Alpaydin ch 5.

Multivariate Methods.

Multivariate Data

$$\underline{X} = \begin{bmatrix} X_1^1 & X_2^1 & \cdots & X_d^1 \\ X_1^2 & & \cdots & X_d^2 \\ \vdots & & & \\ X_1^N & \cdots & \cdots & X_d^N \end{bmatrix}$$

d-columns – d variables
(inputs / features / attributes)
N-rows – i.i.d.
observations / examples / instances.

EG. Loan application –
customer observation vector – age, marital status, income, etc.
Typically, these variables are correlated
If not, great simplification.

Goals: simplification / summarization – explain data by few parameters.
exploratory, – generate hypotheses about data.

Parameter Estimation

$$E[\underline{x}] = \underline{\mu} = [\mu_1 \ldots \mu_d]$$

Variance of X_i is σ_i^2, covariance $\sigma_{ij} = \text{Cov}(X_i, X_j)$
$= E[(x_i - \mu_i)(x_j - \mu_j)]$.

$\underline{\Sigma}$ is a matrix, $\Sigma_{ii} = \sigma_i^2$, $\Sigma_{ij} = \sigma_{ij}$.

$$\underline{\Sigma} = E[(\underline{X} - \underline{\mu})(\underline{X} - \underline{\mu})^T] = E[\underline{X}\,\underline{X}^T] - \underline{\mu}\,\underline{\mu}^T$$

Correlation $\quad \text{Corr}(X_i, X_j) = \rho_{ij} = \dfrac{\sigma_{ij}}{\sigma_i \sigma_j}$.

If variables are indep, then correlation and covariance is zero

Parameter Estimation (cont'd)

Given multivariate sample

sample mean. $\underline{m} = \sum_{t=1}^{N} \underline{x}^t / N$, $m_i = \sum_{t=1}^{N} x_i^t / N$, $i = 1, \dots, d$

sample covariance $\underline{S}$ with entries:

$$s_i^2 = \sum_{t=1}^{N} (x_i^t - m_i)^2 / N$$

$$s_{ij} = \sum_{t=1}^{N} (x_i^t - m_i)(x_j^t - m_j) / N.$$

sample correlation. $r_{ij} = s_{ij} / s_i s_j$.

Missing Values — some variables may be missing in observations.

imputation — estimate the missing variable

- mean imputation — substitute the mean of other observations.

imputation by regression — predict these values from existing ones.

but — data may not be missing at random
— if so, need to model why it is missing
(e.g. heights of Americans — Army data)

Multivariate Normal

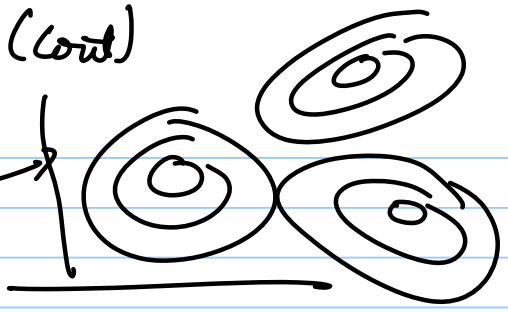
$$p(\underline{x}) = \frac{1}{(2\pi)^{d/2} |\underline{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\underline{x} - \underline{\mu})^T \underline{\Sigma}^{-1} (\underline{x} - \underline{\mu})}$$

Mahalanobis distance $(\underline{x} - \underline{\mu})^T \underline{\Sigma}^{-1} (\underline{x} - \underline{\mu})$

Level sets $(\underline{x} - \underline{\mu})^T \underline{\Sigma}^{-1} (\underline{x} - \underline{\mu}) = c^2$ are d-dim hyperellipsoids centered on $\underline{\mu}$, shape & orientation determined by $\underline{\Sigma}$.

Multivariate Normal (cont)

Bivariate case ($d=2$)



If $\underline{x} \sim N_d(\underline{\mu}, \underline{\Sigma})$

then each dimension of $\underline{x}$ is a univariate normal (converse not necessarily true).

