CH. 5: Multivariate Methods

-- Methods for dealing with multivariate data.

5.1 Multivariate Data

- There are various types of data:

  i) Numerical data: length, width, height, size,
  volume, weight, speed, temperature, ...

  ii) Symbolic data: tag, label, index, name, title, .

  iii) Abstract data: concept, idea, knowledge,
  thought, sensation, expression, feeling, ...

  iv) Entity data: human, animal, car, building, ….

- There are diverse media of data: text, graph, figure, voice, image, video, …………
- Data are often represented in the form of multivariate format, e.g., vector, matrix, tensor, collectively called **pattern**.

Example: $\boldsymbol{x} = (x_1, x_2, \cdots, x_d)^T$: d-D data vector

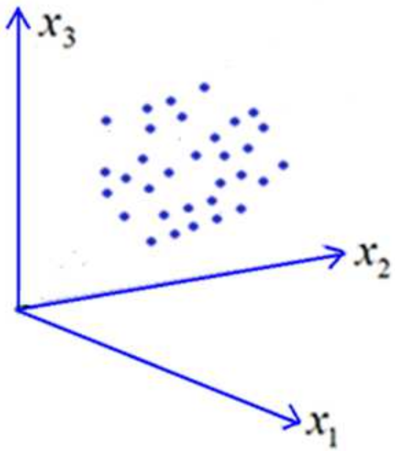
where $x_1, x_2, \cdots, x_d$: **attributes/features**

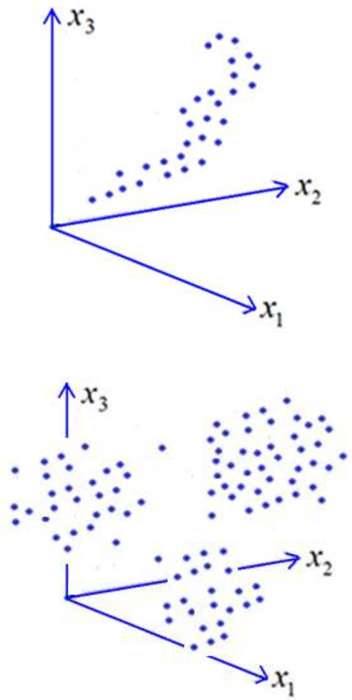
e.g., $\textbf{patient} = (\text{age, gender, height, weight, blood presure, blood sugar, cholesterol}, \cdots)^T$

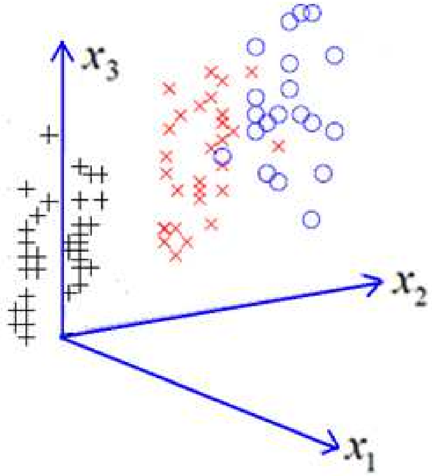
$\textbf{customer} = (\text{age, marage, income, saving, installment payment}, \cdots\cdots)^T$

- Graphical representation (visualization)

  A **pattern** $\boldsymbol{x} = (x_1, x_2, \cdots, x_d)^T$ can be represented as a point in a d-D space. A **sample** $X = \{\boldsymbol{x}_i\}_{i=1}^{N}$ can be represented as a set of points in the space.







- Matrix representation (computation)

A **sample** $X = \{\boldsymbol{x}_i\}_{i=1}^{N}$ can be represented as a matrix.

