Multivariate Random Vector : Some Basic concepts

Multivariate random vector : $\underset{\sim}{X} = (X_1, \ldots, X_K)'$

X_i's are r.v. on $(\Omega, \mathcal{F}, P)$

Joint distribution fn

$$F_{X_1,\ldots,X_K}(x_1, \ldots, x_K) = P(X_1 \le x_1, \ldots, X_K \le x_K)$$

$$\forall\ (x_1, \ldots, x_K) \in \mathbb{R}^K$$

For discrete multivariate distributions

$$F_{X_1,\ldots,X_K}(x_1, \ldots, x_K) = \sum_{i_1 \le x_1} \cdots \sum_{i_K \le x_K} P(X_1 = i_1, \ldots, X_K = i_K)$$

$P(X_1 = x_1, \ldots, X_K = x_K)$ – jt p.m.f.

$\forall\ (x_1, \ldots, x_K) \in$ support of mult dist^n.

Marginal p.m.f. of X_1

$$P(X_1 = x_1) = \sum_{x_2} \cdots \sum_{x_K} P(X_1 = x_1, X_2 = x_2, \ldots X_K = x_K)$$

sly marginal for X_i $(i = 2(1)k)$

Joint marginal of X_i, X_j (or any subset)

$$\sum \cdots \sum_{\text{except } x_i, x_j} P(X_1 = x_1, \ldots, X_K = x_K)$$

For continuous multivariate dist'n's

jt. d.f.

$$F_{X_1,\ldots X_K}(x_1,\ldots x_K) = \int_{-\infty}^{x_1}\cdots\int_{-\infty}^{x_K} f_{X_1,\ldots X_K}(u_1,\ldots u_K)\prod_{i=1}^{K} du_i$$

$f_{X_1,\ldots X_K}(x_1,\ldots,x_K)$: jt p.d.f.

Marginal p.d.f. of X_i

$$f_{X_i}(x_i) = \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} f(u_1,\ldots u_K)\prod_{\substack{j=1\\ j\neq i}}^{K} du_j$$

($K-1$ fold, except over x_i)

jt marginal p.d.f. of (X_i, X_j) (or any subset)

$$f_{X_i,X_j}(x_i,x_j) = \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} f(u_1,\ldots u_K)\prod_{\substack{l=1\\ l\neq i;\ l\neq j}}^{K} du_l$$

($K-2$ fold, except over x_i & x_j)

Expectation vector of $\underset{\sim}{X}$

$$E(\underset{\sim}{X})_{K\times 1} = \begin{pmatrix} EX_1 \\ \vdots \\ EX_K \end{pmatrix} = \underset{\sim}{\mu}$$

Let $\underset{q\times 1}{\underset{\sim}{Y}} = \underset{q\times K}{A}\ \underset{K\times 1}{\underset{\sim}{X}} + \underset{q\times 1}{\underset{\sim}{b}}$; A : matrix of constants

$\underset{\sim}{b}$: vector of constants

$$E(\underset{\sim}{Y}) = A\,E(\underset{\sim}{X}) + \underset{\sim}{b}$$

$$E(\underset{\sim}{\alpha}'\underset{\sim}{X}) = \underset{\sim}{\alpha}'E(\underset{\sim}{X}) \qquad \underset{\sim}{\alpha}\in\mathbb{R}^K$$

Note:
More generally, for a random matrix

$$\underset{p\times q}{Z} = ((Z_{ij}))$$ Z_{ij} s are r.v.s. on $(\Omega, \mathcal{F}, P)$

p.d.f. of Z is jt p.d.f of Z_{ij} s $\left(\begin{matrix} i = 1(1)p \\ j = 1(1)q \end{matrix}\right)$

$$E(Z) = ((E Z_{ij}))$$

If $Z \to \underset{m\times p}{B}\, Z\, \underset{q\times n}{C} + \underset{m\times n}{D}$

then $E Z = B E(Z) C + D$

Covariance matrix of $\underset{\sim}{X}$

$$Cov(\underset{\sim}{X}) = E(\underset{\sim}{X} - E(\underset{\sim}{X}))(\underset{\sim}{X} - E(\underset{\sim}{X}))'$$

$$\Sigma = E(\underset{\sim}{X} - \underset{\sim}{\mu})(\underset{\sim}{X} - \underset{\sim}{\mu})'$$

$(i,j)^{th}$ element of Σ is

$$\sigma_{ij} = E(X_i - \mu_i)(X_j - \mu_j)$$

i.e $\sigma_{ij} = Cov(X_i, X_j)$

$$\sigma_{ii} = E(X_i - \mu_i)^2 = Var(X_i)$$

Note: (i) Total variation in $\underset{\sim}{X}$ $= tr\,\Sigma = \sum_{i=1}^{k} \sigma_{ii}$

(ii) Generalized variance of $\underset{\sim}{X} = |\Sigma|$

Let $\underset{\sim}{X}_{k\times 1}$ & $\underset{\sim}{Y}_{p\times 1}$ be two random vectors ∋

$E(\underset{\sim}{X}) = \underset{\sim}{\mu}_X$ & $E(\underset{\sim}{Y}) = \underset{\sim}{\mu}_Y$

$Cov(\underset{\sim}{X}, \underset{\sim}{Y}) = E(\underset{\sim}{X} - \underset{\sim}{\mu}_X)(\underset{\sim}{Y} - \underset{\sim}{\mu}_Y)'$

Note: $Cov(\underset{\sim}{X}) = \Sigma$ is always a sym matrix

Result: Characterization of a covariance matrix

Any $p\times p$ real sym matrix Σ is a covariance matrix iff Σ is positive semi definite (i.e. $\underset{\sim}{\alpha}'\Sigma\underset{\sim}{\alpha} \geq 0$ $\forall\ \underset{\sim}{\alpha} \in \mathbb{R}^p$).

Pf: Suppose Σ is cov matrix of $\underset{\sim}{X}$, then

$\forall\ \underset{\sim}{\alpha} \in \mathbb{R}^p$

$$0 \leq V(\underset{\sim}{\alpha}'\underset{\sim}{X}) = E(\underset{\sim}{\alpha}'\underset{\sim}{X} - \underset{\sim}{\alpha}'\underset{\sim}{\mu})^2 \quad ; E(\underset{\sim}{X}) = \underset{\sim}{\mu}$$

$$= E(\underset{\sim}{\alpha}'(\underset{\sim}{X} - \underset{\sim}{\mu}))^2$$

$$= E(\underset{\sim}{\alpha}'(\underset{\sim}{X} - \underset{\sim}{\mu}))(\underset{\sim}{\alpha}'(\underset{\sim}{X} - \underset{\sim}{\mu}))'$$

$$= E(\underset{\sim}{\alpha}'(\underset{\sim}{X} - \underset{\sim}{\mu})(\underset{\sim}{X} - \underset{\sim}{\mu})'\underset{\sim}{\alpha})$$

$$= \underset{\sim}{\alpha}'\Sigma\underset{\sim}{\alpha}$$

$\Rightarrow \underset{\sim}{\alpha}'\Sigma\underset{\sim}{\alpha} \geq 0\ \forall\ \underset{\sim}{\alpha} \in \mathbb{R}^p$

$\Rightarrow \Sigma$ is p.s.d.

Alternatively, suppose Σ is p.s.d. with rank r ($\leq p$).

Then, $\Sigma = CC'$; C is $p \times r$ matrix of rank r

Let $\underset{\sim}{Y}$ be $r \times 1$ vector of indep r.v.s $\Rightarrow$

$$E(\underset{\sim}{Y}) = \underset{\sim}{0} \quad \text{and} \quad Cov(\underset{\sim}{Y}) = I_r$$

Transform $\underset{\sim}{Y} \rightarrow \underset{\sim}{X} = C\underset{\sim}{Y}$

then $E(\underset{\sim}{X}) = C\,E(\underset{\sim}{Y}) = \underset{\sim}{0}$

$$Cov(\underset{\sim}{X}) = E\,\underset{\sim}{X}\underset{\sim}{X}' = E(C\underset{\sim}{Y})(C\underset{\sim}{Y})'$$

$$= C\,E(\underset{\sim}{Y}\underset{\sim}{Y}')C' = CC' = \Sigma$$

$$= \Sigma$$

$\Rightarrow \Sigma$ is a covariance matrix

Note: $\underset{\sim}{X} \Rightarrow E(\underset{\sim}{X}) = \underset{\sim}{\mu}$; $Cov(\underset{\sim}{X}) = \Sigma$

Transform $\underset{\sim}{X} \rightarrow \underset{\sim}{Y} = A\underset{\sim}{X} + \underset{\sim}{b}$

$$Cov(\underset{\sim}{Y}) = E(\underset{\sim}{Y} - E(\underset{\sim}{Y}))(\underset{\sim}{Y} - E(\underset{\sim}{Y}))'$$

$$= E\left(A\underset{\sim}{X} + \underset{\sim}{b} - (A\underset{\sim}{\mu} + \underset{\sim}{b})\right)\left(A\underset{\sim}{X} + \underset{\sim}{b} - (A\underset{\sim}{X} + \underset{\sim}{b})\right)'$$

$$= E(A\underset{\sim}{X} - A\underset{\sim}{\mu})(A\underset{\sim}{X} - A\underset{\sim}{\mu})'$$

$$= A\,E(\underset{\sim}{X} - \underset{\sim}{\mu})(\underset{\sim}{X} - \underset{\sim}{\mu})'A' = A\Sigma A'$$

Sp. case: Suppose $V = \text{diag}(\sigma_{11}, \ldots, \sigma_{kk})$

with $\sigma_{ii} > 0$

Take $A = (V^{1/2})^{-1}$

$\underset{\sim}{X} \to \underset{\sim}{Y} = A \underset{\sim}{X}$

$$\text{Cov}(\underset{\sim}{Y}) = (V^{1/2})^{-1} \Sigma (V^{1/2})^{-1} = \Sigma_Y$$

$(i,j)^{th}$ entry of $\Sigma_Y = \dfrac{\sigma_{ij}}{(\sigma_{ii}\sigma_{jj})^{1/2}} = \rho_{ij}$

ρ_{ij} is Corrl^n (X_i, X_j)

i.e. Σ_Y = Corrl^n matrix of $\underset{\sim}{X}$

i.e. Corrl^n $(\underset{\sim}{X}) = (V^{1/2})^{-1} \Sigma (V^{1/2})^{-1} = \rho$.

Note: If Cov matrix of $\underset{\sim}{X}$ is not p.d., then w.p. 1, components of $\underset{\sim}{X}$ are linearly related.

Pf: If $\Sigma \neq 0$, then $\exists$ an $\underset{\sim}{\alpha} \in \mathbb{R}^p$ $(\underset{\sim}{\alpha} \neq \underset{\sim}{0}) \ni$

$$0 = \underset{\sim}{\alpha}' \Sigma \underset{\sim}{\alpha} = V(\underset{\sim}{\alpha}' \underset{\sim}{X})$$

i.e. $P(\underset{\sim}{\alpha}' \underset{\sim}{X} = \underset{\sim}{\alpha}' \underset{\sim}{\mu}) = 1$

i.e $P(\underset{\sim}{\alpha}' (\underset{\sim}{X} - \underset{\sim}{\mu}) = 0) = 1$

i.e. $\sum_i \alpha_i (X_i - \mu_i) = 0$ w.p. 1 for not all $\alpha_i = 0$

i.e. w.p 1 X_i's are linearly related.

Note: Partitions of covariance matrix

$\underset{k\times 1}{\underset{\sim}{X}}$ is $\Rightarrow E(\underset{\sim}{X}) = \underset{\sim}{\mu}$ & $Cov(\underset{\sim}{X}) = \Sigma$

Consider the partition

$$\underset{\sim}{X} = \begin{pmatrix} \underset{\sim}{X}^{(1)}_{p\times 1} \\ \underset{\sim}{X}^{(2)}_{k-p\times 1} \end{pmatrix}; \quad \underset{\sim}{\mu} = \begin{pmatrix} \underset{\sim}{\mu}^{(1)} \\ \underset{\sim}{\mu}^{(2)} \end{pmatrix}.$$

$$Cov(\underset{\sim}{X}) = E(\underset{\sim}{X} - \underset{\sim}{\mu})(\underset{\sim}{X} - \underset{\sim}{\mu})'$$

$$= E\begin{pmatrix} (\underset{\sim}{X}^{(1)} - \underset{\sim}{\mu}^{(1)})(\underset{\sim}{X}^{(1)} - \underset{\sim}{\mu}^{(1)})' & (\underset{\sim}{X}^{(1)} - \underset{\sim}{\mu}^{(1)})(\underset{\sim}{X}^{(2)} - \underset{\sim}{\mu}^{(2)})' \\ (\underset{\sim}{X}^{(2)} - \underset{\sim}{\mu}^{(2)})(\underset{\sim}{X}^{(1)} - \underset{\sim}{\mu}^{(1)})' & (\underset{\sim}{X}^{(2)} - \underset{\sim}{\mu}^{(2)})(\underset{\sim}{X}^{(2)} - \underset{\sim}{\mu}^{(2)})' \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$$

$\Sigma_{11} = Cov(\underset{\sim}{X}^{(1)})$; $\Sigma_{22} = Cov(\underset{\sim}{X}^{(2)})$

$\Sigma_{12} = Cov(\underset{\sim}{X}^{(1)}, \underset{\sim}{X}^{(2)}) = \Sigma_{21}'$

$= (Cov(\underset{\sim}{X}^{(2)}, \underset{\sim}{X}^{(1)}))'$

If elements of $\underset{\sim}{X}^{(1)}$ are indep of the elements of $\underset{\sim}{X}^{(2)}$ then $\Sigma_{12} = 0$, i.e $\Sigma = \begin{pmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{pmatrix}$

Converse is NOT true in general.

Random sampling from multivariate pop^n's

Multivariate pop^n with mean vector $\underset{\sim}{\mu}$ and covariance Σ

$\underset{\sim}{\mu}$ & Σ usually unknown

# of unknowns $p + \frac{p(p+1)}{2}$ (for p dimensional pop^n)

Let $\underset{\sim}{X}_1 = \begin{pmatrix} X_{11} \\ X_{21} \\ \vdots \\ X_{p1} \end{pmatrix}, \ldots \underset{\sim}{X}_n = \begin{pmatrix} X_{1n} \\ X_{2n} \\ \vdots \\ X_{pn} \end{pmatrix}$ be n random samples from the pop^n (i.i.d)

Random Sample matrix

$$X_{p \times n} = \begin{pmatrix} X_{11} & \cdots & X_{1n} \\ X_{21} & \cdots & X_{2n} \\ \cdots & \cdots & \cdots \\ X_{p1} & \cdots & X_{pn} \end{pmatrix}$$

Observation matrix / data matrix

$$\mathcal{X}_{p \times n} = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots \\ x_{p1} & \cdots & x_{pn} \end{pmatrix}$$

$$= (\underset{\sim}{x}_1, \ldots, \underset{\sim}{x}_n) = \begin{pmatrix} \underset{\sim}{y}_1' \\ \vdots \\ \underset{\sim}{y}_p' \end{pmatrix}$$

$\underset{\sim}{x}_j = \begin{pmatrix} x_{1j} \\ \vdots \\ x_{pj} \end{pmatrix}$ j^{th} obsn vector, $j = 1(1)n$

$\underset{\sim}{y}_i' = (y_{i1}, \ldots, y_{in})$ – i^{th} variable obsns, $i = 1(1)p$

$\underset{\sim}{\bar{x}}$: observed sample mean vector

$$\underset{\sim}{\bar{x}} = \begin{pmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_p \end{pmatrix} = \frac{1}{n}\begin{pmatrix} \underset{\sim}{y}_1' \underset{\sim}{1}_n \\ \vdots \\ \underset{\sim}{y}_p' \underset{\sim}{1}_n \end{pmatrix} = \frac{1}{n}\begin{pmatrix} \sum_{j=1}^{n} x_{1j} \\ \vdots \\ \sum_{j=1}^{n} x_{pj} \end{pmatrix} = \frac{1}{n}\, \mathcal{X}\, \underset{\sim}{1}_n$$

Observed sample variance covariance matrix:

$$S_n = \frac{1}{n}\begin{pmatrix} \sum_{j=1}^{n}(x_{1j}-\bar{x}_1)^2 & \cdots & \sum_{1}^{n}(x_{1j}-\bar{x}_1)(x_{pj}-\bar{x}_p) \\ & \sum_{j=1}^{n}(x_{2j}-\bar{x}_2)^2 & \cdots & \sum_{1}^{n}(x_{2j}-\bar{x}_2)(x_{pj}-\bar{x}_p) \\ & & \cdots & \cdots \\ & & & \sum_{1}^{n}(x_{pj}-\bar{x}_p)^2 \end{pmatrix}$$

or $S_{n-1} = \frac{n}{n-1} S_n$

Generalized sample variance : $|S_n|$ (or $|S_{n-1}|$)

Total sample variation : $\operatorname{tr} S_n$ (or $\operatorname{tr} S_{n-1}$)

Note that $(n-1)S_{n-1} = n S_n$

$$= \begin{pmatrix} \sum x_{1j}^2 & \sum x_{1j}x_{2j} & \cdots & \sum x_{1j}x_{pj} \\ & \sum x_{2j}^2 & \cdots & \sum x_{2j}x_{pj} \\ & & \cdots & \\ & & & \sum x_{pj}^2 \end{pmatrix}$$

$$- n \begin{pmatrix} \bar{x}_1^2 & \bar{x}_1\bar{x}_2 & \cdots & \bar{x}_1\bar{x}_p \\ & \bar{x}_2^2 & \cdots & \bar{x}_2\bar{x}_p \\ & & \cdots & \\ & & & \bar{x}_p^2 \end{pmatrix}$$

i.e. $n S_n = \mathcal{X}\mathcal{X}' - n \underset{\sim}{\bar{x}}\,\underset{\sim}{\bar{x}}'$

$$= \mathcal{X}\mathcal{X}' - n\left(\frac{1}{n}\mathcal{X}\underline{1}_n\right)\left(\frac{1}{n}\mathcal{X}\underline{1}_n\right)'$$

$$= \mathcal{X}\mathcal{X}' - \frac{1}{n}\mathcal{X}\underline{1}_n\underline{1}_n'\mathcal{X}'$$

$$= \mathcal{X}\left(I_n - \frac{1}{n}1_n 1_n'\right)\mathcal{X}' = (n-1)S_{n-1}$$

Alternately,

$$n S_n = \mathcal{X}\mathcal{X}' - n\,\underset{\sim}{\bar{x}}\,\underset{\sim}{\bar{x}}'$$

$$= (\underset{\sim}{x}_1 : \cdots : \underset{\sim}{x}_n)\begin{pmatrix} \underset{\sim}{x}_1' \\ \vdots \\ \underset{\sim}{x}_n' \end{pmatrix} - n\,\underset{\sim}{\bar{x}}\,\underset{\sim}{\bar{x}}'$$

$$= \sum \underset{\sim}{x}_j \underset{\sim}{x}_j' - n\,\underset{\sim}{\bar{x}}\,\underset{\sim}{\bar{x}}'$$

$$= \sum_{j=1}^{n} (\underset{\sim}{x}_j - \underset{\sim}{\bar{x}})(\underset{\sim}{x}_j - \underset{\sim}{\bar{x}})' = (n-1)S_{n-1}$$

Note: Corresponding random matrices

$$n S_n = (n-1) S_{n-1}$$

$$= X\left(I_n - \frac{1}{n}1_n 1_n'\right)X'$$

$$= \sum (\underline{X}_j - \underline{\bar{X}})(\underline{X}_j - \underline{\bar{X}})'$$

Multivariate Normal distⁿ

Defⁿ: Let $\underset{\sim}{X}_{p\times 1}$ be a random vector with $E(\underset{\sim}{X}) = \underset{\sim}{\mu}$ and $\text{Cov}(\underset{\sim}{X}) = \Sigma$. We say that $\underset{\sim}{X}$ has a multivariate normal distⁿ $\left(\underset{\sim}{X} \sim N_p(\underset{\sim}{\mu}, \Sigma)\right)$ iff $\forall\, \underset{\sim}{a} \in \mathbb{R}^p$ $(\underset{\sim}{a} \neq \underset{\sim}{0})$, $\underset{\sim}{a}'\underset{\sim}{X}$ has univariate normal distⁿ

Some results

If $\underset{\sim}{X} \sim N_p(\underset{\sim}{\mu}, \Sigma)$, then

(i) $\underset{\sim}{X} - \underset{\sim}{\mu} \sim N_p(\underset{\sim}{0}, \Sigma)$

(ii) each component of $\underset{\sim}{X} \sim N_1$

(iii) Any subvector of $\underset{\sim}{X}$ has mult normal

(iv) $\underset{q\times p}{A}\,\underset{\sim}{X} + \underset{q\times 1}{\underset{\sim}{b}} \sim N_q(A\underset{\sim}{\mu} + \underset{\sim}{b},\ A\Sigma A')$

(v) If $\Sigma > 0$, then p.d.f. of $\underset{\sim}{X}$ is

$$f_{\underset{\sim}{X}}(\underset{\sim}{x}) = (2\pi)^{-p/2}\,|\Sigma|^{-1/2}\exp\left(-\tfrac{1}{2}(\underset{\sim}{x} - \underset{\sim}{\mu})'\Sigma^{-1}(\underset{\sim}{x} - \underset{\sim}{\mu})\right)$$

(vi) If $\Sigma \not> 0$, $\nexists$ p.d.f. of $\underset{\sim}{X}$

(vii) If $\underset{\sim}{X} = \begin{pmatrix} \underset{\sim}{X}^{(1)} \\ \underset{\sim}{X}^{(2)} \end{pmatrix}_{p-q\times 1}^{q\times 1}$, $\underset{\sim}{\mu} = \begin{pmatrix} \underset{\sim}{\mu}^{(1)} \\ \underset{\sim}{\mu}^{(2)} \end{pmatrix}$, $\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$

$\underset{\sim}{X}^{(1)}$, $\underset{\sim}{X}^{(2)}$ indep iff $\Sigma_{12} = 0$

Random sampling from multivariate normal

$\underset{\sim}{X}_1, \ldots, \underset{\sim}{X}_n$ r.s. from $N_p(\underset{\sim}{\mu}, \Sigma)$, $\Sigma > 0$

Unbiased estimators:

$$\bar{\underset{\sim}{X}} = \frac{1}{n}\sum_{i=1}^{n} \underset{\sim}{X}_i \quad ; \quad E\,\bar{\underset{\sim}{X}} = \underset{\sim}{\mu}$$

$$S_{n-1} = \frac{1}{n-1}\sum_{i=1}^{n} (\underset{\sim}{X}_i - \bar{\underset{\sim}{X}})(\underset{\sim}{X}_i - \bar{\underset{\sim}{X}})'$$

$$E\,S_{n-1} = \frac{1}{n-1} E\left(\Sigma \underset{\sim}{X}_i \underset{\sim}{X}_i' - n\,\bar{\underset{\sim}{X}}\,\bar{\underset{\sim}{X}}'\right)$$

$$= \frac{1}{n-1}\left(\sum_{1}^{n} E\,\underset{\sim}{X}_i \underset{\sim}{X}_i' - n\,E\,\bar{\underset{\sim}{X}}\,\bar{\underset{\sim}{X}}'\right)$$

$$= \frac{1}{n-1}\left(n\left(\Sigma + \underset{\sim}{\mu}\,\underset{\sim}{\mu}'\right) - n\left(\Sigma/n + \underset{\sim}{\mu}\,\underset{\sim}{\mu}'\right)\right)$$

$$= \Sigma$$

Thus $\hat{\underset{\sim}{\mu}} = \bar{\underset{\sim}{X}}$ & $\hat{\Sigma} = S_{n-1}$ (This is true for random sampling from any multivariate pop^n)

Maximum likelihood estimators

$\bar{\underset{\sim}{X}}$ is MLE of $\underset{\sim}{\mu}$

$S_n = \frac{1}{n}\Sigma(\underset{\sim}{X}_i - \bar{\underset{\sim}{X}})(\underset{\sim}{X}_i - \bar{\underset{\sim}{X}})'$ is MLE of Σ

Dist^n:

$$\bar{\underset{\sim}{X}} \sim N_p(\underset{\sim}{\mu}, \Sigma/n)$$

$$nS_n = (n-1)S_{n-1} \sim W_p(n-1, \Sigma)$$

→ a Wishart dist^n of order p with d.f. $n-1$ and associated covariance matrix Σ

Also $\bar{\underset{\sim}{X}}$ & S_{n-1} (or S_n) are independently distributed.

Principal Component Analysis (PCA)

$\underset{\sim}{X} = (X_1, \ldots, X_p)'$

$$\Sigma = \mathrm{Cov}(\underset{\sim}{X}) = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1p} \\ & \sigma_{22} & \cdots & \sigma_{2p} \\ & & \ddots & \\ & & & \sigma_{pp} \end{pmatrix} = ((\sigma_{ij}))$$

Total variation in $\underset{\sim}{X} = \mathrm{tr}\,\Sigma = \sum_{i=1}^{p} \sigma_{ii}$

Aim of PCA

Replace $\underset{\sim}{X}$ with $\underset{\sim}{Y} = (Y_1, \ldots, Y_p)'$, where Y_i's are are linear combinations of X_i's $\ni$

(i) Y_i's are uncorrelated, i.e.

$$\mathrm{Cov}(Y_i, Y_j) = 0 \quad \forall\, i \neq j$$

(ii) total variation in $\underset{\sim}{X}$

= total variation in $\underset{\sim}{Y}$

& (iii) total variation in $(Y_1, \ldots, Y_k)'$

$\approx$ total variation of $\underset{\sim}{X}$; $k \ll p$

Note: (iii) may not always be possible

Major Uses of PCA

(i) Data dimension reduction ~~for visualization~~

(ii) Outlier mining

(iii) Detection of clusters in the multidimensional data cloud

(iv) Projection & visualization

(v) ranking of multidimensional data

(vi) Checking for multivariate normality

Def^n: Principal components

PCs are uncorrelated linear combinations $Y_1, \ldots Y_p$ (derived from $X_1, \ldots, X_p$) whose variances are in decreasing order, with Y_1 explaining the maximum variability, Y_2 explaining the second maximum variability and so on.

i.e. the 1st PC, Y_1, is the linear combination $\underset{\sim}{l}_1' \underset{\sim}{X}$ that maximizes $V(\underset{\sim}{l}' \underset{\sim}{X})$ subject to $\underset{\sim}{l}' \underset{\sim}{l} = 1$

the 2nd PC, Y_2, is the linear combination $\underset{\sim}{l}_2' \underset{\sim}{X}$ that maximizes $V(\underset{\sim}{l}' \underset{\sim}{X})$ subject to $\underset{\sim}{l}' \underset{\sim}{l} = 1$ and $\text{cov}(\underset{\sim}{l}_1' \underset{\sim}{X}, \underset{\sim}{l}' \underset{\sim}{X}) = 0$

the ith PC, Y_i, is the linear combination $\underset{\sim}{l}_i' \underset{\sim}{X}$ that maximizes $V(\underset{\sim}{l}' \underset{\sim}{X})$ subject to $\underset{\sim}{l}' \underset{\sim}{l} = 1$ and $\text{cov}(\underset{\sim}{l}_k' \underset{\sim}{X}, \underset{\sim}{l}' \underset{\sim}{X}) = 0 \quad \forall k < i$.

Derivation of PCs

Let Σ be the covariance matrix associated with the random vector $\underset{\sim}{X}_{p\times 1}$. The eigenvalue-eigenvector pairs of Σ are $(\lambda_1, \underset{\sim}{e}_1), \ldots, (\lambda_p, \underset{\sim}{e}_p)$, where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$. (0.n).

The ith PC is given by

$$Y_i = \underset{\sim}{e}_i' \underset{\sim}{X} \qquad i = 1(1)p$$

With $V(Y_i) = \lambda_i$ & $\text{cov}(Y_i, Y_j) = 0 \quad \forall i \neq j$

Pf:

Consider any $\underset{\sim}{l}' \underset{\sim}{X}$, $\underset{\sim}{l} \in \mathbb{R}^p$; $\Rightarrow \underset{\sim}{l}' \underset{\sim}{l} = 1$

$$V(\underset{\sim}{l}' \underset{\sim}{X}) = \underset{\sim}{l}' V(\underset{\sim}{X}) \underset{\sim}{l}$$
$$= \underset{\sim}{l}' \Sigma \underset{\sim}{l}$$

Consider the eigen value - eigen vector decomposition of Σ as

$$\Sigma = P D_\lambda P' \quad ; \quad P = (\underset{\sim}{e}_1 : \cdots : \underset{\sim}{e}_p)$$
$$D_\lambda = \text{diag}(\lambda_1, \cdots, \lambda_p)$$

$$\Rightarrow V(\underset{\sim}{l}' \underset{\sim}{X}) = \underset{\sim}{l}' P D_\lambda P' \underset{\sim}{l}$$
$$= \underset{\sim}{\beta}' D_\lambda \underset{\sim}{\beta}$$
$$= \sum_{i=1}^{p} \beta_i^2 \lambda_i$$

where $\underset{\sim}{\beta} = P' \underset{\sim}{l} \in \mathbb{R}^p$

$$\underset{\sim}{l} = P \underset{\sim}{\beta}$$

Now, $\underset{\sim}{l}' \underset{\sim}{l} = 1 \Rightarrow \underset{\sim}{\beta}' P' P \underset{\sim}{\beta} = \underset{\sim}{\beta}' \underset{\sim}{\beta} = 1$

Thus

$$\max_{\underset{\sim}{l} \ni \underset{\sim}{l}'\underset{\sim}{l} = 1} V(\underset{\sim}{l}' \underset{\sim}{X}) = \max_{\underset{\sim}{\beta} \ni \underset{\sim}{\beta}'\underset{\sim}{\beta} = 1} \underset{\sim}{\beta}' D_\lambda \underset{\sim}{\beta} = \max_{\underset{\sim}{\beta} \ni \underset{\sim}{\beta}'\underset{\sim}{\beta} = 1} \sum_{i=1}^{p} \lambda_i \beta_i^2$$

Realize that,

$$\sum_{i=1}^{p} \lambda_i \beta_i^2 \leq \lambda_1 \sum \beta_i^2 = \lambda_1$$

i.e. $V(\underset{\sim}{l}' \underset{\sim}{X}) \leq \lambda_1$

and $V(\underset{\sim}{e}_1' \underset{\sim}{X}) = \underset{\sim}{e}_1' \Sigma \underset{\sim}{e}_1$

$$= \underset{\sim}{e}_1' P D_\lambda P' \underset{\sim}{e}_1$$
$$= \underset{\sim}{e}_1' (\underset{\sim}{e}_1 : \cdots : \underset{\sim}{e}_p) D_\lambda \begin{pmatrix} \underset{\sim}{e}_1' \\ \vdots \\ \underset{\sim}{e}_p' \end{pmatrix} \underset{\sim}{e}_1$$

i.e. $V(\underset{\sim}{e}_1' \underset{\sim}{X}) = (1, 0, \ldots, 0)\, D_\lambda \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

$= \lambda_1 = \max_{\underset{\sim}{l} \ni \underset{\sim}{l}'\underset{\sim}{l} = 1} V(\underset{\sim}{l}' \underset{\sim}{X})$

$\Rightarrow Y_1 = \underset{\sim}{e}_1' \underset{\sim}{X}$ is the 1st PC

For the second PC, we need $Y_2 = \underset{\sim}{l}_2' \underset{\sim}{X} \ni Y_2$ is uncorrelated with Y_1 and $V(\underset{\sim}{l}_2' \underset{\sim}{X})$ to be the maximum under this constraint.

Now, $\operatorname{Cov}(\underset{\sim}{l}' \underset{\sim}{X}, \underset{\sim}{e}_1' \underset{\sim}{X})$

$= E(\underset{\sim}{l}'(\underset{\sim}{X} - \underset{\sim}{\mu}))(\underset{\sim}{e}_1'(\underset{\sim}{X} - \underset{\sim}{\mu}))'$

$= \underset{\sim}{l}' \Sigma \underset{\sim}{e}_1$

$= \underset{\sim}{l}' \left(\sum_{i=1}^{p} \lambda_i \underset{\sim}{e}_i \underset{\sim}{e}_i' \right) \underset{\sim}{e}_1$

$= \underset{\sim}{l}' (\lambda_1 \underset{\sim}{e}_1) = \lambda_1 \underset{\sim}{l}' \underset{\sim}{e}_1$

Thus $\operatorname{Cov}(\underset{\sim}{l}' \underset{\sim}{X}, \underset{\sim}{e}_1' \underset{\sim}{X}) = 0 \Leftrightarrow \underset{\sim}{l} \perp \underset{\sim}{e}_1$

Thus $\max_{\underset{\sim}{l} \ni \underset{\sim}{l}'\underset{\sim}{l} = 1} V(\underset{\sim}{l}' \underset{\sim}{X})$ subject to $\operatorname{Cov}(\underset{\sim}{l}' \underset{\sim}{X}, Y_1) = 0$ is

equivalent to $\max_{\substack{\underset{\sim}{l} \ni \underset{\sim}{l} \perp \underset{\sim}{e}_1 \\ \underset{\sim}{l}'\underset{\sim}{l} = 1}} V(\underset{\sim}{l}' \underset{\sim}{X})$

$V(\underset{\sim}{l}' \underset{\sim}{X}) = \underset{\sim}{l}' P D_\lambda P'$

$= \underset{\sim}{l}' (\underset{\sim}{e}_1 : \ldots : \underset{\sim}{e}_p) \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_p \end{pmatrix} \begin{pmatrix} \underset{\sim}{e}_1' \\ \vdots \\ \underset{\sim}{e}_p' \end{pmatrix}$

$\forall\ \underset{\sim}{l} \perp \underset{\sim}{e}_1$

$$\underset{\sim}{l}' P D_\lambda P' \underset{\sim}{l} = (\underset{\sim}{0}, b_1, \ldots, b_p) \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_p \end{pmatrix} \begin{pmatrix} 0 \\ b_2 \\ \vdots \\ b_p \end{pmatrix}$$

$$= \sum_{i=2}^{p} b_i^2 \lambda_i$$

$$\leq \lambda_2$$

$$\left(\underset{\sim}{b} = (0, b_2, \ldots, b_p)' = P' \underset{\sim}{l} \right.$$

$$\text{i.e. } \underset{\sim}{l} = P \underset{\sim}{b}$$

$$\left. \underset{\sim}{l}' \underset{\sim}{l} = 1 \Rightarrow \underset{\sim}{b}' \underset{\sim}{b} = 1 \quad \text{i.e. } \sum_{2}^{p} b_i^2 = 1 \right)$$

And $V(\underset{\sim}{e}_2' \underset{\sim}{X}) = \underset{\sim}{e}_2' \Sigma \underset{\sim}{e}_2$

$$= \underset{\sim}{e}_2' (\underset{\sim}{e}_1 : \ldots : \underset{\sim}{e}_p) D_\lambda \begin{pmatrix} \underset{\sim}{e}_1' \\ \vdots \\ \underset{\sim}{e}_p' \end{pmatrix} \underset{\sim}{e}_2$$

$$= (0, 1, 0 \ldots, 0) D_\lambda \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$= \lambda_2$$

$$= \max V(\underset{\sim}{l}' \underset{\sim}{X})$$

$$\underset{\sim}{l} \ni \underset{\sim}{l} \ni \underset{\sim}{e}_1$$

$$\underset{\sim}{l}' \underset{\sim}{l} = 1$$

$\Rightarrow Y_2 = \underset{\sim}{e}_2' \underset{\sim}{X}$ is the 2^{nd} PC

For the $(K+1)^{th}$ PC $(K+1 \leq p)$, we need to

maximise $V(\underset{\sim}{l}' \underset{\sim}{X}) \ni Cov(\underset{\sim}{l}' \underset{\sim}{X}, Y_i) = 0 \ \forall\ i \leq K$

i.e. maximise $V(\underset{\sim}{l}' \underset{\sim}{X}) \ni \underset{\sim}{l} \perp \underset{\sim}{e}_i \ \forall\ i \leq K$

$\forall \underset{\sim}{l} \perp \underset{\sim}{e}_1, \ldots, \underset{\sim}{e}_k$

$$V(\underset{\sim}{l}'\underset{\sim}{X}) = \underset{\sim}{l}'(\underset{\sim}{e}_1 : \ldots : \underset{\sim}{e}_k : \underset{\sim}{e}_{k+1} : \ldots : \underset{\sim}{e}_p)\, D_\lambda \begin{pmatrix} \underset{\sim}{e}_1' \\ \vdots \\ \underset{\sim}{e}_k' \\ \underset{\sim}{e}_{k+1}' \\ \vdots \\ \underset{\sim}{e}_p' \end{pmatrix} \underset{\sim}{l}$$

$$= (0, \ldots 0, \underbrace{c_{k+1}, \ldots, c_p}_{\underset{\sim}{l}'\underset{\sim}{e}_{k+1}}) D_\lambda \begin{pmatrix} 0 \\ \vdots \\ 0 \\ c_{k+1} \\ \vdots \\ c_p \end{pmatrix}$$

$$= \sum_{i=k+1}^{p} c_i^2 \lambda_i$$

$\underset{\sim}{c} = P'\underset{\sim}{l}$ with $\underset{\sim}{c} = (\underbrace{0, \ldots 0}_{k}, c_{k+1}, \ldots, c_p)'$

$\underset{\sim}{l}'\underset{\sim}{l} = 1 \Rightarrow \underset{\sim}{c}'\underset{\sim}{c} = 1$ i.e. $\sum_{k+1}^{p} c_i^2 = 1$

Thus $\forall \underset{\sim}{l} \ni \underset{\sim}{e}_1, \ldots, \underset{\sim}{e}_k$

$$V(\underset{\sim}{l}'\underset{\sim}{X}) = \sum_{k+1}^{p} c_i^2 \lambda_i \leq \lambda_{k+1}$$

and $\lambda_{k+1} = V(\underset{\sim}{e}_{k+1}'\underset{\sim}{X}) \left(= \underset{\sim}{e}_{k+1}' P D_\lambda P' \underset{\sim}{e}_{k+1}\right)$

$$\Rightarrow V(\underset{\sim}{e}_{k+1}'\underset{\sim}{X}) = \lambda_{k+1} = \max_{\underset{\sim}{l} \ni \underset{\sim}{l} \perp \underset{\sim}{e}_1 \ldots \underset{\sim}{e}_k \,\&\, \underset{\sim}{l}'\underset{\sim}{l} = 1} V(\underset{\sim}{l}'\underset{\sim}{X})$$

$\Rightarrow$ (k+1)th PC is

$$Y_{k+1} = \underset{\sim}{e}_{k+1}'\underset{\sim}{X}$$

Note 1 : $Y_1, \ldots, Y_p$ are s.t

$$Cov(\underset{\sim}{Y}) = D_\lambda$$

$$Y_i = \underset{\sim}{e}_i' \underset{\sim}{X}$$

$$Cov(Y_i, Y_j) = Cov(\underset{\sim}{e}_i' \underset{\sim}{X}, \underset{\sim}{e}_j' \underset{\sim}{X})$$

$$= \underset{\sim}{e}_i' \Sigma \underset{\sim}{e}_j$$

$$= \underset{\sim}{e}_i' \left(\sum_{k=1}^{p} \lambda_k \underset{\sim}{e}_k \underset{\sim}{e}_k' \right) \underset{\sim}{e}_j$$

$$= \begin{cases} \lambda_i, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$$

$$\Rightarrow Cov(\underset{\sim}{Y}) = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_p \end{pmatrix} = \Sigma_Y$$

Note 2 : Total variation in $\underset{\sim}{X}$ = Total variation in $\underset{\sim}{Y}$.

