K-Nearest Neighbors

$$X = \begin{pmatrix} wt & ht & age \\ x_{11} & x_{12} & x_{1p} \\ \vdots & & \\ x_{n1} & x_{n2} & x_{np} \end{pmatrix} \xrightarrow{\text{Standardize}} \begin{pmatrix} \frac{x_{11}-\bar{x}_1}{sd(x_1)} & \frac{x_{12}-\bar{x}_2}{sd(x_2)} & \frac{x_{1p}-\bar{x}_p}{sd(x_p)} \\ \vdots & \vdots & \vdots \end{pmatrix}$$

1. Because not paranetic
2. $\text{Distance}(x_1, x_2) = \sqrt{(x_{11}' - x_{21}')^2 + \cdots (x_{1p}' - x_{2p}')^2}$

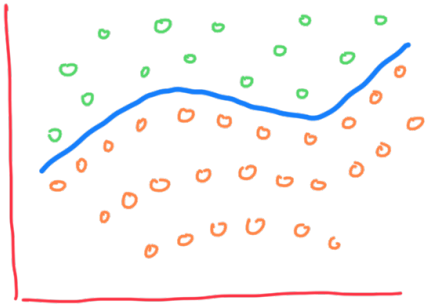
Model complexity $\propto 1/k$

$k = 1$ over-fitting, wiggly, flexible, variance ↑

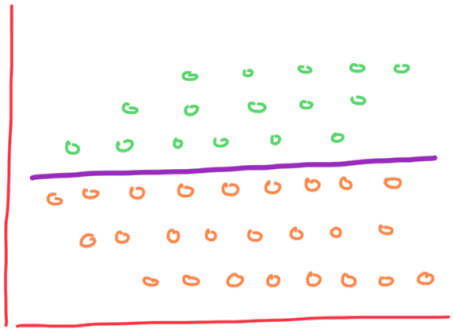
k large inflexible bias ↑

Disadvantage:

1. CURSE OF DIMENSIONALITY: Fail at larger dimensions
2. nearest neighbors tend to be spatially large, bias ↑
3. reducing neighbor by $k\downarrow$ variance ↑



Low value of k is good



Large value of k is good

Q.8. Bias variation derivation for KNN

STATS 503

	Generative $P(Y=C_k \mid x) \propto P(x \mid Y=C_k)$	Discriminative $P(Y=C_k \mid x)$
Parametric	LDA, QDA, NB	Logistic Regression, NN, SVM, DT, RF
Non-parametric	—	kNN

Parametric Methods

1. Make some assumption of functional form
   i.e. come up with some model
   eg. Gaussian distribution for $P(x \mid Y=C_k)$ OR
   Linear model for logit of $P(Y=C_k \mid x)$
2. Use training data to fit model or unknown parameters

Non-Parametric Methods: No explicit assumptions

Advantages

1. Flexible
2. Wider range of shapes

Disadvantages

1. Overfitting
2. Large # observations required

Baye's Theorem

distribution of x given class label; prior probability

$$P(Y=C_k \mid x) = \frac{P(x \mid Y=C_k) \cdot P(Y=C_k)}{P(x)}$$

$$\text{argmax}_k \, P(Y=C_k \mid x) = \text{argmax}_k \left\{ \frac{P(x \mid Y=C_k) \cdot P(Y=C_k)}{P(x) \rightarrow \sum_{k'=1}^{K} \pi_{k'} P(x \mid Y=C_{k'})} \right\}$$

$$\propto \text{argmax}_k \{ P(x \mid Y=C_k) \cdot P(Y=C_k) \} = \text{argmax}_k \, \pi_k p_k(x)$$

independent of k

Generative Methods of classification

$\text{argmax}_k (Y = C_k | X = x) \propto \pi_k p(x | Y = C_k) = \pi_k p_k(x)$

specify a specific direction for $p_k(x)$ (with parameters)

Use training data

$(x_1, y_1) \quad (x_2, y_2) \cdots (x_n, y_n)$

$\Rightarrow \hat{\pi}_k \hat{p}_k(x) \Rightarrow \text{argmax}_k [\hat{\pi}_k \cdot \hat{p}_k(x)] \Rightarrow \hat{C}(x)$ fitted classifier