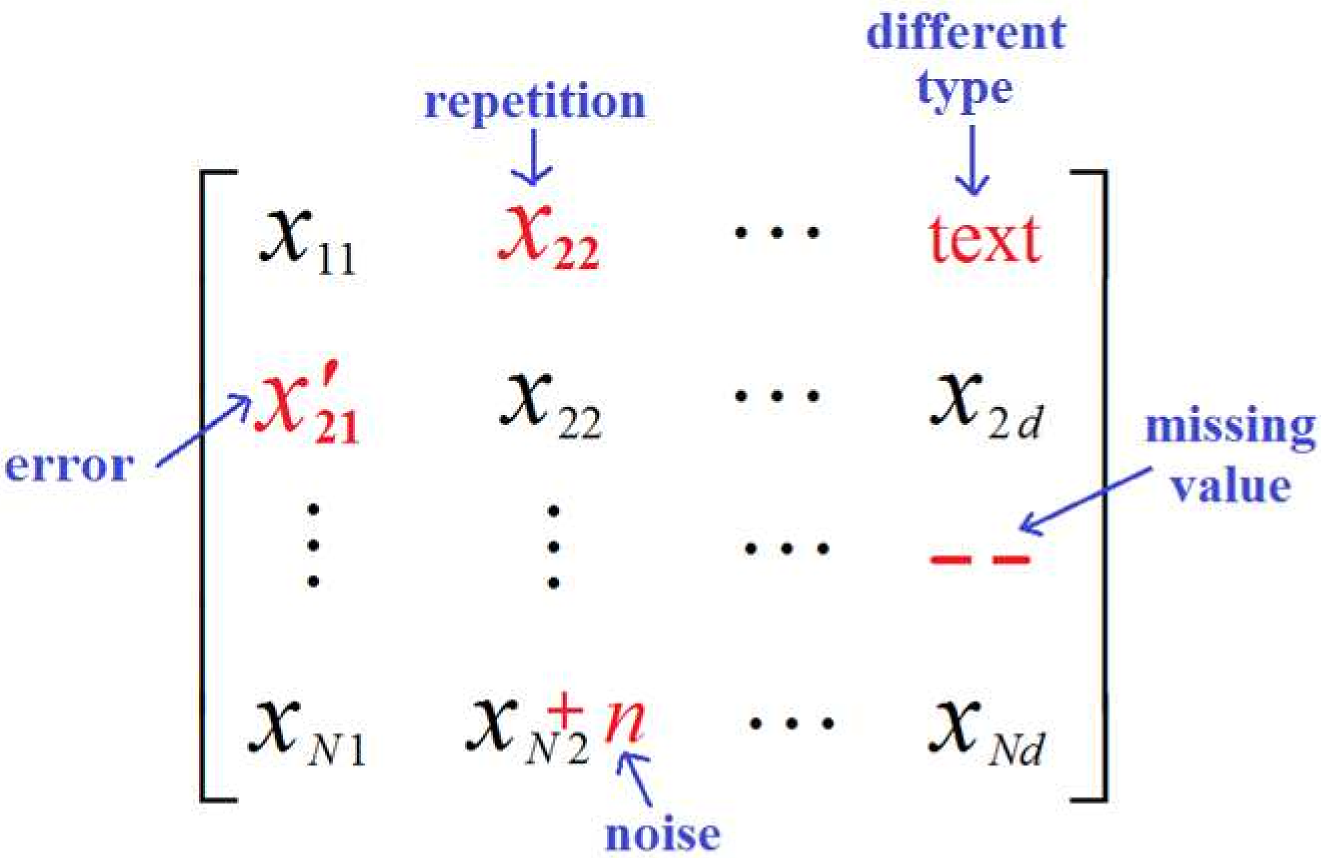
$$X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]^T = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & & & \\ x_{N1} & x_{N2} & \cdots & x_{Nd} \end{bmatrix}$$

e.g., Rotation

$$RX = R\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & & & \\ x_{N1} & x_{N2} & \cdots & x_{Nd} \end{bmatrix}$$

5.2 Data Cleaning

- Data may have noise, error, missing value, repetition, different formats.