Total variation in $\underset{\sim}{X}$ $= tr\, \Sigma$

$$= tr(P D_\lambda P')$$

$$= tr(D_\lambda P' P)$$

$$= tr\, D_\lambda = \sum_{i=1}^{p} \lambda_i$$

Total variation in $\underset{\sim}{Y}$ $= tr\,(Cov(\underset{\sim}{Y}))$

$$= tr\, D_\lambda = \sum_{i=1}^{p} \lambda_i$$

Note 3: If $\underset{\sim}{X} \sim N_p(\underline{\mu}, \Sigma)$, then

(i) $Y_i = \underline{e}_i' \underset{\sim}{X} \quad (\forall\, i = 1(1)p) \sim N_1$

(ii) $$\underset{\sim}{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_p \end{pmatrix} = \begin{pmatrix} \underline{e}_1' \underset{\sim}{X} \\ \vdots \\ \underline{e}_p' \underset{\sim}{X} \end{pmatrix} = \begin{pmatrix} \underline{e}_1' \\ \vdots \\ \underline{e}_p' \end{pmatrix} \underset{\sim}{X}$$

i.e. $\underset{\sim}{Y} = P' \underset{\sim}{X} \sim N_p(P'\underline{\mu}, P'\Sigma P)$

$$P'\Sigma P = P'(P D_\lambda P')P = D_\lambda$$

(iii) $Y_1, \ldots, Y_p$ are independent N_1

(Cov = 0 for joint normal $\Rightarrow$ independence)

Note 4: $\mathrm{Cov}(X_i, Y_j) = ?$

$$Y_j = \underline{e}_j' \underset{\sim}{X}$$

$$\underset{\sim}{Y} = P' \underset{\sim}{X}$$

$$\underset{\sim}{X} = P \underset{\sim}{Y} = (\underset{\sim}{e}_1 : \ldots : \underset{\sim}{e}_p)\, \underset{\sim}{Y}$$

$$= \begin{pmatrix} e_{11} & \cdots & e_{1p} \\ e_{21} & \cdots & e_{2p} \\ \vdots & & \vdots \\ e_{p1} & \cdots & e_{pp} \end{pmatrix} \begin{pmatrix} Y_1 \\ \vdots \\ Y_p \end{pmatrix}$$

i.e. $$X_i = \sum_{k=1}^{p} e_{ik} Y_k$$

$$\mathrm{Cov}(X_i, Y_j) = \mathrm{Cov}\left(\sum_{k=1}^{p} e_{ik} Y_k, Y_j\right) = e_{ij}\lambda_j$$

e_{ij} is the i^{th} element (row) of $\underset{\sim}{e}_j$

$$\mathrm{Corr}^n(X_i, Y_j) = e_{ij}\sqrt{\frac{\lambda_j}{\sigma_{ii}}}$$

$$V(X_i) = \sigma_{ii} \;\&\; V(Y_j) = \lambda_j$$

$$\Rightarrow \text{Corr}^n(X_i, Y_j) = \frac{e_{ij}\lambda_j}{(\sigma_{ii}\lambda_j)^{1/2}} = e_{ij}\sqrt{\frac{\lambda_j}{\sigma_{ii}}}$$

Note 5: Principal components derived from correlation matrix

Consider the standardized variables

$$X_i \to Z_i = \frac{X_i - \mu_i}{\sqrt{\sigma_{ii}}}, \quad i = 1(1)p; \quad \sigma_{ii} > 0$$

define $V = \text{diag}(\sigma_{11}, \ldots, \sigma_{pp})$

$$\underset{\sim}{Z} = (Z_1, \ldots, Z_p)' = V^{-1/2}(\underset{\sim}{X} - \underset{\sim}{\mu})$$

$$E\,\underset{\sim}{Z} = \underset{\sim}{0}; \quad \text{Cov}(\underset{\sim}{Z}) = V^{-1/2}\Sigma V^{-1/2} = \rho$$

Let $(\eta_i, \underset{\sim}{f}_i)$ $(i = 1(1)p)$, $\eta_1 \geq \eta_2 \cdots \geq \eta_p \geq 0$ be the eigen value - eigen vector (e.v.) pairs of ρ matrix, then

(a) i^{th} PC derived from ρ is

$$Y_i = \underset{\sim}{f}_i' \underset{\sim}{Z} = \underset{\sim}{f}_i' V^{-1/2}(\underset{\sim}{X} - \underset{\sim}{\mu})$$

(b) total variation in $\underset{\sim}{Y} = \sum_{i=1}^{p} V(Y_i) = \sum_{i=1}^{p} V(Z_i) = p$

(total variation in $\underset{\sim}{Z}$)

(c) $\rho_{Z_i, Y_j} = f_{ij}\sqrt{\eta_j}$; $i, j = 1(1)p$

Remark: (i) Proportion of total variation in $\underset{\sim}{X}$ explained by the k^{th} PC (based on Σ) is

$$\lambda_k \Big/ \sum_{1}^{p} \lambda_i$$

(ii) Proportion of total variation in $\underset{\sim}{X}$ explained by the first k PCs (based on Σ) is

$$\sum_{i=1}^{k} \lambda_i \Big/ \sum_{i=1}^{p} \lambda_i$$

(iii) If $\sum_{i=1}^{k} \lambda_i \approx \sum_{i=1}^{p} \lambda_i$ for some small k, then it is possible to have a meaningful data dimension reduction as it is enough in such a situation to consider only $(Y_1, \ldots, Y_k)$. Ideally $k \ll p$!

Remark: $e_{ij} \propto \rho_{X_i, Y_j}$ and hence the magnitude of e_{ij} indicates how important is X_i to Y_j, this helps in interpreting Y_j

Remark: In case the units of variables are different or the variables have widely varying ranges, we should obtain PCs from standardized variables i.e. work with ρ instead of Σ to get PCs.

Examples

(1) $$\Sigma = \begin{pmatrix} 1 & -2 & 0 \\ & 5 & 0 \\ & & 2 \end{pmatrix}$$

Soln $|\Sigma - \lambda I| = 0$ gives $\lambda_1 = 5.83$, $\lambda_2 = 2.0$, $\lambda_3 = 0.17$

$\underset{\sim}{e}_1' = (0.383, -.924, 0)$

$\underset{\sim}{e}_2' = (0, 0, 1)$

$\underset{\sim}{e}_3' = (.924, 0.383, 0)$

PCs

$$Y_1 = .383 X_1 - .924 X_2$$

$$Y_2 = X_3$$

$$Y_3 = .924 X_1 + .383 X_2$$

1st PC explains $\frac{5.83}{8} \times 100 = 73\%$ of total variation

1st 2 PCs explain $\frac{5.83 + 2.0}{8} \times 100 = 98\%$ of total variation

$$\rho_{X_1, Y_1} = .383 \sqrt{\frac{5.83}{1}} \approx .925 \qquad \left(\rho_{X_i, X_j} = e_{ij} \sqrt{\frac{\lambda_j}{\sigma_{ii}}} \right)$$

$$\rho_{X_2, Y_1} = (-.924) \sqrt{\frac{5.83}{5}} \approx -.998$$

- - .

Example 2

$$\Sigma = \begin{pmatrix} 1 & 4 \\ 4 & 100 \end{pmatrix} \quad X_1, X_2 \rightarrow \text{widely varying in spread}$$

$\lambda_1 = 100.16$, $\lambda_2 = 0.84$

$\underset{\sim}{e}_1' = (.04, .999)$; $\underset{\sim}{e}_2' = (.999, -0.04)$

$Y_1 = .04X_1 + .999X_2$; $Y_2 = .999X_1 - .04X_2$

Y_1 explains $\frac{100.16}{101} \times 100 \approx 99\%$ of total variation

$$\rho_{X_1, Y_1} = 0.04\sqrt{\frac{100.16}{1}} \approx 0.4$$

$$\rho_{X_2, Y_1} = .999\sqrt{\frac{100.16}{100}} \approx .99$$

X_2 is much more imp to Y_1 than X_1 is to Y_1

$\text{Var}(X_2)$ dominating!

Consider standardized variables

$$Z_1 = \frac{X_1 - \mu_1}{\sqrt{\sigma_{11}}} \qquad Z_2 = \frac{X_2 - \mu_2}{\sqrt{\sigma_{22}}}$$

$$\Sigma_Z = \rho_X = \begin{pmatrix} 1 & .4 \\ .4 & 1 \end{pmatrix}$$

$\eta_1 = 1.4$; $\eta_2 = 0.6$

$\underset{\sim}{f}_1' = (.707, .707)$; $\underset{\sim}{f}_2' = (.707, -.707)$

$Y_1 = .707Z_1 + .707Z_2$

$Y_2 = .707Z_1 - .707Z_2$

$$\rho_{Z_1, Y_1} = .707\sqrt{\frac{1.4}{1}} = \rho_{Z_2, Y_1}$$

Z_1 & Z_2 are equally important

Y_1 explains $\frac{1.4}{2} \times 100 = 70\%$ of total variation

Principal components from data matrix

Observed data matrix

$$\underset{p\times n}{\mathcal{X}} = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ x_{21} & \cdots & x_{2n} \\ \vdots & & \vdots \\ x_{p1} & \cdots & x_{pn} \end{pmatrix}$$

$\underset{\sim}{x}_1, \ldots, \underset{\sim}{x}_n$ n obsns from multivariate pop^n with mean vector μ and cov mat Σ

Sample mean vector: $\bar{\underset{\sim}{x}} = \begin{pmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_p \end{pmatrix} = \frac{1}{n}\,\mathcal{X}\,\underset{\sim}{1}_n$

Sample variance covariance matrix:

$$S_{n-1} = \frac{1}{n-1}\,\mathcal{X}\left(I_n - \frac{1}{n}\,\underset{\sim}{1}_n\,\underset{\sim}{1}_n'\right)\mathcal{X}' = \frac{1}{n-1}\sum(\underset{\sim}{x}_j - \bar{\underset{\sim}{x}})(\underset{\sim}{x}_j - \bar{\underset{\sim}{x}})'$$

$$= \frac{1}{n-1}\begin{pmatrix} \sum(x_{1j}-\bar{x}_1) & \sum(x_{1j}-\bar{x}_1)(x_{2j}-\bar{x}_2) & \cdots & \sum(x_{1j}-\bar{x}_1)(x_{pj}-\bar{x}_p) \\ & & & \\ & & & \\ & & & \sum(x_{pj}-\bar{x}_p)^2 \end{pmatrix}$$

From the observed data matrix, our objective is to construct uncorrelated linear combinations of the measured characteristics that account for as much of the sample variation as possible.

Suppose $\underset{\sim}{l}'\underset{\sim}{x}$ is sample linear combination

We have $\underset{\sim}{l}'\underset{\sim}{x}_1, \ldots \underset{\sim}{l}'\underset{\sim}{x}_n$: n transformed obsns

Sample mean of $(\underset{\sim}{l}'\underset{\sim}{x}_1, \ldots, \underset{\sim}{l}'\underset{\sim}{x}_n)$:

$$\frac{1}{n}\sum_1^n \underset{\sim}{l}'\underset{\sim}{x}_j$$

$$= \frac{1}{n}\sum_{j=1}^n \left(l_1 x_{1j} + l_2 x_{2j} + \cdots + l_p x_{pj}\right)$$

$$= l_1\bar{x}_1 + l_2\bar{x}_2 + \cdots + l_p\bar{x}_p = \underset{\sim}{l}'\bar{\underset{\sim}{x}}$$

Sample variance of $(\underset{\sim}{l}'\underset{\sim}{x}_1, \ldots, \underset{\sim}{l}'\underset{\sim}{x}_n)$:

$$\frac{1}{n-1}\left((\underset{\sim}{l}'\underset{\sim}{x}_1 - \underset{\sim}{l}'\bar{\underset{\sim}{x}})^2 + \cdots + (\underset{\sim}{l}'\underset{\sim}{x}_n - \underset{\sim}{l}'\bar{x})^2\right)$$

$$= \frac{1}{n-1}\left(\left(\underset{\sim}{l}'(\underset{\sim}{x}_1 - \bar{\underset{\sim}{x}})\right)^2 + \cdots + \left(\underset{\sim}{l}'(\underset{\sim}{x}_n - \bar{\underset{\sim}{x}})\right)^2\right)$$

$$= \frac{1}{n-1}\left(\left(\underset{\sim}{l}'(\underset{\sim}{x}_1 - \bar{\underset{\sim}{x}})\right)\left(\underset{\sim}{l}'(\underset{\sim}{x}_1 - \bar{\underset{\sim}{x}})\right)' + \cdots + \left(\underset{\sim}{l}'(\underset{\sim}{x}_n - \bar{\underset{\sim}{x}})\right)\left(\underset{\sim}{l}'(\underset{\sim}{x}_n - \bar{\underset{\sim}{x}})\right)'\right)$$

$$= \frac{1}{n-1}\left(\underset{\sim}{l}'(\underset{\sim}{x}_1 - \bar{\underset{\sim}{x}})(\underset{\sim}{x}_1 - \bar{\underset{\sim}{x}})'\underset{\sim}{l} + \cdots + \underset{\sim}{l}'(\underset{\sim}{x}_n - \bar{\underset{\sim}{x}})(\underset{\sim}{x}_n - \bar{\underset{\sim}{x}})'\underset{\sim}{l}\right)$$

$$= \underset{\sim}{l}'\left(\frac{1}{n-1}\sum_{j=1}^{n}(\underset{\sim}{x}_j - \bar{\underset{\sim}{x}})(\underset{\sim}{x}_j - \bar{\underset{\sim}{x}})'\right)\underset{\sim}{l}$$

$$= \underset{\sim}{l}' S_{n-1}\, \underset{\sim}{l}$$

Further, suppose $\underset{\sim}{l}_i'\underset{\sim}{x}$ & $\underset{\sim}{l}_j'\underset{\sim}{x}$ be 2 linear combinations,

Sample covariance bet^n n pairs

$$(\underset{\sim}{l}_i'\underset{\sim}{x}_1, \underset{\sim}{l}_j'\underset{\sim}{x}_1), \ldots (\underset{\sim}{l}_i'\underset{\sim}{x}_n, \underset{\sim}{l}_j'\underset{\sim}{x}_n)$$

$$\frac{1}{n-1}\left((\underset{\sim}{l}_i'\underset{\sim}{x}_1 - \underset{\sim}{l}_i'\bar{\underset{\sim}{x}})(\underset{\sim}{l}_j'\underset{\sim}{x}_1 - \underset{\sim}{l}_j'\bar{\underset{\sim}{x}}) + \cdots + (\underset{\sim}{l}_i'\underset{\sim}{x}_n - \underset{\sim}{l}_i'\bar{\underset{\sim}{x}})(\underset{\sim}{l}_j'\underset{\sim}{x}_n - \underset{\sim}{l}_j'\bar{\underset{\sim}{x}})\right)$$

$$= \frac{1}{n-1}\sum_{t=1}^{n}\left(\underset{\sim}{l}_i'(\underset{\sim}{x}_t - \bar{\underset{\sim}{x}})(\underset{\sim}{x}_t - \bar{\underset{\sim}{x}})'\underset{\sim}{l}_j'\right)$$

$$= \underset{\sim}{l}_i'\left(\frac{1}{n-1}\sum_{t=1}^{n}(\underset{\sim}{x}_t - \bar{\underset{\sim}{x}})(\underset{\sim}{x}_t - \bar{\underset{\sim}{x}})'\right)\underset{\sim}{l}_j$$

$$= \underset{\sim}{l}_i' S_{n-1}\, \underset{\sim}{l}_j$$

Defⁿ: Sample Principal Components

The sample PCs are uncorrelated linear combinations of observation vectors, have sample variances in decreasing order. The 1st sample PC is the lin comb $\underset{\sim}{l}_1' \underset{\sim}{x}_j$ which maximizes the sample variance of $\underset{\sim}{l}' \underset{\sim}{x}_j$ subject to $\underset{\sim}{l}' \underset{\sim}{l} = 1$ (i.e. sample variance of $(\underset{\sim}{l}_1' \underset{\sim}{x}_1, \ldots \underset{\sim}{l}_1' \underset{\sim}{x}_n)$

$= \max_{\underset{\sim}{l} \ni \underset{\sim}{l}'\underset{\sim}{l}=1} \underset{\sim}{l}' S_{n-1} \underset{\sim}{l}$).

The 2nd sample PC is the linear comb $\underset{\sim}{l}_2' \underset{\sim}{x}_j \ni$ it maximizes sample variance of $\underset{\sim}{l}' \underset{\sim}{x}_j \ni \underset{\sim}{l}' \underset{\sim}{l} = 1$ and zero sample covariance for the pairs $(\underset{\sim}{l}_1' \underset{\sim}{x}_j, \underset{\sim}{l}' \underset{\sim}{x}_j); j = 1(1)n$

The i^{th} sample PC is the linear comb $\underset{\sim}{l}_i' \underset{\sim}{x}_j \ni$ it maximizes the sample variance of $\underset{\sim}{l}' \underset{\sim}{x}_j$ and having zero sample covariance for all pairs $(\underset{\sim}{l}' \underset{\sim}{x}_j, \underset{\sim}{l}_k' \underset{\sim}{x}_j)$

$\forall k < i : j = 1(1)n$.

S_{n-1}: Sample variance covariance matrix

$(\hat{\lambda}_1, \hat{\underset{\sim}{e}}_1), \ldots, (\hat{\lambda}_p, \hat{\underset{\sim}{e}}_p)$: eigenvalue – eigenvector(o.n.) pairs of S_{n-1}

$\max_{\underset{\sim}{l} \ni \underset{\sim}{l}'\underset{\sim}{l}=1} \underset{\sim}{l}' S_{n-1} \underset{\sim}{l} = \hat{\lambda}_1 =$ sample variance of

$(\hat{\underset{\sim}{e}}_1' \underset{\sim}{x}_1, \ldots, \hat{\underset{\sim}{e}}_1' \underset{\sim}{x}_n)$

$\left(= \hat{\underset{\sim}{e}}_1' S_{n-1} \hat{\underset{\sim}{e}}_1 \right.$

$\left. = \hat{\underset{\sim}{e}}_1' \left(\sum_1^n \hat{\lambda}_j \hat{\underset{\sim}{e}}_j \hat{\underset{\sim}{e}}_j' \right) \hat{\underset{\sim}{e}}_1 = \hat{\lambda}_1 \right)$

1st sample PC is $\hat{e}_1' x_j = \hat{y}_{1(j)}$

For 2nd sample PC we need to find

$$\max_{l \ni l \perp \hat{e}_1,\ l'l = 1} l' S_{n-1} l = \hat{\lambda}_2 = \text{Sample variance of } (\hat{e}_2' x_1, \ldots, \hat{e}_2' x_n)$$

2nd sample PC is $\hat{e}_2' x_j = \hat{y}_{2(j)}$

k^{th} $(k \le p)$ sample PC is $\hat{e}_k' x_j = \hat{y}_{k(j)}$

Note (i) Sample variance of $\hat{y}_k = \hat{\lambda}_k$

(ii) Sample covariance of $\hat{y}_i$ & $\hat{y}_k = 0 \quad \forall\ i \neq k$

(Sample covariance of $(\hat{e}_i' x_j, \hat{e}_k' x_j)$ pairs

$$= \hat{e}_i' S_{n-1} \hat{e}_k$$

$$= \hat{e}_i' \left(\sum_{j=1}^{n} \hat{\lambda}_j \hat{e}_j \hat{e}_j' \right) \hat{e}_k = 0$$

(iii) total sample variation $= \sum^{p} s_{ii}$

$=$ total sample variation in sample PCs

$$= \sum_{i=1}^{p} \hat{\lambda}_i$$

(iv) $$r_{x_i, \hat{y}_k} = \hat{e}_{ik} \sqrt{\frac{\hat{\lambda}_k}{s_{kk}}}$$

- - - -

Uses of PCA

Sample PC's $\quad \hat{y}_i = \hat{e}_i' \underset{\sim}{x} \; ; \; i = 1(1)p$

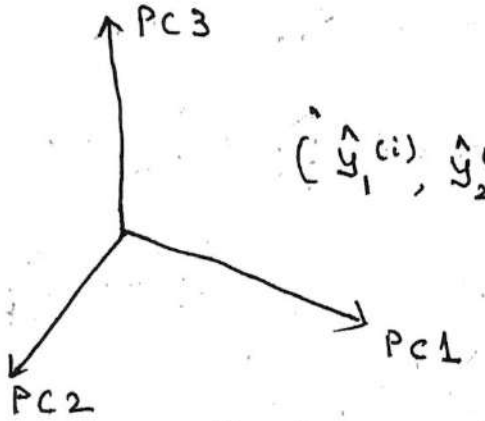
$\underset{\sim}{x}_1 \longrightarrow (\hat{y}_1^{(1)}, \hat{y}_2^{(1)}, \ldots, \hat{y}_p^{(1)})$

$\underset{\sim}{x}_2 \longrightarrow (\hat{y}_1^{(2)}, \hat{y}_2^{(2)}, \ldots, \hat{y}_p^{(2)})$

$\underset{\sim}{x}_n \longrightarrow (\hat{y}_1^{(n)}, \hat{y}_2^{(n)}, \ldots, \hat{y}_p^{(n)})$

(i) Data projection & visualization

Projection onto PC plane



$(\hat{y}_1^{(i)}, \hat{y}_2^{(i)}, \hat{y}_3^{(i)}) \quad i = 1(1)n$

- Meaningful when $(\hat{y}_1, \hat{y}_2, \hat{y}_3)$ collectively explains a "significant" amount of total sample variation
- a similar 2d plot on (PC1, PC2) plane, provided the first 2 PC collectively captures "significant" proportion of total sample variation

(ii) Outlier detection : from 2d / 3d plot (PC plane)

(iii) Cluster detection : from above plots

(iv) Data dimension reduction

through $(\hat{y}_1, \ldots, \hat{y}_K)$ for $K < p$

(v) Ranking of cases

$\underset{\sim}{x}_1 \rightarrow \hat{y}_1^{(1)}$

$\underset{\sim}{x}_2 \rightarrow \hat{y}_1^{(2)}$

$\vdots$

$\underset{\sim}{x}_n \rightarrow \hat{y}_1^{(n)}$

} rank based on $\hat{y}_1^{(i)}$ Coeffs of variables (original) play a crucial role

(vi) variable clustering

- $$r_{\hat{y}_K, x_i} = \hat{e}_{iK} \sqrt{\frac{\hat{\lambda}_K}{s_{ii}}}$$
- Represent each of the original p variables by $\left(\hat{r}_{\hat{y}_1, x_i}, \hat{r}_{\hat{y}_2, x_i}, \hat{r}_{\hat{y}_3, x_i}\right)$ $i = 1(1)p$
- Identify variable clusters from this plot

(vii) checking for multivariate normality

- Check for univariate normality of each of the p PCs
- For i^{th} PC use $\left(\hat{y}_i^{(1)}, \ldots, \hat{y}_i^{(n)}\right)$ to check for univariate normality
- If any of the p checks for univariate normality fails, N_p assumption will be rejected

Cluster Analysis

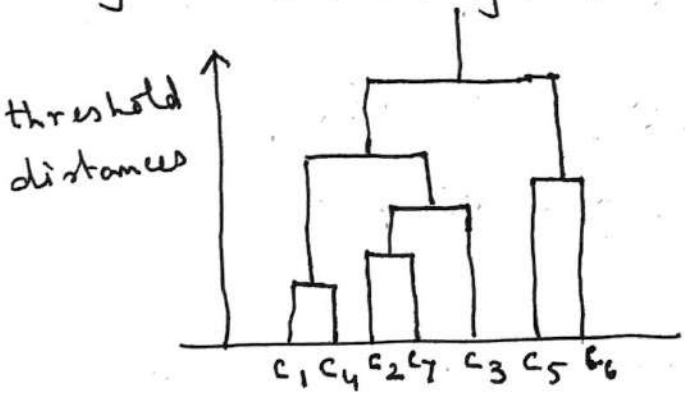
Grouping of objects (with multidimensional feature vectors) based on self similarities ("closeness").

Two approaches

- Hierarchical clustering
- non-hierarchical clustering

Hierarchical clustering

In HCA the observation vectors (cases) are grouped together on the basis of mutual distances in such a manner so as to ensure hierarchy in cluster formation. HCA output is usually visualized through a hierarchical tree, which is a nested set of partitions represented by a tree diagram called dendogram.



Dendogram tree

Characteristics

(i) Sectioning a tree at a particular level produces a partition into g ~~gr~~ disjoint groups

(ii) If two groups are chosen from different partitions (i.e. the results of partitioning at different levels) then either the groups are disjoint or one group contains the other

(iii) a numerical value is associated with each partition up/down the tree where branches join. This is a measure of distance between 2 merged clusters.

(iv) We get different trees corresponding to different distance measures (bet^n cases).

Two algorithms for HCA

- Agglomerative clustering algorithm
- Divisive clustering algorithm

Agglomerative clustering algorithm

- Operates with successive merger of objects
- Starts with n clusters, each containing a single data point.
- At each stage merges the 2 most similar groups to form a new cluster, thus reducing the number of clusters by 1.
- The process of merger of group of objects continue till all subgroups are fused together to form a single cluster

Divisive clustering algorithm

- Operates by successively splitting groups of objects
- starts with a single cluster having all the n objects
- Initial group split into 2 clusters ⇒ the distance bet^n the split groups ~~are~~ is maximum, i.e the subgroups are as far as possible.

- Process of splitting continues till there are n clusters, each having a single object (leaf node)
- DCA is computationally inefficient.

Note: Result from agglomerative (or divisive) clustering algorithm displayed through dendogram.

Steps in agglomerative hierarchical algorithm

Step I : n clusters, each having a single object and an $n \times n$ symmetric matrix of distances ($D = (\!(d_{ij})\!)$)

Step II : Search the distance matrix for the nearest (most similar) pair of clusters Let the distance bet^n the most similar clusters, say U and V, be d_{UV}.

Step III : Merge U & V to form cluster (U, V), at a merger (fusion) level d_{UV}.
Update the distance matrix by
(i) deleting the rows & col^{m}s corresponding to clusters U & V
and (ii) adding a row & col^m giving the distance bet^n (U, V) and the remaining clusters (existing ones other than U & V).

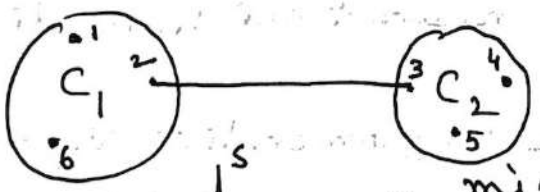
Step IV : Repeat (II) & (III) n-1 times, recording the identity of the clusters that are merged and the merger levels (i.e the

distances at which mergers take place).

Step 2 : Construct the dendogram tree from the information on mergers and level of mergers.

Remark : step iii requires calculation of distance bet^n clusters (for distance matrix updation). The following are commonly used distance measures bet^n clusters of objects

(i) Single link distance (minimum distance or nearest neighbor distance)



$$d^s_{C_1,C_2} = \min_{\substack{i \in C_1 \\ j \in C_2}} d_{ij}$$

(ii) Complete link distance (maximum distance or farthest distance)

$$d^c_{C_1,C_2} = \max_{\substack{i \in C_1 \\ j \in C_2}} d_{ij}$$

(iii) Average link distance

N_{C_1} : # of objects in C_1

N_{C_2} : # of objects in C_2

$$d^A_{C_1,C_2} = \frac{1}{N_{C_1} N_{C_2}} \sum_{\substack{i \in C_1 \\ j \in C_2}} d_{ij}$$

(iv) Centroid distance : Distance bet^n cluster means

(v) Median distance : Distance bet^n cluster medians

Remark : Single linkage agglomerative algorithm

In step iii (distance matrix updation stage), once compute the distance bet^n (U, V) and any existing cluster W as

$$d_{(U,V),W} = \min(d_{UW}, d_{VW})$$

(d_{UW}, d_{VW} present in prev round)

d_{UW} & d_{VW} are nearest neighbor distances

Remark : Complete linkage agglomerative algorithm

In Step iii, compute distance bet^n (U, V) and any existing W as

$$d_{(U,V),W} = \max(d_{UW}, d_{VW})$$

Remark : Average linkage agglomerative algorithm

In step iii, compute distance bet^n (U, V) and W computed as

$$d_{(U,V),W} = \frac{\sum_{i \in (U,V)} \sum_{K \in W} d_{iK}}{N_{(U,V)} \, N_W}$$

d_{iK} : distance bet^n object i in the cluster (U, V) and object K in cluster W

$N_{(U,V)}$: # of objects in (U, V) & N_W : # of objects in W

Example: Single linkage clustering

Five objects c_1, c_2, c_3, c_4, c_5

Distance matrix

$$D = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{array} \begin{pmatrix} 0 & & & & \\ 9 & 0 & & & \\ 3 & 7 & 0 & & \\ 6 & 5 & 9 & 0 & \\ 11 & 10 & 2 & 8 & 0 \end{pmatrix} \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{array}$$

$d_{c_3 c_5} = \min d_{ij} = 2$

$\Rightarrow$ Merge c_3 & c_5 to form cluster (c_3, c_5) at merger level 2 – total 4 clusters

Row col^m deletion containing c_3 & c_5

$$\begin{array}{c} \\ c_3 \\ c_5 \end{array} \begin{pmatrix} 0 & & & & \\ 9 & 0 & & & \\ 3 & 7 & 0 & & \\ 6 & 5 & 9 & 0 & \\ 11 & 10 & 2 & 8 & 0 \end{pmatrix} \begin{array}{c} \\ c_3 \\ c_5 \end{array}$$

Computation of $d_{(c_3, c_5), c_i}$ for $i = 1, 2, 4$

$$d_{(c_3, c_5) c_1} = \min(d_{c_1 c_3}, d_{c_1 c_5}) = \min(3, 11) = 3$$

$$d_{(c_3, c_5) c_2} = \min(d_{c_2 c_3}, d_{c_2 c_5}) = \min(7, 10) = 7$$

$$d_{(c_3, c_5) c_4} = \min(d_{c_3 c_4}, d_{c_4 c_5}) = \min(9, 8) = 8.$$

Updated distance matrix

$$\begin{array}{c} \\ (C_3, C_5) \\ C_1 \\ C_2 \\ C_4 \end{array} \begin{pmatrix} 0 & & & \\ \boxed{3} & 0 & & \\ 7 & 9 & 0 & \\ 8 & 6 & 5 & 0 \end{pmatrix}$$

$d_{(C_3, C_5) C_1}$ is min

Merge (C_3, C_5) with C_1 to form cluster (C_1, C_3, C_5) at merger level 3 – total 3 clusters

Row column deletion step

$$\begin{array}{c} \\ (C_3, C_5) \\ C_1 \\ C_2 \\ C_4 \end{array} \begin{pmatrix} 0 & & & \\ 3 & 0 & & \\ 7 & 9 & 0 & \\ 8 & 6 & 5 & 0 \end{pmatrix}$$

Calculate distance bet^n (C_3, C_5, C_1) & C_2 and (C_3, C_5, C_1) & C_4

$$d_{(C_1, C_3, C_5) C_2} = \min\left(d_{C_1 C_2}, d_{(C_3, C_5) C_2}\right)$$

$$= \min(9, 7) = 7$$

$$d_{(C_1, C_3, C_5) C_4} = \min\left(d_{C_1 C_4}, d_{(C_3, C_5)(C_4)}\right)$$

$$= \min(6, 8) = 6$$

Update distance matrix

$$\begin{matrix}(C_1, C_3, C_5) \\ C_2 \\ C_4\end{matrix}\begin{pmatrix} 0 & & \\ 7 & 0 & \\ 6 & \boxed{5} & 0 \end{pmatrix}$$

$d_{C_2 C_4}$ is min

Merge C_2 & C_4 to form cluster (C_2, C_4) at merger level 5 – total 2 clusters (C_1, C_3, C_5) & (C_2, C_4)

Row column deletion

$$\begin{matrix}(C_1, C_3, C_5) \\ C_2 \\ C_4\end{matrix}\begin{pmatrix} 0 & | & || \\ 7 & 0 & | \\ 6 & 5 & 0 \end{pmatrix}$$

Distance matrix updation

$$\begin{matrix}(C_1, C_3, C_5) \\ (C_2, C_4)\end{matrix}\begin{pmatrix} 0 & \\ ? & 0 \end{pmatrix}$$

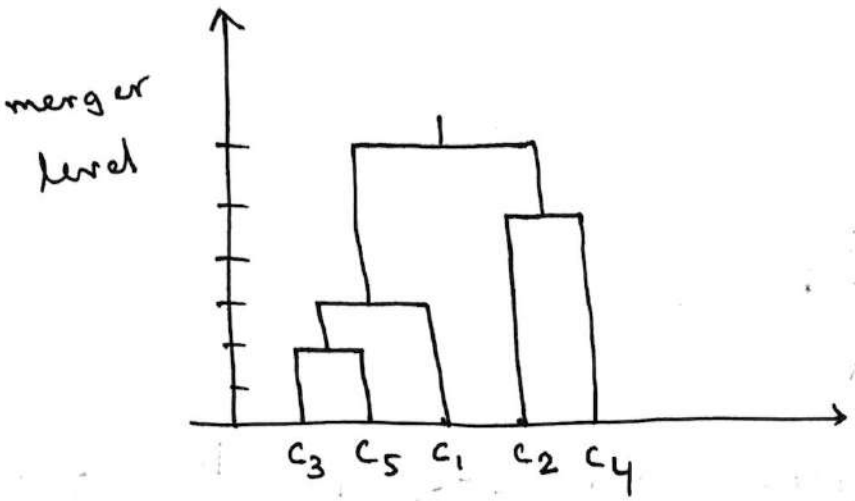
$$d_{(C_1, C_3, C_5)(C_2, C_4)} = \min\left(d_{(C_1, C_3, C_5) C_2}, d_{(C_1, C_3, C_5) C_4}\right)$$

$$= \min(7, 6) = 6$$

$$\begin{matrix}(C_1, C_3, C_5) \\ (C_2, C_4)\end{matrix}\begin{pmatrix} 0 & \\ 6 & 0 \end{pmatrix}$$

⇒ Merge (C_1, C_3, C_5) & (C_2, C_4) at merger level 6 – Single cluster

Dendogram



Measures of dissimilarity

$\underset{p\times n}{\mathcal{X}} = (\underset{\sim}{x}_1 : \cdots : \underset{\sim}{x}_n)$

$\mathcal{X} \to \underset{p\times p}{D}$: dissimilarity matrix

$D = ((d_{rs}))$ ∋

(i) $d_{rs} \geq 0 \quad \forall\, r, s$

(ii) $d_{rr} = 0 \quad \forall\, r$

(iii) $d_{rs} = d_{sr} \quad \forall\, r, s$

Remark: (i) Symmetry cond^n may not always be satisfied

$D \to (D + D')/2$

(ii) If in addition to the cond^{n}s (i)–(iii), d_{rs} also satisfies triangle inequality

$d_{rs} \leq d_{rt} + d_{ts} \quad \forall\, r, s, t$

then the dissimilarity measure is a metric and we use the term "distance"

Some commonly used dissimilarity measures for numeric values

$\underset{\sim}{x}_i$, $\underset{\sim}{x}_j$: $p\times 1$ feature vector $\underset{\sim}{x}_i = \begin{pmatrix} x_{1i} \\ \vdots \\ x_{li} \\ \vdots \\ x_{pi} \end{pmatrix}$

(a) Euclidean distance

$$d_e = \left(\sum_{l=1}^{p} (x_{li} - x_{lj})^2 \right)^{1/2}$$

(b) Absolute value distance or City-block distance

$$d_{cb} = \sum_{l=1}^{p} |x_{li} - x_{lj}|$$

(c) Chebyshev distance

$$d_{ch} = \max_{l} |x_{li} - x_{lj}|$$

(d) Pearson distance

$$d_p = \left(\sum_{l=1}^{p} (x_{li} - x_{lj})^2 / s_l^2 \right)^{1/2}$$

s_l^2 : sample variance for l^{th} variable

(e) Quadratic distance

$$d_q = \left((\underset{\sim}{x}_i - \underset{\sim}{x}_j)' Q (\underset{\sim}{x}_i - \underset{\sim}{x}_j) \right)^{1/2}$$

Q is p.d.

e.g $Q = S^{-1}$: S = within group covariance matrix

Case of feature vector with ordinal values

Realizable values are ordered set.

e.g Academic grades: A, B, C, D, E, F

opinion ~~Opinion~~ : strongly disagree, disagree, neutral, agree, strongly agree

One approach is to transform M (say) ordinal values to

$$\frac{j - \frac{1}{2}}{M} \quad ; j = 1(1)M$$

in the prescribed order of their orginal values.

Treat the transformed values as ~~as~~ quantitative numeric values.

Only

Case of nominal (categorical, unordered) / ordinal values

Transform to binary variables set.

Nominal variable with K states

$\longrightarrow$ represent as K binary variables

nominal variable at m^{th} state

$$(0, \ldots \underset{\uparrow}{1}, \ldots 0)$$
m^{th}

Sly for ordinal

Similarity measures for binary variables

Let $\underset{\sim}{x}_{p\times 1}$ & $\underset{\sim}{y}_{p\times 1}$ be 2 vectors of binary variables.

Define

a : # of occurrences of $x_i = 1$ and $y_i = 1$

b : # of occurrences of $x_i = 0$ & $y_i = 1$

c : . . . . . $x_i = 1$ & $y_i = 0$

d : . . . . . $x_i = 0$ & $y_i = 0$

$a + b + c + d = p$, # of variables.

Some similarity measures:

(a) Simple matching coefficient:

$$s_{SM} = \frac{a+d}{a+b+c+d}$$

(b) Russell & Rao

$$s_{RR} = \frac{a}{a+b+c+d}$$

(c) Jaccard $\quad s_J = \frac{a}{a+b+c}$

(d) Czekanowski:

$$s_{cz} = \frac{2a}{2a+b+c}$$

Note:
dissimilarity measure
e.g. $d_{xy} = 1 - s_{xy}$ for s_{sm}

Mixed type : dimensions have both numerical / categorical

Transform all to binary.

Example

	Height*	Weight⊛	Eye color	Hair color	Handedness	Gender
Ind 1	68	140	green	blond	R	F
Ind 2	73	185	brown	brown	R	M
Ind 3	67	165	blue	blond	R	M
Ind 4	64	120	brown	brown	R	F
Ind 5	76	210	brown	brown	L	M

* : in inch

⊛ : in lb

Define, $X_{1i} = \begin{cases} 1, & ht \geq 72 \text{ in} \\ 0, & o/w. \end{cases}$ $X_{2i} = \begin{cases} 1, & wt \geq 150 \text{ lb} \\ 0, & o/w \end{cases}$

$X_{3i} = \begin{cases} 1, & \text{brown eyes} \\ 0, & o/w \end{cases}$ $X_{4i} = \begin{cases} 1, & \text{blond hair} \\ 0, & o/w \end{cases}$

$X_{5i} = \begin{cases} 1, & R \\ 0, & L \end{cases}$ $X_{6i} = \begin{cases} 1, & \text{female} \\ 0, & \text{male} \end{cases}$

	X_1	X_2	X_3	X_4	X_5	X_6
Ind 1	0	0	0	1	1	1
Ind 2	1	1	1	0	1	0

→

Ind 2 \ Ind 1	1	0
1	1 (a)	2 (b)
0	3 (c)	0 (d)

$$S_{sm}^{(1,2)} = \frac{1}{6}$$

Compute $\forall (i,j)$ $i \neq j$ pairs and get Similarity matrix

$$S = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 1 & & & & & \\ 1/6 & 1 & & & & \\ - & - & 1 & & & \\ - & - & - & 1 & & \\ - & & - & - & 1 & \\ - & - & - & - & - & 1 \end{pmatrix} \end{array}$$

Distance b/t groups of objects

(A) Methods based on feature vectors

Already discussed (single linkage, complete linkage, average linkage, centroid distance, median distance)

(B) Methods based on probabilistic distance

Distance measures use complete information about the structure of groups (classes) provided by the class conditional densities. The distance measure, say J, satisfies

(i) $J = 0$ if $p(\underline{x}|\pi_1) = p(\underline{x}|\pi_2)$

(ii) $J \geq 0$

(iii) J attains the maximum value when the classes are disjoint

Commonly used probabilistic distances

(I) Bhattacharya distance

$$J_B = -\log\left(\int \left\{ p(\underline{x}|\pi_1)\, p(\underline{x}|\pi_2) \right\}^{1/2} d\underline{x} \right)$$

(II) Chernoff distance

$$J_C = -\log\left(\int \left\{ p(\underline{x}|\pi_1) \right\}^{s} \left\{ p(\underline{x}|\pi_2) \right\}^{1-s} d\underline{x} \right)$$

$s \in [0,1]$

(III) Divergence distance

$$J_D = \int \left(p(\underline{x}|\pi_1) - p(\underline{x}|\pi_2) \right) \log\left(\frac{p(\underline{x}|\pi_1)}{p(\underline{x}|\pi_2)} \right) d\underline{x}$$

(iv) Patrick - Fischer distance

$$J_{PF} = \left(\int \left\{ p_1 p(\underset{\sim}{x}|\pi_1) - p_2 p(\underset{\sim}{x}|\pi_2) \right\}^2 d\underset{\sim}{x} \right)^{1/2}$$

p_1 & p_2 are prior probabilities of the classes.

