFICATION

M12: FINAL
08.04.2019

UNSUPERVISED (PCA, clustering → kmean / → hclust)

REGRESSION

continuous

additive errors

$$Y_i = f(x_i) + \varepsilon_i$$

systematic   random

where $E(\varepsilon_i)$ and $Var(\varepsilon_i) = \sigma^2$

and (x_i, Y_i) $i = 1,..,n$

independent pairs

AIM: estimate $f(x)$

Core: $(Y - f(x))^2$
squared error

Bias-variance trade-off (test mse at an obs x_0)

$$E\left[(Y - \hat{f}(x_0))^2\right] = \cdots = \left[E(\hat{f}(x_0)) - f(x_0)\right]^2 + Var(\hat{f}(x_0)) + Var(\varepsilon)$$

new obs at x_0

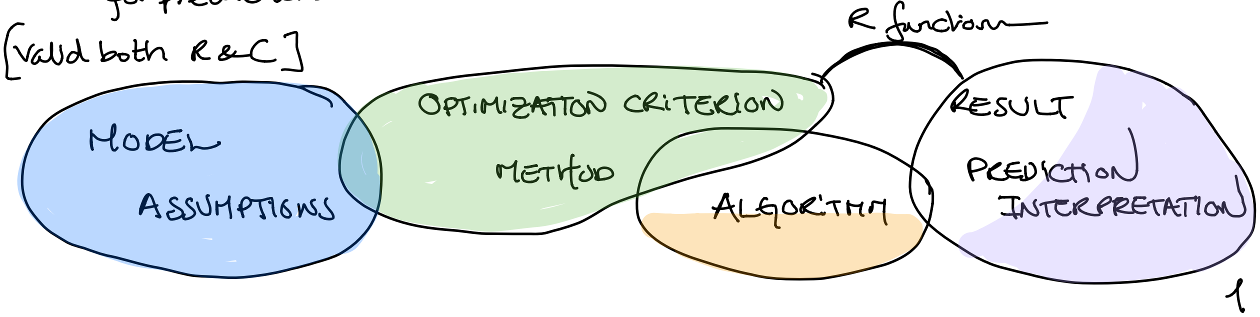
bias^2

irreducible error

Q: how state this orally

- there is random error that you never can fit (not a deterministic world)
- some times an unbiased answer is the best, but if this has high variance, maybe a biased answer is better for prediction?

[valid both R&C]



$f(x)$: how complex? and how much data do we have?
→ guide our choice of solution

1) $Y_i = \overbrace{x_i^T \beta}^{f(x_i)} + \varepsilon_i$ and often $\varepsilon_i \sim N$ $\boxed{M3}$

$$\underset{n\times 1}{Y} = \underset{n\times(p+1)}{X}\,\underset{(p+1)\times 1}{\beta} + \underset{n\times 1}{\varepsilon}$$

"core model"

linear in parameters

& linear in covariates x

$\hat{y}_i = x_i^T \hat{\beta}$

$f(x)$ is a hyperplane

a)

- $\underset{\beta}{\operatorname{argmin}}$ RSS (train MSE)

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \leftarrow \text{least squares estimation} \quad \boxed{M3}$$

- $\underset{\beta}{\operatorname{argmax}}\ \ell(\beta, \sigma^2)$ maximum likelihood $\varepsilon_i \sim N$

$\ell(\beta,\sigma^2) \propto$ RSS + C

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

$$\hat{Y} = X\hat{\beta} = \underbrace{X(X^T X)^{-1} X^T}_{H} Y = HY$$

b) - $\underset{\beta}{\operatorname{argmin}}\ \text{RSS} + \lambda \cdot \sum_{j=1}^{p} \beta_j^2$ $\boxed{M6}$

ridge

$$\hat{\beta} = (X^T X + \lambda I)^{-1} X^T Y$$

c) - $\underset{\beta}{\operatorname{argmin}}\ \text{RSS} + \lambda \cdot \sum_{j=1}^{p} |\beta_j|$ lasso

- regularization: add penalty term
- penalty (hyper) parameter λ ← cross-validation

FIGURES

- dual optimization problem

Alternative to regularization: subset selection $\boxed{M6}$

training data → RSS + penalty (AIC, BIC, R^2_{adj}) ← training MSE

validation data ← MSE, or CV $\boxed{M5}$ (not need AIC etc.)

2) Keep linearity in parameters – but go nonlinear in ~~original~~ covariates!