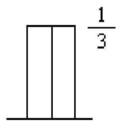


- Cleaning methods – for compensating for shortcomings of data

(a) Noisy and error data -- smoothing

Linear smoothing

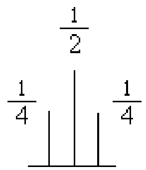
Averaging filter

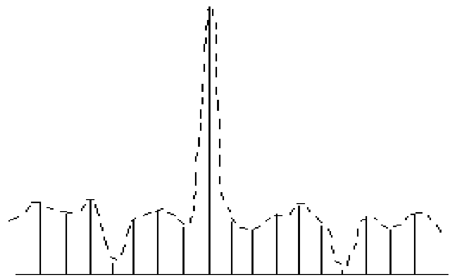


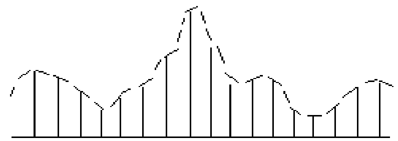
Input data

Output data

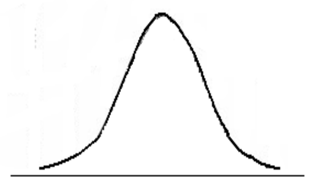
Weighted filter







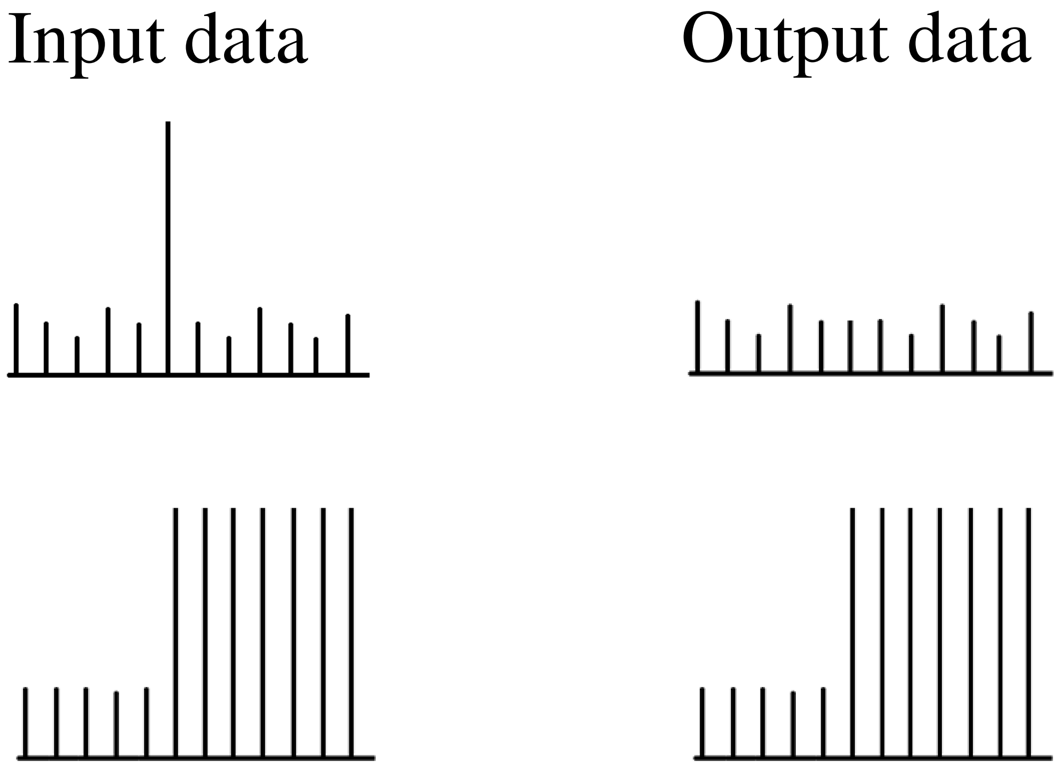
Gaussian filter



$$g(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\bar{x})^2}{2\sigma^2}}$$

Nonlinear smoothing

- Median filter



- *K*-nearest neighbors (K-NN)

(b) Missing values -- imputation

Mean imputation: Substitute the mean of the available data of the missing attribute