Assumptions

1) QDA $p_k(x) \sim MVN(\mu_k, \Sigma_k)$

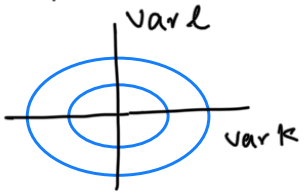
2) LDA $p_k(x) \sim MVN(\mu_k, \Sigma_k)$ $\Sigma_k = \Sigma$ same covariance

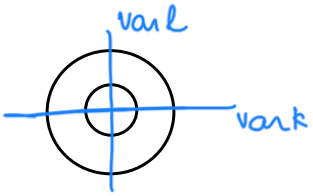
3) NB $p_k(x) \sim MVN(\mu_k, \Sigma_k)$ $\sigma_{ij} = 0 \; ; i \neq j$

$\begin{bmatrix} \sigma_{11} & & \phi \\ & \sigma_{22} & \\ \phi & & \sigma_{kk} \end{bmatrix}$ Σ_k: diagonal matrix no correlation b/w variables

Σ_i denotes the dispersion of the i^{th} class

$$\Sigma_i = \begin{pmatrix} \text{var}(X_1) & \cdots & \text{Cov}(X_p, X_1) \\ \vdots & \text{Var}(X_2) & \vdots \\ \text{cov}(X_1, X_p) & \cdots & \text{var}(X_p) \end{pmatrix}_{p \times p}$$





LDA: same dispersion for all the k classes

NB: when correlation b/w variables = 0 $\Rightarrow$ same dispersion for all classes

Bias: QDA (flexible) < LDA (rigid) < NB

Variance: QDA > LDA > NB

parameters: $O(kp^2) > O(p^2) > O(kp)$ $\because p > k$

Use Cases

QDA: variances are different among classes and $n > p$ enough observations

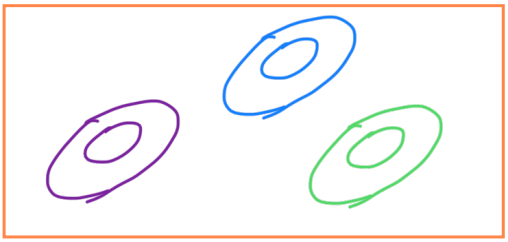
LDA: variances are similar OR NOT enough observations to get variance $n > p$

NB: when p (no of dimensions) is large $> n$ (sample size)

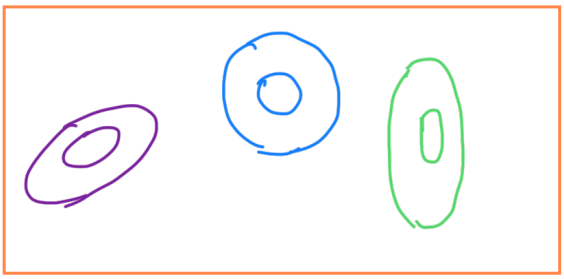
eg. count of words in word vectors / language models

p = dimension = 2

k = classes = 3



LDA distribution better



QDA distribution better

L.5. How to plot LDA, QDA, NB curves ?

Logistic Regression

$\log \frac{p(x)}{1-p(x)} = x^T\beta$ $\quad x = (1, x_1 \cdots x_p)$ $\quad \beta = (\beta_0, \beta_1 \cdots \beta_p)$

$p(x) \in [0,1]$ $\quad x^T\beta \in [-\infty, \infty]$

$$P(Y=1|x) \left[p(x) = \frac{e^{x^T\beta}}{1+e^{x^T\beta}} \right] \qquad P(Y=0|x) \left[1-p(x) = \frac{1}{1+e^{x^T\beta}} \right.$$

Classification Boundary

$x^T\beta = 0$ $\quad$ if $x^T\beta > 0 \Rightarrow C_1$ $\qquad x^T\beta < 0 \Rightarrow C_2$

Q.1. what is outcome of L.R ?