Remark: Probabilistic distances are computed under specific distributional setups.

Example: Gr I objects $- N_p(\underset{\sim}{\mu}_1, \Sigma_1)$ $\quad \Sigma_i > 0$

Gr II objects $- N_p(\underset{\sim}{\mu}_2, \Sigma_2)$

Divergence distance calculations

$$J_D = \int \left((2\pi)^{-p/2} |\Sigma_1|^{-1/2} \exp\left(-\tfrac{1}{2}(\underset{\sim}{x} - \underset{\sim}{\mu}_1)' \Sigma_1^{-1} (\underset{\sim}{x} - \underset{\sim}{\mu}_1)\right) \right.$$

$$\left. - (2\pi)^{-p/2} |\Sigma_2|^{-1/2} \exp\left(-\tfrac{1}{2}(\underset{\sim}{x} - \underset{\sim}{\mu}_2)' \Sigma_2^{-1} (\underset{\sim}{x} - \underset{\sim}{\mu}_2)\right) \right)$$

$$\left(\log \frac{|\Sigma_2|^{1/2}}{|\Sigma_1|^{1/2}} + \log \left(\exp\left(-\tfrac{1}{2} \left\{ (\underset{\sim}{x} - \underset{\sim}{\mu}_1)' \Sigma_1^{-1} (\underset{\sim}{x} - \underset{\sim}{\mu}_1) - (\underset{\sim}{x} - \underset{\sim}{\mu}_2)' \Sigma_2^{-1} (\underset{\sim}{x} - \underset{\sim}{\mu}_2) \right\} \right) \right) \right)$$

$$= \log \frac{|\Sigma_2|^{1/2}}{|\Sigma_1|^{1/2}} \int \left(p(\underset{\sim}{x}|\pi_1) - p(\underset{\sim}{x}|\pi_2) \right) d\underset{\sim}{x}$$

$$+ \int \left(-\tfrac{1}{2}(\underset{\sim}{x} - \underset{\sim}{\mu}_1)' \Sigma_1^{-1} (\underset{\sim}{x} - \underset{\sim}{\mu}_1) \right) \left(p(\underset{\sim}{x}|\pi_1) - p(\underset{\sim}{x}|\pi_2) \right) d\underset{\sim}{x}$$

$$+ \int \left(\tfrac{1}{2}(\underset{\sim}{x} - \underset{\sim}{\mu}_2)' \Sigma_2^{-1} (\underset{\sim}{x} - \underset{\sim}{\mu}_2) \right) \left(p(\underset{\sim}{x}|\pi_1) - p(\underset{\sim}{x}|\pi_2) \right) d\underset{\sim}{x}$$

$$= \left(1 - 1\right) \log \frac{|\Sigma_2|^{1/2}}{|\Sigma_1|^{1/2}}$$

$$+ \left(-\frac{1}{2}\right) \int \left(\underset{\sim}{x}' \Sigma_1^{-1} \underset{\sim}{x} + \underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{\mu}_1 - 2 \underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{x}\right)$$
$$\left(p(\underset{\sim}{x}|\pi_1) - p(\underset{\sim}{x}|\pi_2)\right) d\underset{\sim}{x}$$

$$+ \left(\frac{1}{2}\right) \int \left(\underset{\sim}{x}' \Sigma_2^{-1} \underset{\sim}{x} + \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{\mu}_2 - 2 \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{x}\right)$$
$$\left(p(\underset{\sim}{x}|\pi_1) - p(\underset{\sim}{x}|\pi_2)\right) d\underset{\sim}{x}$$

$$= -\frac{1}{2} \left(\left\{ E_{\pi_1}\left(\underset{\sim}{X}' \Sigma_1^{-1} \underset{\sim}{X}\right) + \underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{\mu}_1 - 2 \underset{\sim}{\mu}_1' \Sigma_1^{-1} E_{\pi_1}(\underset{\sim}{X}) \right\} \right.$$
$$\left. - \left\{ E_{\pi_2}\left(\underset{\sim}{X}' \Sigma_1^{-1} \underset{\sim}{X}\right) + \underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{\mu}_1 - 2 \underset{\sim}{\mu}_1' \Sigma_1^{-1} E_{\pi_2}(\underset{\sim}{X}) \right\} \right)$$

$$+ \frac{1}{2} \left(\left\{ E_{\pi_1}\left(\underset{\sim}{X}' \Sigma_2^{-1} \underset{\sim}{X}\right) + \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{\mu}_2 - 2 \underset{\sim}{\mu}_2' \Sigma_2^{-1} E_{\pi_1}(\underset{\sim}{X}) \right\} \right.$$
$$\left. - \left\{ E_{\pi_2}\left(\underset{\sim}{X}' \Sigma_2^{-1} \underset{\sim}{X}\right) + \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{\mu}_2 - 2 \underset{\sim}{\mu}_2' \Sigma_2^{-1} E_{\pi_2}(\underset{\sim}{X}) \right\} \right)$$

— *

Note that

(i) $E_{\pi_i}(\underset{\sim}{X}) = \underset{\sim}{\mu}_i \quad ; i = 1, 2$

and we need to ~~do~~ calculate terms of type

$$E_{\pi_i}\left(\underset{\sim}{X}' \Sigma_j^{-1} \underset{\sim}{X}\right) \qquad i, j = 1, 2$$

$= ?$

If $\underset{\sim}{X} \sim N_p(\underset{\sim}{\mu}, \Sigma)$, then

$$E(\underset{\sim}{X}' \Sigma^{-1} \underset{\sim}{X}) = E\left(tr\left(\underset{\sim}{X}' \Sigma^{-1} \underset{\sim}{X}\right)\right)$$

$$= E\left(tr\left(\Sigma^{-1} \underset{\sim}{X}\, \underset{\sim}{X}'\right)\right)$$

$$= tr\left(E\left(\Sigma^{-1} \underset{\sim}{X}\, \underset{\sim}{X}'\right)\right)$$

$$= tr\left(\Sigma^{-1} E\left(\underset{\sim}{X}\, \underset{\sim}{X}'\right)\right)$$

$$= tr\left(\Sigma^{-1}\left(\Sigma + \underset{\sim}{\mu}\, \underset{\sim}{\mu}'\right)\right)$$

$$= tr\, \Sigma^{-1}\Sigma + tr\left(\Sigma^{-1} \underset{\sim}{\mu}\, \underset{\sim}{\mu}'\right)$$

$$= tr\, I_p + tr\left(\underset{\sim}{\mu}' \Sigma^{-1} \underset{\sim}{\mu}\right)$$

$$= tr\, I_p + \underset{\sim}{\mu}' \Sigma^{-1} \underset{\sim}{\mu} \quad — (**)$$

In general,

$$E(\underset{\sim}{X}' A\, \underset{\sim}{X}) = tr\, A\Sigma + \underset{\sim}{\mu}' A\, \underset{\sim}{\mu}$$

From (*),

$$J_D = \left(\underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{\mu}_1 + \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{\mu}_2\right)$$

$$- \frac{1}{2}\left(E_{\pi_1}(\underset{\sim}{X}' \Sigma_1^{-1} \underset{\sim}{X}) - E_{\pi_2}(\underset{\sim}{X}' \Sigma_1^{-1} \underset{\sim}{X}) - E_{\pi_1}(\underset{\sim}{X}' \Sigma_2^{-1} \underset{\sim}{X}) + E_{\pi_2}(\underset{\sim}{X}' \Sigma_2^{-1} \underset{\sim}{X})\right)$$

$$= \left(\underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{\mu}_1 + \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{\mu}_1 \right)$$

$$- \frac{1}{2} \left(\left(\text{tr}\, \Sigma_1^{-1} \Sigma_1 + \underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{\mu}_1 \right) - \left(\text{tr}\, \Sigma_1^{-1} \Sigma_2 + \underset{\sim}{\mu}_2' \Sigma_1^{-1} \underset{\sim}{\mu}_2 \right) \right.$$

$$\left. - \left(\text{tr}\, \Sigma_2^{-1} \Sigma_1 + \underset{\sim}{\mu}_1' \Sigma_2^{-1} \underset{\sim}{\mu}_1 \right) + \left(\text{tr}\, \Sigma_2^{-1} \Sigma_2 + \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{\mu}_2 \right) \right)$$

$$= \left(\underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{\mu}_1 + \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{\mu}_1 \right.$$

$$\left. - \frac{1}{2} \underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{\mu}_1 + \frac{1}{2} \underset{\sim}{\mu}_2' \Sigma_1^{-1} \underset{\sim}{\mu}_2 + \frac{1}{2} \underset{\sim}{\mu}_1' \Sigma_2^{-1} \underset{\sim}{\mu}_1 - \frac{1}{2} \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{\mu}_2 \right)$$

$$+ \left(- \frac{1}{2} \text{tr}\, I_p + \frac{1}{2} \text{tr}\, \Sigma_1^{-1} \Sigma_2 + \frac{1}{2} \text{tr}\, \Sigma_2^{-1} \Sigma_1 - \frac{1}{2} \text{tr}\, I_p \right)$$

$$= \left(\frac{1}{2} \underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{\mu}_1 + \frac{1}{2} \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{\mu}_2 + \frac{1}{2} \underset{\sim}{\mu}_2' \Sigma_1^{-1} \underset{\sim}{\mu}_2 + \frac{1}{2} \underset{\sim}{\mu}_1' \Sigma_2^{-1} \underset{\sim}{\mu}_1 \right.$$

$$\left. - \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{\mu}_2 \right)$$

$$+ \frac{1}{2} \left(\text{tr}\, \Sigma_1^{-1} \Sigma_2 + \text{tr}\, \Sigma_2^{-1} \Sigma_1 - 2 \text{tr}\, I_p \right)$$

$$= \frac{1}{2} \left(\underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{\mu}_1 + \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{\mu}_2 + \underset{\sim}{\mu}_2' \Sigma_1^{-1} \underset{\sim}{\mu}_2 + \underset{\sim}{\mu}_1' \Sigma_2^{-1} \underset{\sim}{\mu}_1 \right.$$

$$\left. - 2 \underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{\mu}_2 - 2 \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{\mu}_1 \right)$$

$$+ \frac{1}{2} \left(\text{tr}\, \Sigma_1^{-1} \Sigma_2 + \text{tr}\, \Sigma_2^{-1} \Sigma_1 - 2 \text{tr}\, I_p \right)$$

$$= \frac{1}{2} \left(\underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_1 \right)' \left(\Sigma_1^{-1} + \Sigma_2^{-1} \right) \left(\underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_1 \right)$$

$$+ \frac{1}{2} \text{tr} \left(\Sigma_1^{-1} \Sigma_2 + \Sigma_2^{-1} \Sigma_1 - 2 I_p \right).$$

Note : If $\Sigma_1 = \Sigma_2$, then

$$J_D = (\underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_1)' \Sigma^{-1} (\underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_1)$$

→ square of Mahalanobis distance

Note : For $N_p(\underset{\sim}{\mu}_1, \Sigma_1)$ & $N_p(\underset{\sim}{\mu}_2, \Sigma_2)$ setup.

Chernoff distance

$$J_c = \frac{1}{2} s(1-s) (\underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_1)' \Sigma_s^{-1} (\underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_1) + \frac{1}{2} \log \left(\frac{|\Sigma_s|}{|\Sigma_1|^{1-s} |\Sigma_2|^s} \right)$$

$$\Sigma_s = (1-s)\Sigma_1 + s\Sigma_2 \quad ; \quad s \in [0, 1]$$

Note : If $\Sigma_1 = \Sigma_2$ i.e. $N_p(\underset{\sim}{\mu}_1, \Sigma)$ & $N_p(\underset{\sim}{\mu}_2, \Sigma)$

$$J_D = 8 J_B = (\underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_1)' \Sigma^{-1} (\underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_1).$$

Note : For $N_p(\underset{\sim}{\mu}_1, \Sigma_1)$ & $N_p(\underset{\sim}{\mu}_2, \Sigma_2)$ setup

$$J_{PF} = (2\pi)^{-p/2} \left(|2\Sigma_1|^{-1/2} + |2\Sigma_2|^{-1/2} - 2 |\Sigma_1 + \Sigma_2|^{-1/2} \right.$$

Patrick-Fischer $\left. \exp\left(-\frac{1}{2} (\underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_1)' (\Sigma_1 + \Sigma_2)^{-1} (\underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_1)' \right) \right)$

Non-hierarchical clustering methods

Features

- no distance/dissimilarity matrix calculations
- no hierarchy of clusters
- starts from either
  (i) a random initial partition of items into groups
  or (ii) an initial set of randomly selected seed points, which will form the nuclei of clusters.

K-means clustering or iterative relocation method

K-means method is an iterative algorithm that assigns each item to the cluster having the nearest centroid (mean)

Steps for K-means algorithm

(I) Partition the items into K initial clusters.

(II) Reassign items to the cluster whose centroid is nearest (in Euclidean sense mostly). Recalculate the centroids for the cluster receiving the new item and for the cluster losing that item.

(III) Repeat step (II) till no more reassignment can take place.

Note: Rather than starting with a partition of items into K initial groups in step (I), we can also specify

K initial seed points (centroids) and then proceed to step (II) after a walk through the data.

Remark: Let's try to understand the logic behind K-means method!!

Let there be N objects to be put into K clusters.

Suppose $c(i) = k$ denote that object i in cluster k

$k = 1(1)K \; ; \; i = 1(1)N$

A natural "loss function" (criterion) for clustering:

$$W(c) = \frac{1}{2} \sum_{k=1}^{K} \sum_{c(i)=k} \sum_{c(i')=k} d_{ii'} \quad \text{— (1)}$$

where, $d_{ii'} = d(\underset{\sim}{x}_i, \underset{\sim}{x}_{i'})$

$W(c)$ is within cluster point scatter, a measure of compactness of clusters. It characterizes the extent to which observations assigned, under a partition, to same clusters tend to be close to one another.

Further, consider the total point scatter

$$T = \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} d_{ii'}$$

i.e. $$T = \frac{1}{2} \sum_{k=1}^{K} \sum_{c(i)=k} \left(\sum_{c(i')=k} d_{ii'} + \sum_{c(i') \neq k} d_{ii'} \right)$$

i.e. $$T = \frac{1}{2} \sum_{k=1}^{K} \sum_{c(i)=k} \sum_{c(i')=k} d_{ii'} + \frac{1}{2} \sum_{k=1}^{K} \sum_{c(i)=k} \sum_{c(i') \neq k} d_{ii'}$$

i.e. $T = W(C) + B(C)$

$$B(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i') \neq k} d_{ii'}$$ is between-cluster point scatter. A measure of separation of clusters; large when different clusters are far apart.

Note:

$W(C)$ & $B(C)$ depends on cluster assignment

T is indep of cluster assignment

Note: It is natural to maximize $B(C)$ or equivalently minimize $W(C)$ over all possible cluster assignments

Note: The combinatorial optimization is not feasible even for moderate N & K.

The number of distinct cases (assignments) is

$$S(N,K) = \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \binom{K}{k} k^N$$

e.g. $S(10,4) = 34,105$

$S(19,4) \approx 10^{10}$

Feasible strategies are based on iterative greedy descent - K-means algorithm is one such algorithm.

K-means algorithm

Iterative descent clustering algorithm

Let
$$d(\underset{\sim}{x}_i, \underset{\sim}{x}_{i'}) = d_{ii'} = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = \| \underset{\sim}{x}_i - \underset{\sim}{x}_{i'} \|^2$$

Within-cluster point scatter
$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} \| \underset{\sim}{x}_i - \underset{\sim}{x}_{i'} \|^2$$
$$= \sum_{k=1}^{K} N_k \sum_{C(i)=k} \| \underset{\sim}{x}_i - \bar{\underset{\sim}{x}}_k \|^2$$

where $N_k = \sum_{i=1}^{N} I_{(C(i)=k)}$

$\bar{\underset{\sim}{x}}_k$: mean of of k^{th} cluster

Objective is thus to get C^* :
$$C^* = \underset{C}{\operatorname{argmin}} \sum_{k=1}^{K} N_k \sum_{C(i)=k} \| \underset{\sim}{x}_i - \bar{\underset{\sim}{x}}_k \|^2$$

and we can obtain C^* by enlarged optimization problem
$$\min_{C, \underset{\sim}{m}_1, \ldots \underset{\sim}{m}_K} \sum_{k=1}^{K} N_k \sum_{C(i)=k} \| \underset{\sim}{x}_i - \underset{\sim}{m}_k \|^2 \quad \text{— (2)}$$

as for any set of S observations
$$\bar{\underset{\sim}{x}}_S = \underset{m}{\operatorname{argmin}} \sum_{i=1}^{S} \| \underset{\sim}{x}_i - \underset{\sim}{m} \|^2$$

K-mean algorithm is an alternating optimization procedure for minimization of (2)

K-means clustering algorithm

Step I. For a given cluster assignment C, the total within cluster variability.

$$\sum_{k=1}^{K} N_k \sum_{C(i)=k} \| \underset{\sim}{x}_i - \underset{\sim}{m}_k \|^2$$

is minimized w.r.t. $\underset{\sim}{m}_1, \ldots \underset{\sim}{m}_k$ by taking means of the currently assigned clusters

Step II: Given a current set of cluster means, i.e. $m_k = \bar{\underset{\sim}{x}}_k$

$$\sum_{k=1}^{N} N_k \sum_{C(i)=k} \| \underset{\sim}{x}_i - \bar{\underset{\sim}{x}}_k \|^2$$

is minimized by assigning each obsn to the closest cluster mean

i.e. $$C(i) = \underset{1 \le k \le K}{\operatorname{argmin}} \| \underset{\sim}{x}_i - \underset{\sim}{m}_k \|^2$$

Step I & II are iterated until the assignments do not change.

Remark: The result may represent a suboptimal local minimum. One should start the algorithm with many initial random choices and choose the soln having minimum objective function value.

Example : K-means

Cases	Variables x_1	x_2
A	(5,	3)
B	(-1,	1)
C	(1,	-2)
D	(-3,	-2)

$K = 2$

Step I : Arbitrary partition (A, B) (C, D)

Cluster centroids

Cluster	$\bar{x}_1$	$\bar{x}_2$
(A, B)	2	2
(C, D)	-1	-2

Step II : Compute distances from cluster centroids and reassign each item to the nearest group

(If an item is moved from the initial configuration, the cluster centroid must be updated before we can proceed further)

A

$$d^2(A, (A,B)) = 3^2 + 1^2 = 10$$

$$d^2(A, (C,D)) = 6^2 + 5^2 = 61$$

$\Rightarrow$ A is closer to (A, B) than (C, D)

$\Rightarrow$ No reassignment required.

B

$$d^2(B, (A,B)) = 3^2 + 1^2 = 10$$

$$d^2(B, (C,D)) = 0^2 + 3^2 = 9$$

⇒ Reassign B to (C, D) to form (B, C, D)

Cluster centroid updation

Cluster	Coordinates of centroid $\bar{x}_1$	$\bar{x}_2$
A	5	3
(B, C, D)	-1	-1

$d^2(A, A) = 0$; $d^2(A, (B, C, D)) = 52$

$d^2(B, A) = 40$; $d^2(B, (B, C, D)) = 4$

$d^2(C, A) = 41$, $d^2(C, (B, C, D)) = 5$

$d^2(D, A) = 89$, $d^2(D, (B, C, D)) = 5$

⇒ no reassignment is necessary and relocation iteration stops.

(A), (B, C, D) is the final clustering.

Comparing different cluster partitions

Objective is to have a criterion function for comparing different cluster partitions of the data and hence to choose the best cluster partition.

Let the N data points be given by $\underset{\sim}{x}_1, \ldots, \underset{\sim}{x}_N$.

The sample variance covariance matrix $\hat{\Sigma}$ is given by

$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (\underset{\sim}{x}_i - \underset{\sim}{m})(\underset{\sim}{x}_i - \underset{\sim}{m})'$$

where $\underset{\sim}{m} = \frac{1}{N} \sum_{i=1}^{N} \underset{\sim}{x}_i = \underset{\sim}{\bar{x}}$

Let there be K clusters and define

$$z_{ji} = \begin{cases} 1, & \text{if } \underset{\sim}{x}_i \in \text{cluster } j \\ 0, & \text{o/w} \end{cases}$$

$$\underset{\sim}{m}_j = \frac{1}{N_j} \sum_{i=1}^{N} z_{ji} \underset{\sim}{x}_i \left(= \underset{\sim}{\bar{x}}_j\right) : \text{mean of cluster } j$$

$$N_j = \sum_{i=1}^{N} z_{ji} : \#\text{ of } \underset{\sim}{x}_i\text{'s in cluster } j$$

The within-cluster sum of squares and cross product (SSCP) scatter matrix is

$$S_W = \frac{1}{N} \sum_{j=1}^{K} \sum_{i=1}^{N} z_{ji} (\underset{\sim}{x}_i - \underset{\sim}{m}_j)(\underset{\sim}{x}_i - \underset{\sim}{m}_j)'$$

(the pooled within cluster scatter matrix over K clusters)

Note that

$$\hat{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\underset{\sim}{x}_i - \underset{\sim}{m})(\underset{\sim}{x}_i - \underset{\sim}{m})'$$

$$= \frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N} z_{ji}(\underset{\sim}{x}_i - \underset{\sim}{m})(\underset{\sim}{x}_i - \underset{\sim}{m})'$$

$$= \frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N} z_{ji}(\overline{\underset{\sim}{x}_i - \underset{\sim}{m}_j} + \overline{\underset{\sim}{m}_j - \underset{\sim}{m}})(\underset{\sim}{x}_i - \underset{\sim}{m}_j + \underset{\sim}{m}_j - \underset{\sim}{m})'$$

$$= \frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N} z_{ji}\Big((\underset{\sim}{x}_i - \underset{\sim}{m}_j)(\underset{\sim}{x}_i - \underset{\sim}{m}_j)' + (\underset{\sim}{m}_j - \underset{\sim}{m})(\underset{\sim}{m}_j - \underset{\sim}{m})' + \cancel{(\underset{\sim}{x}_i - \underset{\sim}{m}_j)(\underset{\sim}{m}_j - \underset{\sim}{m})'}^{\,0} + \cancel{(\underset{\sim}{m}_j - \underset{\sim}{m})(\underset{\sim}{x}_i - \underset{\sim}{m}_j)'}^{\,0}\Big)$$

i.e. $$\hat{\Sigma} = \frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N} z_{ji}(\underset{\sim}{x}_i - \underset{\sim}{m}_j)(\underset{\sim}{x}_i - \underset{\sim}{m}_j)' + \frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N} z_{ji}(\underset{\sim}{m}_j - \underset{\sim}{m})(\underset{\sim}{m}_j - \underset{\sim}{m})'$$

(e.g. $\frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N} z_{ji}(\underset{\sim}{m}_j - \underset{\sim}{m})(\underset{\sim}{x}_i - \underset{\sim}{m}_j)'$

$= \frac{1}{N}\sum_{j=1}^{K}(\underset{\sim}{m}_j - \underset{\sim}{m})\sum_{i=1}^{N} z_{ji}(\underset{\sim}{x}_i - \underset{\sim}{m}_j)'$

$= \frac{1}{N}\sum_{j=1}^{K}(\underset{\sim}{m}_j - \underset{\sim}{m})(N_j\underset{\sim}{m}_j - N_j\underset{\sim}{m}_j)' = 0$)

Sly $\frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N} z_{ji}(\underset{\sim}{x}_i - \underset{\sim}{m}_j)(\underset{\sim}{m}_j - \underset{\sim}{m})' = 0$

i.e.
$$\hat{\Sigma} = S_W + \frac{1}{N}\sum_{j=1}^{K} N_j (\underset{\sim}{m}_j - \underset{\sim}{m})(\underset{\sim}{m}_j - \underset{\sim}{m})'$$

$$= S_W + S_B$$

S_B : between-cluster sum of squares and cross product (SSCP) scatter matrix

(indicates the scatter of the cluster means about total grand mean)

Popular optimum clustering criteria are based on univariate (scalar) functions of above matrices, $S_W, S_B, \hat{\Sigma}$

e.g. (a) Minimization of $tr(S_W)$

$$tr(S_W) = tr\left(\frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N} z_{ji}(\underset{\sim}{x}_i - \underset{\sim}{m}_j)(\underset{\sim}{x}_i - \underset{\sim}{m}_j)'\right)$$

$$= \frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N} z_{ji}\, tr\left((\underset{\sim}{x}_i - \underset{\sim}{m}_j)(\underset{\sim}{x}_i - \underset{\sim}{m}_j)'\right)$$

$$= \frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N} z_{ji}\, tr\left((\underset{\sim}{x}_i - \underset{\sim}{m}_j)'(\underset{\sim}{x}_i - \underset{\sim}{m}_j)\right)$$

$$= \frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N} z_{ji} \| \underset{\sim}{x}_i - \underset{\sim}{m}_j \|^2$$

i.e. $$tr(S_W) = \frac{1}{N}\sum_{j=1}^{K} S_j$$

$$S_j = \sum_{i=1}^{n} z_{ji} \| \underset{\sim}{x}_i - \underset{\sim}{m}_j \|^2$$

The within-cluster sum of squares for j^{th} cluster

Thus minimization of $tr(S_W)$ is same as minimization of total within cluster sum of squares about the K centroids. (equiv to max $tr\, S_B$)

(b) Min $\frac{|S_W|}{|\hat{\Sigma}|}$

(c) Min $tr(\hat{\Sigma}^{-1} S_W)$

(d) Max $tr(S_W^{-1} S_B)$

Density estimation

Objective is to find

$f(\underset{\sim}{x})$ - density of feature vector

Approaches

(i) non-parametric approach

(ii) parametric approach

Non-parametric density estimation methods

(A) Histogram method

For one-dimensional data:

$$\hat{p}(x) = \frac{n_j}{\left(\sum_{j=1}^{N} n_j\right) dx}$$

Where,

n_j : # of samples in the histogram cell of width dx the contains the pt x

N : # of cells in the histogram

dx : width of the cell

For multidimensional feature vector:

$$\hat{p}(\underset{\sim}{x}) = \frac{n_j}{\left(\sum_{j=1}^{N} n_j\right) dv}$$

dv : volume of the j^{th} bin

(B) K-nearest neighbor method

Note that if X is a r.v. (cont type),

$$P(x \le X \le x+\Delta x) = F(x+\Delta x) - F(x)$$

$$\lim_{\Delta x \to 0} \frac{F(x+\Delta x) - F(x)}{\Delta x} = p(x) \quad \text{p.d.f}$$

i.e. $F(x+\Delta x) - F(x) = P(x \le X \le x+\Delta x)$

$$\approx p(x)\,\Delta x \quad \text{for small } \Delta x$$

For multivariate set up $\underset{\sim}{X}$ with p.d.f $p(\underset{\sim}{x})$

$P(\underset{\sim}{X}$ will fall in a given region C, centered at say $\underset{\sim}{x})$

$$= \int_C p(\underset{\sim}{x})\, d\underset{\sim}{x} \approx V(C)\, p(\underset{\sim}{x}),$$ if we assume C $\ni$ volume of $V(C)$ is small and $p(\underset{\sim}{x})$ does not vary appreciably within region C

let $\theta = V(C)\, p(\underset{\sim}{x})$

Realize that θ can also be approximated by the proportion of samples falling within C

i.e $\theta \approx \frac{K}{n}$;

where ; K, the number of samples falling within C out of total n samples

i.e. $\frac{K}{n} \approx p(\underset{\sim}{x})\, V$

$$\Rightarrow \hat{p}(\underset{\sim}{x}) = \frac{K}{n V}$$

K-nearest neighbor approach fixes K and then

determines the volume V which contains K samples centered at the point $\underset{\sim}{x}$.

If $\underset{\sim}{x}_K$ is the K^{th} nearest neighbor pt to $\underset{\sim}{x}$, then C may be taken to be sphere centered at $\underset{\sim}{x}$ with radius $\|\underset{\sim}{x} - \underset{\sim}{x}_K\|$. The volume of such a sphere in p dimension is $2 r^p \pi^{p/2} / p \Gamma_{p/2}$

Remark: This approach is different from the histogram approach wherein bin size is fixed.

(C) Kernel methods (Parzen methods)

Consider a 1-dimensional sample, $x_1, \ldots x_n$

An estimate of cumulative dist^n f^n at x is

$$\hat{F}(x) = \frac{\#\text{ of observations} \le x}{n}$$

estimate of p.d.f at x:

$$\hat{p}(x) = \frac{\hat{F}(x+h) - \hat{F}(x-h)}{2h}$$

↗ proportion of obsns falling within the interval $[x-h, x+h]/2h$

i.e. Using a Kernel f^n (rectangular Kernel)

$$K(z) = \begin{cases} \frac{1}{2}, & |z| \le 1 \\ 0, & \text{o/w} \end{cases}$$

$$\hat{p}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

$= \frac{1}{2}$ (# of observations with h distance from x)

i.e. pts within h distance from x contribute $\frac{1}{2nh}$ to the density and pts outside this distance contribute 0.

Remark: 'h' is referred to as spread or smoothing parameter (or bandwidth)

Remark: Examples of popular univariate Kernel f^ns.

(i) Rectangular: $K(z) = \begin{cases} 1/2, & |z| \le 1 \\ 0 & \text{o/w} \end{cases}$

(ii) Triangular: $K(z) = \begin{cases} 1 - |z|, & \text{for } |z| \le 1 \\ 0, & \text{o/w} \end{cases}$

(iii) Gaussian: $K(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2} \quad \forall z$

(iv) Bi-weight / quartic: $K(z) = \begin{cases} \frac{15}{16}(1 - z^2)^2, & |z| \le 1 \\ 0, & \text{o/w}. \end{cases}$

(v) Bartlett - Epanechnikov :

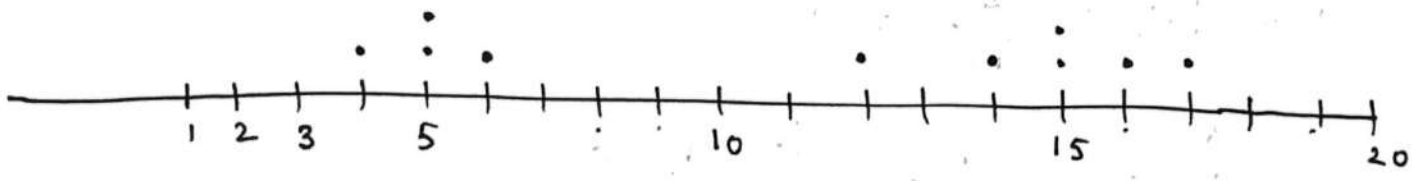
$$K(z) = \begin{cases} \frac{3}{4}\left(1 - z^2/5\right)/\sqrt{5}, & |z| \le \sqrt{5} \\ 0, & \text{o/w} \end{cases}$$

Example : $x = \{4, 5, 5, 6, 12, 14, 15, 15, 16, 17\}$

$n = 10$ samples

(a) Knn density estimate with $K = 4$

$$\hat{p}(3) = \frac{4}{10}\left(V_4(3)\right)^{-1} \qquad \left(\hat{p}(x) = \frac{k}{n} V^{-1} \right)$$



For $V_4(3)$, $r = 3 \Rightarrow V_4(3) = 2r = 6$

$$\hat{p}(3) = \frac{4}{10 \times 6} = \frac{1}{15}$$

$$\hat{p}(10) = \frac{4}{10}\left(V_4(10)\right)^{-1}$$

For $V_4(10)$, $r = 5 \Rightarrow V_4(10) = 10$

$$\Rightarrow \hat{p}(10) = \frac{4}{10 \times 10} = \frac{1}{25}$$

Sly $\hat{p}(15) = \frac{4}{10 \times 2} = \frac{1}{5}$ $(r = 1)$

Kernel density estimate with rectangular Kernel

with $h = 4$ bandwidth

$$\hat{p}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right); \quad K(z) = \begin{cases} 1/2, & |z| \le 1 \\ 0, & \text{o/w} \end{cases}$$

i.e $$\hat{p}(x) = \frac{1}{nh} \sum_{i=1}^{n} \left(\frac{1}{2} I_{(|x - x_i| \le h)}\right)$$

e.g.

$$\hat{p}(3) = \frac{1}{10 \times 4} \left(\sum_{i=1}^{10} \frac{1}{2} I_{(|3 - x_i| \le 4)} \right)$$

i.e. $$\hat{p}(3) = \frac{1}{40} \left(\frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + 0 + 0 + \cdots + 0 \right)$$

i.e $$\hat{p}(3) = \frac{2}{40} = \frac{1}{20}$$

$$\hat{p}(10) = \frac{1}{10 \times 4} \left(\sum_{i=1}^{10} \frac{1}{2} I_{(|10 - x_i| \le 4)} \right)$$

i.e $$\hat{p}(10) = \frac{1}{40} \times \frac{3}{2} = \frac{3}{80}.$$

$$\hat{p}(15) = \frac{1}{40} \times \left(6 \times \frac{1}{2}\right) = \frac{3}{40}$$

- - -

Multivariate Kernel density estimate

Approach I : Assume independence of the component variables and estimate univariate Kernel density estimates for the components and get

$$\hat{p}(\underset{\sim}{x}) = \prod_{i=1}^{p} \hat{p}_i(x_i)$$

Approach II : Generalization of univariate approach for multivariate case.

$$\hat{p}(\underset{\sim}{x}) = \frac{1}{n h^p} \sum_{i=1}^{n} K\left(\frac{\underset{\sim}{x} - \underset{\sim}{x}_i}{h}\right)$$

i.e. $$\hat{p}(\underset{\sim}{x}) = \frac{1}{n h^p} \sum_{i=1}^{n} K\left(\frac{x_1 - x_{i_1}}{h}, \ldots, \frac{x_p - x_{i_p}}{h}\right)$$

Remark : A more general form is using different bandwidth

$$\hat{p}(\underset{\sim}{x}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\prod_{j=1}^{p} h_j} K\left(\frac{x_1 - x_{i_1}}{h_1}, \ldots, \frac{x_p - x_{i_p}}{h_p}\right)$$

Remark : A simple approach is to use a product Kernel

$$\hat{p}(\underset{\sim}{x}) = \frac{1}{n h^p} \sum_{i=1}^{n} \left(\prod_{j=1}^{p} \tilde{K}\left(\frac{x_j - x_{i_j}}{h}\right)\right)$$

or $$\frac{1}{n} \sum_{i=1}^{n} \left(\prod_{j=1}^{p} \frac{1}{h_j} \tilde{K}\left(\frac{x_j - x_{i_j}}{h_j}\right)\right).$$

↑ using diff bandwidth

or $\frac{1}{n}\sum_{i=1}^{n}\left(\prod_{j=1}^{p}\frac{1}{h_j}\tilde{K}_j\left(\frac{x_j - x_{ij}}{h_j}\right)\right)$

↗ using diff bandwidth & diff kernel f^n

$\tilde{K}$ (or $\tilde{K}_j$) is taken as one of the univariate kernels discussed earlier

Remark: Alternatively, one can use a genuine multivariable kernel

e.g. multivariate Gaussian Kernel

$$K(\underset{\sim}{y}) = \frac{1}{(2\pi)^{p/2}} \exp\left(-\frac{1}{2}\underset{\sim}{y}'\underset{\sim}{y}\right)$$

multivariate Epanechnikov Kernel, multivariate quartic kernel are other choices of mult kernel.