Imputation by regression: Predict based on other attributes

$$
\begin{bmatrix}
x_{11} & x_{12} & \cdots\cdots\cdots & x_{1d} \\
x_{21} & x_{22} & \cdots\cdots\cdots & x_{2d} \\
\vdots & & & \vdots \\
x_{N1} & x_{N2} & \cdots\cdots\cdots & x_{Nd}
\end{bmatrix}
$$

(c) Different formats -- quantization

e.g.,

2 no 4 0.2 1 春 4 3 2 4

⇓ ⇓

yes=1, no=0 春=1, 夏=2, 秋=3, 冬=4

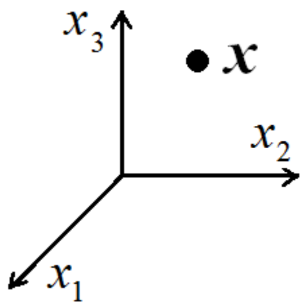
⇓ ⇓

2 **0** 4 0.2 1 **1** 4 3 2 4

5.3 Linear vs Nonlinear Space

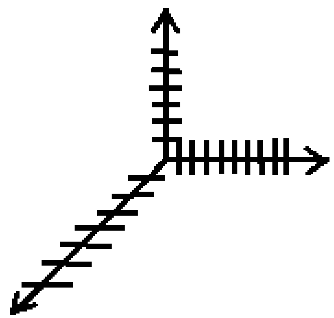
Data vector: $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$

- Linear space

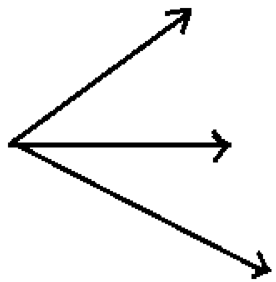


- Nonlinear space

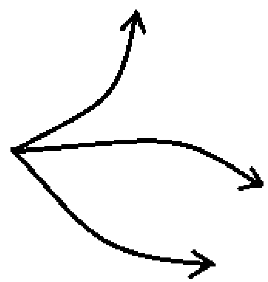
(a) Different scales of attributes



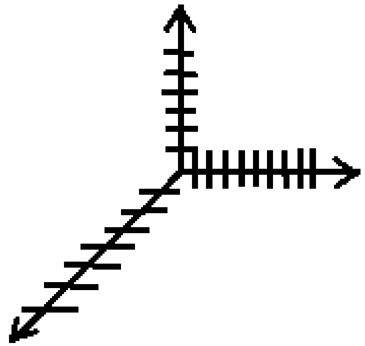
(b) Correlated attributes



(c) Curvilineal attributes



(a) Different Scales -- Normalization



Normalization: $x_i = \dfrac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}$

Example: $\boldsymbol{x}_1 = (2,\ 3,\ 4,\ 2,\ 1,\ 15)^T$,

$\boldsymbol{x}_2 = (1,\ 2,\ 2,\ 4,\ 3,\ 51)^T$, $\boldsymbol{x}_3 = (1,\ 4,\ 3,\ 2,\ 2,\ 35)^T$.

$$d(\boldsymbol{x}_1,\boldsymbol{x}_2)=\sqrt{(2-1)^2+(3-2)^2+(4-2)^2+(2-4)^2+(1-3)^2+(15-51)^2} \approx |15-51|$$

$$d(\boldsymbol{x}_2,\boldsymbol{x}_3)=\sqrt{(1-1)^2+(2-4)^2+(2-3)^2+(4-2)^2+(3-2)^2+(51-35)^2} \approx |51-35|$$

$$d(\boldsymbol{x}_3,\boldsymbol{x}_1)=\sqrt{(1-2)^2+(4-3)^2+(3-4)^2+(2-2)^2+(2-1)^2+(35-15)^2} \approx |35-15|$$

$d(\boldsymbol{x}_1,\boldsymbol{x}_2),\ d(\boldsymbol{x}_2,\boldsymbol{x}_3),\ d(\boldsymbol{x}_3,\boldsymbol{x}_1)$ are dominated by feature x_6

