Naive Bayes Classifier

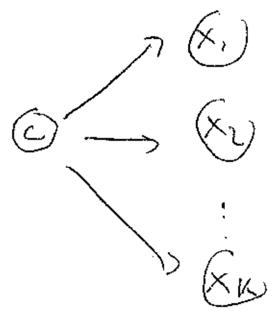
Do this earlier

Have k-tuple observation x which we assign to 1 of C classes: $1 \cdots C$.

x can be real-valued, binary, ...

Naive Bayes (NB) models

$$p(x,c) = p(c)\,p(x|c) \overset{NB}{=} p(c) \prod_{k=1}^{K} p(x_k|c)$$



$\Longleftrightarrow$ $x_1 \cdots x_K$ cond. indep. given c.

Note If we don't assume cond. indep. must learn k-dimen. dist for each class. C.I means we learn K 1-d dists. for each class.

Ex Suppose $x|c \sim \mathcal{N}(\mu_c, \Sigma_c)$ $\quad K^2 + K$ params for each class

vs. $x_k|c \sim \mathcal{N}(\mu_{kc}, \sigma^2_{kc})$ $\quad 2k$ params " " ".

$x_1 \cdots x_K$ cond. indep. $| \; c$

Fewer parameters means more accurate estimation,

Restrictive assump. (if wrong) means less accurate model

This is fundamental tradeoff of ML.

Discrete ~~Binary~~ Features + Naive Bayes

Say $X = \begin{pmatrix} x_1 \\ \vdots \\ x_K \end{pmatrix}$ has $x_k \in \{1, 2, \dots M\}$

Need (according to NB) $p(x_k = m \mid c)$

$k = 1 \dots K$
$m = 1 \dots M$
$c = 1 \dots C$.

Let $\underset{n \times k}{X}$ be data matrix

$$X = \begin{array}{c} \begin{array}{cc} \text{1st feat} & \text{last feat} \\ \downarrow & \downarrow \end{array} \\ \begin{pmatrix} x_{11} & \cdots & x_{1k} \\ x_{21} & \cdots & x_{2k} \\ \vdots & & \\ x_{n1} & \cdots & x_{nk} \end{pmatrix} \end{array} \begin{array}{l} \leftarrow \text{1st obs.} \\ \\ \\ \leftarrow \text{nth obs.} \end{array}$$

$$C = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}$$

How to estimate $p(x_k = m \mid c)$?

Let $n_{kmc} = \left| \{ i : x_{ik} = m, c_i = c \} \right|$

(iii)

(Estimating Probabilities for Discrete Events)

we expt. that has M outcomes $\{1 \dots M\}$ with probs $p_1 \cdots p_M$ $\left(\sum p_m = 1\right)$

Want to estimate $p_1 \cdots p_M$.

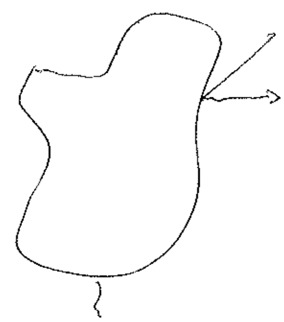
Have n independent observations of expt. $X_1 \cdots X_n$ $\quad X_i \in \{1 \cdots M\}$.

The maximum likelihood est. satisfies

Let $n_m = \left|\{i : x_i = m\}\right|$ $\quad \sum_{m=1}^{M} n_m = n$

$$\hat{p}_1 \cdots \hat{p}_M = \underset{\substack{p_1 \cdots p_M \\ \text{s.t. } \sum p_m = 1}}{\arg\max} \prod_{i=1}^{n} p_{x_i} = \underset{\substack{p_1 \cdots p_M \\ \text{s.t. } \sum p_m = 1}}{\arg\max} \prod_{m=1}^{M} p_m^{n_m}$$

$$= \underset{\substack{p_1 \cdots p_M \\ \text{s.t. } \sum p_m = 1}}{\arg\max} \sum_{m=1}^{M} n_m \log p_m$$



$g(x) = c$

Lagrange multipliers

$$\nabla \sum_{m=1}^{M} n_m \log p_m = \lambda \nabla \sum p_m = 1$$

$$\Longleftrightarrow$$

$$\begin{pmatrix} \frac{n_1}{p_1} \\ \vdots \\ \frac{n_M}{p_M} \end{pmatrix} = \lambda \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \Longrightarrow \quad \begin{array}{l} ① \; p_m = n_m / \lambda \\ \\ ② \; p_m = n_m / n \end{array}$$

(Answer is obvious guess but MLE suggests guess)

Discrete Features + Naive Bayes

Have classification problem with data $\underset{n\times K}{X}$ and class $\underset{n\times 1}{c}$

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1K} \\ x_{21} & x_{22} & \cdots & x_{2K} \\ \vdots & & & \\ x_{n1} & x_{n2} & \cdots & x_{nK} \end{pmatrix} \begin{array}{l} \leftarrow \text{obs } 1 \\ \\ \\ \leftarrow \text{obs } n \end{array} \qquad c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}$$

$$\begin{array}{cccc} \uparrow & & & \uparrow \\ \text{feat } 1 & & & \text{feat } k \end{array}$$

$$c_i \in \{1 \cdots C\} \qquad x_{ik} \in \{1, \cdots M\}$$

NB requires we have $\hat{p}(x_k = m \mid c)$ $\quad k = 1 \cdots K$, $m = 1 \cdots M$, $c = 1 \cdots C$

Let $n_{kmc} = \left| \{ i : x_{ik} = m,\ c_i = c \} \right|$ = # times get class c with k^{th} feat = m.

By prev. argument

$$\hat{p}(x_k = m \mid c) = \frac{n_{kmc}}{\sum_{m'=1}^{M} n_{km'c}}$$

And also

$$\hat{p}(c) = \frac{\left| \{ i : c_i = c \} \right|}{n}$$

Review

$$\sigma(t) = \frac{1}{1+e^{-t}} \qquad \text{Model} \quad p(c=1 \mid x) = \sigma(w^t x)$$

Have data vectors $x_1, x_2, \dots, x_n$. We know

$$\frac{\partial \log p(c_1 \dots c_n \mid x_1 \dots x_n)}{\partial w_j} = \sum_{i=1}^{n} \left(c_i - \sigma(w^t x_i) \right) x_{ij} \iff \nabla \log p(c_1 \dots c_n \mid x_1 \dots x_n) = \sum \left(c_i - \sigma(w^t x_i) \right) x_i$$