Epanechnikov Kernel

$$K(\underset{\sim}{y}) = \begin{cases} (1 - \underset{\sim}{y}'\underset{\sim}{y})^{(p+2)/2} c_p & \text{for } |\underset{\sim}{x}| \le 1 \\ 0, & \text{o/w} \end{cases}$$

$$c_p = \pi^{p/2} \Big/ \Gamma(p/2+1) = 2\pi^{p/2} \Big/ p\Gamma(p/2)$$

Parametric density estimation

Most commonly used assumption : multivariate Gaussian
or a mixture of mult Gaussian

Multivariate Gaussian :

$\underset{\sim}{x}_1, \ldots, \underset{\sim}{x}_n$ realizations of $N_p(\underset{\sim}{\mu}, \Sigma)$; $\Sigma > 0$

Use $\underset{\sim}{x}_1, \ldots, \underset{\sim}{x}_n$ to find $\hat{f}(\underset{\sim}{x})$ $\underset{\sim}{x} \in \mathbb{R}^p$.

$$f(\underset{\sim}{x}) = (2\pi)^{-p/2} |\Sigma|^{-1/2} \exp\left(-\tfrac{1}{2}(\underset{\sim}{x} - \underset{\sim}{\mu})' \Sigma^{-1} (\underset{\sim}{x} - \underset{\sim}{\mu})\right)$$

$\theta = (\underset{\sim}{\mu}, \Sigma)$ set of unknown parameters

Likelihood fn

$$L(\theta) = \prod_{j=1}^{n} f(\underset{\sim}{x}_j)$$

$$L(\theta) = (2\pi)^{-np/2} |\Sigma|^{-n/2} \exp\left(-\tfrac{1}{2} \sum_{j=1}^{n} (\underset{\sim}{x}_j - \underset{\sim}{\mu})' \Sigma^{-1} (\underset{\sim}{x}_j - \underset{\sim}{\mu})\right)$$

Note that

$$\sum_{j=1}^{n} (\underset{\sim}{x}_j - \underset{\sim}{\mu})' \Sigma^{-1} (\underset{\sim}{x}_j - \underset{\sim}{\mu})$$

$$= \sum_{j=1}^{n} (\underset{\sim}{x}_j - \bar{\underset{\sim}{x}} + \bar{\underset{\sim}{x}} - \underset{\sim}{\mu})' \Sigma^{-1} (\underset{\sim}{x}_j - \bar{\underset{\sim}{x}} + \bar{\underset{\sim}{x}} - \underset{\sim}{\mu})$$

$$= \sum_{j=1}^{n} (\underset{\sim}{x}_j - \bar{\underset{\sim}{x}})' \Sigma^{-1} (\underset{\sim}{x}_j - \bar{\underset{\sim}{x}}) + n(\bar{\underset{\sim}{x}} - \underset{\sim}{\mu})' \Sigma^{-1} (\bar{\underset{\sim}{x}} - \underset{\sim}{\mu})$$

$$+ 2 \sum_{j=1}^{n} (\underset{\sim}{x}_j - \bar{\underset{\sim}{x}})' \Sigma^{-1} (\bar{\underset{\sim}{x}} - \underset{\sim}{\mu}) \quad (= 0)$$

$$= \sum_{j=1}^{n} (\underset{\sim}{x}_j - \hat{\underset{\sim}{x}})' \Sigma^{-1} (\underset{\sim}{x}_j - \hat{\underset{\sim}{x}}) + n(\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})' \Sigma^{-1} (\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})$$

$$= \operatorname{tr}\left(\sum_{j=1}^{n} (\underset{\sim}{x}_j - \hat{\underset{\sim}{x}})' \Sigma^{-1} (\underset{\sim}{x}_j - \hat{\underset{\sim}{x}}) \right) + n(\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})' \Sigma^{-1} (\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})$$

$$= \sum_{j=1}^{n} \operatorname{tr} (\underset{\sim}{x}_j - \hat{\underset{\sim}{x}})' \Sigma^{-1} (\underset{\sim}{x}_j - \hat{\underset{\sim}{x}}) + n(\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})' \Sigma^{-1} (\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})$$

$$= \sum_{j=1}^{n} \operatorname{tr} \Sigma^{-1} (\underset{\sim}{x}_j - \hat{\underset{\sim}{x}})(\underset{\sim}{x}_j - \hat{\underset{\sim}{x}})' + n(\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})' \Sigma^{-1} (\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})$$

$$= \operatorname{tr} \Sigma^{-1} \sum_{j=1}^{n} (\underset{\sim}{x}_j - \hat{\underset{\sim}{x}})(\underset{\sim}{x}_j - \hat{\underset{\sim}{x}})' + n(\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})' \Sigma^{-1} (\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})$$

$$= \operatorname{tr} \Sigma^{-1} (n-1)S + n(\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})' \Sigma^{-1} (\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})$$

$$= \operatorname{tr} \Sigma^{-1} A + n(\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})' \Sigma^{-1} (\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})$$

$$\Rightarrow L(\theta) = (2\pi)^{-np/2} |\Sigma|^{-n/2} \exp\left(-\frac{1}{2} \operatorname{tr} \Sigma^{-1} A - \frac{n}{2} (\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})' \Sigma^{-1} (\hat{\underset{\sim}{x}} - \underset{\sim}{\mu}) \right) \quad \text{—— (*)}$$

Note that for a fixed $\Sigma > 0$

$$(\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})' \Sigma^{-1} (\hat{\underset{\sim}{x}} - \underset{\sim}{\mu}) \geq 0 \text{ and } 0 \text{ only for } \hat{\underset{\sim}{x}} = \underset{\sim}{\mu}$$

$L(\underset{\sim}{\mu}, \Sigma)$ for a fixed $\Sigma (>0)$ is max if exponent is max w.r.t $\underset{\sim}{\mu}$

i.e. If $(\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})' \Sigma^{-1} (\hat{\underset{\sim}{x}} - \underset{\sim}{\mu})$ is min w.r.t $\underset{\sim}{\mu}$

i.e. If $\underset{\sim}{\mu} = \hat{\underset{\sim}{x}}$ ← indep of the fixed level of Σ

$$\Rightarrow \hat{\underset{\sim}{\mu}}_{MLE} = \hat{\underset{\sim}{X}}$$

The log-likelihood at $\underset{\sim}{\mu} = \bar{\underset{\sim}{x}}$

$$l(\hat{\underset{\sim}{\mu}}, \Sigma) = \log L(\hat{\underset{\sim}{\mu}}, \Sigma)$$

$$= -\frac{np}{2}\log 2\pi + \frac{n}{2}\log|\Sigma^{-1}| - \frac{1}{2}\operatorname{tr}\Sigma^{-1}A \quad -(*)$$

maximisation of $(*)$ w.r.t. Σ is equiv to maximisation

of $\quad \frac{n}{2}\log|\Sigma^{-1}| - \frac{1}{2}\operatorname{tr}\Sigma^{-1}A$

$$= \frac{n}{2}\log|\Sigma^{-1}A| - \frac{1}{2}\operatorname{tr}\Sigma^{-1}A - \frac{n}{2}\log|A|$$

i.e. maximisation of

$$\frac{n}{2}\log|\Sigma^{-1}A| - \frac{1}{2}\operatorname{tr}\Sigma^{-1}A \quad -(**)$$

Let $\lambda_1, \ldots \lambda_p$ be eigenvalues of $\Sigma^{-1}A$

$$(**) = \frac{n}{2}\log\prod_1^p \lambda_j - \frac{1}{2}\sum_1^p \lambda_j$$

$$= \frac{n}{2}\sum_1^p \log\lambda_j - \frac{1}{2}\sum\lambda_j$$

$$= \frac{1}{2}\sum_{j=1}^p \left(n\log\lambda_j - \lambda_j\right) \quad -(***)$$

$(***)$ is maximised w.r.t λ_j at $\lambda_j = n \;\forall\, n$

i.e. $\quad \Sigma^{-1}A = P\, nI_p\, P' = nI_p.$

$$\Rightarrow \Sigma^{-1} = nA^{-1} \qquad \text{i.e. } \Sigma = \frac{1}{n}A \text{ maximises likelihood w.r.t } \Sigma$$

$$\Rightarrow \hat{\Sigma}_{MLE} = \frac{1}{n}\sum_{j=1}^n (\underset{\sim}{x}_j - \hat{\underset{\sim}{x}})(\underset{\sim}{x}_j - \hat{\underset{\sim}{x}})'$$

Based $\underset{\sim}{x}_1, \ldots, \underset{\sim}{x}_n$ obtain

$$\hat{\underset{\sim}{\mu}} = \bar{\underset{\sim}{x}} \text{ and } \hat{\Sigma} = \frac{1}{n} A = S_n$$

Estimate density as

$$\hat{f}(\underset{\sim}{x}) = (2\pi)^{-p/2} |S_n|^{-1/2} \exp\left(-\frac{1}{2}(\underset{\sim}{x} - \bar{\underset{\sim}{x}})' S_n^{-1} (\underset{\sim}{x} - \bar{\underset{\sim}{x}})\right)$$

Mixture Normal setup.

One of the most widely used assumption

$$f(\underset{\sim}{x}) = \sum_{j=1}^{g} \pi_j \, f(\underset{\sim}{x} | \theta_j)$$

g: # of mixing densities

π_j: mixing proportion for j^{th} group/component

$f(\underset{\sim}{x} | \theta_j)$: density for j^{th} component in the mixture

j^{th} component $N_p(\underset{\sim}{\mu}_j, \Sigma_j)$; $j = 1(1)g$, $\Sigma_j > 0$

$\theta_j = (\underset{\sim}{\mu}_j, \Sigma_j)$

$\Phi = (\pi_1, \ldots, \pi_g, \underset{\sim}{\mu}_1, \Sigma_1, \ldots, \underset{\sim}{\mu}_g, \Sigma_g)$

unknown parameters

Likelihood f^n

$$L(\Phi) = \prod_{i=1}^{n} \left(\sum_{j=1}^{g} \pi_j \, f(\underset{\sim}{x}_i | \theta_j) \right)$$

Remark: $(\underset{\sim}{x}_1, \ldots, \underset{\sim}{x}_n)$ is incomplete data

Use E-M algorithm

$\underset{\sim}{x}$: incomplete data without class labels

$\underset{\sim}{y}' = (\underset{\sim}{x}', \underset{\sim}{z}')$ complete data

where, $\underset{\sim}{z}$ = indicator vector of length g with 1 at the k^{th} position

$$g(\underset{\sim}{y}|\underline{\Phi}) = g(\underset{\sim}{x}, \underset{\sim}{z}|\underline{\Phi}) = \frac{p(\underset{\sim}{x}, \underset{\sim}{z}, \underline{\Phi})}{p(\underset{\sim}{z}, \underline{\Phi})} \cdot \frac{p(\underset{\sim}{z}, \underline{\Phi})}{p(\underline{\Phi})}$$

$$= p(\underset{\sim}{x}|\underset{\sim}{z}, \underline{\Phi})\ p(\underset{\sim}{z}|\underline{\Phi})$$

$$= p(\underset{\sim}{x}|\theta_k)\ \pi_k$$

i.e. $g(\underset{\sim}{y}|\underline{\Phi}) = \left(p(\underset{\sim}{x}|\theta_1)\pi_1\right)^0 \cdots \left(p(\underset{\sim}{x}|\theta_k)\pi_k\right)^1 \cdots \left(p(\underset{\sim}{x}|\theta_g)\pi_g\right)^0$

Let $z_j = \begin{cases} 1, & \text{if } j=k \\ 0, & \text{o/w} \end{cases}$

$$g(\underset{\sim}{y}|\underline{\Phi}) = \prod_{j=1}^{g} \left(p(\underset{\sim}{x}|\theta_j)\pi_j\right)^{z_j}$$

Consider for simplicity $g = 2$

$\underset{\sim}{z} = (z_1, z_2)$ $\quad z_1 = 1$ if $\underset{\sim}{x}$ corresp to component 1

$z_2 = 1$ if $\underset{\sim}{x}$ corresp to component 2

Let $\pi_2 = \pi$, $\pi_1 = 1 - \pi$

$$g(\underset{\sim}{y}|\underline{\Phi}) = \left(p(\underset{\sim}{x}|\theta_1)\pi_1\right)^{z_1} \left(p(\underset{\sim}{x}|\theta_2)\pi_2\right)^{z_2}$$

i.e. $g(\underset{\sim}{y}|\underline{\Phi}) = \left(p(\underset{\sim}{x}|\theta_1)(1-\pi)\right)^{z_1} \left(p(\underset{\sim}{x}|\theta_2)\pi\right)^{z_2}$

$$g(\underset{\sim}{y}_1, \ldots, \underset{\sim}{y}_n|\underline{\Phi}) = \prod_{i=1}^{n} \prod_{j=1}^{2} \left(p(\underset{\sim}{x}_i|\theta_j)\pi_j\right)^{z_{ji}}$$

$$= \prod_{i=1}^{n} \left(\left\{p(\underset{\sim}{x}_i|\theta_1)(1-\pi)\right\}^{z_{1i}} \left\{p(\underset{\sim}{x}_i|\theta_2)\pi\right\}^{z_{2i}}\right.$$

Note: For a general 'g',

$$g(\underset{\sim}{y}_1, \ldots, \underset{\sim}{y}_n \mid \Phi) = \prod_{i=1}^{n} \left(\prod_{j=1}^{g} \left(p(\underset{\sim}{x}_i \mid \theta_j)\pi_j \right)^{z_{ji}} \right).$$

log likelihood fn

$$\ell(\Phi) = \log g(\underset{\sim}{y}_1, \ldots, \underset{\sim}{y}_n \mid \Phi)$$

$$= \sum_{i=1}^{n} \log \left(\left\{ p(\underset{\sim}{x}_i \mid \theta_1)(1-\pi) \right\}^{z_{1i}} \left\{ p(\underset{\sim}{x}_i \mid \theta_2)\pi \right\}^{z_{2i}} \right)$$

i.e $$\ell(\Phi) = \sum_{i=1}^{n} \Big(z_{1i} \log \left(p(\underset{\sim}{x}_i \mid \theta_1)(1-\pi) \right) + z_{2i} \log \left(p(\underset{\sim}{x}_i \mid \theta_2)\pi \right) \Big).$$

$$= \sum_{i=1}^{n} \left(z_{1i} \log \left(p(\underset{\sim}{x}_i \mid \theta_1) \right) + z_{2i} \log \left(p(\underset{\sim}{x}_i \mid \theta_2) \right) \right)$$

$$+ \sum_{i=1}^{n} \left(z_{1i} \log(1-\pi) + z_{2i} \log \pi \right)$$

Note: For a general 'g'

$$\ell(\Phi) = \sum_{i=1}^{n} \left(\sum_{j=1}^{g} z_{ji} \log \left(p(\underset{\sim}{x}_i \mid \theta_j) \right) \right)$$

$$+ \sum_{i=1}^{n} \left(\sum_{j=1}^{g} z_{ji} \log \pi_j \right).$$

Remark:

Note that if (z_{1i}, z_{2i}) is known $\forall i$, the MLE is simple

$$\left\{ \begin{array}{l} \underset{\sim}{\mu}_1 \to \bar{x}_1 \\ \Sigma_1 \to S_1 \end{array} \right\} \text{ from all } \underset{\sim}{x}_i \ni z_{1i} = 1$$

$$\left\{ \begin{array}{l} \underset{\sim}{\mu}_2 \to \bar{x}_2 \\ \Sigma_2 \to S_2 \end{array} \right\} \text{ from all } \underset{\sim}{x}_i \ni z_{2i} = 1$$

But $\underset{\sim}{z}_i$'s are unknown!

E-M algorithm steps

E-step :

$$E\left(z_{ji} \,\middle|\, \underset{\sim}{x}_i, \underline{\Phi}^{(m)}\right) = P\left(z_{ji}=1 \,\middle|\, \underset{\sim}{x}_i, \underline{\Phi}^{(m)}\right) = \omega_{ji}$$

ω_{ji} : prob that $\underset{\sim}{x}_i \in$ group j given current estimates $\underline{\Phi}^{(m)}$

$$\omega_{ji} = \frac{\pi_j^{(m)}\, p\left(\underset{\sim}{x}_i \mid \theta_j^{(m)}\right)}{\pi_1^{(m)}\, p\left(\underset{\sim}{x}_i \mid \theta_1\right) + \pi_2^{(m)}\, p\left(\underset{\sim}{x}_i \mid \theta_2^{(m)}\right)}$$

Form the f^n

$$Q\left(\underline{\Phi}, \underline{\Phi}^{(m)}\right) = \sum_{i=1}^{n} \left(\omega_{1i} \log\left(p\left(\underset{\sim}{x}_i \mid \theta_1\right)\right) + \omega_{2i} \log\left(p\left(\underset{\sim}{x}_i \mid \theta_2\right)\right)\right)$$

$$+ \sum_{i=1}^{n} \sum_{j} \omega_{ji} \log \pi_j$$

Note that $Q\left(\underline{\Phi}, \underline{\Phi}^{(m)}\right) = E\left(\log g\left(\underset{\sim}{y}_1, \ldots \underset{\sim}{y}_n\right) \,\middle|\, \underset{\sim}{x}_1, \ldots \underset{\sim}{x}_n, \underline{\Phi}\right)$

M-step : Maximise Q w.r.t. π_i & θ_i

Maximisation of Q w.r.t. π_i subject to $\Sigma \pi_j = 1$

$$\tilde{Q} = Q - \lambda\left(\Sigma \pi_j - 1\right)$$

$$\frac{\partial \tilde{Q}}{\partial \pi_j} = \frac{\partial}{\partial \pi_j}\left(\sum_{i=1}^{n} \sum_{j} \omega_{ji} \log \pi_j\right) - \frac{\partial}{\partial \pi_j}\left(\lambda\left(\Sigma \pi_j - 1\right)\right)$$

$$= \sum_{i=1}^{n} \frac{\omega_{ji}}{\pi_j} - \lambda = 0$$

$$\Rightarrow \quad \lambda = \sum_{j=1}^{n} \omega_{ji} \Big/ \pi_j \quad \text{i.e.} \quad \lambda\, \pi_j = \sum_{i=1}^{n} \omega_{ji}$$

$$\lambda \pi_j = \sum_i \omega_{ji}$$

$$\lambda \sum_j \pi_j = \sum_j \sum_i \omega_{ji} = \sum_{i=1}^{n} \left(\sum_j \omega_{ji} \right)$$

i.e. $\lambda = n$

& $\hat{\pi}_j = \frac{1}{n} \sum_{i=1}^{n} \omega_{ji}$

Also for θ_j

$$\hat{\mu}_j = \frac{\sum_{i=1}^{n} \omega_{ji} \underset{\sim}{x}_i}{\sum_{i=1}^{n} \omega_{ji}} = \frac{1}{n \hat{\pi}_j} \sum_{i=1}^{n} \omega_{ji} \underset{\sim}{x}_i$$

& $$\hat{\Sigma}_j = \frac{\sum_{i=1}^{n} \omega_{ji} (\underset{\sim}{x}_i - \hat{\underset{\sim}{\mu}}_j)(\underset{\sim}{x}_i - \hat{\underset{\sim}{\mu}}_j)'}{\sum_{i=1}^{n} \omega_{ji}}$$

i.e. $$\hat{\Sigma}_j = \frac{1}{n \hat{\pi}_j} \sum_{i=1}^{n} \omega_{ji} (\underset{\sim}{x}_i - \hat{\underset{\sim}{\mu}}_j)(\underset{\sim}{x}_i - \hat{\underset{\sim}{\mu}}_j)'$$

The E-M algorithm alternates bet^n E-step of estimating ω_{ji} and M-step of calculating $\hat{\pi}_j$, $\hat{\underset{\sim}{\mu}}_j$ and $\hat{\Sigma}_j$, given ω_{ji}.

The iteration continues till convergence of likelihood.

Example : $p = 1$; $g = 2$

Comp 1 : $N(\mu_1, \sigma_1^2)$; $\theta = (\mu_1, \sigma_1^2)$

Comp 2 : $N(\mu_2, \sigma_2^2)$; $\theta = (\mu_2, \sigma_2^2)$

$\Phi = (\pi, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)$

$$Q(\Phi, \Phi^{(m)}) = \sum_{i=1}^{n} \left(w_{1i} \log p(x_i|\theta_1) + w_{2i} \log p(x_i|\theta_2) \right)$$

E-step :

$$+ \sum_{i=1}^{n} \left(\sum_{j=1}^{2} w_{ji} \log \pi_j \right) \quad - (*)$$

$$w_{1i} = \frac{\pi_1^{(m)} p(x_i|\theta_1^{(m)})}{\pi_1^{(m)} p(x_i|\theta_1^{(m)}) + (1-\pi_1^{(m)}) p(x_i|\theta_2^{(m)})}$$

Starting r.h.s. can be obtained from cluster analysis output.

M-Step : $\hat{\pi}_j = \dfrac{\sum_{i=1}^{n} w_{ji}}{n}$; $j = 1, 2$

and $$\frac{\partial Q}{\partial \mu_j} = \sum_{i=1}^{n} w_{ji} \frac{\partial}{\partial \mu_j} \left(-\frac{1}{2} \log 2\pi\sigma_j^2 - \frac{1}{2\sigma_j^2}(x_i - \mu_j)^2 \right)$$

$$= \sum_{i=1}^{n} w_{ji} \left(\frac{1}{2\sigma_j^2} (x_i - \mu_j) \right) = 0$$

i.e. $$\sum_{i=1}^{n} w_{ji} x_i = \mu_j \sum_{i=1}^{n} w_{ji}$$

$$\Rightarrow \hat{\mu}_j = \frac{\sum_i w_{ji} x_i}{\sum_i w_{ji}} = \frac{\sum_i w_{ji} x_i}{n \hat{\pi}_j}$$

$$\frac{\partial Q}{\partial \sigma_j^2} = \sum_{i=1}^{n} w_{ji} \frac{\partial}{\partial \sigma_j^2} \left(-\frac{1}{2}\log 2\pi - \frac{1}{2}\log \sigma_j^2 - \frac{1}{2\sigma_j^2}(x_i - \mu_j)^2 \right)$$

$$= \sum_{i=1}^{n} w_{ji} \left(-\frac{1}{2\sigma_j^2} + \frac{1}{2(\sigma_j^2)^2}(x_i - \mu_j)^2 \right)$$

$$\left.\begin{aligned} \frac{\partial Q}{\partial \mu_j} &= 0 \\ \frac{\partial Q}{\partial \sigma_j^2} &= 0 \end{aligned}\right\} \Rightarrow \quad \hat{\mu}_j = \frac{1}{\sum_i w_{ji}} \sum_i w_{ji} x_i$$

$$\hat{\sigma}_j^2 = \frac{1}{\sum_i w_{ji}} \sum_i w_{ji} (x_i - \hat{\mu}_j)^2$$

Start with initial $\left(\pi_1^{(0)}, \theta_1^{(0)}, \theta_2^{(0)}\right)$ → obtain

(w_{1i}, w_{2i}) $i = 1(1)n$ ⟶ obtain $(\hat{\pi}_1, \hat{\mu}_1, \hat{\sigma}_1^2,$

$\hat{\mu}_2, \hat{\sigma}_2^2) = \left(\pi_1^{(1)}, \theta_1^{(1)}, \theta_2^{(1)}\right)$ → alternate

bet^n E-step & M-step till convergence of the likelihood.

Association Rule Mining

Market basket analysis: to find associations between different itemsets that customers place in shopping basket

Applications: Cross marketing, catalog/floor design, design attractive packs, web log analysis for e-commerce,

Rule generation:

Antecedent ⇒ Consequent (support, confidence)

Data structure:

$T = \{t_1, \ldots, t_n\}$ set of transcations

Each t_K is an itemset

$I = \{i_1, \ldots, i_m\}$

Typical data

Cid	A1	A2	A3	. . .	A_K
C_1	0	0	1	0 1 1 - - -	0
C_2	1	0	0	1 1 0 - - -	1
:					
C_n	-	-	-	- - -	.

Aim: • Find frequent patterns, i.e. associations among sets of items in T

• Represent these relationships as association rules of the form

$X \Rightarrow Y$ (support, confidence)

Important definitions:

Support count : # of occurrences of an itemset in the database

$\sigma(\{\text{itemset}\})$

Support : Fraction of transactions containing the itemset

$$s(\text{itemset}) = \frac{\sigma(\text{itemset})}{|T|}$$

Frequent itemset : An itemset whose support $\geq$ a threshold, minsup

Confidence : A measure of how often B appears in [Rule: $(A \Rightarrow B)$] transactions containing A.

% of transactions containing A which also contains B

$$c(A \Rightarrow B) = \frac{s(A,B)}{s(A)} \quad \left(\text{est of conditional prob } B|A\right)$$

Example:

Tid	items
T1	bread, egg, peanut-butter
T2	bread, peanut-butter
T3	bread, milk, peanut-butter
T4	beer, bread
T5	beer, milk

Rule	Support	Confidence
bread ⇒ peanut-butter	$\frac{3}{5} = 0.6$	$\frac{3}{4} = 0.75$
peanut-butter ⇒ bread	$\frac{3}{5} = 0.6$	$\frac{3}{3} = 1.0$
beer ⇒ bread	$\frac{1}{5} = 0.2$	$\frac{1}{2} = 0.5$
peanut-butter ⇒ egg	$\frac{1}{5} = 0.2$	$\frac{1}{3} = 0.33$
egg ⇒ peanut-butter	$\frac{1}{5} = 0.2$	$\frac{1}{1} = 1$
egg ⇒ milk	0	0

ARM task: Given a set of transactions, the goal of ARM is to find all rules ⇒

(i) support ≥ min sup

& (ii) confidence ≥ min conf

(min sup, min conf) : fixed apriori

Brute force approach

- List all possible association rules
- Compute support & confidence for each rule
- Prune as per threshold

Approach is computationally prohibitive

The Apriori Algorithm

Agrawal & Srikant: "Fast algorithms for mining association rules in large databases", Int Conf on VLDB, 1994.

2-Step ARM of Apriori algorithm

S1: Generate all frequent itemsets with support ≥ min sup

S2: Generate association rules using these frequent itemsets

Anti-monotonicity property of Apriori algorithm

Downward closure property

- Any subset of a frequent itemset is frequent

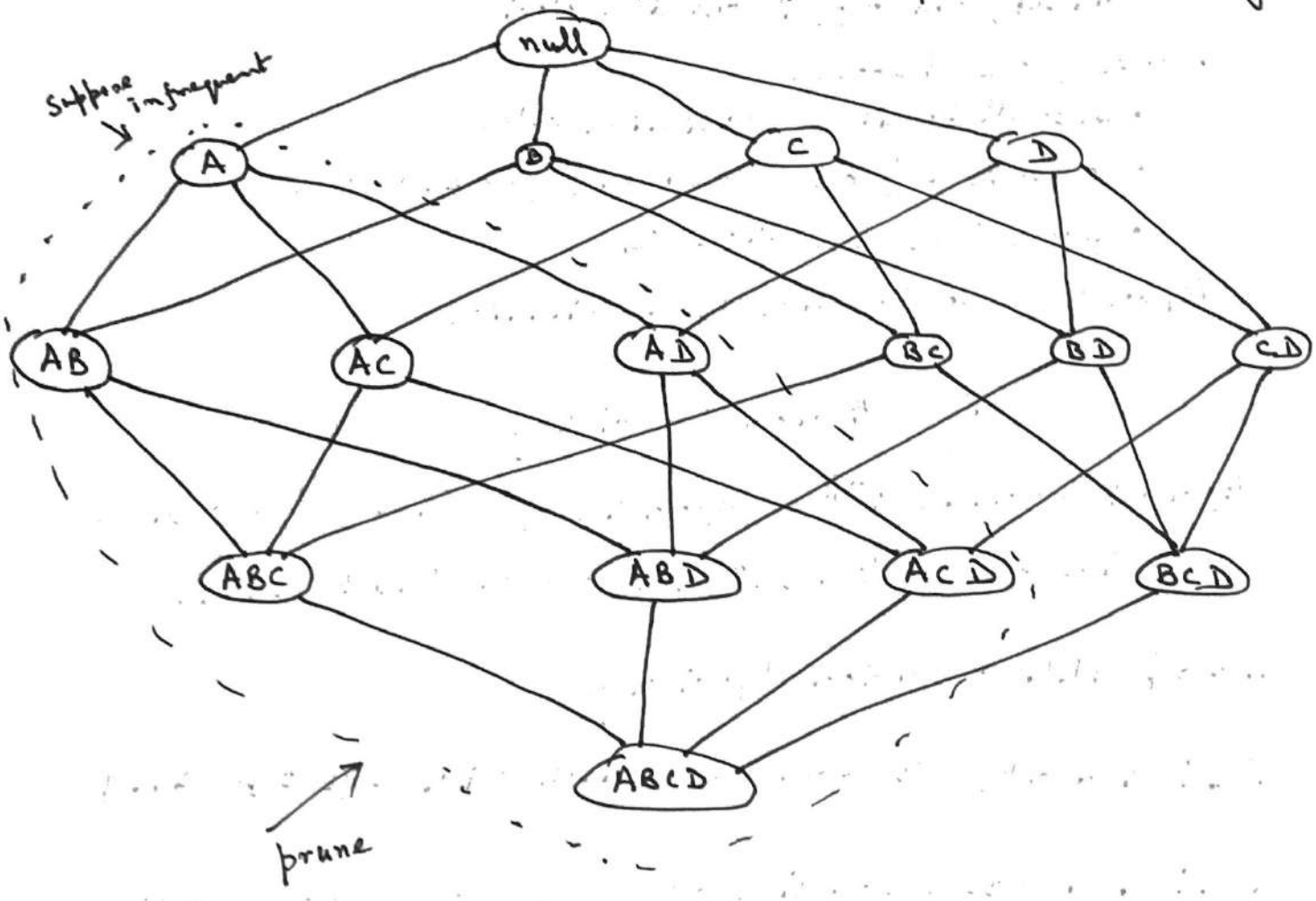
  i.e. $\forall\ X \subseteq Y$

  $$S(X) \geq S(Y) \quad \forall\ X, Y$$

⇒ If an itemset is not frequent, none of it's supersets can be frequent (⇒ prune all such sets)

⇒ If an itemset is not frequent, there is no need to explore its supersets.

Example : Itemset lattice → apriori principle of pruning



- Suppose B is found to be in frequent (support < min sup)

  ⇒ all itemsets with B will also be in frequent

  ⇒ prune all such branches. (i.e. all it's supersets)

  i.e. exclude itemset AB, BC, BD, ABC, ABD, BCD, ABCD

- suppose AB is found to be in frequent

  ⇒ all itemsets with AB will also be in frequent

  ⇒ prune all such branches (i.e. all supersets)

  i.e. exclude itemsets, ABC, ABD, ABCD.

Apriori Step-1 in pseudo codes

- K = 1
- Generate frequent itemsets of length 1
- Prune itemsets of higher orders (i.e. supersets), if necessary
- Generate itemsets of length K+1 from frequent itemsets of length K
- Compute the support of new candidate itemsets w.r.t. min sup. Prune if necessary.
- K = K+1
- Repeat until no frequent itemsets are found.

Generation (step) of itemsets for next level

Let L_K denote the frequent itemsets at level K and C_K denote the set of all candidates at level K

- Items in L_{K-1} are listed in an order

Step 1: Self joining $L_{K-1} * L_{K-1}$

i.e. joining of 2 items from L_{K-1}

$$\{p.item_1, p.item_2 \ldots\ldots p.item_{K-1}\}$$
$$\&\quad \{q.item_1\ q.item_2 \ldots\ldots q.item_{K-1}\}$$

(=, =, =, <) under the given order

insert into C_K the itemset $p.item_1\ p.item_2 \ldots p.item_{K-2}\ p.item_{K-1}\ q.item_{K-1}$

Step 2: Pruning of C_K set

$\forall$ itemsets c in C_K and

$\forall$ $(K-1)$ order subsets s of c (under the given order)

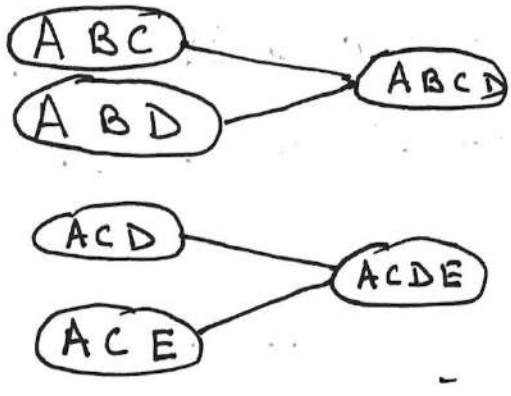
If (s is not in L_{K-1}) delete c from C_K

Justification is anti-monotone property

Example

$$L_3 = \{ABC, ABD, ACD, ACE, BCD\}$$

Self joining step: $L_3 * L_3$



Prunning step:

Consider ACDE → subsets ACD, ACE, CDE, ADE

CDE/ADE is not in L_3

⇒ Prune ACDE from C_4

Consider ABCD → subsets ABC, ABD, ACD, BCD

all subsets in L_3

⇒ Prunning not req'd for ABCD

⇒ Pass ABCD to C_4

Example

T1 : {A, C, D}

T2 : {B, C, E}

T3 : {A, B, C, E}

T4 : {B, E}

min sup = .5

1st Scan →

C_1

item	support count
A	2
B	3
C	3
~~D~~	~~1~~ .. prune
E	3

→ Freq itemset L_1

A	2
B	3
C	3
E	3

→ C_2

AB
AC
AE
BC
BE
CE

2nd Scan →

Itemset	Support Count
~~AB~~	~~1~~
AC	2
~~AE~~	~~1~~
BC	2
BE	3
CE	2

prune

L_2

AC	2
BC	2
BE	3
CE	2

C_3

BCE

all other joins prunned.

L_3

$S(BCE) = 2$

Under C_3, the only Itemset BCE has support count 2, i.e $\geq$ minsup

Step 2 of apriori algorithm

Generate association rules using frequent itemsets

Given any frequent itemset L;

- find all non-empty subsets F of L
- output each rule $F \Rightarrow \{L - F\}$ that satisfies the threshold on confidence (min conf)

Example: Let $L = \{A, B, C\}$ is a frequent itemset

Candidate rules are

$AB \Rightarrow C$; $AC \Rightarrow B$; $BC \Rightarrow A$

$A \Rightarrow BC$; $B \Rightarrow AC$; $C \Rightarrow AB$

In general, there are $2^{|L|} - 2$ candidate rules.

Remark: Efficiency in rule generation

Confidence of rules generated from the same Itemset have anti-monotone property

$$c(ABC \Rightarrow D) \geq c(AB \Rightarrow CD) \geq c(A \Rightarrow BCD)$$

Sly $c(ABC \Rightarrow D) \geq c(AC \Rightarrow BD) \geq c(C \Rightarrow ABD)$

- - - .

[Consider, e.g., $ABC \Rightarrow D$ & $AB \Rightarrow CD$

$AB \subseteq ABC$

$s(AB) \geq s(ABC)$

$\Rightarrow \frac{1}{s(ABC)} \not\geq \frac{1}{s(AB)}$

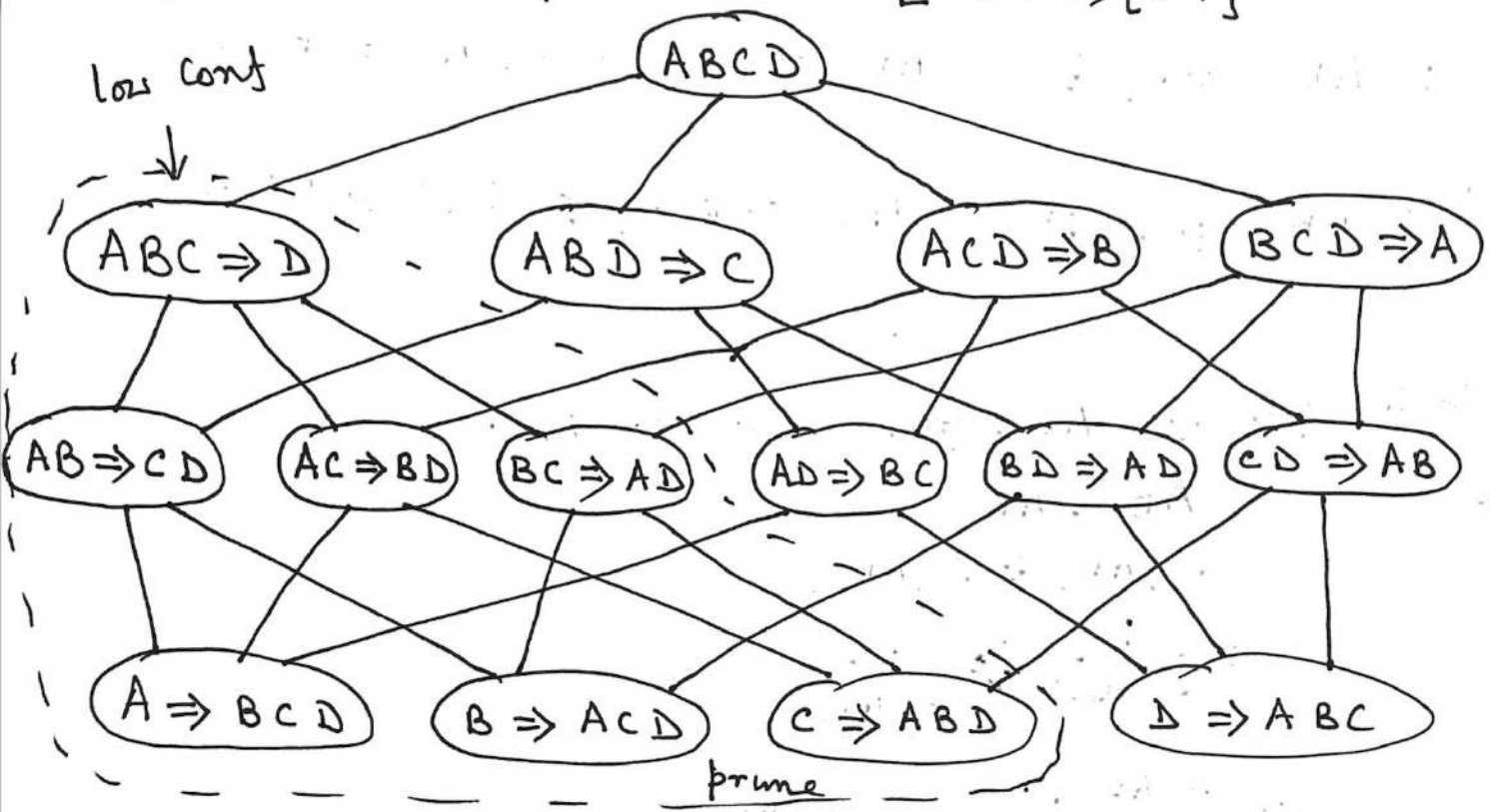
$\Rightarrow \frac{s(ABCD)}{s(ABC)} \geq \frac{s(ABCD)}{s(AB)}$

i.e. $c(ABC \Rightarrow D) \geq c(AB \Rightarrow CD)$]

We apply this property to prune rule generation sequence.

Example of pruning using anti-monotone property

Suppose ABCD is a frequent itemset and look at possible rules for ARM $L \rightarrow F \Rightarrow \{L-F\}$



Suppose the rule ABC ⇒ D is low conf
i.e. C(ABC ⇒ D) < min conf

Then by anti-monotone property, all sub rules below that in the lattice will be < min conf and

can be prunned, i.e.

AB ⇒ CD, AC ⇒ BD, BC ⇒ AD

A ⇒ BCD, B ⇒ ACD & C ⇒ ABD

are prunned.

Example

T1 : {A, C, D}

T2 : {B, C, E}

T3 : {A, B, C, E}

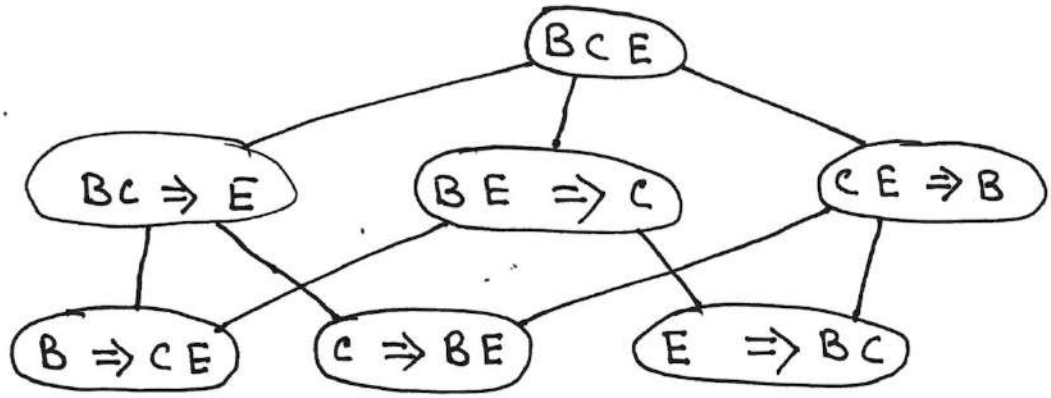
T4 : {B, E}

Frequent Itemsets with min sup ≥ 0.5 (i.e. $s(\cdot) = 2$) are

BCE , AC, BC, BE, CE

$s(\cdot)=2$ $s(\cdot)=2$ $s(\cdot)=3$ $s(\cdot)=2$

Consider rules for BCE Itemset



$$C(BC \Rightarrow E) = \frac{S(BCE)}{S(BC)} = \frac{2}{2} = 1$$

$\Rightarrow BC \Rightarrow E\ (.5, 1)$

$$C(BE \Rightarrow C) = \frac{S(BCE)}{S(BE)} = \frac{2}{3}$$

if min conf = 0.75, then $BE \Rightarrow C$

along with $B \Rightarrow CE$ & $E \Rightarrow BC$

are prunned.

$$C(CE \Rightarrow B) = \frac{S(BCE)}{S(CE)} = \frac{2}{2} = 1$$

$$C(C \Rightarrow BE) = \frac{S(BCE)}{S(C)} = \frac{2}{3}$$ prune at min conf 0.75

Discrimination & Classification

Discriminant analysis: optimal way to separate heterogeneous populations

Classification: Classification of new observation vector in one of the possible populations (classes) using discriminant function.

Examples / Applications

Medical diagnostics, alert systems for adverse events, credit card fraud detection, loan classification, - - - - .

Data structure

Learning sample

$$\mathcal{L} = \{(\underset{\sim}{x}_1, j_1), \ldots, (\underset{\sim}{x}_n, j_n)\}$$

n preclassified examples

$\underset{\sim}{x}_i \in \mathcal{X}$, feature space $\forall\ i = 1(1)n$

$j_i \in \{1, 2, \ldots J\}\ \forall\ i = 1(1)n$

J : possible number of classes.

$C = \{1, 2, \ldots, J\}$ - set of classes

aim: Given $\underset{\sim}{x} \in \mathcal{X}$; to find a systematic way of predicting class membership

i.e. assigns one of the classes in $\{1, 2, \ldots J\}$ to $\underset{\sim}{x} \in \mathcal{X}$.

Def^n : Classifier

A classifier or a classification rule is a f^n $d(.)$ defined on $\mathcal{X} \ni \forall\ \underset{\sim}{x}$, $d(\underset{\sim}{x})$ is equal to one of the numbers $1, 2, \ldots J$.

Note : An alternate way to look at a classifier is through partitions

Let $A_j = \{ \underset{\sim}{x} : d(\underset{\sim}{x}) = j \}$

$A_1, \ldots, A_J$ are $\ni$

$$A_i \cap A_j = \phi \quad \forall\ i \neq j$$

& $\bigcup_i A_i = \mathcal{X}$

Alt def^n : A classifier is a partition of $\mathcal{X}$ into J disjoint set $A_1, \ldots, A_J \ni \forall\ \underset{\sim}{x} \in A_j$ the assigned class membership is j.