Normalization: $x_i = \dfrac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)},\ i = 1,\cdots,6$

$$\boldsymbol{x}_1' = (1,\ 0.5,\ 1,\ 0,\ 0,\ 0),\quad \boldsymbol{x}_2' = (0,\ 0,\ 0,\ 1,\ 1,\ 1),$$

$$d(\boldsymbol{x}_1',\boldsymbol{x}_2') = \sqrt{(1-0)^2+(0.5-0)^2+(1-0)^2+(0-1)^2+(0-1)^2+(0-1)^2}$$

Z-normalization:

$$x_i' = \frac{x_i - \mu_{x_i}}{\sigma_{x_i}} \sim Z(0,1)$$

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & & & \\ x_{N1} & x_{N2} & \cdots & x_{Nd} \end{bmatrix}$$

(b) Correlated attributes -- PCA

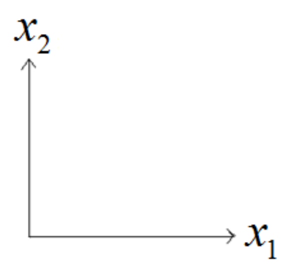
Principal Component Analysis

-- Linearly transforms a number of correlated features $\{x_1, \cdots, x_n\}$ into the same number of uncorrelated features $\{y_1, \cdots, y_n\}$.

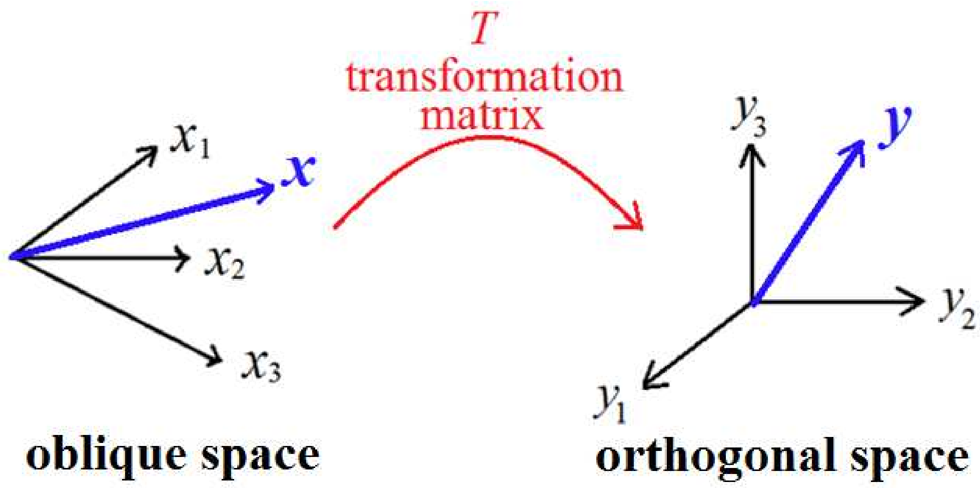
- Correlation coefficient: $\rho(x_i, x_j) = \dfrac{\sigma_{ij}}{\sigma_i \sigma_j}$

  Uncorrelation: $\rho = 0 \Rightarrow \sigma_{ij} = \mathrm{Cov}(x_i, x_j) = 0.$

  Graphically, the axes corresponding to uncorelated x_i and x_j are orthogonal.



- The transformation from an oblique to an orthogonal space is accomplished through a transformation matrix T.

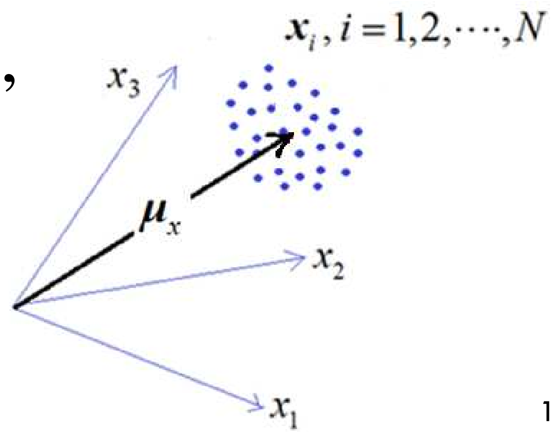


- **PCA derive *T***

Data vectors $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{id})^T,$

$$i = 1, 2, \cdots, N$$

Mean vectors: $\boldsymbol{\mu}_x = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i,$