Writing

$$\underset{n \times k}{X} = \begin{pmatrix} x_{11} & \cdots & x_{1k} \\ x_{21} & \cdots & x_{2k} \\ \vdots & & \\ x_{n1} & \cdots & x_{nk} \end{pmatrix} \qquad c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} \qquad p = \begin{pmatrix} \sigma(w^t x_1) \\ \vdots \\ \sigma(w^t x_n) \end{pmatrix}$$

$$\nabla \log p(c_1 \dots c_n \mid x_1 \dots x_n) = X^t (c - p)$$

$$\frac{\partial^2 \log p(c_1 \dots c_n \mid x_1 \dots x_n)}{\partial w_j \partial w_l} = -\sum_{i=1}^{n} \sigma(w^t x_i)\left(1 - \sigma(w^t x_i)\right) x_{ij} x_{il}$$

$$= -\left(X^t C X \right)_{jl}$$

$$C = \begin{pmatrix} \sigma(w^t x_1)(1-\sigma(w^t x_1)) & & 0 \\ & \ddots & \\ 0 & & \sigma(w^t x_n)(1-\sigma(w^t x_n)) \end{pmatrix} \qquad \text{i.e. } -X^t C X \text{ is Hessian}$$

NR: $$w^{new} = w^{old} + H^{-1} \nabla f(w^{old}) = w^{old} + (X^t C X)^{-1} X^t (c - p)$$

[NB. C and p depend on w^{old} so must be recomputed each iter

Regression

In classification data are labeled samples:

$$(x_1, c_1),\ (x_2, c_2) \ldots (x_n, c_n)$$

where $\{x_i\}$ are vectors usually $\in \mathbb{R}^d$ and $c_i \in \underbrace{\{1, \ldots C\}}_{\text{our classes}}$

In regression have $(x_1, y_1) \ldots (x_n, y_n)$ where

$x_i \in \mathbb{R}^d$ (as before) but $y_i \in \mathbb{R}$ (or $\mathbb{R}^k$)

The $\{x_i\}$ are the predictors and the $\{y_i\}$ are the response.

Formulation with Loss Function

Let $\hat{y} = \hat{y}(x)$ be prediction of y based on x

Suppose we adopt loss function $L(y, \hat{y}) = (y - \hat{y})^2$

~~meaning~~ giving "cost" for estimating $\hat{y}$ when truth is y.

Observe x and seek $\hat{y}$ that minimizes expected loss

$$E\,L = \int (y - \hat{y})^2\, p(y|x)\, dy$$

Know $\hat{y} = E(y|x) = \int y\, p(y|x)\, dy$.

so $\hat{y}(x) = E(y|x)$ is obvious choice for regression.

$\hat{y}(x) = E(y|x)$ is known as regression function

Linear Regression

In linear regression predict y as linear fn of x

$$\hat{y} = \hat{y}(x) = w^t x = w_1 x_1 + w_2 x_2 + \cdots w_k x_k .$$

Ex

$$x = \overset{x_1}{\text{amount of education}},\ \overset{x_2}{\text{salary of parents}},\ \overset{x_3}{\text{age}}$$

y = salary of individual

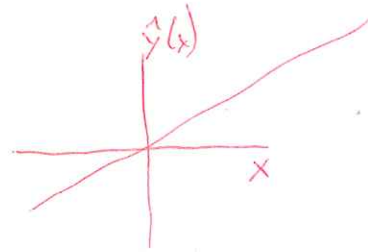
Estimate $\hat{y} = w_1 x_1 + w_2 x_2 + w_3 x_3 = w^t x$.

In linear regression common to augment observed predictors with other variables derived from observations.

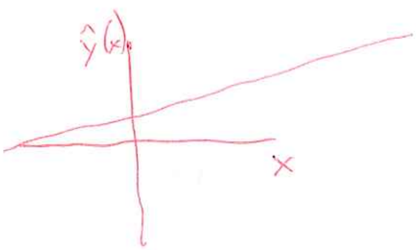
Ex

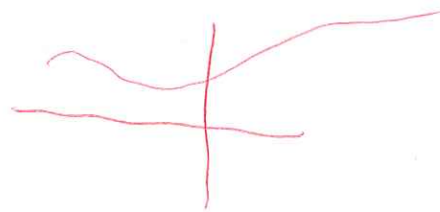
Suppose data look like

Straight linear regression requires $\hat{y}(x) = wx$ (line through origin)



But could view predictors as $1, x$ so $\hat{y}(x) = w_0 1 + w_1 x$





or perhaps $1, x, x^2, x^3 \cdots x^k$ so $\hat{y}(x) = w_0 1 + w_1 x + w_2 x^2 + \cdots + w_k x^k$

This is still linear regression!

Geometric View of Regression

Have data $(x_1, y_1) \cdots (x_n, y_n)$ $\quad x_i \in \mathbb{R}^k$, $y_i \in \mathbb{R}$.

x_i includes whatever features we derive from obs including 1.

Use linear prediction: $\hat{y}_i = w^t x_i$ and want to minimize sum of squared errors (SSE) between $\hat{y}_i$ and y_i

$$\hat{w} = \arg\min_w \sum_{i=1}^{n} (y_i - w^t x_i)^2$$

Another View

Let

$$X = \begin{array}{r c} & \begin{array}{cccc} \text{1st feat} & & & \text{kth feat.} \end{array} \\ \begin{array}{r} \text{1st obs} \rightarrow \\ \\ \\ \text{nth obs} \rightarrow \end{array} & \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & & & \\ x_{n1} & x_{n2} & \cdots & x_{nk} \end{pmatrix} \end{array} \qquad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

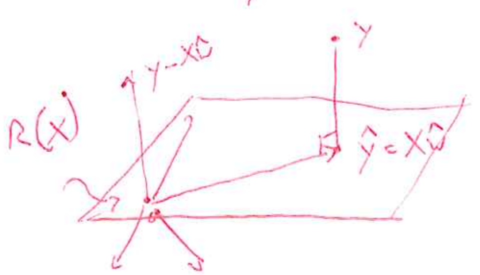
X is data matrix.

$$\hat{w} = \arg\min_w \sum_{i=1}^{n} \left(y_i - (Xw)_i\right)^2 = \arg\min_w \|y - Xw\|^2$$

Geometric Picture

Write $R(X) = \{Xw : w \in \mathbb{R}^k\}$ = range of X.

Seek $\hat{y} \in R(X)$ s.t. $\|y - \hat{y}\|^2$ minimized.