Fisher Linear Discriminant Function (FLDF)

Consider a 2-class problem

i.e. 2 pop^{n}s π_1 & π_2

$\underset{\sim}{X} | \pi_1 \sim \underset{\sim}{\mu}_1, \Sigma$; $\underset{\sim}{\mu}_1 = E(\underset{\sim}{X}|\pi_1)$, $\Sigma = Cov(\underset{\sim}{X}|\pi_1)$

$\Sigma > 0$;

$\underset{\sim}{X} | \pi_2 \sim \underset{\sim}{\mu}_2, \Sigma$; $\underset{\sim}{\mu}_2 = E(\underset{\sim}{X}|\pi_2)$, $\Sigma = Cov(\underset{\sim}{X}|\pi_2)$

Aim :
- To find a f^{n} $g(.)$ $\ni$ if $\underset{\sim}{x}_1$ is from π_1 & $\underset{\sim}{x}_2$ is from π_2, then $g(\underset{\sim}{x}_1)$ & $g(\underset{\sim}{x}_2)$ should look as "different as possible". Such a $g(.)$ will be the constructed discriminant f^{n}.
- Given a new obsn $\underset{\sim}{x}$, use $g(.)$ to classify it to one of π_1 & π_2.

FLDF approach : change π_1 & π_2 into univariate pop^{n} by transforming $\underset{\sim}{X}$ to $\underset{\sim}{l}' \underset{\sim}{X}$

$$\Rightarrow \quad \underset{\sim}{l}' \underset{\sim}{X} | \pi_1 \sim \underset{\sim}{l}' \underset{\sim}{\mu}_1, \ \underset{\sim}{l}' \Sigma \underset{\sim}{l}$$

$$\& \ \underset{\sim}{l}' \underset{\sim}{X} | \pi_2 \sim \underset{\sim}{l}' \underset{\sim}{\mu}_2, \ \underset{\sim}{l}' \Sigma \underset{\sim}{l}$$

Note that

Separate out the 2 univariate pop^{n} as much as possible

$\Leftrightarrow$ maximization of distance bet^{n} the 2 pop^{n}s w.r.t. $\underset{\sim}{l}$

Take the distance bet^n the 2 univariate pop^{n}s as

$$\frac{(\underset{\sim}{l}'\underset{\sim}{\mu}_1 - \underset{\sim}{l}'\underset{\sim}{\mu}_2)^2}{\underset{\sim}{l}'\Sigma\,\underset{\sim}{l}} \quad - (*)$$

Maximize $(*)$ w.r.t. $\underset{\sim}{l}$

i.e. find $\underset{\sim}{l}^* = \arg\max_{\underset{\sim}{l}} \dfrac{(\underset{\sim}{l}'(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2))^2}{\underset{\sim}{l}'\Sigma\,\underset{\sim}{l}}$

Define $\underset{\sim}{a}' = \underset{\sim}{l}'\Sigma^{1/2}$

Then $\dfrac{(\underset{\sim}{l}'(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2))^2}{\underset{\sim}{l}'\Sigma\,\underset{\sim}{l}}$

$$= \frac{(\underset{\sim}{a}'\Sigma^{-1/2}(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2))^2}{\underset{\sim}{a}'\underset{\sim}{a}} \quad - (*^2)$$

By C-S inequality

$$(*)^2 \le \frac{(\underset{\sim}{a}'\underset{\sim}{a})(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)'\Sigma^{-1}(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)}{\underset{\sim}{a}'\underset{\sim}{a}}$$

$$= (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)'\Sigma^{-1}(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)$$

→ Square of Mahalanobis distance

Note that the equality in the above is attained

at $\underset{\sim}{a}' = (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)'\Sigma^{-1/2}$

i.e. $\underset{\sim}{l}'\Sigma^{1/2} = (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)'\Sigma^{-1/2}$

i.e. $\underset{\sim}{l}' = (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1}$

So we get the FLDF as

$$g(\underset{\sim}{X}) = (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{X}$$ as the discriminant f^n.

Next task is to frame the classification rule based on FLDF

i.e. given a new obsn $\underset{\sim}{x}_0$, to assign it to π_1 or π_2

Realize that

$$E\left((\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{X} \mid \pi_1 \right) = (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{\mu}_1 = m_1, \text{ say}$$

& $$E\left((\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{X} \mid \pi_2 \right) = (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{\mu}_2 = m_2, \text{ say}$$

and $$m_1 - m_2 = (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)$$

$\Rightarrow$ $m_1 - m_2 \geq 0$; equal to 0 only if $\underset{\sim}{\mu}_1 = \underset{\sim}{\mu}_2$

i.e. $m_1 \geq m_2$

Let $y_0 = (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{x}_0$

We will assign $\underset{\sim}{x}_0$ to π_1 if

y_0 is closer to m_1 than to m_2

i.e. If $y_0 \geq \frac{m_1+m_2}{2}$; we assign $\underset{\sim}{x}_0$ to π_1

i.e If $y_0 \geq \frac{1}{2}(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} (\underset{\sim}{\mu}_1 + \underset{\sim}{\mu}_2)$

and assign $\underset{\sim}{x}_0$ to π_2 if

$$y_0 < \frac{1}{2}(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} (\underset{\sim}{\mu}_1 + \underset{\sim}{\mu}_2)$$

Thus the classification partition associated with FLDF is (R_1, R_2)

$$R_1 = \left\{ \underset{\sim}{x} : (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{x} \geq \frac{1}{2}(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} (\underset{\sim}{\mu}_1 + \underset{\sim}{\mu}_2) \right\}$$

$$R_2 = \left\{ \underset{\sim}{x} : (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{x} < \frac{1}{2}(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} (\underset{\sim}{\mu}_1 + \underset{\sim}{\mu}_2) \right\}$$

Remark: Usually $\underset{\sim}{\mu}_1$, $\underset{\sim}{\mu}_2$, Σ are unknown. We use the learning sample $\mathcal{L}$, of preclassified examples, to estimate $\underset{\sim}{\mu}_1$, $\underset{\sim}{\mu}_2$, Σ and get sample analogue of $(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{x}$ as

$$(\bar{\underset{\sim}{x}}_1 - \bar{\underset{\sim}{x}}_2)' S^{-1} \underset{\sim}{x}$$

This is called the Fisher sample linear discriminant function.

where, $(\underset{\sim}{\bar{x}}_1, S_1)$ are sample mean vector and sample variance covariance matrix computed from pre-classified examples of π_1 class.

Slg, $(\underset{\sim}{\bar{x}}_2, S_2)$ → computed from pre-classified examples of π_2 class.

$$S = \frac{(n_1-1)S_1 + (n_2-1)S_2}{n_1+n_2-2}$$

n_1 : # of pre-classified π_1 cases in $\mathcal{L}$

n_2 : # of pre-classified π_2 cases in $\mathcal{L}$

$$\hat{R}_1 = \left\{ \underset{\sim}{x} : (\underset{\sim}{\bar{x}}_1 - \underset{\sim}{\bar{x}}_2)' S^{-1} \underset{\sim}{x} \geq \frac{1}{2} (\underset{\sim}{\bar{x}}_1 - \underset{\sim}{\bar{x}}_2)' S^{-1} (\underset{\sim}{\bar{x}}_1 + \underset{\sim}{\bar{x}}_2) \right\}$$

$$\hat{R}_2 = \left\{ \underset{\sim}{x} : (\underset{\sim}{\bar{x}}_1 - \underset{\sim}{\bar{x}}_2)' S^{-1} \underset{\sim}{x} < \frac{1}{2} (\underset{\sim}{\bar{x}}_1 - \underset{\sim}{\bar{x}}_2)' S^{-1} (\underset{\sim}{\bar{x}}_1 + \underset{\sim}{\bar{x}}_2) \right\}$$

General Classification problem

Two-class problem

π_1 & π_2 : two classes

Suppose,

$\underset{\sim}{X} | \pi_1$ has support $\mathcal{X}_1$ $\qquad \mathcal{X}_1 \cup \mathcal{X}_2 = \mathcal{X}$

$\underset{\sim}{X} | \pi_2$ has support $\mathcal{X}_2$

if $\mathcal{X}_1 \cap \mathcal{X}_2 = \phi$, then there is no misclassification

But usually $\mathcal{X}_1 \cap \mathcal{X}_2 \neq \phi$ and there is a chance that observations can be misclassified.

Let

$c(i|j)$ denote the cost when an obsn from π_j is misclassified as from π_i

$c(1|1) = c(2|2) = 0$

Suppose the class conditional densities are

$f_i(\underset{\sim}{x})$; $i = 1, 2$

i.e. $\underset{\sim}{X} | \pi_i$ has p.d.f. $f_i(\underset{\sim}{x})$ $\qquad i = 1, 2$

and the prior probabilities for π_1 and π_2 be p_1 and p_2, respy.

Let

$P(i|j)$: probability of misclassifying an observation from π_j into π_i

Suppose the classification partition is $\{R_1, R_2\}$, (i.e. if $\underset{\sim}{x} \in R_i$, we classify it to π_i). Then

$$P(i|j) = \int_{R_i} f_j(\underset{\sim}{x})\, d\underset{\sim}{x}$$

e.g. $$P(2|1) = \int_{R_2} f_1(\underset{\sim}{x})\, d\underset{\sim}{x}$$

& $$P(1|2) = \int_{R_1} f_2(\underset{\sim}{x})\, d\underset{\sim}{x}$$

Also P(an obsn comes from π_i and is misclassified to π_j) $= p_i\, P(j|i)$

Define

- Total Probability of misclassification (TPM).

$$= p_1 P(2|1) + p_2 P(1|2)$$

- Expected cost of misclassification (ECM)

$$= c(2|1)\left(p_1 P(2|1)\right) + c(1|2)\left(p_2 P(1|2)\right)$$

Optimal Strategies

To find the partition $\{R_1^*, R_2^*\}$, say, ∋

TPM or ECM is minimised.

(I) TPM minimizing partition

$$TPM = p_2 \int_{R_1} f_2(\underset{\sim}{x})\, d\underset{\sim}{x} + p_1 \int_{R_2} f_1(\underset{\sim}{x})\, d\underset{\sim}{x}$$

$$= p_2 \int_{R_1} f_2(\underset{\sim}{x})\, d\underset{\sim}{x} + p_1 \int_{\mathcal{X}-R_1} f_1(\underset{\sim}{x})\, d\underset{\sim}{x}$$

$$= p_2 \int_{R_1} f_2(\underset{\sim}{x})\, d\underset{\sim}{x} + \int_{\mathcal{X}} p_1 f_1(\underset{\sim}{x})\, d\underset{\sim}{x} - \int_{R_1} p_1 f_1(\underset{\sim}{x})\, d\underset{\sim}{x}$$

i.e.

$$TPM = \int_{R_1} \left(p_2 f_2(\underset{\sim}{x}) - p_1 f_1(\underset{\sim}{x})\right) d\underset{\sim}{x} + \underbrace{\int_{\mathcal{X}} p_1 f_1(\underset{\sim}{x})\, d\underset{\sim}{x}}_{\text{indep of partition}}$$

Minimization of TPM w.r.t. $\{R_1, R_2\}$

$\Leftrightarrow$

minimization of $\int_{R_1} \left(p_2 f_2(\underset{\sim}{x}) - p_1 f_1(\underset{\sim}{x})\right) d\underset{\sim}{x}$

Note that $\forall\, \underset{\sim}{x} \in \mathcal{X}$: $\left(p_2 f_2(\underset{\sim}{x}) - p_1 f_1(\underset{\sim}{x})\right)$ is either ≤ 0 or > 0.

Thus, the TPM is minimized by choosing

$$R_1^* = \{ \underset{\sim}{x} : p_2 f_2(\underset{\sim}{x}) \leq p_1 f_1(\underset{\sim}{x}) \}$$

i.e.

$$R_1^* = \left\{ \underset{\sim}{x} : \frac{f_1(\underset{\sim}{x})}{f_2(\underset{\sim}{x})} \geq \frac{p_2}{p_1} \right\}$$

& $$R_2^* = \left\{ \underset{\sim}{x} : \frac{f_1(\underset{\sim}{x})}{f_2(\underset{\sim}{x})} < \frac{p_2}{p_1} \right\}$$

In other words,

TPM minimizing classification rule is

assign $\underset{\sim}{x}$ to Π_1 if $\frac{f_1(\underset{\sim}{x})}{f_2(\underset{\sim}{x})} \geq \frac{p_2}{p_1}$

& to Π_2 if o/w.

Remark: Consider testing of

$H_0: \underset{\sim}{x} \sim f_1(\underset{\sim}{x})$ ag $H_A: \underset{\sim}{x} \sim f_2(\underset{\sim}{x})$

By Neyman-Pearson lemma, the MP(α) test has critical region

$$W = \left\{ \underset{\sim}{x} : \frac{f_2(\underset{\sim}{x})}{f_1(\underset{\sim}{x})} > K \right\}$$

where K is $\ni$

$$P_{f_1}\left(\frac{f_2(\underset{\sim}{x})}{f_1(\underset{\sim}{x})} > K \right) = \alpha$$

Thus we observe that

$$R_2^* = \left\{ \underline{x} : \frac{f_2(\underline{x})}{f_1(\underline{x})} > \frac{p_1}{p_2} \right\}$$

Corresponds to the critical region of an MP test of a fixed size (fixed by $\frac{p_1}{p_2}$)

$$P_{f_1}\left(\frac{f_2(\underline{x})}{f_1(\underline{x})} > \frac{p_1}{p_2} \right) = \alpha^*, \text{ say.}$$

Example :

Π_1

$f_1(\underline{x})$

x_2 \ x_1	1	2	3
1	.1	.05	.15
2	.25	.2	.25

Π_2

$f_2(\underline{x})$

x_2 \ x_1	1	2	3
1	.2	.2	.2
2	.2	.1	.1

$$p_1 = p_2 = \frac{1}{2}$$

TPM minimizing rule is

assign $\underline{x}$ to Π_1 If $f_1(\underline{x}) \geq f_2(\underline{x})$

i.e. $R_1^* = \left\{ \underline{x} : f_1(\underline{x}) \geq f_2(\underline{x}) \right\}$

$$\Rightarrow R_1^* = \{ (1,2), (2,2), (3,2) \}$$

$$R_2^* = \{ (1,1), (2,1), (3,1) \}$$

$$P(1|2) = \sum_{\underset{\sim}{x} \in R_1^*} f_2(\underset{\sim}{x})$$

$$= 0.2 + 0.1 + 0.1 = 0.4$$

$$P(2|1) = \sum_{\underset{\sim}{x} \in R_2^*} f_1(\underset{\sim}{x})$$

$$= 0.1 + 0.05 + 0.15 = 0.3$$

$$TPM(R_1^*, R_2^*) = \frac{1}{2}\left(P(1|2) + P(2|1)\right)$$

$$= 0.5(0.4 + 0.3)$$

Note: $p_1 \neq p_2$

If $p_1 = 0.4$ & $p_2 = 0.6$ then

$$R_1^* = \{\underset{\sim}{x} : 0.4\, f_1(\underset{\sim}{x}) \geq 0.6\, f_2(\underset{\sim}{x})\}$$

$$R_2^* = \{\underset{\sim}{x} : 0.4\, f_1(\underset{\sim}{x}) < 0.6\, f_2(\underset{\sim}{x})\}$$

. . . .

Example 2:

$$\pi_1 \equiv N_p(\underset{\sim}{\mu}_1, \Sigma)$$

$$\pi_2 = N_p(\underset{\sim}{\mu}_2, \Sigma) \; ; \; \Sigma > 0$$

$p_1 = p_2$

Optimum TPM rule is

$$R_1^* = \left\{\underset{\sim}{x} : \frac{f_1(\underset{\sim}{x})}{f_2(\underset{\sim}{x})} \geq 1\right\}$$

$$R_2^* = \left\{\underset{\sim}{x} : \frac{f_1(\underset{\sim}{x})}{f_2(\underset{\sim}{x})} < 1\right\}$$

Now, $\dfrac{f_1(\underset{\sim}{x})}{f_2(\underset{\sim}{x})} \geq 1$.

$\Leftrightarrow$

$$(2\pi)^{-p/2}|\Sigma|^{-1/2}\exp\left(-\tfrac{1}{2}(\underset{\sim}{x}-\underset{\sim}{\mu}_1)'\Sigma^{-1}(\underset{\sim}{x}-\underset{\sim}{\mu}_1)\right)$$
$$\geq (2\pi)^{-p/2}|\Sigma|^{-1/2}\exp\left(-\tfrac{1}{2}(\underset{\sim}{x}-\underset{\sim}{\mu}_2)'\Sigma^{-1}(\underset{\sim}{x}-\underset{\sim}{\mu}_2)\right)$$

i.e. $(\underset{\sim}{x}-\underset{\sim}{\mu}_1)'\Sigma^{-1}(\underset{\sim}{x}-\underset{\sim}{\mu}_1) \leq (\underset{\sim}{x}-\underset{\sim}{\mu}_2)'\Sigma^{-1}(\underset{\sim}{x}-\underset{\sim}{\mu}_2)$

i.e. $\underset{\sim}{x}'\Sigma^{-1}\underset{\sim}{x} + \underset{\sim}{\mu}_1'\Sigma^{-1}\underset{\sim}{\mu}_1 - 2\underset{\sim}{\mu}_1'\Sigma^{-1}\underset{\sim}{x}$
$\leq \underset{\sim}{x}'\Sigma^{-1}\underset{\sim}{x} + \underset{\sim}{\mu}_2'\Sigma^{-1}\underset{\sim}{\mu}_2 - 2\underset{\sim}{\mu}_2'\Sigma^{-1}\underset{\sim}{x}$

i.e. $-2(\underset{\sim}{\mu}_1-\underset{\sim}{\mu}_2)'\Sigma^{-1}\underset{\sim}{x} \leq \underset{\sim}{\mu}_2'\Sigma^{-1}\underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_1'\Sigma^{-1}\underset{\sim}{\mu}_1$

i.e. $-2(\underset{\sim}{\mu}_1-\underset{\sim}{\mu}_2)'\Sigma^{-1}\underset{\sim}{x} \leq \underset{\sim}{\mu}_2'\Sigma^{-1}\underset{\sim}{\mu}_2 - \underset{\sim}{\mu}_1'\Sigma^{-1}\underset{\sim}{\mu}_1$
$+ \underset{\sim}{\mu}_2'\Sigma^{-1}\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_1'\Sigma^{-1}\underset{\sim}{\mu}_2$

i.e. $-2(\underset{\sim}{\mu}_1-\underset{\sim}{\mu}_2)'\Sigma^{-1}\underset{\sim}{x} \leq -(\underset{\sim}{\mu}_1-\underset{\sim}{\mu}_2)'\Sigma^{-1}(\underset{\sim}{\mu}_1+\underset{\sim}{\mu}_2)$

i.e. $(\underset{\sim}{\mu}_1-\underset{\sim}{\mu}_2)'\Sigma^{-1}\underset{\sim}{x} \geq \tfrac{1}{2}(\underset{\sim}{\mu}_1-\underset{\sim}{\mu}_2)'\Sigma^{-1}(\underset{\sim}{\mu}_1+\underset{\sim}{\mu}_2)$

Hence the TPM minimizing rule is

Assign $\underset{\sim}{x}$ to π_1 if

$$(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{x} \geq \frac{1}{2}(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} (\underset{\sim}{\mu}_1 + \underset{\sim}{\mu}_2)$$

to π_2 if o/w

Remark:

The above rule is same as FLDF

Remark: If $\pi_1 \equiv N_p(\underset{\sim}{\mu}_1, \Sigma)$

$\pi_2 \equiv N_p(\underset{\sim}{\mu}_2, \Sigma)$ $\quad \Sigma > 0$

Then

$$R_1^* = \{\underset{\sim}{x} : (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{x} \geq \frac{1}{2}(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} (\underset{\sim}{\mu}_1 + \underset{\sim}{\mu}_2) + \log\left(\frac{p_2}{p_1}\right)\}$$

$$R_2^* = \{\underset{\sim}{x} : (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{x} < \frac{1}{2}(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} (\underset{\sim}{\mu}_1 + \underset{\sim}{\mu}_2) + \log\left(\frac{p_2}{p_1}\right)\}$$

Remark: If, the learning sample is to be used to obtain estimated partition $(\hat{R}_1^*, \hat{R}_2^*)$

$$\hat{R}_1^* = \{\underset{\sim}{x} : (\bar{\underset{\sim}{x}}_1 - \bar{\underset{\sim}{x}}_2)' S^{-1} \underset{\sim}{x} \geq \frac{1}{2}(\bar{\underset{\sim}{x}}_1 - \bar{\underset{\sim}{x}}_2)' S^{-1} (\bar{\underset{\sim}{x}}_1 + \bar{\underset{\sim}{x}}_2) + \log\left(\frac{p_2}{p_1}\right)\}$$

Remark:

$(\bar{\underset{\sim}{x}}_1 - \bar{\underset{\sim}{x}}_2)' S^{-1} \underset{\sim}{x} - \frac{1}{2}(\bar{\underset{\sim}{x}}_1 - \bar{\underset{\sim}{x}}_2)' S^{-1} (\bar{\underset{\sim}{x}}_1 + \bar{\underset{\sim}{x}}_2)$ is called Anderson's classification statistic

Optimum classification rules (ECM)

Recall that we have earlier derived optimum classification rule minimizing "Total Probability of Misclassification (TPM)". We now consider the problem of deriving optimum classification rule minimizing "Expected Cost of Misclassification (ECM)".

You may recall that for a 2-class problem, ECM is given by

$$ECM = p_1 P(2|1)\, c(2|1) + p_2 P(1|2)\, c(1|2)$$

p_1 & p_2 are prior probabilities of classes π_1 & π_2

$c(i|j)$: cost of misclassifying an obsn from π_j into π_i

$P(i|j)$: prob of misclassifying an obsn from π_j into π_i

$$P(1|2) = \int_{R_1} f_2(\underset{\sim}{x})\, d\underset{\sim}{x} \; ; \; P(2|1) = \int_{R_2} f_1(\underset{\sim}{x})\, d\underset{\sim}{x}$$

$f_i(\underset{\sim}{x})$: class conditional density of class π_i

$\{R_1, R_2\}$ is the classification partition

$$ECM = p_1 c(2|1) \int_{R_2} f_1(\underset{\sim}{x})\, d\underset{\sim}{x} + p_2 c(1|2) \int_{R_1} f_2(\underset{\sim}{x})\, d\underset{\sim}{x}$$

$$= p_1 c(2|1) \int_{\mathcal{X} - R_1} f_1(\underset{\sim}{x})\, d\underset{\sim}{x} + p_2 c(1|2) \int_{R_1} f_2(\underset{\sim}{x})\, d\underset{\sim}{x}$$

$$= \int_{R_1} \left(p_2 c(1|2) f_2(\underset{\sim}{x}) - p_1 c(2|1) f_1(\underset{\sim}{x}) \right) d\underset{\sim}{x} + \left(p_1 c(2|1) \int_{\mathcal{X}} f_1(\underset{\sim}{x})\, dx \right)$$

← indep of partition $\{R_1, R_2\}$.

The above ECM is minimized w.r.t. $\{R_1, R_2\}$ by choosing

$$R_1^* = \left\{ \underset{\sim}{x} : p_2 C(1|2) f_2(\underset{\sim}{x}) \leq p_1 C(2|1) f_1(\underset{\sim}{x}) \right\}$$

$$R_2^* = \mathcal{X} - R_1^* = \left\{ \underset{\sim}{x} : p_2 C(1|2) f_2(\underset{\sim}{x}) > p_1 C(2|1) f_1(\underset{\sim}{x}) \right\}$$

(logic is same as used for deriving opt TPM rule in class)

Remark: If $C(1|2) = C(2|1)$, then opt ECM rule is same as TPM rule.

Remark: The opt ECM partition $\{R_1^*, R_2^*\}$ correspond to a Most Powerful (MP) test for testing

$H_0: \underset{\sim}{X} \sim f_1(\underset{\sim}{x})$ ag $H_A: \underset{\sim}{X} \sim f_2(\underset{\sim}{x})$

This MP test is of fixed size.

(Recall the logic given in class during discussion on connection bet" opt TPM partition and MP test using Neyman-Pearson lemma)

Example 1: $\pi_1 : N_p(\underset{\sim}{\mu}_1, \Sigma)$

$\pi_2 : N_p(\underset{\sim}{\mu}_2, \Sigma) \quad \Sigma > 0$

$$p_1 f_1(\underset{\sim}{x}) C(2|1) \geq p_2 f_2(\underset{\sim}{x}) C(1|2)$$

$$\Leftrightarrow \quad \frac{f_1(\underset{\sim}{x})}{f_2(\underset{\sim}{x})} \geq \frac{p_2 C(1|2)}{p_1 C(2|1)}$$

$$\Leftrightarrow \quad \frac{(2\pi)^{-p/2} |\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(\underset{\sim}{x} - \underset{\sim}{\mu}_1)' \Sigma^{-1} (\underset{\sim}{x} - \underset{\sim}{\mu}_1)\right)}{(2\pi)^{-p/2} |\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(\underset{\sim}{x} - \underset{\sim}{\mu}_2)' \Sigma^{-1} (\underset{\sim}{x} - \underset{\sim}{\mu}_2)\right)} \geq \frac{p_2 C(1|2)}{p_1 C(2|1)}$$

$\Leftrightarrow$

$$-\frac{1}{2}\left((\underset{\sim}{x}-\underset{\sim}{\mu}_1)'\Sigma^{-1}(\underset{\sim}{x}-\underset{\sim}{\mu}_1)-(\underset{\sim}{x}-\underset{\sim}{\mu}_2)'\Sigma^{-1}(\underset{\sim}{x}-\underset{\sim}{\mu}_2)\right) \geq \log\left(\frac{p_2\,C(1|2)}{p_1\,C(2|1)}\right)$$

$\Leftrightarrow$

$$-\frac{1}{2}\left(\underset{\sim}{x}'\Sigma^{-1}\underset{\sim}{x}+\underset{\sim}{\mu}_1'\Sigma^{-1}\underset{\sim}{\mu}_1-2\underset{\sim}{\mu}_1'\Sigma^{-1}\underset{\sim}{x}-\underset{\sim}{x}'\Sigma^{-1}\underset{\sim}{x}-\underset{\sim}{\mu}_2'\Sigma^{-1}\underset{\sim}{\mu}_2+2\underset{\sim}{\mu}_2'\Sigma^{-1}\underset{\sim}{x}\right) \geq \log\left(\frac{p_2\,C(1|2)}{p_1\,C(2|1)}\right)$$

$\Leftrightarrow$

$$-\frac{1}{2}\left(\underbrace{-2(\underset{\sim}{\mu}_1-\underset{\sim}{\mu}_2)'\Sigma^{-1}\underset{\sim}{x}}_{\text{FLDF}}-\underbrace{(\underset{\sim}{\mu}_2'\Sigma^{-1}\underset{\sim}{\mu}_2-\underset{\sim}{\mu}_1'\Sigma^{-1}\underset{\sim}{\mu}_1+\underset{\sim}{\mu}_2'\Sigma^{-1}\underset{\sim}{\mu}_1-\underset{\sim}{\mu}_1'\Sigma^{-1}\underset{\sim}{\mu}_2)}_{=0}\right) \geq \log\left(\frac{p_2\,C(1|2)}{p_1\,C(2|1)}\right)$$

$\Leftrightarrow$

$$-\frac{1}{2}\left(-2(\underset{\sim}{\mu}_1-\underset{\sim}{\mu}_2)'\Sigma^{-1}\underset{\sim}{x}-(\underset{\sim}{\mu}_2-\underset{\sim}{\mu}_1)'\Sigma^{-1}(\underset{\sim}{\mu}_1+\underset{\sim}{\mu}_2)\right) \geq \log\left(\frac{p_2\,C(1|2)}{p_1\,C(2|1)}\right)$$

i.e. $$R_1^* = \left\{\underset{\sim}{x} : +\frac{1}{2}\left(2(\underset{\sim}{\mu}_1-\underset{\sim}{\mu}_2)'\Sigma^{-1}\underset{\sim}{x}-(\underset{\sim}{\mu}_1-\underset{\sim}{\mu}_2)'\Sigma^{-1}(\underset{\sim}{\mu}_1+\underset{\sim}{\mu}_2)\right) \geq \log\left(\frac{p_2\,C(1|2)}{p_1\,C(2|1)}\right)\right\}$$

$$R_2^* = \mathcal{X} - R_1^*$$

Sp. Case : If $p_2\,C(1|2) = p_1\,C(2|1)$, then

$$\tilde{R}_1^* = \left\{\underset{\sim}{x} : (\underset{\sim}{\mu}_1-\underset{\sim}{\mu}_2)'\Sigma^{-1}\underset{\sim}{x} \geq \frac{1}{2}(\underset{\sim}{\mu}_1-\underset{\sim}{\mu}_2)'\Sigma^{-1}(\underset{\sim}{\mu}_1+\underset{\sim}{\mu}_2)\right\}$$

which is same as classification partition based on FLDF.

Remark : R_1^* given above estimated from learning sample $\mathcal{L}$ of preclassified examples.

Example 2

$$\pi_1 : N_p(\underset{\sim}{\mu}_1, \Sigma_1)$$

$$\pi_2 : N_p(\underset{\sim}{\mu}_2, \Sigma_2) \; ; \; \Sigma_i > 0$$

$$R_1^* = \{ \underset{\sim}{x} : p_1 f_1(\underset{\sim}{x}) C(2|1) \geq p_2 f_2(\underset{\sim}{x}) C(1|2) \}$$

Proceeding as in example 1, we get

$$p_1 f_1(\underset{\sim}{x}) C(2|1) \geq p_2 f_2(\underset{\sim}{x}) C(1|2)$$

$\Leftrightarrow$

$$-\frac{1}{2}\left((\underset{\sim}{x} - \underset{\sim}{\mu}_1)' \Sigma_1^{-1} (\underset{\sim}{x} - \underset{\sim}{\mu}_1) - (\underset{\sim}{x} - \underset{\sim}{\mu}_2)' \Sigma_2^{-1} (\underset{\sim}{x} - \underset{\sim}{\mu}_2) \right)$$
$$- \frac{1}{2} \log \left(\frac{|\Sigma_1|}{|\Sigma_2|} \right) \geq \log \left(\frac{p_2 C(1|2)}{p_1 C(2|1)} \right)$$

$\Leftrightarrow$

$$-\frac{1}{2}\left(\underset{\sim}{x}' (\Sigma_1^{-1} - \Sigma_2^{-1}) \underset{\sim}{x} \right) + (\underset{\sim}{\mu}_1' \Sigma_1^{-1} - \underset{\sim}{\mu}_2' \Sigma_2^{-1}) \underset{\sim}{x}$$
$$\geq \log \left(\frac{p_2 C(1|2)}{p_1 C(2|1)} \right) + \frac{1}{2} \log \left(\frac{|\Sigma_1|}{|\Sigma_2|} \right) + \frac{1}{2}(\underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{\mu}_2)$$

(indep of $\underset{\sim}{x}$)

The L.h.s. in above expression is called the quadratic discriminant fn (for obvious reason).

Assignment rule:

For a new $\underset{\sim}{x}_0$;

Assign $\underset{\sim}{x}_0$ to π_1 if

$$-\frac{1}{2} \underset{\sim}{x}_0' (\Sigma_1^{-1} - \Sigma_2^{-1}) \underset{\sim}{x}_0 + (\underset{\sim}{\mu}_1' \Sigma_1^{-1} - \underset{\sim}{\mu}_2' \Sigma_2^{-1}) \underset{\sim}{x}_0$$
$$\geq \log \left(\frac{p_2 C(1|2)}{p_1 C(2|1)} \right) + \frac{1}{2} \log \left(\frac{|\Sigma_1|}{|\Sigma_2|} \right) + \frac{1}{2} (\underset{\sim}{\mu}_1' \Sigma_1^{-1} \underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2' \Sigma_2^{-1} \underset{\sim}{\mu}_2)$$

Remark: Note that we have to use the learning sample $\mathcal{L}$ to have sample counterpart $\{\hat{R}_1, \hat{R}_2\}$

If in $\mathcal{L}$ there are n_1 preclassified obsns from Π_1 & n_2 preclassified cases from Π_2 ($n_1 + n_2 = n \leftarrow$ total preclassified cases in $\mathcal{L}$), then use n_1 obsns of preclassified Π_1 examples to calculate $\hat{\underset{\sim}{\mu}}_1 = \bar{\underset{\sim}{x}}_{(1)}$ & $\hat{\Sigma}_1 = S_{(1)}$

Ily $\hat{\underset{\sim}{\mu}}_2 = \bar{\underset{\sim}{x}}_{(2)}$ & $\hat{\Sigma}_2 = S_{(2)}$ from n_2 preclassified cases from Π_2.

Note that in this case no pooling is required as would be necessary in example 1 for estimation of common covariance matrix.

Example 3 : Discrete dist^n case

π_1 p.m.f.

x_2 \ x_1	0	1	2
0	.1	.05	.15
1	.25	.2	.25

Prior prob

$p_1 = 0.4$

π_2 p.m.f

x_2 \ x_1	0	1	2
0	.2	.2	.2
1	.2	.1	.1

Prior prob

$p_2 = 0.6$

Misclassification cost

$$c(1|2) = 5 \qquad c(2|1) = 10$$

$$c(1|1) = c(2|2) = 0 \text{ (always)}$$

$\forall$ possible pairs $(0,0), (0,1), (1,0), (1,1)$
$(2,0), (2,1)$;

Check the assignment

e.g. $(0,0) = \underset{\sim}{x}$

$$p_1 f_1(\underset{\sim}{x}) c(2|1) = 0.4 \times 0.1 \times 10 = 0.4$$

& $$p_2 f_2(\underset{\sim}{x}) c(1|2) = 0.6 \times 0.2 \times 5 = 0.6$$

Recall that $$R_1^* = \{\underset{\sim}{x} : p_1 f_1(\underset{\sim}{x}) c(2|1) \geq p_2 f_2(\underset{\sim}{x}) c(1|2)\}$$

Δ $R_2^* = \{\underset{\sim}{x} : \quad ,, \quad < \quad ,, \quad \}$

Hence assign rule for $(0,0)$ is π_2

Similarly assignment rules for all other possible (x_1, x_2) can be computed.

Remark: For all the 3 examples ECM (or TPM) of the ECM minimizing classification rules can be calculated once we calculate $P(1|2)$ & $P(2|1)$

e.g: Consider "Example 1" of $\pi_1 : N_p(\underset{\sim}{\mu}_1, \Sigma_0)$

$\pi_2 : N_p(\underset{\sim}{\mu}_2, \Sigma)$

$$P(1|2) = P_{\pi_2}\left(\underset{\sim}{X} \in R_1^*\right)$$

$$= P_{\pi_2}\left((\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{X} \geq \log\left(\frac{p_2 C(1|2)}{p_1 C(2|1)}\right) + \frac{1}{2}(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} (\underset{\sim}{\mu}_1 + \underset{\sim}{\mu}_2)\right)$$

Note that under π_2:

$$Z_1 = (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{X} \sim N_1\left((\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{\mu}_2, \underbrace{(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)}_{= \Delta^2 \leftarrow \text{Sq of Mahalanobis distance}}\right)$$

$$\Rightarrow P(1|2) = P_{\pi_2}\Big(\underbrace{\frac{Z_1 - (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{\mu}_2}{\Delta}}_{\text{under } \pi_2 \sim N(0,1)} \geq \Big[\frac{1}{2}(\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} (\underset{\sim}{\mu}_1 + \underset{\sim}{\mu}_2) - (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{\mu}_2 + \log\left(\frac{p_2 C(1|2)}{p_1 C(2|1)}\right)\Big] \Delta^{-1}\Big)$$

Let $Y = \dfrac{Z_1 - (\underset{\sim}{\mu}_1 - \underset{\sim}{\mu}_2)' \Sigma^{-1} \underset{\sim}{\mu}_2}{\Delta}$

$$P(1|2) = P_{\pi_2}\left(Y \geq \frac{1}{2}\Delta + \frac{1}{\Delta} \log\left(\frac{p_2 C(1|2)}{p_1 C(2|1)}\right)\right)$$

$$\Rightarrow P(1|2) = 1 - \Phi\left(\frac{\Delta}{2} + \frac{1}{\Delta}\log\left(\frac{p_2 C(1|2)}{p_1 C(2|1)}\right)\right)$$

Sly $P(2|1)$ can be calculated

Note: TPM / ECM can be estimated by estimating Δ from learning set $\mathcal{L}$.

Remark: Performance measure for classifier

Apparent error rate (APER)

- Calculate the "confusion matrix"

Actual class \ Predicted class	π_1	π_2	
π_1	n_{1C}	n_{1M}	$(n_{1C} + n_{1M} = n_1)$
π_2	n_{2M}	n_{2C}	$(n_{2M} + n_{2C} = n_2)$

n_{iM} : # of cases from π_i misclassified by classifier

n_{ic} : # of - - - - π_i correctly classified

$$APER = \frac{n_{1M} + n_{2M}}{n_1 + n_2}$$

Classification in multiclass problem

Let $\pi_1, \pi_2, \ldots, \pi_C$ be C classes with prior probabilities $p(\pi_1), p(\pi_2), \ldots p(\pi_C)$, respectively.