Special case — if the components of $\underline{x}$ are independent then the correlations (covariances) are 0.

$$p(\underline{x}) = \prod_{i=1}^{d} p_i(x_i) = \frac{1}{(2\pi)^{d/2} \prod_{i=1}^{d} \sigma_i} e^{-\frac{1}{2}\sum_{i=1}^{d}\left(\frac{x_i-\mu_i}{\sigma_i}\right)^2}.$$

(Note: we can always pick a coordinate rep for which this is true → by rotating space to diagonalize $\underline{\underline{\Sigma}}$ e.g. $\underline{\underline{\Sigma}} \to \underline{\underline{\Phi}}^T \underline{\underline{\Sigma}} \underline{\underline{\Phi}}$ $\underline{\underline{\Phi}}$ rotation

$\underline{\underline{\Phi}}$ can be obtained from eigenvectors/eigenvalues of $\underline{\underline{\Sigma}}$)

Importance Property

the projection of a d-dim normal onto a vector $\underline{w}$ is also a Gaussian.

or is the projection onto a subspace $\underline{\underline{W}}$.

$$\underline{w}^T \underline{x} = w_1 x_1 + \cdots + w_d x_d \sim N(\underline{w}^T \underline{\mu}, \underline{w}^T \underline{\underline{\Sigma}} \underline{w})$$

$$\underline{\underline{W}}^T \underline{x} \sim N_k(\underline{\underline{W}}^T \underline{\mu}, \underline{\underline{W}}^T \underline{\underline{\Sigma}} \underline{\underline{W}})$$

$\underline{\underline{W}}$ $d \times k$ matrix. rank $k < d$.

(4)

Multivariate classification

$$p(\underline{x}|C_i) = \frac{1}{(2\pi)^{d/2}|\underline{\Sigma}_i|^{1/2}} e^{-\frac{1}{2}(\underline{x}-\underline{\mu}_i)^T \underline{\Sigma}_i^{-1}(\underline{x}-\underline{\mu}_i)}$$

(analyte simplicity)

Discriminant function

$$g_i(\underline{x}) = \log p(\underline{x}|C_i) + \log P(C_i)$$

$$g_i(\underline{x}) = -\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\underline{\Sigma}_i| - \frac{1}{2}(\underline{x}-\underline{\mu}_i)^T \underline{\Sigma}_i^{-1}(\underline{x}-\underline{\mu}_i) + \log P(C_i)$$

Given training samples $\mathcal{X} = \{\underline{x}^t, r^t\}$

Estimate the distribution: $\hat{P}(C_i) = \sum_t r_i^t / N$

$$\underline{m}_i = \sum_t r_i^t \underline{x}^t / \sum_t r_i^t$$

$$\underline{S}_i = \sum_t r_i^t (\underline{x}^t - \underline{m}_i)(\underline{x}^t - \underline{m}_i)^T / \sum_t r_i^t$$

Plug into discriminant function:

$$g_i(x) = -\frac{1}{2}\log|\underline{S}_i| - \frac{1}{2}(\underline{x}-\underline{m}_i)^T \underline{S}_i^{-1}(\underline{x}-\underline{m}_i) + \log \hat{P}(C_i)$$

(dropping constant term)

Quadratic Discriminant

Can be written as

$$g_i(\underline{x}) = \underline{x}^T \underline{W}_i \underline{x} + \underline{w}_i^T \underline{x} + w_{i0}$$

$$\underline{W}_i = -\frac{1}{2}\underline{S}_i^{-1}, \quad \underline{w}_i = \underline{S}_i^{-1}\underline{m}_i,$$

$$w_{i0} = -\frac{1}{2}\underline{m}_i^T \underline{S}_i^{-1}\underline{m}_i - \frac{1}{2}\log|\underline{S}_i| + \log \hat{P}(C_i).$$

No. parameters: $K \cdot d$ mean + $K \cdot d(d+1)/2$ covariance

(5) Multivariate Classification (cont)

For large d, samples are small, Σ_i may be singular and Σ^{-1} won't exist.