Pictorially $y - \hat{y}$ should be orthogonal to $R(X)$

$$(Xw, y - X\hat{w}) = 0 \quad \forall w$$

$$\Longleftrightarrow$$

$$(w, X^t(y - X\hat{w})) = 0 \quad \forall w \Longleftrightarrow X^t(y - X\hat{w}) = 0$$

$$\Uparrow$$

$$X^t X \hat{w} = X^t y \quad \text{(normal eqns)}$$

$$\Uparrow$$

$$\hat{w} = (X^t X)^{-1} X^t y.$$

Easy to remember version

Would like to solve $Xw = y$

Think of

$$\underset{n \times k}{\begin{pmatrix} \\ \\ X \\ \\ \\ \end{pmatrix}} \underset{k \times 1}{\begin{pmatrix} w \end{pmatrix}} = \underset{n \times 1}{\begin{pmatrix} \\ \\ y \\ \\ \\ \end{pmatrix}}$$

Could have many (n) ='ns, but few (k) unknowns so can't solve this usually.

But, can solve $X^t X w = X^t y$ (Normal ='ns)

Ex Simple_linear_regression.r

Projection

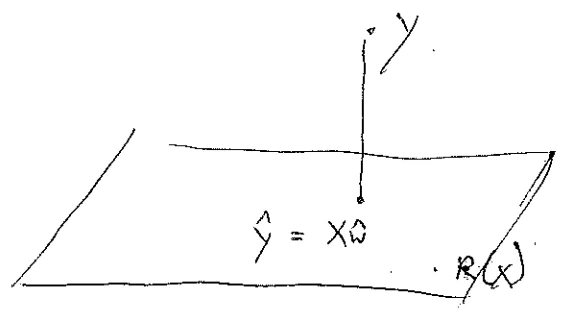
We saw

$\hat{w} = \arg\min_w \| y - Xw \|$

given by

① $X^t X \hat{w} = X^t y$ (Normal ='ns)

② $\hat{w} = (X^t X)^{-1} X^t y$



$\hat{y} = X\hat{w}$ is "projection" of y onto $R(X)$.

Projection Matrices

If L is a linear space (eg plane or line containing origin) the "projection" of y onto L $P_L y$ is closest point in L to y.

A projection matrix P has

① $P^2 = P$ (idempotent)

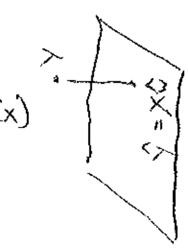
② $P^t = P$ (symmetric)

Easy to see $X(X^tX)^{-1}X^t$ satisfies ① + ②

① $\left(X(X^tX)^{-1}X^t\right)\left(X(X^tX)^{-1}X^t\right) = X(X^tX)^{-1}X^t$

② $\left(X(X^tX)^{-1}X^t\right)^t = X\left((X^tX)^{-1}\right)^t X^t = X(X^tX)^{-1}X^t$

Since $R\left(X(X^tX)^{-1}X^t\right) = R(X)$ $\quad X(X^tX)^{-1}X^t = P_{R(X)}$



Thm of Pythagoras for proj

Let P be projection. For any y

$$y = Py + (I-P)y \implies$$

$$\|y\|^2 = \|Py + (I-P)y\|^2 = \left(Py + (I-P)y,\ Py + (I-P)y\right)$$

$$= \|Py\|^2 + \|(I-P)y\|^2$$

Note $(Py, (I-P)y) = (y, P(I-P)y) = (y, 0y) = 0$

$\|y\|^2 = \|\hat{y}\|^2 + \|y - \hat{y}\|^2$

For our special case of $\hat{y} = X\hat{\beta}$

Statistical View of Regression

~~Model~~ Data $(x_1, y_1) \dots (x_n, y_n)$ $\quad x_i \in \mathbb{R}^k, \; y_i \in \mathbb{R}$

Model

$$y_i = w^t x_i + \varepsilon_i \qquad i = 1 \dots n$$

where ~~w~~ w is vector of unknowns $w \in \mathbb{R}^k$

$\varepsilon_1 \dots \varepsilon_n \overset{iid}{\sim} N(0, \sigma^2)$ $\quad \sigma^2$ unknown. $\quad \left[w, \sigma^2 \text{ parameters} \right]$

Equivalently

$$\underset{n \times k}{X} = \begin{pmatrix} x_{11} & \cdots & x_{1k} \\ x_{21} & \cdots & x_{2k} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{nk} \end{pmatrix} \qquad Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \qquad w = \begin{pmatrix} w_1 \\ \vdots \\ w_k \end{pmatrix} \qquad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

$$Y = Xw + \varepsilon \qquad \varepsilon \sim N(0, \sigma^2 I)$$

Defn

An estimator, $\hat{\theta}$, for param θ is unbiased if $E\hat{\theta} = \theta$ (on average estimate is correct)

Ex If $X_1 \dots X_n$ are sequence of indep. S-F trials

$P(X_i = 1) = p$

$P(X_i = 0) = 1 - p$.

We saw the MLE for p was $\hat{p} = \frac{1}{n} \sum_{i=1}^{n} X_i$.

Easy to see $\hat{p}$ is unbiased for p.

$E X_i = 1p + 0(1-p) = p$

$E\hat{p} = E \frac{1}{n} \sum_{i=1}^{n} X_i = \frac{1}{n} \sum_{i=1}^{n} E X_i = \frac{1}{n} \sum_{i=1}^{n} p = p$

We have used $\hat{w} = (X^t X)^{-1} X^t y$ as estimate for w.

In fact $\hat{w}$ is unbiased for w.

Have

$$y = Xw + \varepsilon \qquad \varepsilon \sim N(0, \sigma^2 I)$$

$$E\hat{w} = E(X^tX)^{-1}X^t y = E(X^tX)^{-1}X^t(Xw+\varepsilon)$$

$$= E(X^tX)^{-1}X^tXw + E(X^tX)^{-1}X^t\varepsilon$$

$$= w + (X^tX)^{-1}X^t \underbrace{E\varepsilon}_{0}$$

$$= w \implies \hat{w} \text{ unbiased for } w.$$

Variance of Unbiased Estimator

If $\hat{\theta}$ is unbiased for θ, then

$$V(\hat{\theta}) = E(\hat{\theta} - \theta)^2 = \text{expected sq. error},$$

If $\hat{\theta}_1$ and $\hat{\theta}_2$ are UE's for θ and $V(\theta_1) < V(\theta_2)$

then $\hat{\theta}_1$ has less sq error on average, and is better in this regard.

This variance often used as goodness meas. for UEs.

Gauss-Markov Thm

Informally, $\hat{w} = (X^tX)^{-1}X^t y$ has smallest variance (sq. error) of all UE's.