Bayes classifier

Assign $\underset{\sim}{x}$ to π_j if the posterior prob of the class π_j given the obsn $\underset{\sim}{x}$, i.e., $p(\pi_j | \underset{\sim}{x})$ is the highest over all classes $\pi_1, \ldots, \pi_C$

i.e. Assign $\underset{\sim}{x}$ to π_j if

$$p(\pi_j | \underset{\sim}{x}) \geq p(\pi_k | \underset{\sim}{x}) \; ; \; k = 1, \ldots, C, \; k \neq j$$

$$\Leftrightarrow \quad \frac{p(\pi_j)\, p(\underset{\sim}{x} | \pi_j)}{\sum_{i=1}^{C} p(\pi_i) p(\underset{\sim}{x} | \pi_i)} \geq \frac{p(\pi_k)\, p(\underset{\sim}{x} | \pi_k)}{\sum_{i=1}^{C} p(\pi_i)\, p(\underset{\sim}{x} | \pi_i)}$$

$$\Leftrightarrow \quad p(\pi_j)\, p(\underset{\sim}{x} | \pi_j) \geq p(\pi_k)\, p(\underset{\sim}{x} | \pi_k)$$

$\Rightarrow$ as in the 2-class problem (done in class), Bayes classifier for a $C (\geq 2)$-class problem can be expressed in terms of prior and class conditional densities

TPM minimizing classifier

TPM for a general c-class problem can be defined as

$$TPM = \sum_{i=1}^{c} p(\pi_i)\, P(\text{error} \mid \pi_i)$$

↑ misclassification

i.e. $P(\text{error} \mid \pi_i)$: prob of misclassifying a pattern from π_i

Let $\{R_1, R_2, \ldots, R_c\}$ be the classification partition, then

$$P(\text{error} \mid \pi_i) = \int_{R_i^c} f(\underset{\sim}{x} \mid \pi_i)\, d\underset{\sim}{x}$$

$$\Rightarrow TPM = \sum_{i=1}^{c} \int_{\mathcal{X}-R_i} f(\underset{\sim}{x} \mid \pi_i)\, p(\pi_i)\, d\underset{\sim}{x}$$

$$= \sum_{i=1}^{c} p(\pi_i) \int_{\mathcal{X}-R_i} f(\underset{\sim}{x} \mid \pi_i)\, d\underset{\sim}{x}$$

$$= \sum_{i=1}^{c} p(\pi_i) \left(\int_{\mathcal{X}} f(\underset{\sim}{x} \mid \pi_i)\, d\underset{\sim}{x} - \int_{R_i} f(\underset{\sim}{x} \mid \pi_i)\, d\underset{\sim}{x} \right)$$

$$= \sum_{i=1}^{c} p(\pi_i) \left(1 - \int_{R_i} f(\underset{\sim}{x} \mid \pi_i)\, d\underset{\sim}{x} \right.$$

i.e.

$$TPM = 1 - \sum_{i=1}^{C} \int_{R_i} p(\pi_i) f(\underset{\sim}{x} | \pi_i) \, d\underset{\sim}{x}$$

$\Rightarrow$ Minimizing the TPM w.r.t. partition $\{R_1, \ldots R_C\}$ is equivalent to maximizing

$$\sum_{i=1}^{C} \int_{R_i} p(\pi_i) f(\underset{\sim}{x} | \pi_i) \, d\underset{\sim}{x} \quad \text{w.r.t. } \{R_1, \ldots, R_C\}$$

This can be viewed as prob of correct classification

Maximization of the above is achieved by selecting R_i to be the region for which $p(\pi_i) f(\underset{\sim}{x} | \pi_i)$ is the largest among all classes (i.e among all $i = 1(1)C$)

i.e. the TPM minimizing classification rule is

Assign $\underset{\sim}{x}$ to π_i if

$$p(\pi_i) f(\underset{\sim}{x} | \pi_i) \geq p(\pi_K) f(\underset{\sim}{x} | \pi_K) \quad \forall K \neq i$$

Remark: As in the 2-class problem, the TPM minimizing rule is equivalent to the one obtained by maximizing the "posterior prob", i.e. the Bayes classifier

Minimum ECM classification rule

Let $p(\pi_i) = p_i \quad i = 1(1)C$

class conditional densities $f(\underset{\sim}{x} | \pi_i) ; i = 1(1)C$

$c(k|i)$: cost of misclassifying an item/pattern from π_i to π_k ; $k, i = 1(1)C$

$c(i|i) = 0 \quad \forall \; i = 1(1)C$

$\{R_1, \ldots, R_C\}$: classification partition

$$P(k|i) = P(\text{classifying an item to } \pi_k | \pi_i)$$

$$= \int_{R_k} f(\underset{\sim}{x} | \pi_i) \, d\underset{\sim}{x}$$

$$P(i|i) = \int_{R_i} f(\underset{\sim}{x} | \pi_i) \, d\underset{\sim}{x} = \int_{\mathcal{X} - \bigcup_{k \neq i} R_k} f(\underset{\sim}{x} | \pi_i) \, d\underset{\sim}{x}$$

$$= 1 - \sum_{\substack{k=1 \\ k \neq i}}^{C} P(k|i)$$

Note that the conditional ECM of $\underset{\sim}{x}$ from π_1 into π_2 or π_3 or ... or π_C is

$$ECM_1 = P(2|1)\,c(2|1) + P(3|1)\,c(3|1) + \ldots + P(C|1)\,c(C|1)$$

$$= \sum_{k=2}^{C} P(k|1)\, c(k|1)$$

The above ECM_1 is with prob $p(\pi_1) = p_1$

Thus for other conditional ECM's ECM_i

we can write similar expressions w.p. p_i

(as below)

$\Rightarrow$

$$ECM = p_1 ECM_1 + p_2 ECM_2 + \cdots + p_c ECM_c$$

i.e. $$ECM = p_1 \left(\sum_{K=2}^{c} P(K|1) C(K|1) \right)$$

$$+ p_2 \left(\sum_{\substack{K=1 \\ K \neq 2}}^{c} P(K|2)\, C(K|2) \right) + \cdots$$

$$\cdots + p_c \left(\sum_{K=1}^{c-1} P(K|c)\, C(K|c) \right)$$

i.e. $$ECM = \sum_{i=1}^{c} p_i \sum_{\substack{K=1 \\ K \neq i}}^{c} P(K|i)\, C(K|i)$$

$$= \sum_{K=1}^{c} \sum_{\substack{i=1 \\ i \neq K}}^{c} p_i\, P(K|i)\, C(K|i)$$

$$= \sum_{K=1}^{c} \sum_{\substack{i=1 \\ i \neq K}}^{c} \int_{R_K} p_i\, C(K|i)\, f(\underset{\sim}{x}|\pi_i)\, d\underset{\sim}{x}$$

$$\text{i.e. } ECM = \sum_{K=1}^{C} \int_{R_K} \left(\sum_{\substack{i=1 \\ i \neq K}}^{C} p_i \, c(K|i) \, f(\underset{\sim}{x}|\pi_i) \right) d\underset{\sim}{x}$$

$$= \sum_{K=1}^{C} \int_{R_K} h_K(\underset{\sim}{x}) \, d\underset{\sim}{x}$$

Note that

$$\sum_{K=1}^{C} \int_{R_K} \left(h_K(\underset{\sim}{x}) - \min_j h_j(\underset{\sim}{x}) \right) d\underset{\sim}{x} \geq 0$$

with equality only if

$$h_K(\underset{\sim}{x}) = \min_j h_j(\underset{\sim}{x}) \quad \forall \underset{\sim}{x} \text{ in } R_K$$

Thus the ECM minimizing classification rule is

Assign $\underset{\sim}{x}$ to π_K if

$$\sum_{\substack{i=1 \\ i \neq K}}^{C} p_i \, c(K|i) \, f(\underset{\sim}{x}|\pi_i)$$ is minimum among all such c expressions

Remark: under equal cost setup, i.e. all misclassification costs are same

we assign $\underset{\sim}{x}$ to π_K if

$$\sum_{\substack{i=1 \\ i \neq K}}^{C} p_i \, f(\underset{\sim}{x}|\pi_i)$$ is the smallest

i.e. $$\sum_{\substack{i=1 \\ i \neq K}}^{c} p_i f(\underset{\sim}{x}|\pi_i) \leq \sum_{\substack{i=1 \\ i \neq j}}^{c} p_i f(\underset{\sim}{x}|\pi_i) \quad -(*) \qquad \forall\, j \neq K$$

Subtracting $\sum_{\substack{i=1 \\ i \neq K,j}}^{c} p_i f(\underset{\sim}{x}|\pi_i)$ from both sides of $(*)$, we get

$$\sum_{\substack{i=1 \\ i \neq K}}^{c} p_i f(\underset{\sim}{x}|\pi_i) - \sum_{\substack{i=1 \\ i \neq K,j}}^{c} p_i f(\underset{\sim}{x}|\pi_i)$$

$$\leq \sum_{\substack{i=1 \\ i \neq j}}^{c} p_i f(\underset{\sim}{x}|\pi_i) - \sum_{\substack{i=1 \\ i \neq K,j}}^{c} p_i f(\underset{\sim}{x}|\pi_i)$$

i.e. $p_j f(\underset{\sim}{x}|\pi_j) \leq p_K f(\underset{\sim}{x}|\pi_K) \quad \forall\, j \neq K$

i.e the rule is

assign $\underset{\sim}{x}$ to π_K if

$p_K f(\underset{\sim}{x}|\pi_K)$ is largest

This is the same as TPM minimizing rule (what we expect under equal misclassification cost).

Further, the above is (under equal cost) equivalent to Bayes classifier.

Multiclass classification problem

Example 1: 3-class problem

Consider the following cost-prior table

	True membership π_1	π_2	π_3
π_1	$c(1\|1)=0$	$c(1\|2)=500$	100
classified as π_2	10	0	50
π_3	50	200	0
prior prob	$p_1 = 0.05$	$p_2 = 0.60$	$p_3 = 0.35$

← misclassification cost matrix

Let $\underset{\sim}{x}_0$ be a new obsn $\Rightarrow$

$$f(\underset{\sim}{x}_0 | \pi_1) = 0.01 \; ; \; f(\underset{\sim}{x}_0 | \pi_2) = 0.85$$
$$f(\underset{\sim}{x}_0 | \pi_3) = 2$$

ECM minimizing classifier

Compute $\sum_{\substack{i=1 \\ i \neq K}}^{3} p_i \, c(K|i) \, f(\underset{\sim}{x}_0 | \pi_i) \quad \forall K = 1, 2, 3$

$K = 1$: $\sum_{i=2}^{3} p_i \, c(1|i) \, f(\underset{\sim}{x}_0 | \pi_i)$

$$= p_2 \, c(1|2) \, f(\underset{\sim}{x}_0 | \pi_2) + p_3 \, f(\underset{\sim}{x}_0 | \pi_3) \, c(1|3)$$
$$= 325$$

$K=2$: $\sum_{\substack{i=1 \\ i\neq 2}}^{3} p_i\, c(2|i)\, f(\underset{\sim}{x}_0|\pi_i)$

$= p_1\, c(2|1)\, f(\underset{\sim}{x}_0|\pi_1) + p_3\, c(2|3)\, f(\underset{\sim}{x}_0|\pi_3)$

$= 35.06$

$K=3$: $\sum_{i=1}^{2} p_i\, c(3|i)\, f(\underset{\sim}{x}_0|\pi_i) = 102.03$

Since $\sum_{\substack{i=1 \\ i\neq 2}}^{3} p_i\, c(2|i)\, f(\underset{\sim}{x}_0|\pi_i)$ is the smallest

we assign $\underset{\sim}{x}_0$ to π_2

Note: Suppose in the same example, we take equal cost $\Rightarrow$ TPM minimizing rule

Calculate $p_1 f(\underset{\sim}{x}_0|\pi_1) = 0.0005$

$p_2 f(\underset{\sim}{x}_0|\pi_2) = 0.510$

$p_3 f(\underset{\sim}{x}_0|\pi_3) = 0.7$

Since $p_3 f(\underset{\sim}{x}_0|\pi_3) > p_i f(\underset{\sim}{x}_0|\pi_i)$ $i=1,2$

$\underset{\sim}{x}_0$ is allocated to π_3

Bayes classifier would also have done the same assignment

Example 3 : Discrete populations

Consider 3 bivariate discrete populations with the following j.p.m.f.s :

Π_1

x_2 \ x_1	1	2
1	0.5	0.2
2	0.1	0.2

Π_2

x_2 \ x_1	1	2
1	0.2	0.1
2	0.3	0.4

Π_3

x_2 \ x_1	1	2
1	0.25	0.25
2	0.25	0.25

Prior probabilities are equal, i.e. $p(\pi_i) = \frac{1}{3}$; $i = 1,2,3$ $(= p_i)$

Misclassification costs are

$$c(1|i) = 1 \; ; \; i = 2,3$$

$$c(2|i) = 2 \; ; \; i = 1,3$$

$$c(3|i) = 3 \; ; \; i = 1,2$$

Let us denote by $f_i(\underset{\sim}{x})$ to be class conditional prob for class π_i ; $i = 1,2,3$.

ECM classification rule is :

Assign $\underset{\sim}{x}$ to π_K if

$$\sum_{\substack{i=1 \\ i \neq K}}^{3} p_i \, f_i(\underset{\sim}{x}) \, c(K|i) \quad \text{is smallest}$$

Based on the above find rule for all pairs $\underset{\sim}{x}$

Pair (1,1)

$\underset{\sim}{x} = (1,1)$

$K=1$ $\quad \sum_{\substack{i=2}}^{3} p_i f_i(\underset{\sim}{x}) c(1|i)$

$= p_2 f_2(\underset{\sim}{x}) c(1|2) + p_3 f_3(\underset{\sim}{x}) c(1|3)$

$= \frac{1}{3}\left(f_2(\underset{\sim}{x}) + f_3(\underset{\sim}{x})\right)$

$= \frac{1}{3}(0.2 + 0.25) = 0.15$

$K=2$ $\quad \sum_{\substack{i=1 \\ i\neq 2}}^{3} p_i f_i(\underset{\sim}{x}) c(2|i)$ $\quad \underset{\sim}{x} = (1,1)$

$= p_1 f_1(\underset{\sim}{x}) c(2|1) + p_3 f_3(\underset{\sim}{x}) c(2|3)$

$= \frac{2}{3}\left(f_1(\underset{\sim}{x}) + f_3(\underset{\sim}{x})\right) = \frac{2}{3}(0.5 + 0.25)$

$= 0.5$

$K=3$ $\quad \sum_{i=1}^{2} p_i f_i(\underset{\sim}{x}) c(3|i)$

$= p_1 f_1(\underset{\sim}{x}) c(3|1) + p_2 f_2(\underset{\sim}{x}) c(3|2)$

$= \frac{3}{3}\left(f_1(\underset{\sim}{x}) + f_2(\underset{\sim}{x})\right) = 0.7$

$\sum_{\substack{i=1 \\ i\neq 1}}^{3} p_i f_i(\underset{\sim}{x}) c(1|i)$ is smallest

$\Rightarrow$ Assign $\underset{\sim}{x} = (1,1)$ to π_1

i.e. $(1,1) \in R_1$ partition

Pair (1,2)

$K=1 \rightarrow \frac{1}{3}\left(f_2(1,2) + f_3(1,2)\right) = \frac{1}{3}[0.55)$ ← smallest

$K=2 \rightarrow \frac{2}{3}\left(f_1(\overset{(1,2)}{\overset{\swarrow}{x}}) + f_3(1,2)\right) = \frac{2}{3}[0.35)$

$K=3 \rightarrow \frac{3}{3}\left(f_1(x,2) + f_2(1,2)\right) = 0.4$

$K=1$ gives smallest

$\Rightarrow (1,2) \in R_1$ partition

Pair (2,1)

$K=1 \rightarrow \frac{1}{3}\left(f_2(2,1) + f_3(2,1)\right) = \frac{1}{3}(0.35)$

$K=2 \rightarrow \frac{2}{3}\left(f_1(2,1) + f_3(2,1)\right) = \frac{2}{3}(0.45)$

$K=3 \rightarrow \frac{3}{3}\left(f_1(2,1) + f_2(2,1)\right) = 0.3$

$K=1$ gives smallest

$\Rightarrow (2,1) \in R_1$ partition

Pair (2,2)

$K=1 \rightarrow \frac{1}{3}\left(f_2(2,2) + f_3(2,2)\right) = \frac{1}{3}(0.65)$

$K=2 \rightarrow \frac{2}{3}\left(f_1(2,2) + f_3(2,2)\right) = \frac{2}{3}[0.45)$

$K=3 \rightarrow \frac{3}{3}\left(f_1(2,2) + f_2(2,2)\right) = 0.6$

$\Rightarrow (2,2) \in R_1$

$\Rightarrow$ ECM minimizing classification rule is

x_2 \ x_1	1	2
1	π_1	π_1
2	π_1	π_1.

Suppose we have a set $\mathcal{L}$ of preclassified examples

$$\mathcal{D} = \{((1,2),\pi_1), ((1,1),\pi_1), ((1,1),\pi_1), ((1,1),\pi_3), ((2,2),\pi_2), ((2,1),\pi_3), ((1,2),\pi_3), ((2,2),\pi_3)\}$$

If we apply ECM rule on $\mathcal{D}$, we get

classified into \ coming from	π_1	π_2	π_3
π_1	3	1	4
π_2	0	0	0
π_3	0	0	0

Misclassification rate would be $\frac{5}{8}$!!

↑ not good at all!

(i) Try to find the TPM opt rule

(ii) Try to calculate ECM/TPM of the two optimum rule

Example 2: Multiclass Gaussian

$$\pi_i \equiv N_p(\underset{\sim}{\mu}_i, \Sigma_i)$$

Assume for simplicity that $C(k|i)$ s are all same

We know that under this equal cost, the rule is

Assign $\underset{\sim}{x}$ to π_k if

$$p_k f(\underset{\sim}{x}|\pi_k) = \max_i p_i f(\underset{\sim}{x}|\pi_i)$$

i.e if $\log\left(p_k f(\underset{\sim}{x}|\pi_k)\right) = \max_i \log\left(p_i f(\underset{\sim}{x}|\pi_i)\right)$

with $f(\underset{\sim}{x}|\pi_k) = (2\pi)^{-p/2}|\Sigma_k|^{-1/2}\exp\left(-\frac{1}{2}(\underset{\sim}{x}-\underset{\sim}{\mu}_k)'\Sigma_k^{-1}(\underset{\sim}{x}-\underset{\sim}{\mu}_k)\right)$

Define quadratic discriminant score for the ith popn, π_i, as

$$\delta_i^{QDS}(\underset{\sim}{x}) = -\frac{1}{2}\log|\Sigma_i| - \frac{1}{2}(\underset{\sim}{x}-\underset{\sim}{\mu}_i)'\Sigma_i^{-1}(\underset{\sim}{x}-\underset{\sim}{\mu}_i) + \log p_i$$

Classification Rule

Assign $\underset{\sim}{x}$ to π_k if

$$\delta_k^{QDS}(\underset{\sim}{x}) = \max_i \delta_i^{QDS}(\underset{\sim}{x})$$

Note: With a given learning sample $\mathcal{L}$ of preclassified examples, we calculate the quadratic discriminant score for each class

$$\hat{\delta}_i^{QDS}(\underset{\sim}{x}) = -\frac{1}{2}\log|S_i| - \frac{1}{2}(\underset{\sim}{x} - \bar{\underset{\sim}{x}}_i)' S_i^{-1} (\underset{\sim}{x} - \bar{\underset{\sim}{x}}_i) + \log p_i$$

S_i : sample variance-covariance matrix based on preclassified examples corresponding to class π_i in $\mathcal{L}$

$\bar{\underset{\sim}{x}}_i$: sample mean vector based on preclassified examples corresponding to class π_i in $\mathcal{L}$.

Remark: Sp case: suppose π_i is $N_p(\underset{\sim}{\mu}_i, \Sigma)$ $i = 1(1)3$ $\Sigma > 0$

$$\log p_i f_i(\underset{\sim}{x}) = \underbrace{-\frac{1}{2}|\Sigma| - \frac{1}{2}\underset{\sim}{x}'\Sigma^{-1}\underset{\sim}{x}}_{\text{same } \forall \text{ pop's}} + \underset{\sim}{\mu}_i'\Sigma^{-1}\underset{\sim}{x} - \frac{1}{2}\underset{\sim}{\mu}_i'\Sigma^{-1}\underset{\sim}{\mu}_i + \log p_i$$

Define the linear discriminant score as

$$\delta_i^{LDS}(\underset{\sim}{x}) = \underset{\sim}{\mu}_i'\Sigma^{-1}\underset{\sim}{x} - \frac{1}{2}\underset{\sim}{\mu}_i'\Sigma^{-1}\underset{\sim}{\mu}_i + \log p_i$$

Assignment rule is:

Assign $\underset{\sim}{x}$ to π_K if

$$\delta_K^{LDS}(\underset{\sim}{x}) = \max_i \delta_i^{LDS}(\underset{\sim}{x})$$

We have to use $\mathcal{L}$ to get the following estimated LDS

$$\hat{\delta}_i^{LDS}(\underset{\sim}{x}) = \bar{\underset{\sim}{x}}_i' S^{-1} \underset{\sim}{x} - \frac{1}{2} \bar{\underset{\sim}{x}}_i' S^{-1} \bar{\underset{\sim}{x}}_i + \log p_i$$

$\bar{\underset{\sim}{x}}_i$ for $i = 1(1)c$ calculated from pre-classified examples of π_i class in $\mathcal{L}$

$$\mathcal{L}\left(\sum_{i=1}^{c} n_i - c\right) S = \sum_{i=1}^{c} (n_i - 1) S_i$$

S_i for $i = 1(1)c$ calculated from pre-classified examples of π_i class in $\mathcal{L}$.

(divisor $(n_i - 1)$ sample covariance matrix)

Estimated classification rule is:

Assign $\underset{\sim}{x}$ to π_K if

$$\hat{\delta}_K^{LDS}(\underset{\sim}{x}) = \max_i \hat{\delta}_i^{LDS}(\underset{\sim}{x})$$

Note: The above rule is equivalent to

Assign $\underset{\sim}{x}$ to π_K if

$$-\frac{1}{2}(\underset{\sim}{x} - \bar{\underset{\sim}{x}}_K)' S^{-1} (\underset{\sim}{x} - \bar{\underset{\sim}{x}}_K) + \log p_K$$

i.e assign $\underset{\sim}{x}$ to pop^n closest in terms of sqd Mahalanobis distance penalised by $\log p_K$.

Note: We may also need to estimate p_i's from $\mathcal{L}$.

Nearest-neighbor classifier

It's a non-parametric classifier approach.

NN classifier uses observations in k closest to the given $\underset{\sim}{x}$, closest in the feature vector space.

Let $N_K(\underset{\sim}{x})$ denote the neighborhood of $\underset{\sim}{x}$ defined by the K closest points $\underset{\sim}{x}_i$ s in the training set.

Obtain a "majority vote rule" for the K nearest neighbors, i.e. pts within $N_K(\underset{\sim}{x})$; voting w.r.t. their class membership. The class which is the winner (having maximum counts for ~~etc~~ cases inside $N_K(\underset{\sim}{x})$) is the assigned class.

Logistic discrimination model

Logistic regression based classification model
As opposed to the previous methods (TPM, ECM), this approach does not require strong parametric assumption regarding class conditional densities ($f(\underset{\sim}{x}|\pi_K)$).

Two-class problem

Let us first consider a 2-class classification problem

Suppose the 2 classes are π_1 & π_2

Logistic discrimination model assumes that the class conditional densities of the 2 classes satisfy

$$\log\left(\frac{f(\underset{\sim}{x}|\pi_1)}{f(\underset{\sim}{x}|\pi_2)}\right) = \beta_0 + \underset{\sim}{\beta}' \underset{\sim}{x}$$

where; $\underset{\sim}{x}_{p\times 1} \in \mathcal{X}$ is the feature vector
$f(\underset{\sim}{x}|\pi_i)$s are class conditional densities
$\beta_0, \underset{\sim}{\beta}_{p\times 1}$: unknown deterministic constants

Note that

$$\log\left(\frac{f(\underset{\sim}{x}|\pi_1)}{f(\underset{\sim}{x}|\pi_2)}\right)$$ is referred to as the log-odds ratio.

Now

$$\log\left(\frac{f(\underset{\sim}{x}|\pi_1)}{f(\underset{\sim}{x}|\pi_2)}\right) = \beta_0 + \underline{\beta}' \underset{\sim}{x}$$

$$\Leftrightarrow \log\left(\frac{\frac{p(\pi_1|\underset{\sim}{x})\, f(\underset{\sim}{x})}{p(\pi_1)}}{\frac{p(\pi_2|\underset{\sim}{x})\, f(\underset{\sim}{x})}{p(\pi_2)}}\right) = \beta_0 + \underline{\beta}' \underset{\sim}{x}$$

$$\Leftrightarrow \log\left(\frac{p(\pi_2)\, p(\pi_1|\underset{\sim}{x})}{p(\pi_1)\, p(\pi_2|\underset{\sim}{x})}\right) = \beta_0 + \underline{\beta}' \underset{\sim}{x}$$

$$\Leftrightarrow \frac{p(\pi_2)}{p(\pi_1)} \cdot \frac{p(\pi_1|\underset{\sim}{x})}{p(\pi_2|\underset{\sim}{x})} = e^{\beta_0 + \underline{\beta}' \underset{\sim}{x}}$$

$$\Leftrightarrow \frac{p(\pi_1|\underset{\sim}{x})}{p(\pi_2|\underset{\sim}{x})} = \frac{p(\pi_1)}{p(\pi_2)}\, e^{\beta_0 + \underline{\beta}' \underset{\sim}{x}}$$

$$\Rightarrow \frac{1 - p(\pi_2|\underset{\sim}{x})}{p(\pi_2|\underset{\sim}{x})} = e^{\log \frac{p(\pi_1)}{p(\pi_2)} + \beta_0 + \underline{\beta}' \underset{\sim}{x}}$$

$$\left(\because p(\pi_1|\underset{\sim}{x}) = 1 - p(\pi_2|\underset{\sim}{x})\right)$$

$$\Rightarrow \frac{1 - p(\pi_2|\underset{\sim}{x})}{p(\pi_2|\underset{\sim}{x})} = e^{\beta_0^* + \underline{\beta}' \underset{\sim}{x}}$$

$$\left(\beta_0^* = \beta_0 + \log \frac{p(\pi_1)}{p(\pi_2)}\right)$$

$$\Rightarrow p(\pi_2|\underset{\sim}{x}) = \frac{1}{1 + e^{\beta_0^* + \underline{\beta}' \underset{\sim}{x}}}$$

and accordingly

$$p(\pi_1|\underset{\sim}{x}) = \frac{e^{\beta_0^* + \underline{\beta}' \underset{\sim}{x}}}{1 + e^{\beta_0^* + \underline{\beta}' \underset{\sim}{x}}}$$

Discrimination bet^{n} the 2-classes depend on the odds-ratio $p(\pi_1|\underset{\sim}{x}) / p(\pi_2|\underset{\sim}{x})$.

The class assignment rule is

Assign $\underset{\sim}{x}$ to π_1 if $\frac{p(\pi_1|\underset{\sim}{x})}{p(\pi_2|\underset{\sim}{x})} \geq 1$

& assign $\underset{\sim}{x}$ to π_2 if $\frac{p(\pi_1|\underset{\sim}{x})}{p(\pi_2|\underset{\sim}{x})} < 1$

i.e. the assignment rule is

Assign $\underset{\sim}{x}$ to π_1 if $\beta_0^* + \underset{\sim}{\beta}' \underset{\sim}{x} \geq 0$

& assign $\underset{\sim}{x}$ to π_2 if $\beta_0^* + \underset{\sim}{\beta}' \underset{\sim}{x} < 0$

Note: The logistic discrimination is thus based on the value of $\beta_0^* + \underset{\sim}{\beta}' \underset{\sim}{x}$.

Remark: Note that when we talk about classification in the above framework based on the value of "$\beta_0^* + \underset{\sim}{\beta}' \underset{\sim}{x}$", $\underset{\sim}{x}$ is the feature vector observed and we need to estimate/compute β_0^* & $\underset{\sim}{\beta}_{p\times 1}$ based on learning set. We would address this point after we look at the multiclass framework of logistic discrimination.

Logistic discrimination : multiclass

Suppose that we have C classes $\pi_1, \ldots, \pi_C$

Assume that log-likelihood ratios for any pair (s,c) $s = 1(1)C-1$ of likelihoods is linear of the form

$$\log\left(\frac{f(\underset{\sim}{x}|\pi_s)}{f(\underset{\sim}{x}|\pi_C)}\right) = \beta_{s0} + \underset{\sim}{\beta}_s' \underset{\sim}{x}$$

$s = 1(1)C-1$

i.e the $C-1$ log-odds specifies the model for discrimination

Note that

$$\log\left(\frac{f(\underset{\sim}{x}|\pi_s)}{f(\underset{\sim}{x}|\pi_C)}\right) = \beta_{s0} + \underset{\sim}{\beta}_s' \underset{\sim}{x} \quad ; \; s = 1(1)C-1$$

$$\Leftrightarrow \log\left(\frac{p(\pi_C)}{p(\pi_s)} \cdot \frac{p(\pi_s|\underset{\sim}{x})}{p(\pi_C|\underset{\sim}{x})}\right) = \beta_{s0} + \underset{\sim}{\beta}_s' \underset{\sim}{x} \; ; \; s = 1(1)C-1$$

$$\Leftrightarrow \frac{p(\pi_C)}{p(\pi_s)} \cdot \frac{p(\pi_s|\underset{\sim}{x})}{p(\pi_C|\underset{\sim}{x})} = e^{\beta_{s0} + \underset{\sim}{\beta}_s' \underset{\sim}{x}} \; ; \; s = 1(1)C-1$$

$$\Leftrightarrow \frac{p(\pi_s|\underset{\sim}{x})}{p(\pi_C|\underset{\sim}{x})} = e^{\beta_{s0}^* + \underset{\sim}{\beta}_s' \underset{\sim}{x}} \; ; \; s = 1(1)C-1$$

$$\left(\beta_{s0}^* = \beta_{s0} + \log\left(\frac{p(\pi_s)}{p(\pi_C)}\right)\right)$$

$$\Leftrightarrow \frac{1}{p(\pi_c|\underset{\sim}{x})} \sum_{s=1}^{c-1} p(\pi_s|\underset{\sim}{x}) = \sum_{s=1}^{c-1} e^{\beta^*_{s0} + \underset{\sim}{\beta}'_s \underset{\sim}{x}}$$

$$\Leftrightarrow \frac{1 - p(\pi_c|\underset{\sim}{x})}{p(\pi_c|\underset{\sim}{x})} = \sum_{s=1}^{c-1} e^{\beta^*_{s0} + \underset{\sim}{\beta}'_s \underset{\sim}{x}}$$

$$\left(\sum_{s=1}^{c} p(\pi_s|\underset{\sim}{x}) = 1 \Rightarrow \sum_{s=1}^{c-1} p(\pi_s|\underset{\sim}{x}) = 1 - p(\pi_c|\underset{\sim}{x}) \right)$$

$$\Rightarrow p(\pi_c|\underset{\sim}{x}) = \frac{1}{1 + \sum_{s=1}^{c-1} e^{\beta^*_{s0} + \underset{\sim}{\beta}'_s \underset{\sim}{x}}}$$

$$\& \quad p(\pi_s|\underset{\sim}{x}) = p(\pi_c|\underset{\sim}{x})\, e^{\beta^*_{s0} + \underset{\sim}{\beta}'_s \underset{\sim}{x}}$$

$$\text{i.e.} \quad p(\pi_s|\underset{\sim}{x}) = \frac{e^{\beta^*_{s0} + \underset{\sim}{\beta}'_s \underset{\sim}{x}}}{1 + \sum_{s=1}^{c-1} e^{\beta^*_{s0} + \underset{\sim}{\beta}'_s \underset{\sim}{x}}}.$$

$s = 1(1)c-1$

The multiclass logistic discrimination rule based on the above set of posterior of the classes is given by:

Assign $\underset{\sim}{x}$ to π_j if

$$p(\pi_j|\underset{\sim}{x}) = \max_i p(\pi_i|\underset{\sim}{x})$$

i.e Assign $\underset{\sim}{x}$ to π_j if $(j = 1(1)c-1)$

$$\begin{cases} \beta^*_{j0} + \underset{\sim}{\beta}'_j \underset{\sim}{x} = \max\limits_{(i=1(1)c-1)} \left\{ \beta^*_{i0} + \underset{\sim}{\beta}'_i \underset{\sim}{x} \right\} & \text{— (i)} \\ \text{and } \beta^*_{j0} + \underset{\sim}{\beta}'_j \underset{\sim}{x} > 0 & \text{— (ii)} \end{cases}$$

Otherwise assign to π_c

[Note that (i) comes from the fact that

$$p(\pi_s | \underset{\sim}{x}) = \frac{e^{\beta^*_{s0} + \underset{\sim}{\beta}'_s \underset{\sim}{x}}}{1 + \sum_{s=1}^{c-1} e^{\beta^*_{s0} + \underset{\sim}{\beta}'_s \underset{\sim}{x}}} \quad \forall\, s = 1(1)c-1$$

So $p(\pi_j | \underset{\sim}{x}) = \max\limits_{i=1(1)c-1} p(\pi_i | \underset{\sim}{x})$

$\Leftrightarrow$ condition (i)

Further (ii) comes from the fact that

$$p(\pi_j | \underset{\sim}{x}) > p(\pi_c | \underset{\sim}{x})$$

$j = 1(1)c-1)$

$$\Leftrightarrow \quad \frac{e^{\beta^*_{j0} + \underset{\sim}{\beta}'_j \underset{\sim}{x}}}{1 + \sum_{s=1}^{c-1} e^{\beta^*_{s0} + \underset{\sim}{\beta}'_s \underset{\sim}{x}}} > \frac{1}{1 + \sum_{s=1}^{c-1} e^{\beta^*_{s0} + \underset{\sim}{\beta}'_s \underset{\sim}{x}}}$$

$$\Leftrightarrow \quad e^{\beta^*_{j0} + \underset{\sim}{\beta}'_j \underset{\sim}{x}} > 1$$

$\Leftrightarrow$ $\beta^*_{j0} + \underset{\sim}{\beta}'_j \underset{\sim}{x} > 0$ if this does not happen then assign $\underset{\sim}{x}$ to π_c]

In other words assignment rule is:

assign $\underset{\sim}{x}$ to π_j if $(j = 1(1)c-1)$

$$\begin{cases} p(\pi_j|\underset{\sim}{x}) > p(\pi_K|\underset{\sim}{x}) & K = 1(1)c-1 \\ & K \neq j \\ \& \; p(\pi_j|\underline{x}) > p(\pi_c|\underline{x}) & \text{otherwise assign } \underset{\sim}{x} \text{ to } \pi_c \end{cases}$$

$\Leftrightarrow$ assign $\underset{\sim}{x}$ to π_j $(j = 1(1)c-1)$ if

$$\beta^*_{j0} + \underset{\sim}{\beta}'_j \underset{\sim}{x} = \max_{s=1(1)c-1} \left\{ \beta^*_{s0} + \underset{\sim}{\beta}'_s \underset{\sim}{x} \right\} \;\&$$

$\beta^*_{j0} + \underset{\sim}{\beta}'_j \underset{\sim}{x} > 0$ otherwise assign $\underset{\sim}{x}$ to π_c

—

Parameter estimation : 2 class problem

Consider a binary variable Y with values 1 and 0. "1" associated with π_1 population and "0" associated with π_2 population.

Learning sample data : $(\underset{\sim}{x}_1, y_1), (\underset{\sim}{x}_2, y_2), \ldots, (\underset{\sim}{x}_n, y_n)$

$$p(\pi_1 | \underset{\sim}{x}) = \frac{e^{\beta_0^* + \underset{\sim}{\beta}' \underset{\sim}{x}}}{1 + e^{\beta_0^* + \underset{\sim}{\beta}' \underset{\sim}{x}}} = \frac{e^{\underset{\sim}{\beta}^{*\prime} \underset{\sim}{x}^*}}{1 + e^{\underset{\sim}{\beta}^{*\prime} \underset{\sim}{x}^*}}$$

$$\underset{\sim}{x}^* = \begin{pmatrix} 1 \\ \underset{\sim}{x} \end{pmatrix} ; \quad \underset{\sim}{\beta}^* = \begin{pmatrix} \beta_0^* \\ \underset{\sim}{\beta} \end{pmatrix}$$

& $p(\pi_2 | \underset{\sim}{x}) = \frac{1}{1 + e^{\underset{\sim}{\beta}^{*\prime} \underset{\sim}{x}^*}}$.

$$f_y(y_i) = P(Y = y_i) = \left(p(\pi_1 | \underset{\sim}{x})\right)^{y_i} \left(1 - p(\pi_1 | \underset{\sim}{x})\right)^{1 - y_i}$$

y_i is 0 or 1

Likelihood fn.

$$L(\underset{\sim}{\beta}^*) = \prod_{i=1}^{n} f_{y_i}(y_i)$$

$$L(\underset{\sim}{\beta}^*) = \prod_{i=1}^{n} \left(p(\pi_1 | \underset{\sim}{x}_i)\right)^{y_i} \left(1 - p(\pi_1 | \underset{\sim}{x}_i)\right)^{1 - y_i}$$

log likelihood

$$\log L = \sum_{i=1}^{n} \left(y_i \log p(\pi_1 | \underset{\sim}{x}_i) + (1 - y_i) \log \left(1 - p(\pi_1 | \underset{\sim}{x}_i)\right) \right)$$

$$= \sum_{i=1}^{n} \left(y_i \log \left(\frac{p(\pi_1 | \underset{\sim}{x}_i)}{1 - p(\pi_1 | \underset{\sim}{x}_i)} \right) + \log \left(1 - p(\pi_1 | \underset{\sim}{x}_i)\right) \right)$$

$$\log L = \sum_{i=1}^{n} \left(y_i \underset{\sim}{\beta}^{*\prime} \underset{\sim}{x}_i^{*} - \log\left(1 + e^{\underset{\sim}{\beta}^{*\prime} \underset{\sim}{x}_i^{*}}\right) \right)$$

$$\frac{\partial \log L}{\partial \underset{\sim}{\beta}^{*}} = \sum_{i=1}^{n} \left(y_i \underset{\sim}{x}_i^{*} - \left(1 + e^{\underset{\sim}{\beta}^{*\prime} \underset{\sim}{x}_i^{*}}\right)^{-1} e^{\underset{\sim}{\beta}^{*\prime} \underset{\sim}{x}_i^{*}} \underset{\sim}{x}_i^{*} \right)$$

$$= \sum_{i=1}^{n} \underset{\sim}{x}_i^{*} \left(y_i - \frac{e^{\underset{\sim}{\beta}^{*\prime} \underset{\sim}{x}^{*}}}{1 + e^{\underset{\sim}{\beta}^{*\prime} \underset{\sim}{x}^{*}}} \right)$$

$$\frac{\partial \log L}{\partial \underset{\sim}{\beta}^{*}} = 0 \Rightarrow \sum_{i=1}^{n} \underset{\sim}{x}_i^{*} \left(y_i - \frac{e^{\underset{\sim}{\beta}^{*} \underset{\sim}{x}_i^{*}}}{1 + e^{\underset{\sim}{\beta}^{*\prime} \underset{\sim}{x}^{*}}} \right) = 0 \quad \text{— (*)}$$

(*) is a system of $p+1$ nonlinear equations which needs to be solved to obtain the MLEs. An Iteratively Reweighted Least Squares (IRLS) method is used for solving (*) to get the MLEs.

CART: Classification & Regression Tree

CART, is a powerful tree based method which can be used for regression modelling and for classification tasks.

- CART is a non-parametric approach and does not require any distributional assumption.

Before we talk about construction of such trees, let us first look at how it works.

Classification Trees

Classification tree (or a decision tree) is an example of a multistage decision process.

Rather than using the complete set of features jointly to make a output decision, different subsets of features are used at different levels of the tree

Consider the following classification tree for for a 3-class problem with 6-dimensional feature vector $\underset{\sim}{x} = (x_1, x_2, x_3, x_4, x_5, x_6)'$.

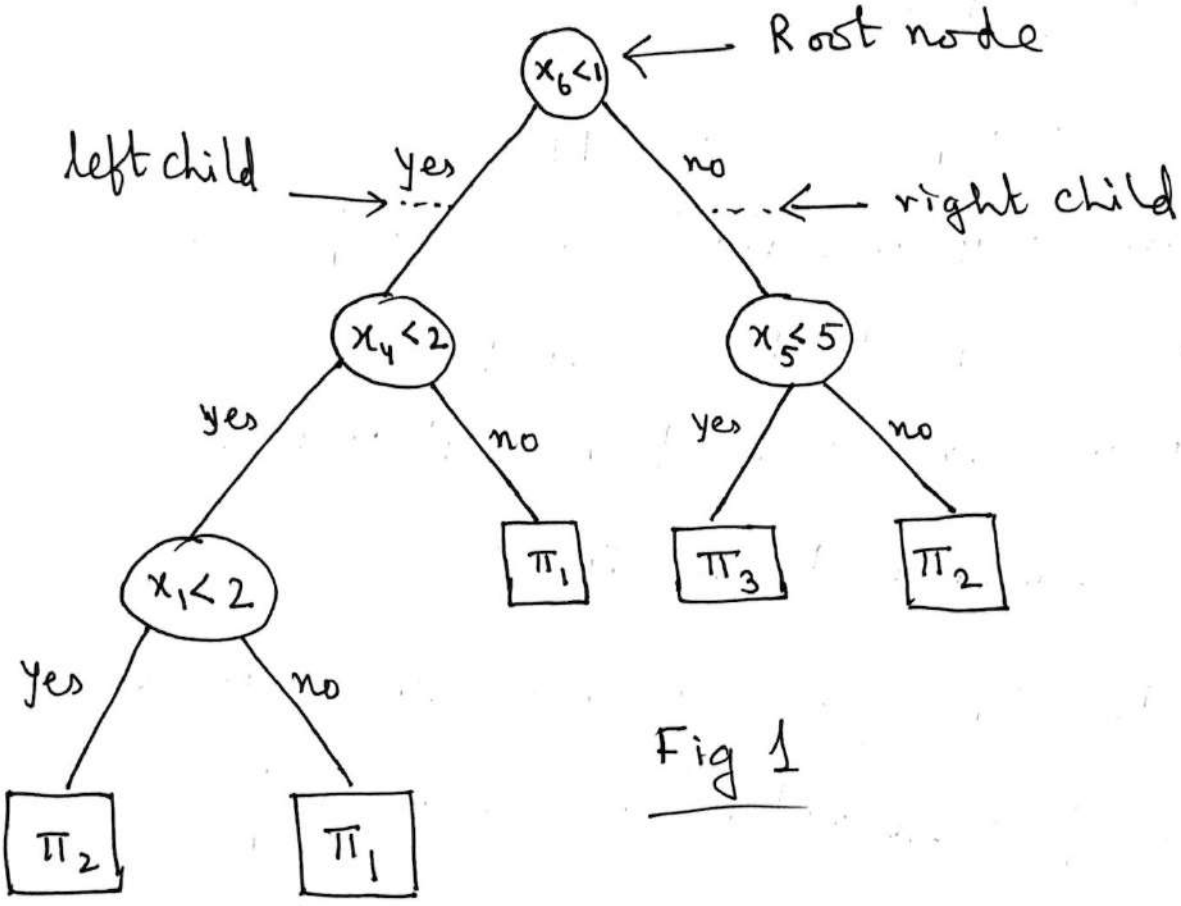


Fig 1

Explanation.