May need to decrease dimensionality (next lecture) or pool data and estimate common covariance matrix: $\Sigma = \sum_i \hat{P}(C_i)\Sigma_i$.

discriminant reduces to $g_i(x) = -\frac{1}{2}(x-m_i)^T \Sigma^{-1}(x-m_i) + \log \hat{P}(C_i)$

Further simplification assumes off-diagonal elements covariance are zero.

naive Bayes classifier: $p(x_j|C_i)$ univariate Gaussian. Σ is diagonal (so is Σ^{-1})

$$g_i(x) = -\frac{1}{2}\sum_{j=1}^{d}\left(\frac{x_j^t - m_{ij}}{s_j}\right)^2 + \log \hat{P}(C_i)$$

m_{ij} is mean of j^{th} component of i^{th} class.

$\rightsquigarrow$ reduces complexity of Σ from $O(n^2)$ to $O(n)$.

Extreme simplification: assume all variances are the same. Then Mahalanobis distance = Euclidean dist.

$$g_i(x) = -\frac{1}{2s^2}\sum_{j=1}^{d}(x_j^t - m_{ij})^2 + \log \hat{P}(C_i).$$

If priors are equal — $g_i(x) = -\|x-m_i\|^2$ nearest mean classifier. $\rightarrow$ prototype/template matching

$g_i(x) = w_i^T x + w_{i0}$, $w_i = m_i$, $w_{i0} = -\frac{1}{2}\|m_i\|^2$

'6) Tuning Complexity

As before, for polynomial regression, we can choose between models with different complexity.

→ Full Covariance $d + \frac{1}{2}d(d+1)$ parameters

→ Indep. Same Variance $d + 1$

Same procedures as before:

(e.g. cross-validation, regularization, etc.)

Discrete Features:

Discrete attributes – color $\in$ {red, blue, green...}

pixel $\in$ {on, off}

Suppose x_j are binary (Bernoulli)

$$p_{ij} = p(x_j = 1 \mid C_i)$$

$$p(\mathbf{x} \mid C_i) = \prod_{j=1}^{d} p_{ij}^{x_j} (1 - p_{ij})^{(1-x_j)}$$

Discriminant $g_i(\mathbf{x}) = \log p(\mathbf{x} \mid C_i) + \log P(C_i)$

$$= \sum_j \{x_j \log p_{ij} + (1 - x_j) \log (1 - p_{ij})\} + \log P(C_i)$$

linear in x

Estimate $\hat{p}_{ij} = \sum_t x_j^t r_i^t / \sum_t r_i^t$

General Case: multinomial x_j choose from $(v_1 \dots v_{n_j})$

p_{ijk} = prob (x_j from class C_i takes value v_k)

$$p_{ijk} = p(z_{jk} = 1 \mid C_i) = p(x_j = v_k \mid C_i)$$

Discrete Features (cont)

$z_{jk}^t = \begin{cases} 1, & \text{if } x_j^t = v_k \\ 0, & \text{otherwise} \end{cases}$

If attributes are independent: $p(\underline{x}|C_i) = \prod_{j=1}^{d} \prod_{k=1}^{n_j} p_{ijk}^{z_{jk}}$

discriminant $g_i(\underline{x}) = \sum_j \sum_k z_{jk} \log p_{ijk} + \log \hat{P}(C_i)$

Max like estimate for p_{ijk} is $\hat{p}_{ijk} = \sum_t z_{jk}^t r_i^t / \sum_t r_i^t$

Multivariate Regression

$r^t = g(\underline{x}^t | w_0 .. w_d) + \epsilon = w_0 + w_1 x_1^t + .. + w_d x_d^t + \epsilon.$

least squares:

$E(w_0, .. w_d | \mathcal{X}) = \frac{1}{2} \sum_t (r^t - w_0 - w_1 x_1^t - .. - w_d x_d^t)^2.$

$\underline{\underline{X}} = \begin{bmatrix} 1 & x_1^1 & .. & x_d^1 \\ \vdots & & & \\ 1 & x_1^N & .. & x_d^N \end{bmatrix}, \quad \underline{w} = \begin{bmatrix} w_0 \\ \vdots \\ w_d \end{bmatrix}, \quad \underline{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r^N \end{bmatrix}$

minimize wrt. w's $\quad \underline{\underline{X}}^T \underline{\underline{X}} \underline{w} = \underline{\underline{X}}^T \underline{r}.$

$\underline{\hat{w}} = (\underline{\underline{X}}^T \underline{\underline{X}})^{-1} \underline{\underline{X}}^T \underline{r}.$