More precisely, let $\tilde{w}$ be a U.E. for w, and $\alpha \in \mathbb{R}^k$. Then

$$V(\alpha^t \hat{w}) \leq V(\alpha^t \tilde{w})$$

For ex, if $\alpha^t = (0 \cdots \overset{i\text{th}}{1} \cdots 0)$ this says $V(\hat{w}_i) \leq V(\tilde{w}_i)$

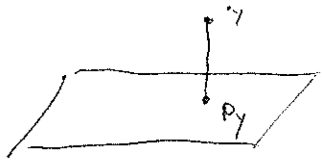
Estimating σ^2

① If $X \sim N(0, \sigma^2)$ $EX^2 = \int x^2 N(x; 0, \sigma^2)\, dx = \sigma^2$

If $\underset{n\times 1}{\varepsilon} \sim N(0, \sigma^2 \underset{n\times n}{I})$

$$E\|\varepsilon^2\|^2 = E\left(\varepsilon_1^2 + \varepsilon_2^2 + \cdots + \varepsilon_n^2\right) = n\sigma^2$$

② Fact Suppose $\underset{n\times n}{P}$ is projection matrix proj onto k-dim space

Eg 

P projects onto 2-d space

Then $E\|P\varepsilon\|^2 = k\sigma^2$

Since $\|\varepsilon\|^2 = \|P\varepsilon\|^2 + \|(I-P)\varepsilon\|^2$

Taking expectations gives:

$$n\sigma^2 = k\sigma^2 + E\|(I-P)\varepsilon\|^2$$

$$\Longrightarrow E\|(I-P)\varepsilon\|^2 = (n-k)\sigma^2$$

③ In regression model $y = Xw + \varepsilon$ let $P_{R(X)}$ be proj onto $R(X)$

$$E\|y - \hat{y}\|^2 = E\|(I - P_{R(X)})\, y\|^2 = E\|(I - P_{R(X)})\, Xw + \varepsilon\|^2$$

$$= E\|(I - P_{R(X)})\,\varepsilon\|^2 = (n-k)\sigma^2$$

$$\Longrightarrow E\, \frac{\|y - \hat{y}\|^2}{n-k} = \sigma^2$$

We will use $\hat{\sigma}^2 = \dfrac{\|y-\hat{y}\|^2}{n-k}$ as our (unbiased) estimate for σ^2.

Do regression_variance.r

Overfitting

Have data $(x_1, y_1) \cdots (x_n, y_n)$ $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$

Ex Want to predict price of Apple stock on particular day.

Choose relevant predictors

① Overall consumer spending

② Advertising expenditures of Apple

③ Investor confidence in tech sector

④ Price of labor in China

$$X = \begin{array}{c|c} & \text{var } 1 \;\cdots\; \text{var } d \\ \begin{array}{c}\text{day } 1\\ \vdots \\ \text{day } n\end{array} & \left(\begin{array}{c} \\ \\ \\ \end{array}\right) \end{array}$$

Measure variables and get $SSE = \|y - \hat{y}\|^2 = \|y - X\hat{w}\|^2$

SSE doesn't seem small enough (predictions, $\hat{y}$, not close to y)

so add new predictors

ⓐ Rainfall in Ecuador on each day

ⓑ Price of Barley futures

ⓒ Dist between closest pair of Jupiter's moons

⋮ ⋮

$$X = \begin{array}{c|c} & \text{Vars} \\ & 1 \;\, 2 \cdots d, a, b, c, \cdots \\ \begin{array}{c}\text{day } 1\\ \vdots \\ \text{day } n\end{array} & \left(\begin{array}{c} \\ \\ \\ \end{array}\right) \end{array}$$

Since new predictors are independent of Apple's stock price, so shouldn't help.

But SSE continually decreases as irrelevant predictors added.

This seems like good news, but with new data $(x_{n+1}, y_{n+1}) \cdots (x_{2n}, y_{2n})$

learned model predicts poorly. That is

X_T, y_T are original data, $\hat{w}_T$ learned weights $\hat{w}_T = (X_T^t X_T)^{-1} X_T^t y_T$

X_N, y_N new data $SSE_N = \|y_N - X_N \hat{w}_T\|^2$ is high. This is known as overfitting

Do overfit_regression.c

Ways to Avoid Overfitting

① Variable selection: Use only subset of vbles.

Use notation

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1d} \\ x_{21} & \cdots & x_{2d} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{nd} \end{pmatrix} \qquad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

$$X(j) = j^{th} \text{ col of } X \qquad \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{pmatrix}$$

$$X(j_1, \cancel{\ } j_2) = \text{cols } j_1, j_2 \qquad \underset{n \times 2}{\begin{pmatrix} x_{1j_1} & x_{1j_2} \\ \vdots & \vdots \\ x_{nj_1} & x_{nj_2} \end{pmatrix}}$$

$$X(j_1, j_2, j_3) = \text{cols } j_1, j_2, j_3 \text{, etc.}$$

Let j_1 be best single predictor.

$$j_1 = \arg\min_j \| y - X(j) \cancel{\ } (X^t(j) X(j))^{-1} X^t(j) y \|^2$$

Chose 2^d predictor, j_2 to be best vble in addition to j_1

$$j_2 = \arg\min_j \| y - X(j_1, j) (X^t(j_1, j) X(j_1, j))^{-1} X^t(j_1, j) y \|^2$$

$$j_3 = \arg\min_j \| y - X(j_1, j_2, j) (X^t(j_1, j_2, j) X(j_1, j_2, j))^{-1} X^t(j_1, j_2, j) y \|^2 \ \ldots$$

Let $SSE_0 = \|y\|^2$

$SSE_1 = \| y - X(j_1) (X^t(j_1) X(j_1))^{-1} X(j_1) y \|^2$

$SSE_2 = \| y - X(j_1, j_2) (X^t(j_1, j_2) X(j_1, j_2))^{-1} X^t(j_1, j_2) y \|^2$

$\ldots$

Can show that under non-pathological conds.

$SSE_0 > SSE_1 > SSE_2 \ldots$

How far should we go?

Add vbles as long as

$SSE_{k-1} - SSE_k > \text{threshold}.$

Ridge Regression

Initial formulation of regression: $\hat{w} = \arg\min_w \|y - Xw\|^2$.

Have seen this formulation is prone to overfitting ~~with~~ with many predictors.

Ridge regression penalizes complex fits of the data:

$$\hat{w}_{Ridge} = \arg\min_w \underbrace{\|y - Xw\|^2}_{\text{data fit}} + \underbrace{\lambda \|w\|^2}_{\text{complexity penalty}}. \qquad \lambda > 0$$

Can show $\hat{w}_{Ridge} = (X^t X + \lambda I)^{-1} X^t y$

When $\lambda = 0$ get old solution $\hat{w}$. As λ increases $\hat{w}_{Ridge}$ "shrinks" to 0.