Q.2. when do we use $Y_i = 1 \Rightarrow \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}}$

$Y_i = 0 \Rightarrow \frac{1}{1+e^{x_i^T\beta}}$

DEFINITION Likelihood :

Training data $(x_1, Y_1) \cdots (x_n, Y_n)$

$$\text{argmax}_\beta \prod_{i=1}^{n} p(x_i, y_i) = \text{argmax}_\beta \prod_{i=1}^{n} [p(x_i)\, p(y_i|x_i)]$$

$$= \text{argmax}_\beta \prod [p(y_i|x_i)] = \text{amax}_\beta \ln\left[\prod p(y_i|x_i)\right]$$

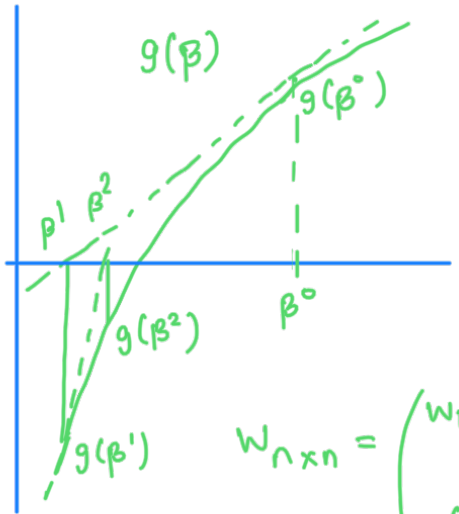
independent of β $\qquad$ one-one monotonic function

$$= \text{amax}_\beta \sum_{i=1}^{n} \ln p(y_i|x_i) = \text{argmax}_\beta \; \ell(\beta)$$

some function of β

How to maximize β_0 : Iterative Newton Raphson method

$\frac{\partial \ell(\beta)}{\partial \beta} = 0 \Rightarrow$ for $p+1$ dimension vector $\quad \frac{\partial \ell(\beta)}{\partial \beta_0} = 0 \quad \frac{\partial \ell(\beta)}{\partial \beta_1} = 0 \cdots \frac{\partial \ell(\beta)}{\partial \beta_p} = 0$



$\ell(\beta)$ is a scalar function

$\beta_{new} = \beta_{old} - g(\beta_{old})/g'(\beta_{old})$

$g(\beta) = \frac{\partial \ell(\beta)}{\partial \beta} = 0$

$\beta_{new} = (X^T W X)^{-1} X^T W z$ $\quad$ repeat until convergence

$$W_{n \times n} = \begin{pmatrix} w_{11} & & & \phi \\ & w_{22} & & \\ & & \ddots & \\ \phi & & & w_{nn} \end{pmatrix} \qquad X = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & & x_{2p} \\ \vdots & \vdots & & & \vdots \\ 1 & x_{n1} & & & x_{np} \end{pmatrix}$$

$W = \text{diag}[p(x_i; \beta_{old})\{1 - p(x_i; \beta_{old})\}]$ $\qquad z = X\beta_{old} + W^{-1}(y - p)$

Comparision with Ordinary least squares

$\min_\beta \sum (z_i - \beta^T x_i)^2 \Rightarrow \hat{\beta} = (X^T X)^{-1} X^T z$

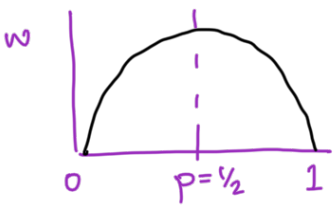
Comparision with Weighted least squares

$\min_\beta \sum w_i (z_i - \beta^T x_i)^2 \Rightarrow \hat{\beta} = (X^T W X)^{-1} X^T W z$

Q.3. What is value of w

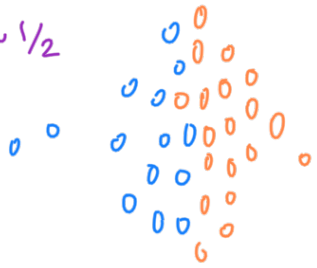
Interpretation of logistic Regression

$w_i = p(x_i)(1 - p(x_i))$



$p \sim 0$ or $p \sim 1$ w: small easy for classification

$p \sim 1/2$ w: large not so easy classification



Regression Error

$\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2$

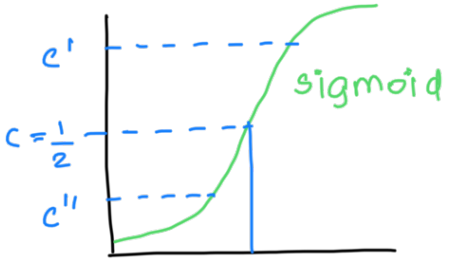
$\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i - \cdots - \hat{\beta}_d x_d)^2$