○ : internal nodes of the tree; associated with each internal node, there is a variable and a threshold

□ : these are called terminal node or leaf: there is a class label associated with each leaf

The top most internal node is the "root" of the tree

Remark: Construction of classification tree would revolve around

(i) how to choose the variable associated with any internal node (including the root node)

(ii) how to find the threshold associated with the variable at any internal node

(iii) how to assign class label at the terminal nodes.

(iv) when to declare a node ~~as~~ terminal

How to use the classification tree?

Suppose we have a feature vector

$$\underset{\sim}{x} = (5, 3, 6, 2, 1, 3)'$$ and

wish to use classification tree of Fig 1 to classify the above feature vector.

S1: We begin with the root node; compare the value of x_6 (i.e. 3) with the threshold (which is 1) and find that threshold is exceeded and

hence take the right side path (called the ~~left~~ right child of the node)

S2: We arrive at the internal node $(x_5 < 5)$

We compare the value of the x_5 variable in the feature (i.e. 1) with the threshold at that node (i.e. 5) and find that the threshold is not exceeded; hence take the left side path (the left child)

S3: We arrive at a terminal node with the last path direction. The terminal node that we have reached has a class label π_3 and hence the allotment of the given feature vector is π_3, i.e. we assign

$\underset{\sim}{x} = (5, 3, 6, 2, 1, 3)'$ to π_3

The rule looks so simple!!

Isn't it!!

Remark: The tree classifier in Fig 1 is an example of a binary decision tree.

Remark: Every internal node induces a partition of feature space. These partitions induced are hyperplanes parallel to the coordinate axes.

Remark: The classification partition is induced by the set of terminal nodes

Remark: For every internal node $t \in T$, there is a subspace $U(t) \in \mathcal{X}$. This $U(t)$ is the union of the subspaces of the terminal nodes that are it's descendants (i.e. the terminal nodes below t in the tree structure).

Example: classification partitions

2-class problem: π_1 & π_2

2-dimensional feature vector

$$\underset{\sim}{x} = (x_1, x_2)$$

Classification tree is given in Fig 2

Remark: The tree classifier in Fig 1 is an example of a "binary decision tree". More generally, the outcome of a decision at any node can also be more than 2.

Remark: Binary trees partition the feature space into 2 parts. The partitions induced (as in Fig 1) are hyperplanes parallel to the coordinate axes.

Let me try to illustrate the concept of partitions induced by such trees with help of a simple example

Example: 2-class problem: Π_1 & Π_2

2-dimensional feature vector

$$\underset{\sim}{x} = (x_1, x_2)'$$

Let the constructed classification tree be given by Fig 2

$[a_1 < a_2$
$b_1 < b_2 < b_3]$

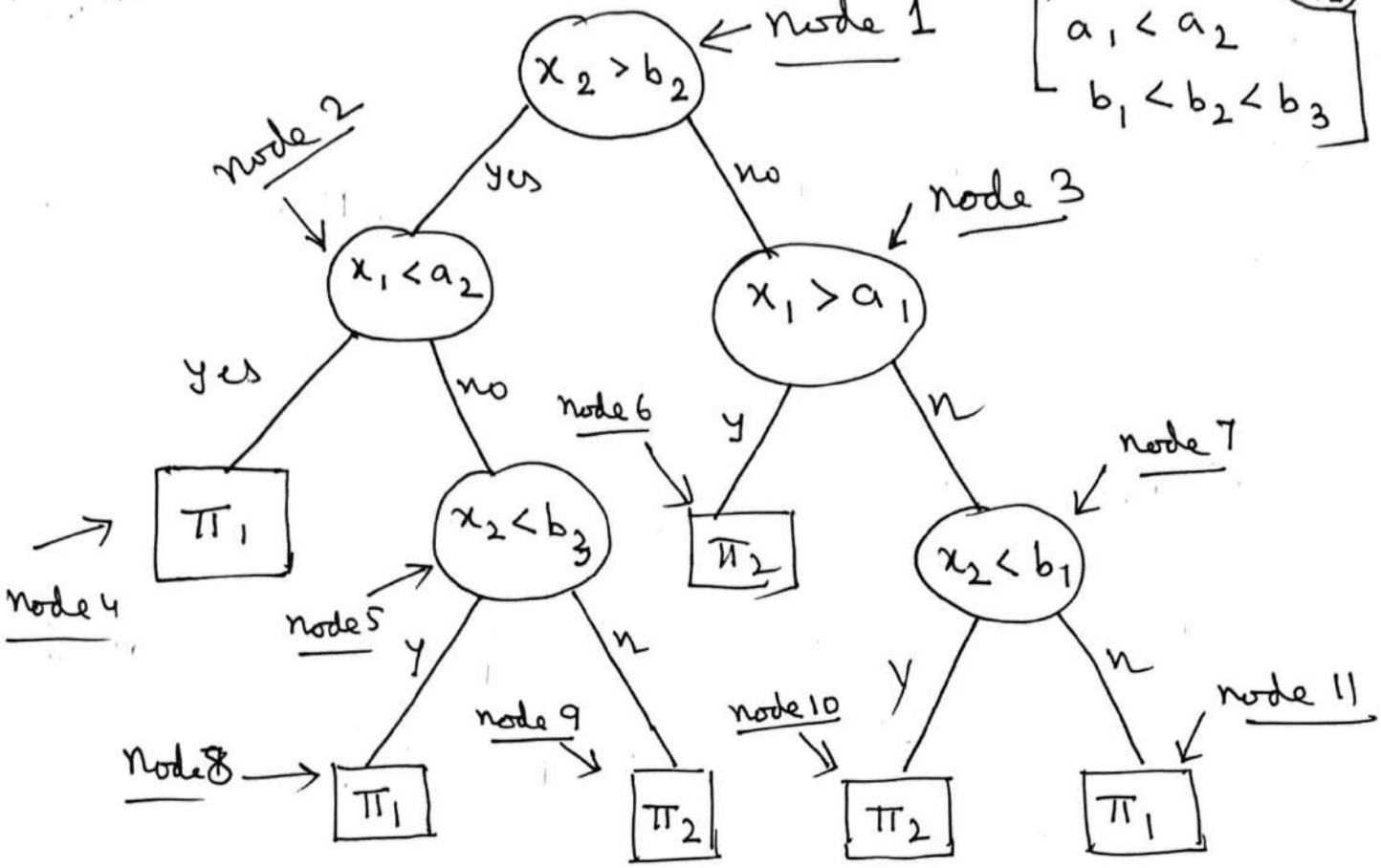


Fig 2.

To see how the above leads to partition of the feature space with class labels for partition regions follow the sequence of figures.

node 1 $\Rightarrow$

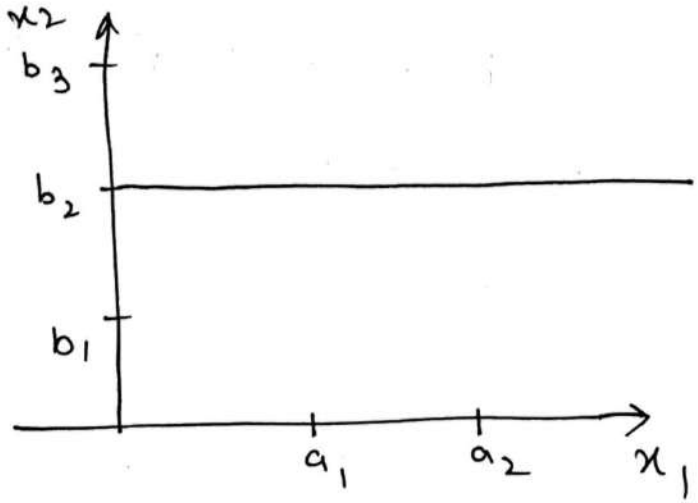


Fig 2.1

Node 2 ⇒

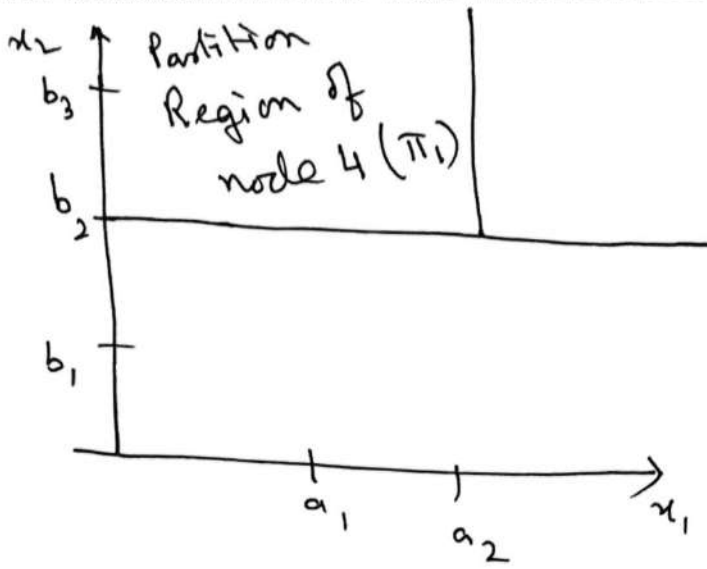


Fig 2.2

Node 5 ⇒

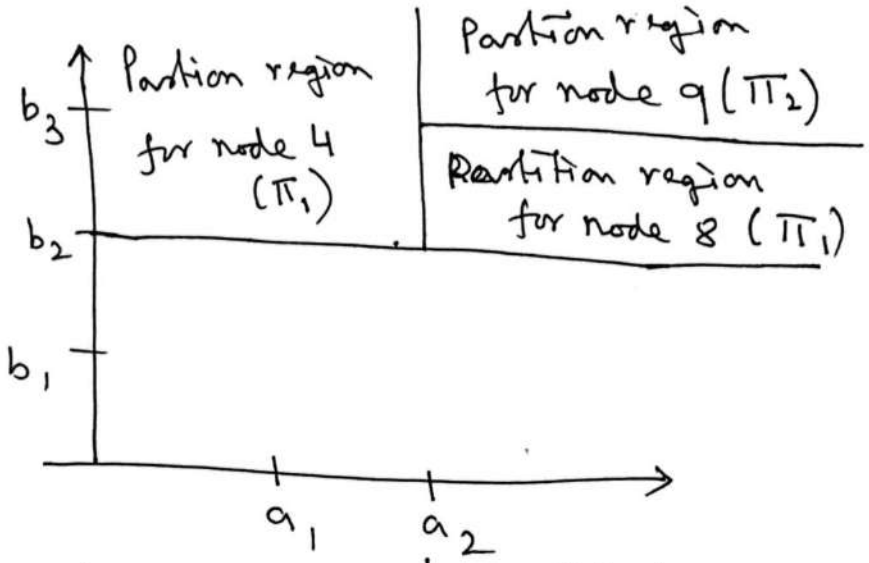


Fig 2.3

Node 3 ⇒

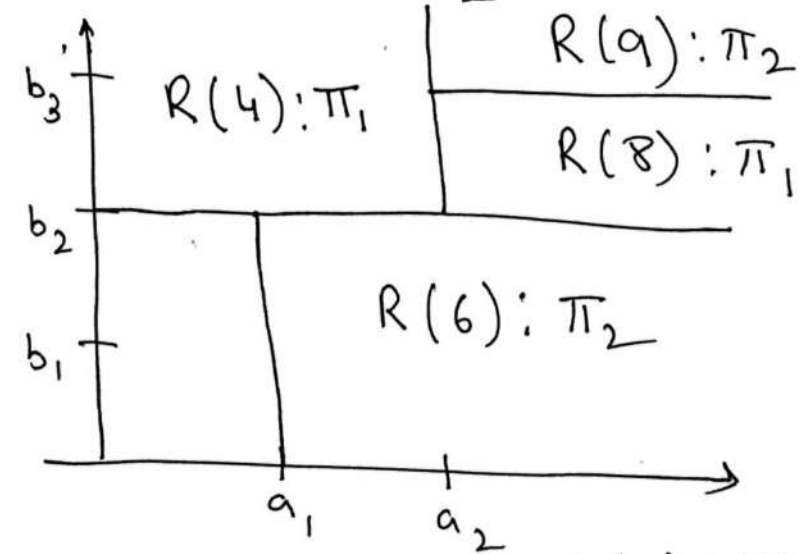


Fig 2.4

Finally, node 7 ⇒

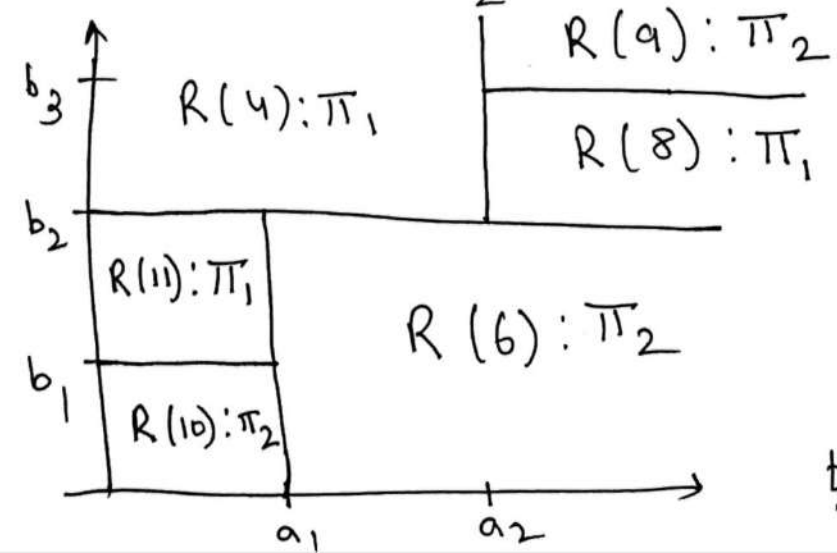
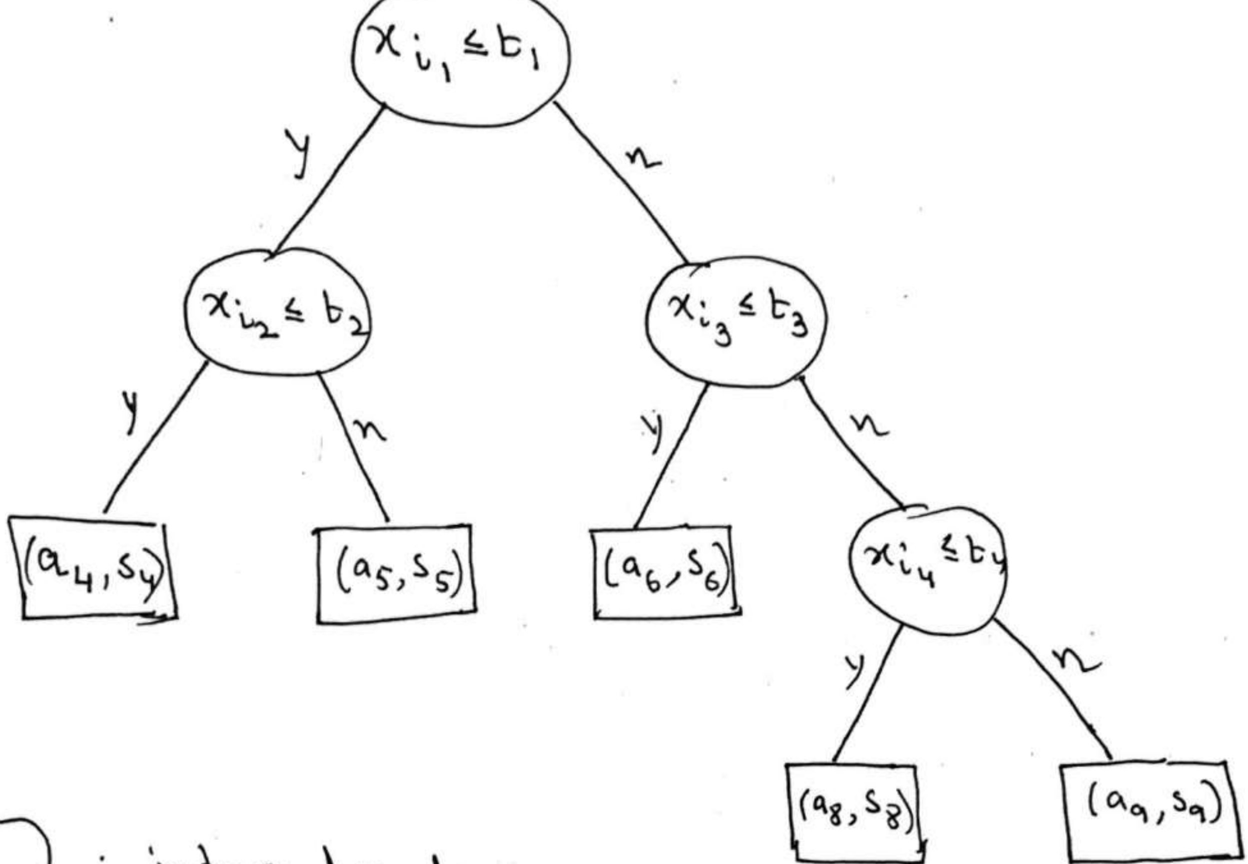


Fig 2.5

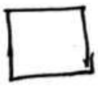
This is the final partition of the feature space that the tree classifier induces.

Regression Trees

Consider the following example of a regression tree



 : internal nodes

 : terminal nodes

i_j : j^{th} split variable

t_j : split point value

a_i : avg of all the response values of patterns in $\mathcal{L}$ reaching terminal node i.

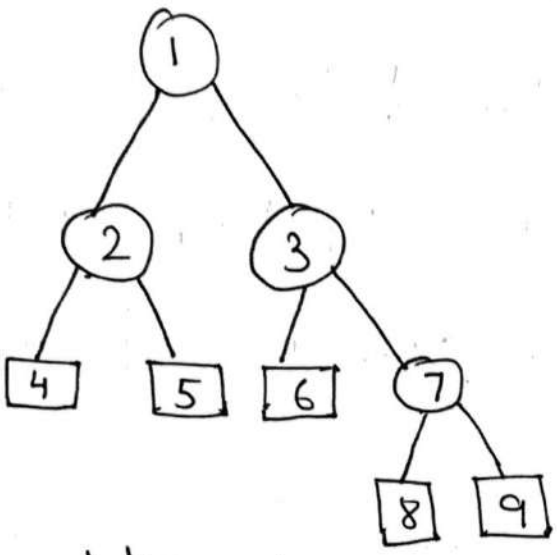
s_i : st. dev of - - - - - - - - - - - - - - - - - - i.

Remark : Regression trees look very much the same as classification trees, with split variables & split points. The terminal nodes are characterized by avg of

response variable values (in contrast to class labels as in classification trees) of patterns in the learning set $\mathcal{L}$ that reaches a particular terminal node.

Remark: As in the classification trees, regression trees also induce a partition of the feature vector space. The predicted value of the response of a feature vector reaching a particular terminal node (i.e. belonging to the corresponding partition) would typically be the average of all the responses in the learning set (under sq error loss), whose feature vectors belongs to that partition.

Remark: Suppose the following is a classification or a regression tree



T : tree defined by the nodes $\{1, 2, \ldots, 9\}$

$\tilde{T}$: set of terminal nodes $= \{4, 5, 6, 8, 9\}$

Partition induced by the tree is defined through the terminal nodes

$$U(4), U(5), U(6), U(8), U(9)$$

each $U(i) \in \mathcal{X} \leftarrow$ feature space

$$U(i) \cap U(j) = \phi$$

$$i \neq j \; ; \; i, j = 4, 5, 6, 8, 9$$

$$U(4) \cup U(5) \cup U(6) \cup U(8) \cup U(9) = \mathcal{X}$$

Towards tree construction

Some basic definitions

Tree: A tree is defined to be a set T of positive integers together with 2 functions $l(.)$ and $r(.)$ from T to $T \cup \{0\}$.

Each member of T corresponds to a node in the tree.

$l(t)$ and $r(t)$ are left node and right node values of t $(\forall\, t \in T) \ni$

(i) $\forall\, t \in T$, either $l(t)$ and $r(t)$ are both 0 (in such a situation t is terminal) or $l(t)$ and $r(t)$ are both > 0 (internal node)

(ii) apart from the root node ($t=1$) there is a unique parent $s \in T \ni \forall\, t \neq 1 \;\exists$ an $s \in T \ni t = l(s)$ or $t = r(s)$.

Subtree A subtree is a non-empty subset T_1 of T together with 2 f^n's $l_1(.)$ and $r_1(.)$ ⇒

$$l_1(t) = \begin{cases} l(t), & \text{if } l(t) \in T_1 \\ 0, & \text{o/w} \end{cases}$$

& $$r_1(t) = \begin{cases} r(t), & \text{if } r(t) \in T_1 \\ 0, & \text{o/w.} \end{cases}$$

and T_1, $l_1(.)$ and $r_1(.)$ form a tree.

Pruned subtree: A subtree that has same root as the original tree

Terminal node set:

$\tilde{T}$: set of terminal nodes.

$\tilde{T} = \{t_1, t_2, \ldots, t_M\}$; $t_i \in T$

Partition induced by a tree T

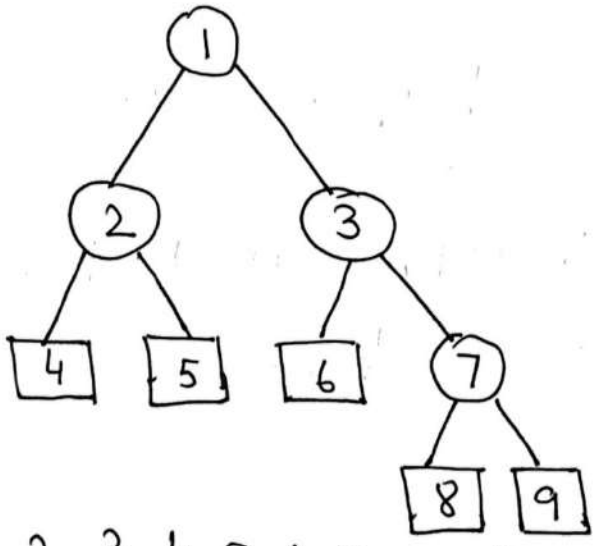
$\{U(t) : t \in \tilde{T}\}$: partition of feature space induced by T with terminal node set $\tilde{T}$

$U(t) \cap U(s) = \phi \;\forall\; t \neq s$; $t, s \in \tilde{T}$

and $\bigcup_{t \in \tilde{T}} U(t) = \mathcal{X}$

Note: Remember that associated with every non-terminal node t, there is a subspace $U(t)$ of $\mathcal{H}$, which is union of the subspaces of the terminal nodes that are its descendants.

Example:



$T = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$\tilde{T} = \{4, 5, 6, 8, 9\}$

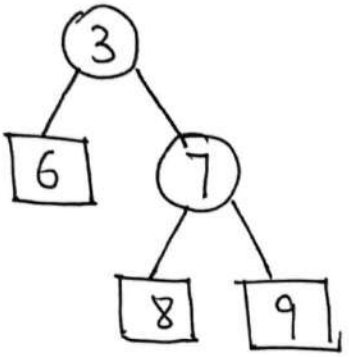
t	$l(t)$	$r(t)$
1	2	3
2	4	5
3	6	7
4	0	0
5	0	0
6	0	0

t	$l(t)$	$r(t)$
7	8	9
8	0	0
9	0	0

enough to know the tree structure

Let
$T_1 = \{3, 6, 7, 8, 9\}$

T_1

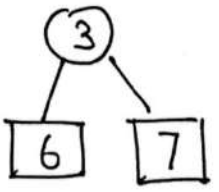


$$l_1(t) = \begin{cases} l(t), & \text{if } l(t) \in T_1 \\ 0, & \text{o/w} \end{cases}$$

$$r_1(t) = \begin{cases} r(t), & \text{if } r(t) \in T_1 \\ 0, & \text{o/w} \end{cases}$$

t	$l_1(t)$	$r_1(t)$
3	6	7
6	0	0
7	8	9
8	0	0
9	0	0

$T_2 = \{3, 6, 7\}$

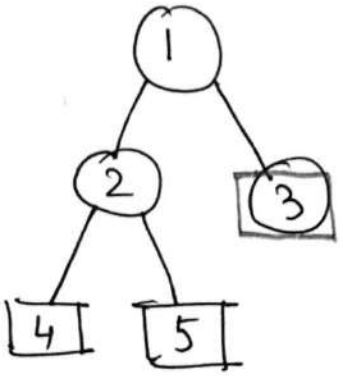


T_2

t	$l_2(t)$	$r_2(t)$
3	6	7
6	0	0
7	0	0

Also, take

$T_3 = \{1, 2, 3, 4, 5\}$



T_3

t	$l_3(t)$	$r_3(t)$
1	2	3
2	4	5
3	0	0
4	0	0
5	0	0

Remark : T_1, T_2, T_3 are all subtrees of T

T_1 is not a prunned subtree of T

T_2 is a prunned subtree of T_1 but is not a prunned subtree of T.

T_3 is a prunned subtree of T

Note : {2,3,4,5} is not a subtree, as 2 and 3 has no parent.

Classification Tree

Let us now look at various issues associated with construction of classification trees.

Let T be a tree with $\tilde{T} = \{t_1, \ldots, t_M\}$ as the set of terminal nodes.

Let $\pi_{j(t)} \in \{\pi_1, \ldots, \pi_c\}$ denote one of the class labels, that is associated with terminal node t

A classification tree consists of T, $\tilde{T}$, $\{\pi_{j(t)}, t \in \tilde{T}\}$ and a partition $\{U(t) : t \in \tilde{T}\}$.

Let the learning sample be $\mathcal{L} = \{(\underset{\sim}{x}_i, y_i) : i = 1(1)N\}$
(y_i s are the class labels)

$\forall t \in T$, define

$N(t)$ = # of sample patterns in $\mathcal{L} \ni \underset{\sim}{x}_i \in U(t)$

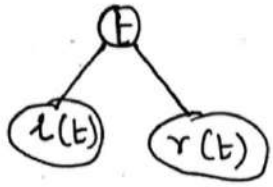
$N_j(t)$ = # - - - - - - - - $\ni \underset{\sim}{x}_i \in U(t)$ and $y_i = \pi_j$

Clearly, $\sum_j N_j(t) = N(t)$ & $\sum_{t \in \tilde{T}} N(t) = N$

$p(t) = \dfrac{N(t)}{N}$; estimate of $P(\underset{\sim}{X} \in U(t))$ based on $\mathcal{L}$

$p(\pi_j | t) = \dfrac{N_j(t)}{N(t)}$; estimate of $P(Y = \pi_j \mid \underset{\sim}{X} \in U(t))$

Recall that $\forall\, t \in T$



let $t_L = l(t)$ & $t_R = r(t)$

$p_L^t = \dfrac{p(t_L)}{p(t)}$; estimate of $P(\underset{\sim}{X} \in U(t_L) \mid \underset{\sim}{X} \in U(t))$

$p_R^t = \dfrac{p(t_R)}{p(t)}$; estimate of $P(\underset{\sim}{X} \in U(t_R) \mid \underset{\sim}{X} \in U(t))$

Rule for label assignment (class label assignment)

Assign label π_j to node t if

$$p(\pi_j \mid t) = \max_i p(\pi_i \mid t)$$

(i.e. majority voting inside the partition)

Splitting rules

Splitting rules at internal nodes are based on "node impurity" measures.

In general "node impurity" for any node $t \in T$ is defined as

$$Imp(t) = \phi\left(p(\pi_1 \mid t), p(\pi_2 \mid t), \ldots, p(\pi_c \mid t)\right)$$

Note:

Such a "node impurity" would be maximum if $p(\pi_j \mid t) = \frac{1}{c}$ $\forall j$ (i.e. equal representation of all classes at t)

and minimum if $p(\pi_j \mid t) = 1$ for some j & $p(\pi_i \mid t) = 0$ $\forall i \neq j$ } it's a pure node

Examples of "node impurity" measures

(i) Gini Index $(t) = \sum_{\substack{i,j \\ i \neq j}} p(\pi_i|t)\, p(\pi_j|t)$

(If $C = 2$ (two-class problem)

Gini Index $(t) = \sum_{i=1}^{2} \sum_{\substack{j=1 \\ j \neq i}}^{2} p(\pi_i|t)\, p(\pi_j|t)$

$= p(\pi_1|t)\, p(\pi_2|t) + p(\pi_2|t)\, p(\pi_1|t)$

$= p(1-p) + (1-p)\, p \quad \left(p(\pi_1|t) = p\right)$

$= 2p(1-p)$

→ max if $p = \frac{1}{2}$ (i.e. $\frac{1}{c}$)

min if p or $1-p = 0/1$)

(ii) Misclassificate error rate at node t

$= \frac{1}{N(t)} \sum_{i:\, \underset{\sim}{x}_i \in U(t)} I\left(y_i \neq \pi_{j(t)}\right)$

$j(t) = \arg\max_i p(\pi_i|t)$

$= \frac{1}{N(t)} \left(N(t) - N_{j(t)}(t)\right)$

$= 1 - \frac{N_{j(t)}(t)}{N(t)} = 1 - p(\pi_j|t)$

(iii) Cross-entropy or deviance

$-\sum_i p(\pi_i|t) \log p(\pi_i|t)$

Remark: The above measures are for node impurity. We can define tree impurity as

$$\text{Imp}(T) = \sum_{t \in \tilde{T}} p(t)\, \text{Imp}(t)$$

How to split a node?

This is by far the most important question!!

Consider a split using variable x_K at level l

say, $S_K^l = \{\underset{\sim}{x} : x_K < l\}$

We define a measure of "goodness of split" at node t as the change in impurity fn. For the split S_K^l at t, this is

$$\Delta \text{Imp}(S_K^l, t) = \text{Imp}(t) - \left(p_L^t \text{Imp}(t_L) + p_R^t \text{Imp}(t_R)\right)$$

(recall that $p_L^t = P(\underset{\sim}{X} \in U(t_L) \mid \underset{\sim}{X} \in U(t))$

& $p_R^t = P(\underset{\sim}{X} \in U(t_R) \mid \underset{\sim}{X} \in U(t))$)

We need to find $(K^*, l^*) \ni \Delta \text{Imp}(S_K^l, t)$ is maximized over all $K = 1(1)p$ (dimension of feature vector) and l is allowed take one of a finite # of values within the range of possible values of the chosen feature variable.

The above "goodness of split" based approach is used to split nodes starting with the root node.

When to stop splitting?

We do not split a node if the change in the impurity due to split is less than a prescribed threshold

OR

Grow the tree till the terminal nodes are all pure (all patterns belonging to the partition have same class label) and then apply pruning of the tree.

What is pruning? How to prune a classification tree?

There are various approaches of pruning. We discuss here 2 important approaches.

Pruning of a grown tree after applying splitting of nodes basically means cutting branches of the tree to get a subtree of the original tree.

Cost-complexity pruning

let $r(t) = 1 - \max_i p(\pi_i | t)$

↑ estimate of prob of misclassification at node t

define

$$R(t) = p(t)\, r(t)$$

Estimate of overall misclassification rate of the tree classifier is

$$R(T) = \sum_{t \in \tilde{T}} p(t) r(t) = \sum_{t \in \tilde{T}} R(t)$$

If α denotes the cost of complexity per terminal node then define

$R_\alpha(t) = R(t) + \alpha$ as cost-complexity criterion for node t

& $R_\alpha(T) = \sum_{t \in \tilde{T}} (R(t) + \alpha)$ (α : tuning parameter / a trade-off parameter)

$$= \sum_{t \in \tilde{T}} R(t) + \alpha |\tilde{T}|$$

where $|\tilde{T}|$: cardinality of terminal node set $\tilde{T}$

Cost-complexity pruning approach : Starting from T_0 (a pure tree, i.e. a tree having all terminal nodes as pure), find the subtree T_α (for a fixed α) $\ni R_\alpha(T)$ is minimum.

Weakest link pruning approach

Let T_t be subtree with root t

& $\{t\}$ is subbranch of T_t consisting of a single node t

Let $R_\alpha(t) = R(t) + \alpha = r(t)\, p(t) + \alpha$ (as defined earlier)

& for T_t; $R_\alpha(T_t) = R(T_t) + \alpha |\tilde{T}_t|$

Now $R_\alpha(T_t) < R_\alpha\{t\}$ (i.e. T_t has smaller cost complexity than $\{t\}$).

if $R_\alpha(T_t) + \alpha|\tilde{T}_t| < R(t) + \alpha$

i.e. if $\alpha(|\tilde{T}_t| - 1) < R(t) - R(T_t)$

i.e. if $\alpha < \dfrac{R(t) - R(T_t)}{|\tilde{T}_t| - 1}$ — (*)

Note that as $\alpha \uparrow$ equality is achieved at (*) and T_t and $\{t\}$ have same cost-complexity and $\{t\}$ is preferred to T_t; i.e. T_t can be pruned at t

We define

$$g(t) = \frac{R(t) - R(T_t)}{|\tilde{T}_t| - 1}$$

as the "strength of link" at node t

Find $t^* \ni g(t^*)$ is min and then prune the tree at this "weakest-link", i.e. at t^*.

At the next step start with the prunned subtree and find the weakest link and prune again.

Continue to reach the root through prunning to get the sequence of prunned subtrees

$$T_1 \supset T_2 \supset \cdots \supset \{t_1\}$$

↑ org root

Under this approach, the prunned subtree with min $R_\alpha(T)$ is the best prunned tree.

Growing a regression tree

How to split nodes?

$\mathcal{L} = \{(\underset{\sim}{x}_i, y_i) : i = 1(1)N\}$

Consider a split variable j and a split pt t

$R_1(j,t) = \{\underset{\sim}{x} : x_j < t\}$ & $R_2(j,t) = \{\underset{\sim}{x} : x_j \geq t\}$

Set the criterion f^n as

$$\underset{j,t}{\text{Min}} \left[\min_{c_1} \sum_{i:\underset{\sim}{x}_i \in R_1(j,t)} (y_i - c_1)^2 + \min_{c_2} \sum_{i:\underset{\sim}{x}_i \in R_2(j,t)} (y_i - c_2)^2 \right] - (*)$$

Note that for a fixed j & t

$$\min \rightarrow \hat{c}_1 = \text{average}\left(y_i \mid \underset{\sim}{x}_i \in R_1(j,t)\right);$$

$$\& \ \hat{c}_2 = \text{average}\left(y_i \mid \underset{\sim}{x}_i \in R_2(j,t)\right)$$

Find (j^*, t^*) for the optimum split at a node $\Rightarrow$

$$\sum_{i:\underset{\sim}{x}_i \in R_1(j,t)} (y_i - \hat{c}_1)^2 + \sum_{i:\underset{\sim}{x}_i \in R_2(j,t)} (y_i - \hat{c}_2)^2 \quad \text{is minimised}$$

here j varies over all $j = 1(1)p$ (feature vector dim)

& t in the grid of the j^{th} variable

Growing a regression tree

How to split nodes?

$\mathcal{L} = \{(\underset{\sim}{x}_i, y_i) : i = 1(1)N\}$

Consider a split variable j and a split pt t

$R_1(j,t) = \{\underset{\sim}{x} : x_j < t\}$ & $R_2(j,t) = \{\underset{\sim}{x} : x_j \geq t\}$

Set the criterion f^n as

$$\underset{j,t}{\text{Min}} \left[\min_{c_1} \sum_{i:\underset{\sim}{x}_i \in R_1(j,t)} (y_i - c_1)^2 + \min_{c_2} \sum_{i:\underset{\sim}{x}_i \in R_2(j,t)} (y_i - c_2)^2 \right] - (*)$$

Note that for a fixed j & t

$$\min \rightarrow \hat{c}_1 = \text{average}\left(y_i \mid \underset{\sim}{x}_i \in R_1(j,t)\right);$$

$$\& \ \hat{c}_2 = \text{average}\left(y_i \mid \underset{\sim}{x}_i \in R_2(j,t)\right)$$

Find (j^*, t^*) for the optimum split at a node $\Rightarrow$

$$\sum_{i:\underset{\sim}{x}_i \in R_1(j,t)} (y_i - \hat{c}_1)^2 + \sum_{i:\underset{\sim}{x}_i \in R_2(j,t)} (y_i - \hat{c}_2)^2 \quad \text{is minimised}$$

here j varies over all $j = 1(1)p$ (feature vector dim)

& t in the grid of the j^{th} variable

When to stop splitting?

(I) Split tree node only if the decrease in sum of squares residual due to split exceeds some pre-assigned threshold

OR

(II) Std dev of responses at terminal nodes is low (below a threshold)

OR

(III) Grow a large tree and stop splitting if the node size (# of patterns reaching that node) is reaches a low threshold level and apply pruning.

Regression tree pruning

$\forall t \in \tilde{T}$ or T define $\hat{c}_t = \frac{1}{N(t)} \sum_{\underset{\sim}{x}_i \in U(t)} y_i$

$N(t)$: # of $\underset{\sim}{x}_i \in U(t)$

Impurity measure:

Mean square error of obsn in $U(t)$ from $\hat{c}$

$$Q_t = \frac{1}{N(t)} \sum_{i : \underset{\sim}{x}_i \in U(t)} (y_i - \hat{c}_t)^2$$

Define cost complexity fn as

$$C_\alpha(T) = \sum_{t \in \tilde{T}} N(t)\, Q_t + \alpha |\tilde{T}|$$

α: tuning parameter which governs the trade-off bet' tree size and goodness of fit

$|\tilde{T}|$: cardinality of $\tilde{T}$.

Prunning can be done based on $C_\alpha(T)$. For a fixed α, let T_α be a subtree of T obtained by prunning T. We try to find

$$T_\alpha \subset T \ni T_\alpha \text{ minimizes } C_\alpha(T)$$

for a fixed α.

(Note: larger the α smaller in the opt T_α)

Weakest link prunning: Collapse internal nodes (to prune) that produces the smallest per node increase in

$$\sum_{t \in \tilde{T}} N(t)\, Q(t)$$ and continue till root node is reached.

Select the optimal from the prunned subtree sequence.

Two-class

Perceptron learning rule for classification

Training set $\mathcal{L} = ((\underset{\sim}{x}_1, i_1), \ldots (\underset{\sim}{x}_n, i_n))$

$i_j \in \{1, 2\}$. $\forall\ j = (1, \ldots) n$; $\underset{\sim}{x}_i \in \mathcal{X}$. $p \times 1$

Pop^{n}s π_1 & π_2.

A linear classifier

$$\underset{\sim}{w}' \underset{\sim}{x} + w_0 \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow \underset{\sim}{x} \in \begin{cases} \pi_1 \\ \pi_2 \end{cases}$$

$\underset{\sim}{v} = (w_0, w_1, \ldots w_p)'$ & $\underset{\sim}{z} = (1, x_1, \ldots x_p)'$.

or

$$\underset{\sim}{v}' \underset{\sim}{z} \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow \underset{\sim}{x} \in \begin{cases} \pi_1 \\ \pi_2 \end{cases}$$

Note : $\underset{\sim}{z}$ can also be $= (1, \phi_1(\underset{\sim}{x}), \ldots \phi_D(\underset{\sim}{x}))'$.

Define

$$\underset{\sim}{y}_i' = (1, \underset{\sim}{x}_i'), \text{ if } \underset{\sim}{x}_i \in \pi_1$$

& $\underset{\sim}{y}_i' = (-1, -\underset{\sim}{x}_i')$, if $\underset{\sim}{x}_i \in \pi_2$.

Ideally, we want a solution for $\underset{\sim}{v}$ ∋

$\underset{\sim}{v}' \underset{\sim}{y} > 0$ for as many samples as possible.