Note Ridge Regression useful when $\underbrace{\underset{k\times n}{X^t}\underset{n\times k}{X}}_{k\times k}$ is singular (eg. $k > n$)

since $(X^t X + \lambda I)$ always invertible.

How to Choose λ? (Cross validation)

Suppose we divide data into "training set" and "validation set"

$$X = \begin{pmatrix} X_{11} & X_{12} & \cdots & X_{1k} \\ X_{21} & X_{22} & \cdots & X_{2k} \\ \vdots & & & \\ \hline \vdots & & & \\ X_{n1} & X_{n2} & \cdots & X_{nk} \end{pmatrix} \begin{matrix} X_T \\ \\ \\ X_V \end{matrix} \qquad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ \hline \vdots \\ y_n \end{pmatrix} \begin{matrix} y_T \\ \\ y_V \end{matrix} = \begin{pmatrix} y_T \\ y_V \end{pmatrix}$$

$$= \begin{pmatrix} X_T \\ X_V \end{pmatrix} \qquad \hat{w}_T(\lambda) = \arg\min_w \|y - X_T w\|^2 + \lambda \|w\|^2$$

$$= (X_T^t X_T + \lambda I)^{-1} X_T^t y$$

$\Rightarrow$ Choose λ giving best performance on validation set.

$$\hat{\lambda} = \arg\min_\lambda \|y_V - X_V \hat{w}_T(\lambda)\|^2$$

LASSO Regression

$$\hat{w}_{LASSO} = \arg\min_{w} \|y - Xw\|^2 + \lambda \|w\|$$ no square.

① As before $\lambda = 0$ gives $\hat{w} = (X^t X)^{-1} X^t y$.

② As λ increases some components of $\hat{w}_{LASSO}$ driven to 0.

thus LASSO acts like variable selection while λ controls # of vbles.

③ Computation of $\hat{w}_{LASSO}$ more complicated + won't discuss.

Gaussian Mixtures + EM Algorithm

Mixtures

① Chose randomly from K classes $\{1 \dots K\}$ with probs $\pi_1 \dots \pi_K$.

② If chose k^{th} class sample x from $x \sim p_k(x)$

$$p(x) = \cancel{\prod_{k=1}^{K} \pi_k} \sum_{k=1}^{K} \pi_k \, p_k(x)$$

For Gaussian Mixture Model (GMM) class cond. dists $(p_k(x))$ are multivariate normal

$$p(x) = \sum_{k=1}^{K} \pi_k \, N(x; \mu_k, \Sigma_k) = \sum_{k=1}^{K} \pi_k \frac{1}{(2\pi)^{d/2} |\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)\Sigma_k^{-1}(x-\mu_k)}$$

Suppose have data:

$$\underset{n \times d}{X} = \begin{pmatrix} x_{11} & \cdots & x_{1d} \\ x_{21} & \cdots & x_{2d} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{nd} \end{pmatrix}$$

Want to estimate GMM $\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$

Intuition

Let $c_i \in \{1 \dots K\}$ be class x_i comes from. (unknown)

If the $\{c_i\}$ known then let

$$n_k = |\{i : c_i = k\}| \; ; \quad \hat{\mu}_k = \frac{1}{n_k} \sum_{i=1}^{n} x_i \, I_{c_i = k} \; ;$$

$$\hat{\Sigma}_k = \frac{1}{n_k} \sum_{i=1}^{n} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^t \, I_{c_i = k}$$

where $I_{c_i = k} = \begin{cases} 1 & \text{if } c_i = k \\ 0 & \text{o.w.} \end{cases}$

But we don't know $\{c_i\}$. Do know

$$p(c_i = k \mid x_i) \overset{\text{Bayes}}{=} \frac{\pi_k \, N(x_i ; \mu_k, \Sigma_k)}{\sum_{k'=1}^{K} \pi_{k'} \, N(x_i ; \mu_{k'}, \Sigma_{k'})} \doteq \gamma_{ik}$$

γ_{ik} is "responsibility" of k^{th} class for ith sample

$$\sum_{k=1}^{K} \gamma_{ik} = 1 \qquad n = \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{ik} = \sum_{k=1}^{K} \underbrace{\left(\sum_{i=1}^{n} \gamma_{ik} \right)}_{\doteq n_k}$$

$$\cancel{\sum_{i=1}^{n} \gamma_{ik} \doteq n_k} \qquad = \sum_{k=1}^{K} n_k$$

n_k is # samples attributed to k^{th} class (not an integer)

Idea: treat x_i as γ_{i1} of a samples from class 1

γ_{i2} " " " 2

$\vdots$

γ_{iK} " " " K

$$\hat{\pi}_k = \frac{n_k}{n}$$

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ik} x_i$$

$$\hat{\Sigma}_k = \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ik} (x_i - \mu_k)(x_i - \mu_k)^t$$

<u>Algorithm</u>

1. Init $\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$
2. Compute $\gamma_{ik} = p(c_i = k \mid x_i)$ and $n_k = \sum_{i=1}^{n} \gamma_{ik}$
3. (Re)Estimate $\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ from updates.
4. Go to 2 until convergence.

Look at gmm.em.r

Revisiting GMM Algorithm Viewed as Max Likelihood

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1d} \\ x_{21} & \cdots & x_{2d} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{nd} \end{pmatrix}$$

Want to fit GMM to X with K-mixture components.

Have

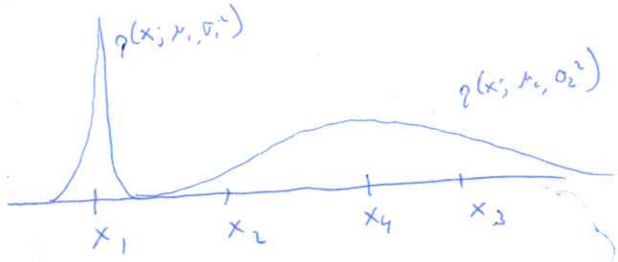
$$p(X; \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k N(x_{i\cdot}; \mu_k, \Sigma_k)$$

Want to estimate params $\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ by MLE.

Aside

Problem not well-posed since degenerate solutions give arbitrarily-high likelihood.