Classification Error

np.mean (y_hat != y_true)

Q.4. accuracy = ↑

$\frac{1}{n} \sum_{i=1}^{n} 1(\hat{c}(x_i) \neq y_i)$



$$\hat{C}(x) = \begin{cases} 1 & \hat{p}(x) > 0.7 \\ 0 & \hat{p}(x) \leq 0.7 \end{cases}$$

Let $c = 0.7$

fpr, tpr, thresholds ← from roc-curve

Plot ROC and get AUC

upper left points and larger AUC are better results

Logistic

- Linearity by construction
- More general
- Robust to outliers
- weight depends on distance from decision boundary

LDA

- Linearity by consequence
- assumes Gaussian density
- Easier to compute, when $N(\mu, \Sigma)$
- Uses all data points

Multinomial logistic regression

$$p(Y = c_1 | x) = \frac{e^{x^T \beta_1}}{1 + e^{x^T \beta_1} + e^{x^T \beta_2} + \cdots + e^{x^T \beta_{k-1}}}$$

$$p(Y = c_2 | x) = \frac{e^{x^T \beta_2}}{1 + e^{x^T \beta_1} + e^{x^T \beta_2} + \cdots + e^{x^T \beta_{k-1}}}$$

$$p(Y = c_K | x) = \frac{1}{1 + e^{x^T \beta_1} + e^{x^T \beta_2} + \cdots + e^{x^T \beta_{k-1}}}$$

Use training data (x_1, y_1) (x_2, y_2) $\cdots$ (x_n, y_n)

$\Rightarrow \hat{\beta}_1, \hat{\beta}_2, \cdots \hat{\beta}_{k-1}$

Bias- Variance Trade off

Model complexity ↑

kNN as k ↓

QDA > LDA > NB

Logistic Regression > LDA

smooth spline $\lambda \downarrow$

LOOCV > k-fold CV as most training data is used

computationally intense, fit each model n times (×)

5/10 fold CV do not suffer high Bias & high Variance

r.v. $E.(z_1, z_2 \cdots z_k)$ $\quad Var(\bar{z}) = Var\left(\frac{1}{k}\sum z_i\right) = \frac{1}{k^2} Var\left(\sum z_i\right) = \frac{1}{k^2}\left(\sum Var(z_i) + \sum Cov(z_j, z_k)\right)$

kC.V E. $\bar{z} = \frac{1}{k}\sum z_i$

$\therefore$ Var $(\bar{z})$ LOOCV is high z_j and z_k are highly correlated for LOOCV

$$(y - \hat{y})^2 = (\underbrace{y - E[y]}_{a} + \underbrace{E(y) - E[\hat{y}]}_{b} + \underbrace{E[\hat{y}] - \hat{y}}_{c})^2$$

$$= (y - E[y])^2 + (E[y] - E[\hat{y}])^2 + (E[\hat{y}] - \hat{y})^2 + \underbrace{2ab + 2bc + 2ca}_{0}$$

$E[y]$ = constant

$E[c] = c$

$$E[ab] = (E[y] - E[\hat{y}])\, E[y - E[y]]$$

$$= (E[y] - E[\hat{y}])(E[y] - E[y]) = 0$$

$$E[(y_0 - \hat{y}_0)^2] = E[(y_0 - E[y_0])^2] + E[(E[y_0] - E[\hat{y}_0])^2] + E[(E[\hat{y}_0] - \hat{y}_0)^2]$$