$\underbrace{\text{keep linearity in parameters}}_{\text{use LS estimation}}$

[if categorical ← dummy variable coding]

$$x_i^T = [1 \quad x_{i1} \quad x_{i2} \quad \dots \quad x_{ip}]$$

x_{ip}: migth be an interaction term

focus on x_{i1}, but same for the others — numerical

replace x_{i1} with

polynomial regr. $\quad x_{i1} \quad x_{i1}^2 \quad x_{i1}^3 \quad \dots \quad x_{i1}^d$

step functions $\quad I(c_1 \le x_{i1} < c_2) \quad I(c_2 \le x_{i1} \le c_3) \quad \dots$

$$(x_{i1} - c_k)^3_+ = \begin{cases} (x_{i1} - c_k)^3 & x_{i1} > c_k \\ 0 & \text{else} \end{cases}$$

truncated power basis

order M spline with K knots [(M-1)+K+1nt]

$$b_j(x_{i1}) = x_{i1}^j \quad j = 1, .., M-1 \qquad x \quad x^2 \quad x^3$$

$$b_{M-1+k}(x_{i1}) = (x_{i1} - c_k)_+^{M-1} \quad k = 1, .., K \text{ — one pr knot}$$

natural cubic splines $\quad \overline{+1nt}$

$$b_1(x_{i1}) = x_{i1}, \quad b_{k+2}(x_{i1}) = d_k(x_{i1}) - d_K(x_{i1}) \quad k = 0, .., K-1$$

$c_0, \dots, c_{K+1}$

$$d_k(x_{i1}) = \frac{(x_{i1} - c_k)^3_+ - (x_{i1} - c_{K+1})^3_+}{c_{K+1} - c_k}$$

k K

3) Local regression and KNN

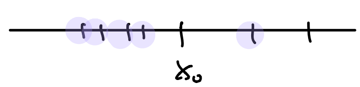
linear in parameters locally

$$\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 x_0^2$$

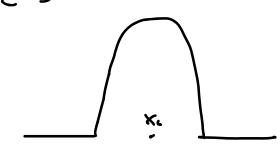
Use $\frac{k}{n}$ closest x_i to x_0 and weigh $K_{i0} = K(x_i, x_0)$ and find $\hat{\beta}_0, \hat{\beta}_1, \beta_2$ by minimizing

weighted LS

$$\sum_{i=1}^{n} K_{i0} \cdot (y_i - \beta_0 - \beta_1 x_i - \beta_2 x_i^2)^2$$



k default



$$\begin{cases} (1-|t|^3)^3 & |t| \le 1 \\ 0 & \end{cases}$$

$$K(x_i, x_0) = \left(1 - \left|\frac{x_0 - x_i}{x_0 - x_k}\right|^3\right)^3_+$$

k closest

tricube (N, Epanechnikov)

KNN: Exam 2018 Probl 1

4) Smoothing spline $\quad Y_i = f(x_i) + \varepsilon_i \quad$ nonlinear

$$\min \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int f''(t)^2 \, dt$$

$\hat{y} = Sy$ where $S = X(X^TX + \lambda\Omega)^{-1} X^T$ $\qquad H = X(X^TX)^{-1}X^TY$

lots of theory here - not in this course

ridge $\quad X(X^TX + \lambda I)^{-1} X^T$

5) Trees

$R_1, \dots, R_J$ non-overlapping regions in predictor space

$\hat{y}_{R_j}$ = mean response of all training samples $x_i \in R_j$

Prediction: $\hat{y}_0 = \hat{y}_{R_j}$ when $x_0 \in R_j$ "step function"

Find R_j's to

$$\text{minimize} \quad \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

$$\left(\text{alt}: \quad \sum_{i=1}^{n} (y_i - \hat{y}_{R_j}(x_i))^2 \right)$$

but do this by greedy recursive binary splitting

- Pruning ← ala lasso penalty
- Bagging & random forest } algorithms
- Boosting

6) Add Linear transformation to MLR → PCR principal comp. regr

Rename: $Y_i = z_i^T \beta + \varepsilon_i$

$$z_i = \begin{bmatrix} e_1^T x_i \\ e_2^T x_i \\ \vdots \\ e_m^T x_i \end{bmatrix} \quad \text{where } S = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T$$

and $S e_i = \lambda_i e_i$

(x_i may be scaled also, so correlation and not covariance matrix used)

7) Neural networks

$$\hat{y}_1(x_i) = \beta_{01} + \sum_{m=1}^{M} \beta_{m1} \cdot z_m(x_i)$$

$$z_m(x_i) = \phi_h\left(\alpha_{0m} + \sum_{j=1}^{p} \alpha_{jm} \cdot x_{ij}\right)$$

$$\phi_h(a) = \max(a, 0) \quad \text{or} \quad \phi_h(a) = \frac{1}{1+\exp(-a)}$$

And, may add more ϕ_h's inside the ϕ_h's = more layers of the NN.