Consider 1-d case with $K = 2$



If we take $\mu_1 = x$ and let $\sigma_1^2 \downarrow 0$ then

$$p(x_1; \mu_1, \sigma_1^2) = \frac{1}{(2\pi\sigma_1^2)^{1/2}} e^{-\frac{1}{2\sigma_1^2}(x_1-\mu_1)^2} = \frac{1}{(2\pi\sigma_1^2)^{1/2}}$$

so $p(x_1; \mu_1, \sigma_1^2) \underset{\sigma_1^2 \downarrow 0}{\longrightarrow} \infty$

Thus we seek non-degenerate solutions.

Now

$$\log p(X; \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i; \mu_k, \Sigma_k)$$

Could optimize by differentiating wrt. μ_k and setting to 0

$$0 = \nabla_{\mu_k} \log p(X; \{\pi_{k'}, \mu_{k'}, \Sigma_{k'}\}_{k'=1}^{K})$$

$$= \sum_{i=1}^{n} \frac{\pi_k \mathcal{N}(x_i; \mu_k, \Sigma_k) \Sigma_k^{-1} \cancel{(x_i-\mu_k)} (\mu_k - x_i)}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}(x_i; \mu_{k'}, \Sigma_{k'})}$$

$$= \sum_{i=1}^{n} \gamma_{ik} \Sigma_k^{-1} (\mu_k - x_i)$$

$$= \Sigma_k^{-1} \sum_{i=1}^{n} \gamma_{ik} (\mu_k - x_i)$$

$$\Longrightarrow \sum_{i=1}^{n} \gamma_{ik} (\mu_k - x_i) = 0$$

$$\Longrightarrow \sum_{i=1}^{n} \gamma_{ik} x_i = \sum_{i=1}^{n} \gamma_{ik} \mu_k = n_k \mu_k$$

$$\Longrightarrow \hat{\mu}_k = \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ik} x_i \qquad \text{(as before)}$$

Can also show, differentiating wrt Σ_k we get

$$\hat{\Sigma}_k = \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ik} (x_i - \hat{\mu}_k)(x_i - \mu_k)^t \qquad \text{(as before)}$$

To estimate $\{\pi_k\}$ since $\sum \pi_k = 1$ need Lagrange:

$$\nabla_{\pi} \log p(X; \{\pi_k, \mu_k, \Sigma_k\}) = \lambda \nabla_{\pi} \sum_{k=1}^{K} \pi_k$$

$$\Longrightarrow \sum_{i=1}^{n} \frac{\mathcal{N}(x_i; \mu_k, \Sigma_k)}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}(x_i; \mu_{k'}, \Sigma_{k'})} = \lambda \qquad k = 1 \cdots K$$

Multiplying both sides by π_k and summing over k gives

$$\sum_{k=1}^{K} \sum_{i=1}^{n} \gamma_{ik} = n = \sum_{k=1}^{K} \pi_k \lambda = \lambda$$

$$\Longrightarrow \pi_k = \frac{1}{n} \sum_{i=1}^{n} \gamma_{ik} = \frac{n_k}{n}$$

Thus same algorithm emerges as MLE.

The EM Algorithm

Algorithm for estimating GMM params is example of EM (Expectation - Maximization) Algorithm.

EM

Data (X, Y) — $\underbrace{(X,Y)}_{\text{complete data.}}$

X = observable (incomplete) data

Y = unobservable data

Ex

$$X \sim p(x) = \sum_{k=1}^{K} \pi_k \, N(x; \mu_k, \Sigma_k) = \text{GMM}$$

X = observed data

Y = which Gaussian $\in \{1 \cdots K\}$ generated X.

EM assumes that

① Have $X \sim p(x|\theta) = \sum_y p(x,y|\theta)$

② Want to estimate θ by MLE.

③ Given complete data (X, Y) easy to compute MLE.

Given X and current θ est, θ^{old}

$p(Y|X, \theta^{old})$ is dist on unobserved given observed

$$Q(\theta, \theta^{old}) = E_{\theta^{old}} \log p(X, Y|\theta) = \sum_y \log p(X, Y=y|\theta) \, p(Y=y|X, \theta^{old})$$

$$\theta^{new} = \arg\max_{\theta} Q(\theta, \theta^{old})$$

EM Produces sequence $\theta^1, \theta^2, \ldots$

Can show

① $p(X|\theta^1) \le p(X|\theta^2) \le \cdots$ with equality iff θ^n is local max.

② Update eqns often tractable.

EM Ex (GMM Revisited)

① $y \in \{1, \dots K\}$ $\quad p(y=k) = \pi_k$ $\qquad p(x) = \sum_{k=1}^{K} \pi_k \, N(x; \mu_k, \Sigma_k)$

② $x \sim N(\mu_k, \Sigma_k)$

$$\underset{\text{observed}}{X} = \begin{pmatrix} \cdots & x_1 & \cdots \\ \cdots & x_2 & \cdots \\ & \vdots & \\ \cdots & x_n & \cdots \end{pmatrix} \qquad \underset{\text{unobserved}}{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

Have random sample $(x_1, y_1) \cdots (x_n, y_n)$ but don't observe y's.

$$\theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$$

$$\log p(X, Y \mid \theta) = \log \prod_{i=1}^{n} p(x_i, y_i \mid \theta) = \sum_{i=1}^{n} \log p(x_i, y_i \mid \theta)$$

Given θ^{old}

$$p(y_i = k \mid x_i, \theta^{old}) = \frac{\pi_k \, N(x_i; \mu_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \, N(x_i, \mu_{k'}, \Sigma_{k'})} = \gamma_{ik}$$

$$\begin{aligned} E_{\theta^{old}} \log p(X, Y \mid \theta) &= E_{\theta^{old}} \sum_{i=1}^{n} \log p(x_i, y_i \mid \theta) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} \log p(x_i, y_i = k \mid \theta) \, p(y_i = k \mid x_i, \theta^{old}) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} \log p(x_i, y_i = k \mid \theta) \, \gamma_{ik} \end{aligned}$$

$$\begin{aligned} \{\pi_k^{new}, \mu_k^{new}, \Sigma_k^{new}\}_{k=1}^{K} &= \underset{\{\pi_k, \mu_k, \Sigma_k\}}{\arg\max} \sum_{i=1}^{n} \sum_{k=1}^{K} \log \pi_k \, N(x_i; \mu_k, \Sigma_k) \, \gamma_{ik} \\ &= \underset{\{\pi_k, \mu_k, \Sigma_k\}}{\arg\max} \sum_{i=1}^{n} \sum_{k=1}^{K} \left(\log \pi_k + \log N(x_i; \mu_k, \Sigma_k) \right) \gamma_{ik} \end{aligned}$$

Note: Can max indep over $\{\pi_k\}_{k=1}^{K}$ and $\{\mu_k, \Sigma_k\}_{k=1}^{K}$