$y = f(x) + \xi$

$E[y_0] = E[f(x_0)] + E[\xi]$

$f(x_0)$ = constant wrt E

$E[\xi] = 0$; assumption

$\therefore E[y_0] = f(x_0)$

$y_0 - E[y_0] = y_0 - f(x_0) = \xi$

$E[\xi^2] = Var(\xi) = \sigma^2$; assumption

Var $(\hat{y}_0)$

$= E[(\hat{y}_0 - E[\hat{y}_0])^2]$

$E[(E[y_0] - E[\hat{y}_0])^2]$

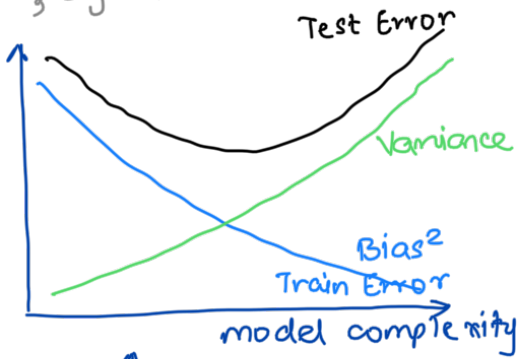
$= (f(x_0) - E[\hat{y}_0])^2$; expectation vanishes

$= Bias(\hat{y}_0)$; by definition

Expected Test error $T.E(x_0) = E[(y_0 - \hat{y}_0)^2]$

$$\therefore T.E.(x_0) = \sigma^2 + bias^2(\hat{y}_0) + var(\hat{y}_0)$$

irreducible error



model complexity ↑ bias² ↓ variance ↑

Assume: Data points $(x_1, y_1) \cdots (x_n\ y_n)$ are iids (R.V.) from a joint distribution $P(X, Y)$

Given:

X : feature vector — predictors → continuous / discrete

Y : class label → Binary / Multiclass

To find: a model / classifier Y as function of X

Goal: minimize error for prediction on unseen "test" data

0-1 Loss:

$$L(y, \hat{y}) = \begin{cases} 1 & \text{if } \hat{y} \neq y \\ 0 & \text{if } \hat{y} = y \end{cases}$$

Symmetric loss, all errors are equal

Sometimes not the best

CROSS-VALIDATION:

1 [1, 2, 20] → mean(y_true != y_hat)

2 [1, 2, 20] → mean(y_true != y_hat)

⋮

10 [1, 2, 20] → mean(y_true != y_hat)

np.mean()

10-fold cross-validation

Data_size = 200

Folders_size = 20

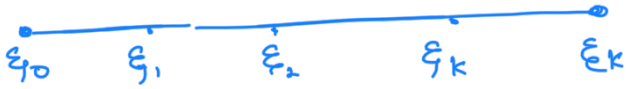
No of observations ↑

No of dimensions ↓

Relation complexity ↑

Model complexity ↑

Splines



$k+1$ knots

Cubic: $ax^3+bx^2+cx+d=0$ — $4(k+1)$ equations

continuity: $S_k(\xi_k) = S_{k+1}(\xi_k)$ — k

First-order cty: $S'_k(\xi_k) = S'_{k+1}(\xi_k)$ — k

Second order cty: $S''_k(\xi_k) = S''_{k+1}(\xi_k)$ — k

$k+4$ equations

Natural cubic: $S''(\xi_0) = S''(\xi_{k+1}) = 0$ — 2

$S'''(\xi_0) = S'''(\xi_{k+1}) = 0$ — 2

k equations

L.6 verify from graph ↑

is this first & second order?

check $TE = var + (bias)^2$

smoothing splines

To predict a function $g(x)$ for the given data:

$$\min_{g(x)} \underbrace{\sum (y_i - g(x_i))^2}_{RSS} + \underbrace{\lambda \int g''(t)^2 dt}_{penalty}$$

1. $g''(t)^2$: Roughness; second order gives curvature
2. squared to get positive outcome
3. $\lambda \geq 0$: tuning parameter

$\lambda = 0$ $\hat{g}(x)$ curve fits all points; over-fitting

$\lambda \to \infty$ $\hat{g}(x)$ becomes smoothest

$\hat{g}''(x)$:. linear regression model

Generalised Additive Models

Advantages

Disadvantages

$$y_i = \beta_0 + f_1(wt_i) + f_2(ht_i) + \cdots + f_p(age_i) + \varepsilon_i$$

Suits for splines with more than one "x" predictors

Non-linear fit but still additive NOT $f(wt, ht, age^2) \cdots$

Easy interpretation of individual predictors

$$\ln\left(\frac{p(y_i = c_1 | x_i)}{1 - p(y_i = c_1 | x_i)}\right) = \beta_0 + \sum_{j=1}^{n} f_j(x_{ij})$$