As before: $$\underset{\alpha,\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \hat{y}(x_i))^2$$

and may add regularization of various types

CLASSIFICATION

$Y = \{1, 2, \ldots, J\}$ or $Y = \{0, 1\}$

Observe (x_i, Y_i) jointly, $i = 1, \ldots, n$

independent pairs

1) Bayes classifier (Utopia)

assign a new obs x_0 to the most likely class

$P(Y = j \mid X = x_0)$ ← this will minimize the 0/1 - loss

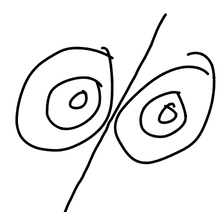
Error rate on obs x_0

$$1 - \max_j P(Y = j \mid X = x_0)$$

Bayes error rate = optimal error rate ← "we know $P(Y=j \mid X)$"

$$= 1 - E_X \left(\max_j P(Y = j \mid X) \right)$$

compare to irreducible error



if you go lower → you prob. have overfitted

2) Two paradigms:

Diagnostic : model $P(Y = j \mid X)$ directly

Sampling : model $P(X \mid Y = j)$ and $P(Y = j)$ and use Bayes theorem to get

$$P(Y = j \mid X = x) = \frac{P(X = x \mid Y = j) P(Y = j)}{P(X = x)}$$

3) Methods:

LDA ④
QDA

$X | Y = j$
$\sim N(\mu_j, \Sigma_j)$

logistic regression ④

KNN-classifier ②

size K

- define neighbourhood
- majority vote

MMC ⑨
SVC
SVM

Classification tree ⑧
Bagging & random forest

Neural nets ⑪

4)

<u>Logistic regression</u> [two-class problem]

$$P(Y=1 \mid x_i) \to p_i = \frac{1}{1+\exp(-x_i^T\beta)}$$

$$\prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}$$

$$\underset{\beta}{\operatorname{argmax}}\ l(\beta) = \underset{\beta}{\operatorname{argmax}} \sum_{i=1}^{n} \left(y_i \log p_i + (1-y_i)\log(1-p_i)\right)$$

odds → IMAGE

5) Classification tree :

$R_1, \dots, R_J$ non-overlapping regions in predictor space

$\hat{y}_{R_j}$ = majority vote in R_j or $\hat{p}_{jk} = n_{jk} / N_j$ ← train in region

when $x_0 \in R_j$ ← train of class k in region j

Find R_j's to minimize

$$G = \sum_{k=1}^{K} \hat{p}_{jk} \cdot (1 - \hat{p}_{jk}) \quad \text{or} \quad D = - \sum_{k=1}^{K} \hat{p}_{jk} \cdot \log \hat{p}_{jk}$$

Gini   cross entropy (closely to deviance)

node impurity

but do this by greedy recursive binary splitting

- Pruning ← ala lasso penalty
- Bagging &
- random forest

} algorithms

6) SVM support vector machine $Y_i \in \{-1, 1\}$

$$\max_{\beta,\alpha,\varepsilon} M \text{ subject to } \sum \beta_j^2 = 1, \ \sum \varepsilon_i \le C, \ \varepsilon_i \ge 0$$

slack budget

$$y_i \cdot f(x_i) \ge M(1-\varepsilon_i)$$

$$f(x) = \beta_0 + \sum_{i \in S} \alpha \cdot K(x, x_i)$$

kernel

alternative way; when $f(x_i) = \beta_0 + x_i^T \beta$ (SVC)

$$\min_\beta \left[\sum_{i=1}^{n} \underbrace{\max(0, 1 - y_i(\beta_0 + x_i^T\beta)}_{\text{hinge loss}}) + \lambda \sum \beta_j^2 \right]$$

compared to logistic regression with (-1,1)

$$l(\beta) \propto \sum_{i=1}^{n} \log\left(1 + e^{-y_i(\beta_0 + x_i^T\beta)}\right)$$

+ add ridge penalty if needed

7) Neural networks

$$\hat{y}_c(x) = \phi_o\left(\beta_{0c} + \sum_{m=1}^{M} \beta_{mc} z_m\right)$$

$$z_m = \phi_h\left(\alpha_{0m} + \sum_{j=1}^{p} \alpha_{jm} x_j\right)$$

one hidden layer

may add more

$$\phi_o(a) = \frac{1}{1+\exp(-a)}$$

$c=2$: binary cross-entropy

$$J(\theta) = -\frac{1}{n}\sum^{n}\left[y_i \cdot \ln \hat{y}_1(x_i) + (1-y_i)\ln(1-\hat{y}_1(x_i))\right]$$

(cut $\frac{1}{2}$)

$$\phi_o(a_j) = \frac{\exp(a_j)}{\sum_{j=1} \exp(a_j)}$$ softmax

$C>2$: categ. c-e.

$$J(\theta) = -\frac{1}{n}\sum^{n}\frac{1}{C}\sum^{C}\left(y_{ic} \cdot \ln \hat{y}_c(x_i)\right)$$

[log vs ln]