① $$\Pi^{new} = \arg\max_{\substack{\Pi \\ \sum \pi_k = 1}} \sum_{i=1}^{n} \sum_{k=1}^{K} \log \pi_k \, \gamma_{ik} = \arg\max_{\substack{\Pi \\ \sum \pi_k = 1}} \sum_{k=1}^{K} \log \pi_k \, n_k$$

~~By Lagrange set~~ $\cancel{\nabla_\pi \sum_{i=1}^{n} \sum_{k=1}^{K} \log \pi_k \gamma_{ik} = \lambda \nabla_\pi \sum \pi_k}$

By Lagrange set $$\nabla_\pi \sum_{k=1}^{K} \log \pi_k \, n_k = \lambda \nabla_\pi \sum_{k=1}^{K} \pi_k$$

$$\Rightarrow \quad \frac{n_k}{\pi_k} = \lambda \mathbb{1} \quad \Longleftrightarrow \quad \sum_k n_k = \sum_k \lambda \pi_k$$

$$n = \lambda$$

$$\Rightarrow \quad \pi_k = \frac{n_k}{n}$$

② $$\mu^{new} = \arg\max_{\mu} \sum_{i=1}^{n} \sum_{k=1}^{K} \log \mathcal{N}(x_i; \mu_k, \Sigma_k) \, \gamma_{ik}$$

$$= \arg\max_{\mu} \sum_{k=1}^{K} \sum_{i=1}^{n} \log \mathcal{N}(x_i; \mu_k, \Sigma_k) \, \gamma_{ik}$$

Can max inner sum for each μ_k $\Rightarrow$

$$\mu_k^{new} = \arg\max_{\mu_k} \sum_{i=1}^{n} \log \mathcal{N}(x_i; \mu_k, \Sigma_k) \, \gamma_{ik}$$

Setting $\nabla_{\mu_k} = 0$ gives

$$0 = \sum_{i=1}^{n} \frac{\cancel{\mathcal{N}(x_i; \mu_k, \Sigma_k)}^{\,1} \, \Sigma_k^{-1} (x_i - \mu_k) \, \gamma_{ik}}{\cancel{\mathcal{N}(x_i; \mu_k, \Sigma_k)}}$$

$$= \cancel{\Sigma_k^{-1}} \sum_{i=1}^{n} (x_i - \mu_k) \, \gamma_{ik}$$

$$\Rightarrow \sum_{i=1}^{n} x_i \gamma_{ik} = \sum_{i=1}^{n} \mu_k \gamma_{ik} = \mu_k n_k \quad \Rightarrow \quad \mu_k^{new} = \frac{1}{n_k} \sum_{i=1}^{n} x_i \gamma_{ik}$$

em_faithful.r

③ Can show $$\Sigma_k^{new} = \frac{1}{n_k} \sum_{i=1}^{n} (x_i - \mu_k^{new})(x_i - \mu_k^{new})^t \gamma_{ik}$$

Do em_faithful.r

Multiclass Logistic Regression

Softmax

$$p(k|x) = \frac{e^{w_k^t \phi(x)}}{\sum_{k'} e^{w_{k'}^t \cdot \phi(x)}} = \text{softmax fn.}$$

Many classification problems lead to posteriors having softmax formulation

Ex Consider generative model

$$P(C=k) = \pi_k \qquad k = 1 \dots K$$

$$X | C=k \sim \text{Binomial}(n, p_k)$$

$$p(C=k|x) = \frac{\pi_k \cancel{\binom{n}{x}} p_k^x (1-p_k)^{n-x}}{\sum_{k'} \pi_{k'} \cancel{\binom{n}{x}} p_{k'}^x (1-p_{k'})^{n-x}} = \frac{f(k,x)}{\sum_{k'} f(k',x)}$$

$$\cancel{\frac{\left(\frac{p_k}{1-p_k}\right)^x \pi_k (1-p_k)^n}{\sum_{k'}}}$$

$$f(k,x) = \pi_k p_k^x (1-p_k)^{n-x}$$

$$= \pi_k \left(\frac{p_k}{1-p_k}\right)^x (1-p_k)^n$$

$$= e^{\log \pi_k \left(\frac{p_k}{1-p_k}\right)^x (1-p_k)^n}$$

$$= e^{x \log \frac{p_k}{1-p_k} + 1 \cdot \log \pi_k (1-p_k)^n} = e^{w_k^t \phi(x)}$$

where $w_k = \begin{pmatrix} \log \frac{p_k}{1-p_k} \\ \log \pi_k (1-p_k)^n \end{pmatrix}$

$$\phi(x) = \begin{pmatrix} x \\ 1 \end{pmatrix}$$

Recall idea of discriminative models

Since many classification problems have softmax form assume softmax and fit.

$$\Rightarrow p(C=k|x) = \frac{e^{w_k^t \phi(x)}}{\sum_{k'} e^{w_{k'}^t \phi(x)}}$$

Multiclass Logistic

Have feature vector $\phi(x)$ to be classified in K class $\{1, \dots k\}$.

Model

$$q_k \doteq p(\text{class} = k \mid \phi(x)) = \frac{e^{w_k^t \phi(x)}}{\sum_{k'} e^{w_{k'}^t \phi(x)}} = \frac{e^{a_k}}{\sum_{k'} e^{a_{k'}}} \qquad a_k = w_k^t \phi(x)$$

$$\frac{\partial q_k}{\partial a_j} = \frac{\left(\sum_{k'} e^{a_{k'}}\right) e^{a_k} I_{kj} - e^{a_k} e^{a_j}}{\left(\sum_{k'} e^{a_{k'}}\right)^2} = q_k I_{kj} - q_k q_j = q_k (I_{kj} - q_j)$$