If ∃ a $\underset{\sim}{v}$ ∋ $\underset{\sim}{v}' \underset{\sim}{y} > 0$ ∀ patterns then the data are said to be 'linearly separable.

$\frac{14}{4}$ -3 $\frac{21}{4}$ $\frac{7}{4}$ +2

If $\underset{\sim}{v}' \underset{\sim}{y}_i < 0$ then a misclassification occur

Define the perceptron criterion function as

$$J_p(\underset{\sim}{v}) = \sum_{\underset{\sim}{y}_i \in Y} (-\underset{\sim}{v}' \underset{\sim}{y}_i)$$

$\underset{\sim}{y}_i \in Y \rightarrow$ set of all misclassified patterns

* $J_p(\underset{\sim}{v})$ is prop to the sum of distances of the misclassified samples to the decision boundary.

Objective is to find. $v \ni J_p(\underset{\sim}{v})$ is minimised.
gradient-descent procedure used to solve $\min_{\underset{\sim}{v}} J_p(\underset{\sim}{v})$

$$\frac{\partial J_p(\underset{\sim}{v})}{\partial \underset{\sim}{v}} = \sum_{\underset{\sim}{y}_i \in Y} (-\underset{\sim}{y}_i)$$

Iteration steps

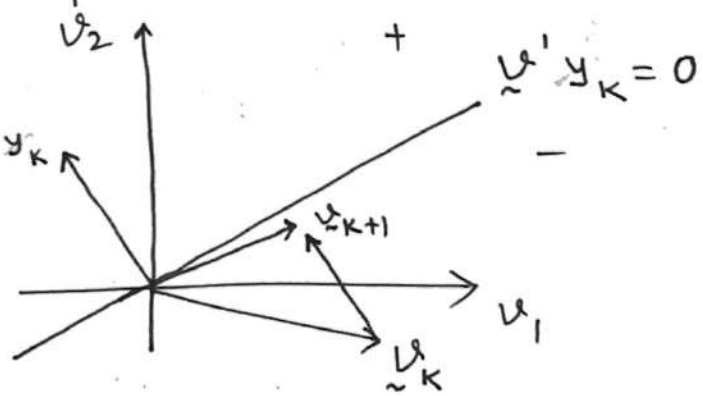
(i) $$v_{K+1} = v_K + \rho \sum_{\underset{\sim}{y}_i \in Y} \underset{\sim}{y}_i$$

batch updation mode or many pattern adaption

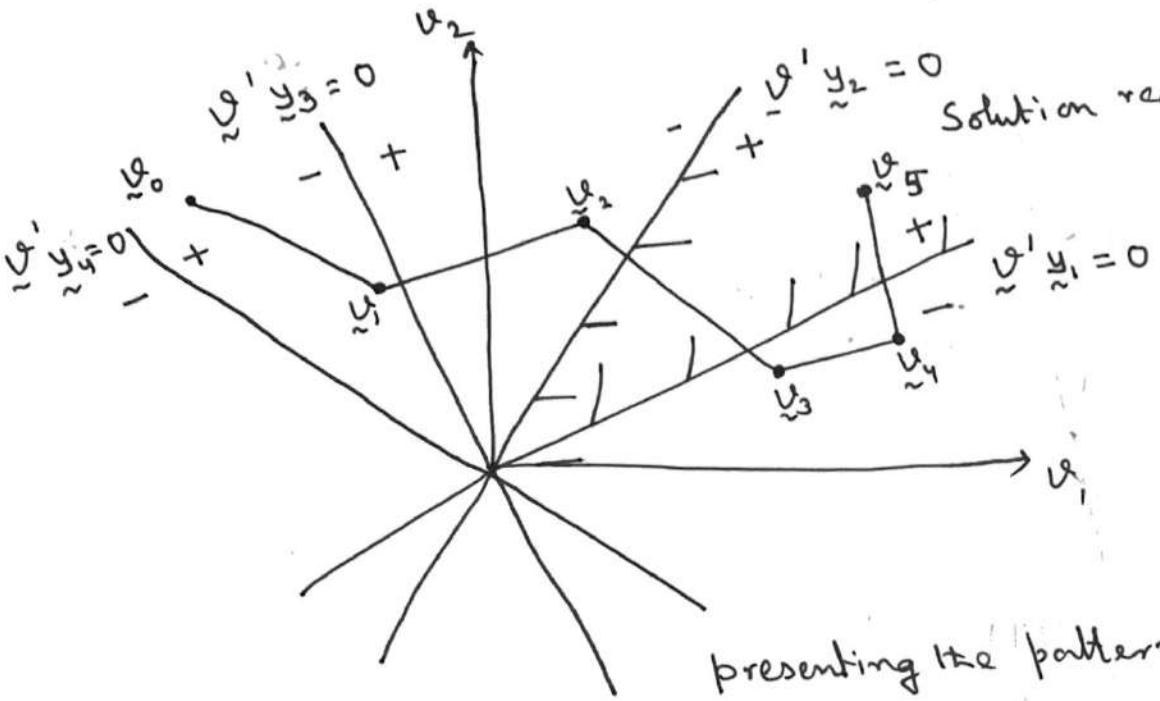
(ii) $$v_{K+1} = v_K + \rho \underset{\sim}{y}_i$$ — ρ: step size
possible to have ρ_K also

$\underset{\sim}{y}_i$ is a training pattern that is misclassified.

instantaneous updation mode or single-pattern adaption
weight updation in weight space



example with 4 patterns on single-pattern adaption mode



presenting the patterns cyclically

$\underset{\sim}{y}_1, \underset{\sim}{y}_2, \underset{\sim}{y}_3, \underset{\sim}{y}_4$

Starting with $\underset{\sim}{v}_0$, first updation for $\underset{\sim}{y}_2$ takes $\underset{\sim}{v}_0$ to $\underset{\sim}{v}_1$; next updation for $\underset{\sim}{y}_3$ ($\underset{\sim}{v}_1 \to \underset{\sim}{v}_2$); next updation for $\underset{\sim}{y}_2$ ($\underset{\sim}{v}_2 \to \underset{\sim}{v}_3$) - - - - .

A solution of $J_p(\underset{\sim}{v}) = 0$. will be obtained for linearly separable patterns.

Note: An important variation of the above perceptron learning criterion is through introduction of a 'margin' $b > 0$.

A weight vector is updated whenever

$$\underset{\sim}{v}'\underset{\sim}{y}_i \leq b$$

$\Rightarrow$ the solution vector v must lie at a distance greater than

(In term of org feature space $\Leftarrow$ $b/|\underset{\sim}{y}_i|$ from hyperplane $\underset{\sim}{v}'\underset{\sim}{y}_i = 0$

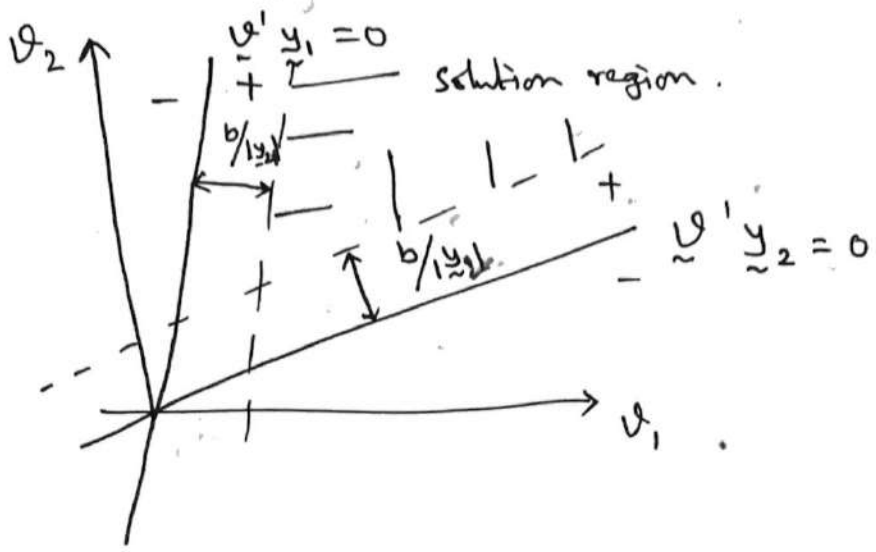
$b/|w|$

Aim of the margin is to improve generalization

without the margin some points may lie too close to the separating boundary ie the separating hyperplane.

Maximal margin classifier $\to$ Support Vector Machine (SVM) classifier.

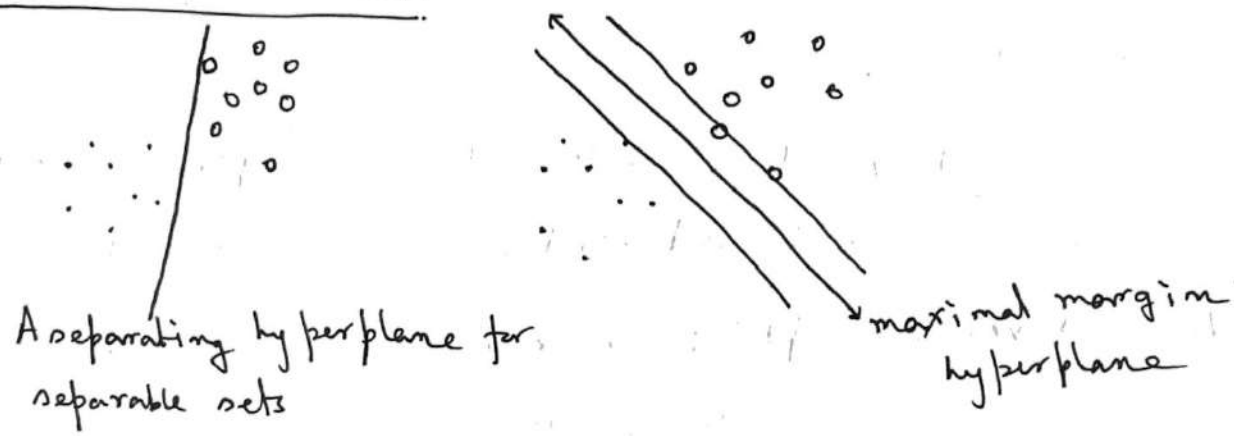
Solution region with a margin



SVM

Construction of 'best' separating hyperplane, the maximal margin hyperplane

Basic SVM model.



Suppose the data is linearly separable

Two classes π_1 & π_2 → labels $y_i = \pm 1$

$$\mathcal{L} = ((\underset{\sim}{x}_1, y_1), \ldots (\underset{\sim}{x}_n, y_n))$$

linear discriminant fn

$$g(\underset{\sim}{x}) = \underset{\sim}{w}' \underset{\sim}{x} + w_0$$

Decision rule

$$\underset{\sim}{w}'\underset{\sim}{x} + w_0 \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow \underset{\sim}{x} \in \begin{cases} \pi_1 \text{ with } y_i = +1 \\ \pi_2 \text{ with } y_i = -1 \end{cases}$$

$\Rightarrow$ Training pts are correctly classified if

$$y_i (\underset{\sim}{w}'\underset{\sim}{x}_i + w_0) > 0 \quad \forall i$$

Let $b > 0$ be a margin $\ni$ we seek a solution $\ni$

$$y_i (\underset{\sim}{w}'\underset{\sim}{x}_i + w_0) \geq b$$

Perceptron learning algorithm yeilds a solution for which all points, $\underset{\sim}{x}_i$ are at a distance greater than or equal to $b/|\underset{\sim}{w}|$ from the separating hyperplane

Note: A scaling of b, w_0 and $\underset{\sim}{w}$ leaves $b/|\underset{\sim}{w}|$ unaltered and

$$y_i (\underset{\sim}{w}'\underset{\sim}{x}_i + w_0) \geq b \text{ is still satisfied}$$

w.l.o.g. we take $b = 1 \rightarrow$ canonical hyperplanes

c H :

$$H_1 : \underset{\sim}{w}'\underset{\sim}{x} + w_0 = +1$$

$$H_2 : \underset{\sim}{w}'\underset{\sim}{x} + w_0 = -1$$

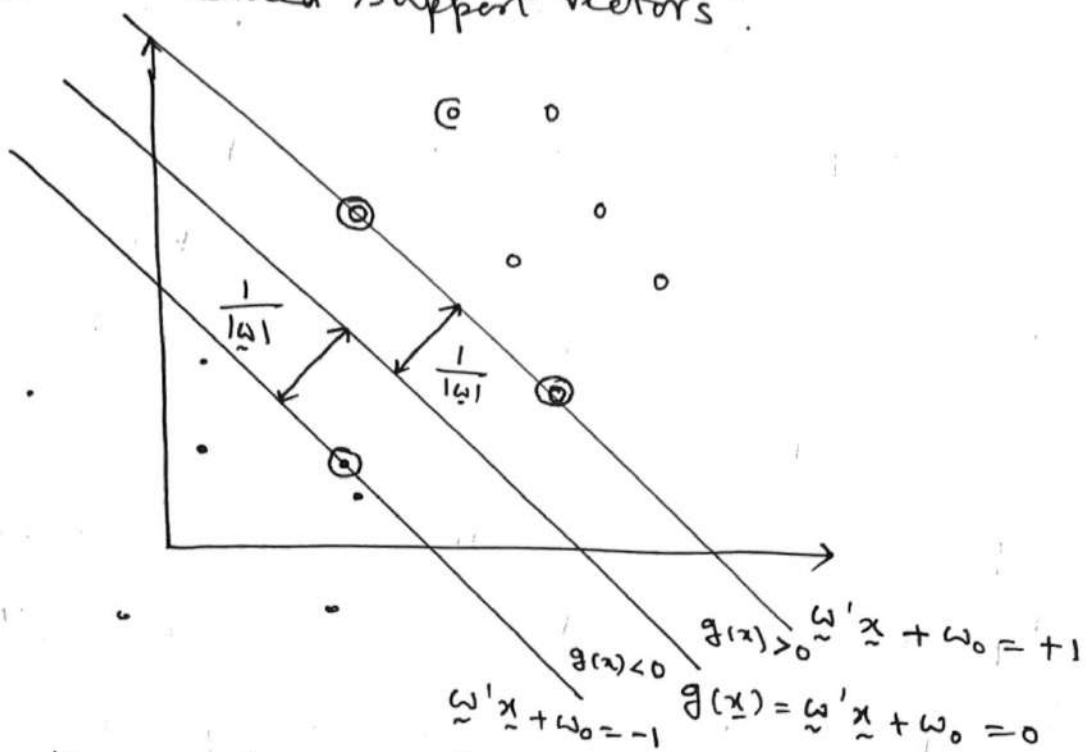
and

$$\underset{\sim}{w}'\underset{\sim}{x}_i + w_0 \geq +1 \quad \text{for } y_i = +1$$

$$\underset{\sim}{w}'\underset{\sim}{x}_i + w_0 \leq -1 \quad \text{for } y_i = -1$$

The distance of H_1 & H_2 from the separating hyperplane $g(\underset{\sim}{x}) = \underset{\sim}{w}'\underset{\sim}{x} + w_0 = 0$ is $1/|\underset{\sim}{w}|$ and is termed the margin bet^n canonical H_1/H_2 hyperplanes and the sep hyperplane

The points that lie on the canonical separating hyperplanes H_1 and H_2 are called 'support vectors'.



Objective

Maximizing the margin subject to the constraint that the separating hyperplane separates the linearly separable groups

i.e. Max $\frac{1}{|\underset{\sim}{w}|}$ i.e. Min $|\underset{\sim}{w}|$ $\ni$ the constraint

$$C : y_i (\underset{\sim}{w}' \underset{\sim}{x}_i + w_0) \geq 1 \quad \forall\ i = 1(1)n$$

$$L_p = \frac{1}{2} \underset{\sim}{w}'\underset{\sim}{w} - \sum_{i=1}^{n} \alpha_i \left(y_i (\underset{\sim}{w}' \underset{\sim}{x}_i + w_0) - 1 \right).$$

where $\alpha_i \geq 0$; $i = 1(1)n$ are Lagrange multiplier.

$(p+1)$ Primal parameter $w_0, w_1, \ldots w_p$.

Note (i) Solution obtained using quadratic programming

(ii) Case of linearly non-separable data solved using appropriate slack variables

(iii) Concepts can be extended for multiclass problem

ANN

- Computational algorithm inspired by the parallelism of human brain
- Tries to mimic the cognitive ability of the human brain to identify patterns and to make decisions and forecasts based on past experience

How NN works?

- Non-linear models that can be trained to map the past and the future values of a time series
- extract hidden structure and relationship governing the data
- powerful pattern recognition capabilities

NN in the context of statistical methods

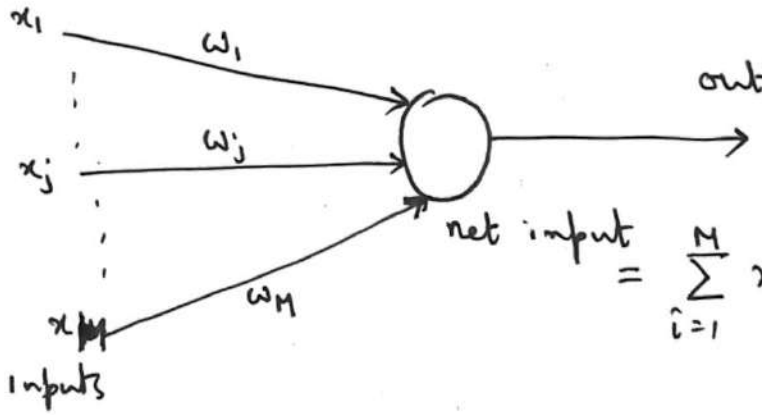
- Multivariate, non-linear, non-parametric inference technique
- completely data-driven and assumption free
- ability to model non-linear dynamics.

Some application areas

Stock market prediction, foreign exchange trading, bankruptcy prediction, volatility prediction, economic indicator forecasting credit rating, prediction of international airline passenger traffic, nowcasting in aviation industry, ozone level prediction

Human brain: Hundred billion neurons - interconnected structure

Basic computational unit of ANN - a neuron - an artificial neuron



x_j : input value from the j^{th} input node

w_j : weight on the connection from the j^{th} input node to the PU.

PU - collects the inputs from all the inputs to get the net input value

$\sum_{i=1}^{M} w_i x_i$; w_i denoting the relative importance of i^{th} input

PU processes the net input value and generates the output

$$Out = f\left(\sum_{i=1}^{M} w_i x_i\right)$$

↑ transfer function / squashing f^n.

The simplest NN

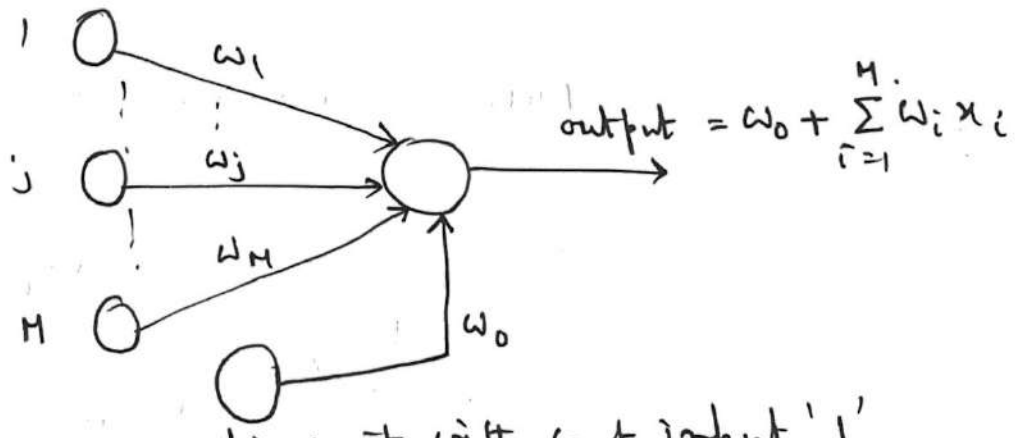
A single PU network.

MLR setup $\mathcal{L} = \{(\underset{\sim}{x}_1, y_1), \ldots (\underset{\sim}{x}_n, y_n)\}$.

Consider a transfer f^n as identity

$$\text{output} = f\left(\sum_{i=1}^{M} w_i x_i\right) = \sum_{i=1}^{M} w_i x_i$$

MLR model without a constant term



Unknown parameters : weight connections $(w_1, \ldots w_n)$

or $(w_0, w_1, \ldots w_n)$

optimum values of $(w_0, w_1, \ldots w_M)$ can be obtained by minimizing residual SSq.

Note: Injecting non-linearity into the model — consider a non-linear transfer f^n.

e.g (i) A sigmoidal transfer f^n.

$$f(x) = \frac{1}{1+e^{-x}}$$

(ii) A tanh transfer f^n

$$f(x) = \tanh(x).$$

For (i), $$\text{output} = \frac{1}{1+e^{-(w_0 + \sum_{i=1}^{M} w_i x_i)}} = g(\underset{\sim}{x})$$

A non-linear model of response in terms of the values of the feature vector components

Let $(\underset{\sim}{x}_p, y_p)$ denote the p^{th} pattern in $\mathcal{L}$.

Network output: $O_p = f\left(w_0 + \sum_{j=1}^{M} x_{pj} w_j\right) \Big/. f\left(\sum_{j=1}^{M} x_{pj} w_j\right)$.

Actual output: y_p =

Error for the p^{th} pattern : $y_p - O_p$

Error square : $\mathcal{E}_p = \frac{1}{2}(y_p - O_p)^2$ ← instantaneous error for p^{th} pattern

Total error over $\mathcal{L}$ $$\mathcal{E} = \frac{1}{2}\sum_{p=1}^{n} \mathcal{E}_p = \frac{1}{2}\sum_{p=1}^{n} (y_p - O_p)^2.$$

Training of the network : determination of optimum weight vector.

Modes of learning : Instantaneous mode or single pattern adaptation mode

or Batch mode or multiple pattern adaptation mode.

Method for obtaining the weight updation : Gradient descent algorithm

Instantaneous mode :

$$\mathcal{E}_p = \frac{1}{2}(y_p - o_p)^2.$$

$$\mathcal{E}_p = \frac{1}{2}\left(y_p - f\left(\sum_{j=1}^{M} w_j x_{pj}\right)\right)^2$$

Gradient of the error $\mathcal{E}_p$ w.r.t. the weights

$$\frac{\partial \mathcal{E}_p}{\partial w_j} = -(y_p - o_p) \cdot \frac{\partial f(\Sigma w_j x_{pj})}{\partial w_j}$$

$$= -(y_p - o_p) \frac{\partial f(\Sigma w_j x_{pj})}{\partial (\Sigma w_j x_{pj})} \cdot \frac{\partial (\Sigma w_j x_{pj})}{\partial w_j}$$

$$= -(y_p - o_p) f'(\Sigma w_j x_{pj}) x_{pj}.$$

e.g. for sigmoidal transfer fn.

$$f(\Sigma w_j x_{pj}) = \frac{1}{1 + e^{-\Sigma w_j x_{pj}}} = o_p.$$

$$\left[f(x) = \frac{1}{1+e^{-x}} \; ; f'(x) = \frac{+e^{-x}}{(\quad)^2} = \frac{1}{1+e^{-x}}\left(1 - \frac{1}{1+e^{-x}}\right) \right]$$

$$f'(\Sigma w_j x_{pj}) = o_p(1 - o_p).$$

$$\Rightarrow \frac{\partial \mathcal{E}_p}{\partial w_j} = -(y_p - o_p) o_p (1 - o_p) x_{pj}$$

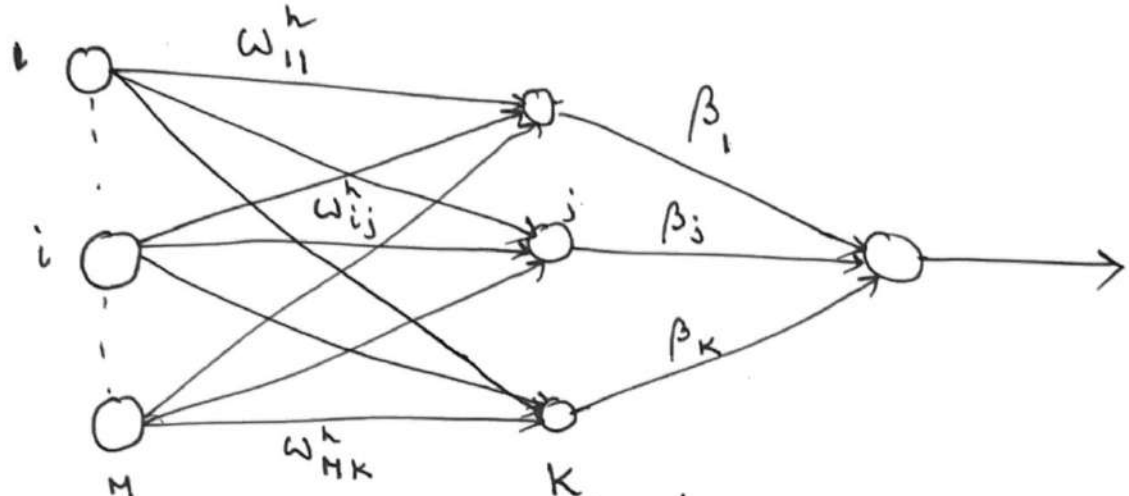
Gradient descent algorithm weight updation

$$w_j^{(k+1)} = w_j^{(k)} + \eta \left(- \frac{\partial \varepsilon_p}{\partial w_j} \right)\Big|_{w_j^{(k)}}$$

$$w_j^{(k+1)} = w_j^{(k)} + \eta \, (y_p - o_p) \, o_p (1 - o_p) \, x_{pj}$$

Present the patterns at random and use the above equation for weight updation until stopping criterion is reached.

Multi-layer Perceptron (MLP)



M input, K hidden layer neurons and one output

M-K-1 MLP.

p^{th} pattern $\rightarrow$ $(\underset{\sim}{x}_p, y_p)$

$$net_{pj}^h = \sum_{i=1}^{M} w_{ij}^h \, x_{pi} \quad \left(\begin{array}{c} \text{may have a} \\ \text{bias unit} \end{array} \right).$$

net input value at j^{th} neuron of hidden layer

output from the j^{th} neuron

$$f(net_{pj}^h) = i_{pj}$$

i_{pj} : j^{th} input to the output unit

net input value at output

$$\sum_{j=1}^{K} \beta_j \, i_{pj} = net_p^o$$

Final network output

$$O_p = f(net_p^o) \quad \left(\begin{array}{l}\text{Also 'f' can be diff}\\ \text{from the f used at}\\ \text{hidden layer neurons}\end{array}\right).$$

i.e. $O_p = f(net_p^o)$

$$= f\left(\sum_{j=1}^{K} \beta_j \, i_{pj}\right)$$

$$= f\left(\sum_{j=1}^{K} \beta_j \, f\left(\sum_{i=1}^{M} w_{ij}^h \, x_{pi}\right)\right).$$

or $$= g\left(\sum_{j=1}^{K} \beta_j \, f\left(\sum_{i=1}^{M} w_{ij}^h \, x_{pi}\right)\right).$$

Note: We can have different 'f' at different hidden layer neurons, in that case

$$O_p = g\left(\sum_{j=1}^{K} \beta_j \, f_j\left(\sum_{i=1}^{M} w_{ij}^h \, x_{pi}\right)\right).$$

Note: Suppose we consider sigmoid transfer fn at all neurons (hidden and output) then

$$O_p = f\left(\sum_{j=1}^{K} \beta_j \left\{1 + \exp\left(-\sum_{i=1}^{M} w_{ij}^h \, x_{pi}\right)\right\}^{-1}\right)$$

$$= \left\{1 + \exp\left(-\sum_{j=1}^{K} \beta_j \left[1 + \exp\left(-\sum_{i=1}^{M} w_{ij}^h \, x_{pi}\right)\right]^{-1}\right)\right\}^{-1}$$

i.e. $O_p = f^*(x_{p1}, \ldots x_{pM}) \rightarrow$ a non-linear model.

Note: The network above is called a feed-forward network.

unknown parameters for an M-K-1 network : weight connections.

input layer → hidden layer : M × K weight matrix

$$\begin{pmatrix} w^h_{11} & w^h_{12} & \cdots & w^h_{1K} \\ - & - & - & - \\ w^h_{M1} & w^h_{M2} & \cdots & w^h_{MK} \end{pmatrix}.$$

hidden layer → output node : K-dimensional weight vector

$$(\beta_j, \ldots \beta_K)'.$$

Total # of unknown weights : $(MK + K)$.

Learning of the network : determination of weights.

Either instantenous mode OR batch mode. Using

Back-propagation algorithm (b-p of errors).

Consider for example learning on an instantenous mode.

$$\mathcal{E}_p = \frac{1}{2}(y_p - o_p)^2 \leftarrow \text{instantenous sq error for } p^{th} \text{ pattern}$$

$$\mathcal{E}_p = \frac{1}{2}\left(y_p - f^o(net^o_p)\right)^2 \quad \Big| \; o_p = f^o\left(\sum_j \beta_j f^o_j(net^h_{pj})\right)$$

Gradient of $\mathcal{E}_p$ w.r.t. ~~output~~ hidden to output layer wts.

$$\frac{\partial \mathcal{E}_p}{\partial \beta^o_j} = -\frac{1}{2}(y_p - o_p)\frac{\partial f^o(net^o_p)}{\partial \beta^o_j}$$

$$= -\frac{1}{2}(y_p - o_p)\frac{\partial f^o(net^o_p)}{\partial net^o_p} \cdot \frac{\partial net^o_p}{\partial \beta^o_j}.$$

$net^o_p = \sum_{j=1}^{K} \beta^o_j i_{pj}$

$i_{pj} = f_j\left(\sum_{i=1}^{M} x_{pi} w^h_{ij}\right)$

$$= -\frac{1}{2}(y_p - o_p) f^{o\prime}(net^o_p)\, i_{pj}.$$

i.e. $$\frac{\partial \mathcal{E}_p}{\partial \beta^o_j} = -\frac{1}{2}(y_p - o_p)\, o_p(1 - o_p)\, i_{pj}$$

for a sigmoid transfer fr at the output node

Sly gradient of E_p w.r.t. input layer → hidden layer wts

$$\frac{\partial E_p}{\partial w_{ij}^h} = \frac{\partial}{\partial w_{ij}^h} \frac{1}{2}\left(y_p - f^o\left(\sum_{j=1}^{k} \beta_j f_j\left(\sum_{i=1}^{M} w_{ij}^h x_{pi}\right)\right)\right)^2$$

$o_p = f^o(net_p^o)$

$net_p^o = \sum_{j=1}^{k} \beta_j^o i_{pj}$

$i_{pj} = f_j\left(\sum_{i=1}^{M} w_{ij}^h x_{pi}\right)$

$i_{pj} = f_j(net_{pj}^h)$

$net_{pj}^h = \sum_{i=1}^{M} w_{ij}^h x_{pi}$

$$= -(y_p - o_p)\frac{\partial f^o(net_p^o)}{\partial net_p^o} \cdot \frac{\partial net_p^o}{\partial i_{pj}} \cdot \frac{\partial i_{pj}}{\partial net_{pj}^h} \cdot \frac{\partial net_{pj}^h}{\partial w_{ij}^h}$$

$$= \underbrace{-(y_p - o_p) f^{o\prime}(net_p^o)}_{\delta_p^o} \beta_j^o f_j'(net_{pj}^h) x_{pi}$$

Define, output layer delta as.

$$\delta_p^o = (y_p - o_p) f^{o\prime}(net_p^o)$$

and hidden layer delta as

$$\delta_{pj}^h = \delta_p^o \beta_j^o f_j'(net_{pj}^h).$$

Weight updation eq^n s

output layer weights

$$\beta_j^o(k+1) = \beta_j^o(k) + \eta \delta_p^o i_{pj}\Big|_{\beta^{(k)}, W^{(k)}}$$

hidden layer weights

$$w_{ij}^h(k+1) = w_{ij}^h(k) + \eta \delta_{pj}^h x_{pi}\Big|_{\beta^{(k+1)}, W^{(k)}}.$$

η: learning rate parameter

Iteration continues till stopping criterion is reached

Either loss error below threshold OR change in wt vector is lower than a threshold.

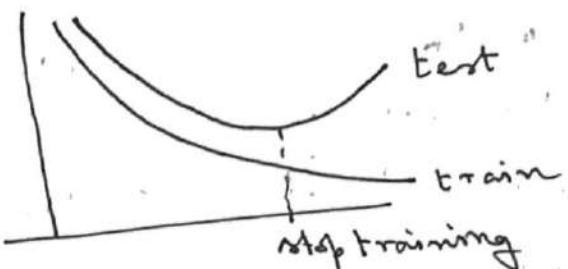
· Note: Information flows from input level to output level → error calculated on output layer → propagated back for adjustment of hidden to output layer wts → with updated hidden to output layer wts error propagates backwards for weight updation of input to hidden layer weights.
→ Back-propagation of error.

Some important points: (after the discussion on variants)

(i) Data splitting – training set and test set

(ii) Preprocessing – normalisation of inputs and output → $\frac{* - \bar{x}_i}{Sd_{x_i}}$; $\frac{* - min}{max - min}$
Along channel / across channel

(iii) Overfitting – • MSE on training & test set



• weight decay – add a penalty term for large weights

$$R(\theta) + \lambda J(\theta)$$

$$R(\theta) = \sum_{p=1}^{n} (y_p - o_p)^2 \qquad \theta: \text{set of weights } (MK + K)$$

$$J(\theta) = \sum_{i,j} w_{ij}^{h\,2} + \sum_{K} \beta_K^{o\,2}$$

λ: tuning parameter, $\lambda \geq 0$
larger λ tend to shrink weights towards 0
(approach similar to Ridge regression

• weight elimination

$$J(\theta) = \sum_{i,j} \frac{w_{ij}^{h\,2}}{1 + w_{ij}^{h\,2}} + \sum_{K} \frac{\beta_K^{o\,2}}{1 + \beta_K^{o\,2}} .$$

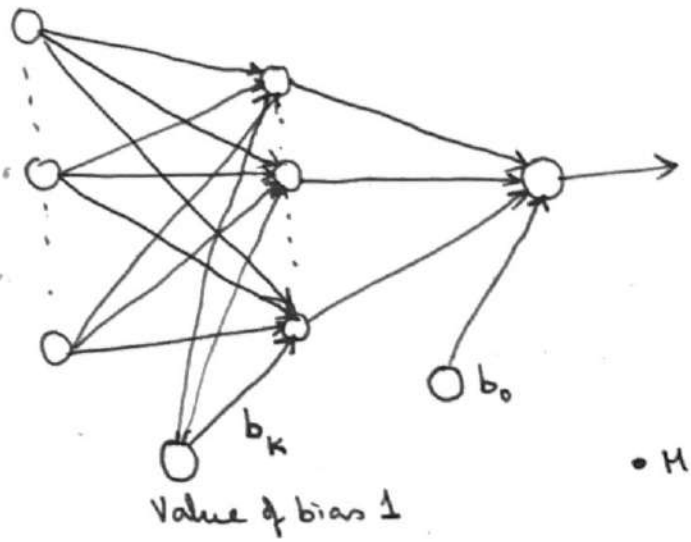
(iv) Number of hidden layers & units

One approach is to resort to experimentation

Systematic approach < simulated annealing
Genetic algorithm

Variants of the feedforward MLP

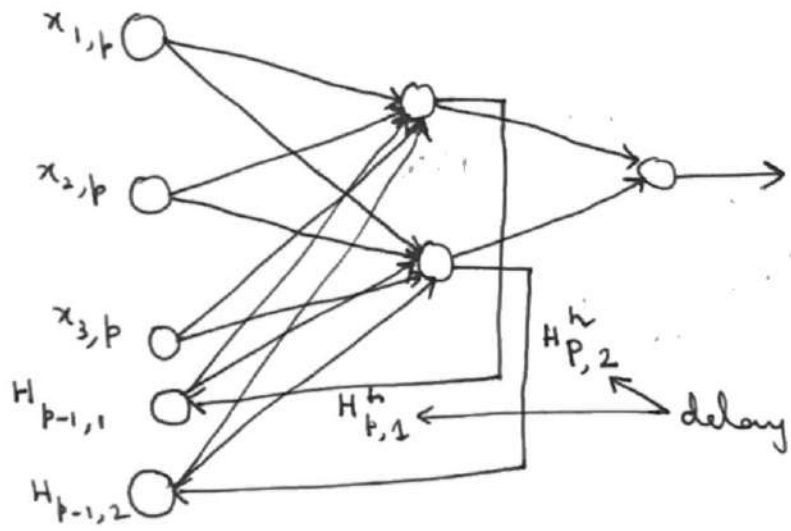
- MLP with bias units



- Multiple output MLP.

- ~~MLP~~ Recurrent MLP (more appropriate for time series set up)

3-2-1

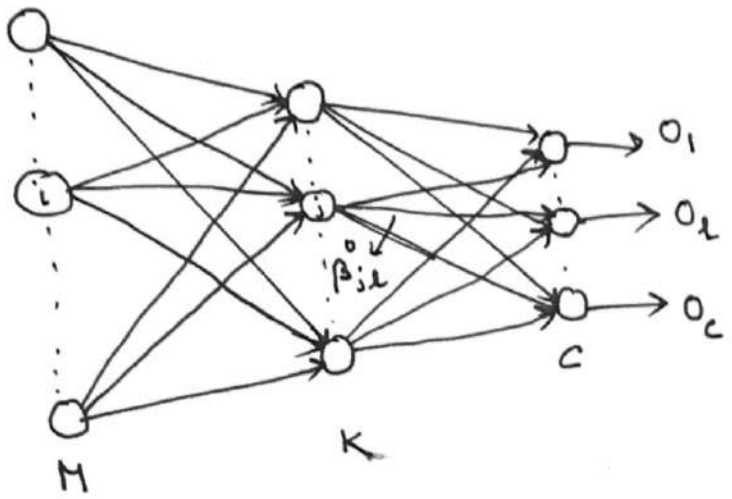


$$\hat{y}_p = \phi\left(\sum_{k=1}^{2} \beta_k^0 \, \psi_k^h \left(\sum_{i=1}^{3} x_{pi} \, w_{ij}^h + \sum_{k=1}^{2} w_k^f \, \psi_k^h \left(\sum_{i=1}^{3} x_{p-1,i} \, w_{ij}^h \right) \right) \right)$$

feedback input reaching hidden units.

ANN classification model

c - class problem



$\beta^0_{1l}\, i_{p1} + \cdots \beta^0_{jl}\, i_{pj} + \cdots$

$$net^h_{pj} = \sum_{i=1}^{M} x_{pi}\, w^h_{ij}$$

$$i_{pj} = f^h_j\left(net^h_{pj}\right) = f^h_j\left(\sum_{i=1}^{M} x_{pi}\, w^h_{ij}\right)$$

At l^{th} output unit

inputs are $i_{p1}, i_{p2}, \ldots i_{pK}$

Net input at l^{th} output $\sum_{j=1}^{K} \beta^0_{jl}\, i_{pj} = T_l \quad l = 1(1)c$

output of l^{th} output unit

$$O_l = \frac{e^{T_l}}{\sum_{s=1}^{c} e^{T_s}} \quad ; \quad l = 1(1)c$$

$$\sum_{l=1}^{c} O_l = 1 \quad ; \quad 0 \le O_l \le 1$$

Corresponding classifier is

$$G(\underset{\sim}{x}) = \underset{l}{\operatorname{argmax}}\; O_l$$

(*) Discuss how training set is to be constructed. This will specify what is y_{iK}.

Total error : $\sum_{i=1}^{n} \sum_{K=1}^{c} \left(y_{iK} - O_K\right)^2$

or deviance $-\sum_{i=1}^{n} \sum_{k=1}^{c} y_{iK} \log O_K(\underset{\sim}{x}_i)$

Steps for ANN model building

(1) Creation of Data records

(2) Splitting of data into training & test set

(3) Preprocessing of inputs and output

(4) Decide network architecture – # of inputs, # of hidden layers, # of neurons in hidden layer, transfer f^n's

(5) Training of the network

(6) Calculate error measure for training & test set