Have training data $(\phi(x_1), c_1), (\phi(x_2), c_2) \dots (\phi(x_n), c_n)$

$c_i \in \{1 \dots K\}$ are classes

$$X = \begin{pmatrix} \text{--} & \phi(x_1) & \text{--} \\ \text{--} & \phi(x_2) & \text{--} \\ \vdots & \vdots & \vdots \\ \text{--} & \phi(x_n) & \text{--} \end{pmatrix} \qquad T = \begin{pmatrix} t_{11} & \text{--} & t_{1k} \\ t_{21} & \text{--} & t_{2k} \\ \vdots & & \vdots \\ t_{n1} & \text{--} & t_{nk} \end{pmatrix}$$

$$t_{ik} = \begin{cases} 1 & c_i = k \\ 0 & \text{o.w.} \end{cases}$$

$c_i = 1 \Rightarrow t_i = 100 \dots 0$

$c_i = 2 \Rightarrow t_i = 010 \dots 0$

$$p(T \mid w_1 \dots w_k) = \prod_{i=1}^{n} \prod_{k=1}^{k} q_{ik}^{t_{ik}} \qquad q_{ik} = \frac{e^{w_k^t \phi(x_i)}}{\sum_{k'} e^{w_{k'}^t \phi(x_i)}}$$

$$\text{Error fn} = E(w_1 \dots w_k) = -\log p(T \mid w_1 \dots w_k) = -\sum_{i=1}^{n} \sum_{k=1}^{K} t_{ik} \log q_{ik}$$

$$\nabla_{w_j} E(w_1 \dots w_k) = -\sum_{i=1}^{n} \sum_{k=1}^{k} \frac{t_{ik}}{q_{ik}} \frac{\partial q_{ik}}{\partial a_j} \nabla_{w_j} a_{ij}$$

$$= -\sum_{i=1}^{n} \sum_{k=1}^{k} \frac{t_{ik}}{q_{ik}} q_{ik} (I_{kj} - q_{ij}) \phi(x_i)$$

$$= -\sum_{i=1}^{n} (t_{ij} - q_{ij}) \phi(x_i)$$

$$= \sum_{i=1}^{n} (q_{ij} - t_{ij}) \phi(x_i)$$

Can show

$$\nabla_{w_k} \nabla_{w_j} E(w_1 \dots w_k) = \sum_{i=1}^{n} q_{ij} (I_{kj} - q_{ik}) \phi(x_i) \phi(x_i)^t$$

Classification Trees

Two common treatments:

CART = Classification + Regression Trees

C4.5

We do CART

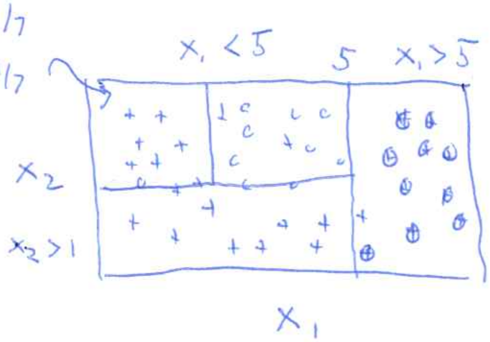
Pictorial View

Have data $(x_1, c_1), (x_2, c_2) \ldots (x_n, c_n)$

$x_i \in \mathbb{R}^2$ (2 continuous features

$c_i \in \{+, o\}$

$p(+) = 6/7$
$p(o) = 1/7$
so $+$

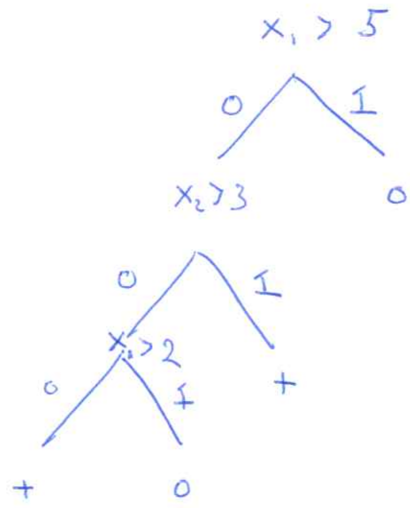


① Choose split $x_k < s$ dividing data into regions that are as pure as possible

② Recursively (and greedily) subdivide to increase purity (will return to stopping later)

③ Label each final region with most frequent class (or prob dist as in logistic regression)

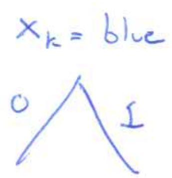
Can visualize splits as tree

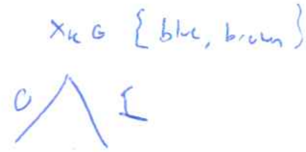


Need not be balanced!

CART works with categorical variables too

$x_k \in \{$ blue, brown, hazel, green $\}$





Always have binary splits in CART

Do tree_prostate_cancer.r

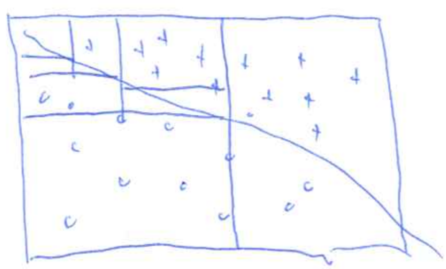
Virtues of Classification Trees

1. No parametric assumptions! (So general)
2. Computationally Efficient (Many variables, Large data)
3. Performs variable selection on-line

Some Drawbacks

1. Some datasets awkward

a)



Splits of form $x_k > s$

b) Binary features $x_1 \cdots x_k$

$$c(x_1 \cdots x_n) = \left(\sum_{i=1}^{k} x_i\right) \bmod 2$$

Can't classify without all features so CART fails here

2. Prone to overfitting, though can be addressed.

Choosing Splits

Splits chosen to maximize purity $\iff$ minimize impurity

Common Impurity Measures

① **Entropy**

Have dist. $p_1 \cdots p_K$ for prob (or proportion) of classes $1 \cdots K$

Eg $\{+ \, o \, o \, + \, o \, o\}$ $\quad p(+) = \frac{1}{3}$; $p(o) = \frac{2}{3}$

$$H(p) = -\sum_{k=1}^{K} p_k \log_2 p_k$$

Can show

a) $H(p) \geq 0$

b) $H(p) = 0 \iff p$ concentrates on 1 class

(Eg $p = (0, 0, 0, 1, 0, 0)$)

c) $H(p)$ is maximal when $p_1 = p_2 = \cdots = p_K = \frac{1}{K}$

② **Gini Index**

$$G(p) = 1 - \sum_{k=1}^{K} p_k^2$$

a) $G(p) \geq 0$

b) $G(p) = 0 \iff p$ concentrates on 1 class

c) $G(p)$ max when $p_1 = \cdots p_K = 1/K$

Suppose a terminal node in CART had $(a, b, a, a, a, b, c) \Rightarrow p = \left(\frac{4}{7}, \frac{2}{7}, \frac{1}{7}\right)$

Rather than labeling as most likely class (a), we choose a $4/7$ of time, b $\frac{2}{7}$ of time, c $\frac{1}{7}$ of time

What is prob. of error?

$P(\text{correct}) = \frac{4}{7}\frac{4}{7} + \frac{2}{7}\frac{2}{7} + \frac{1}{7}\frac{1}{7} = \sum p_k^2$

$\Rightarrow P(\text{error}) = 1 - \sum_k p_k^2 = G(p)$.