Covariance matrix:

$$
\begin{aligned}
C_x &= \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i - \boldsymbol{\mu}_x)(\boldsymbol{x}_i - \boldsymbol{\mu}_x)^T \\
&= \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_i\boldsymbol{x}_i^T - \boldsymbol{x}_i\boldsymbol{\mu}_x^T - \boldsymbol{\mu}_x\boldsymbol{x}_i^T + \boldsymbol{\mu}_x\boldsymbol{\mu}_x^T) \\
&= \frac{1}{N}[\sum_{i=1}^{N}\boldsymbol{x}_i\boldsymbol{x}_i^T - \sum_{i=1}^{N}\boldsymbol{x}_i\boldsymbol{\mu}_x^T - \boldsymbol{\mu}_x\sum_{i=1}^{N}\boldsymbol{x}_i^T + \sum_{i=1}^{N}\boldsymbol{\mu}_x\boldsymbol{\mu}_x^T] \\
&= \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i\boldsymbol{x}_i^T + \boldsymbol{\mu}_x\boldsymbol{\mu}_x^T - \boldsymbol{\mu}_x\boldsymbol{\mu}_x^T - \boldsymbol{\mu}_x\boldsymbol{\mu}_x^T = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i\boldsymbol{x}_i^T - \boldsymbol{\mu}_x\boldsymbol{\mu}_x^T
\end{aligned}
$$

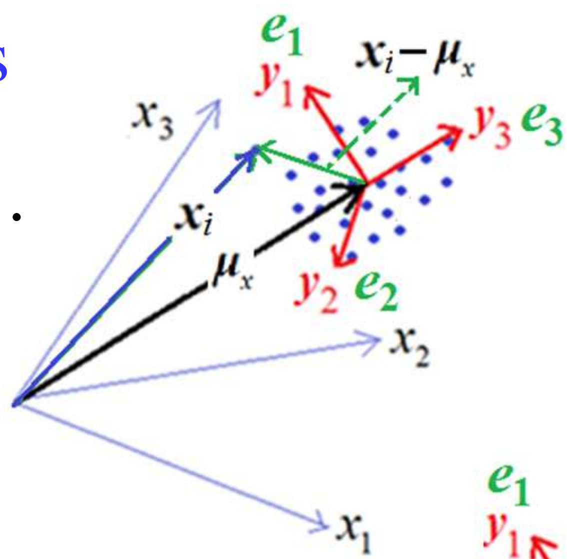
Let λ_i and $\boldsymbol{e}_i$, $i = 1,\cdots,d$, be the eigenvalues and eigenvectors of C_x, i.e., $C_x\boldsymbol{e}_i = \lambda_i\boldsymbol{e}_i$.

$\boldsymbol{e}$'s : principal components

Suppose $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$.

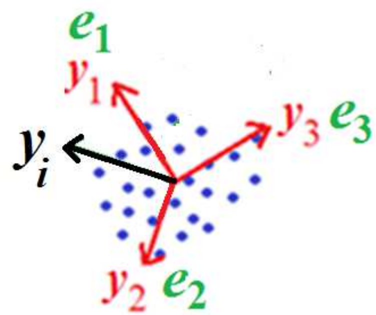
Let $A = [\boldsymbol{e}_1 \ \ \boldsymbol{e}_2 \ \cdots \ \boldsymbol{e}_d]$,

and $\boldsymbol{y}_i = A^T(\boldsymbol{x}_i - \boldsymbol{\mu}_x)$.



y-axes corresponding to eigenvectors

$\boldsymbol{e}$'s are orthogonal, i.e., uncorrelated.



$$\boldsymbol{\mu}_y = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{y}_i = \frac{1}{N}\sum_{i=1}^{N}A^T(\boldsymbol{x}_i - \boldsymbol{\mu}_x) = \frac{1}{N}A^T\left(\sum_{i=1}^{N}\boldsymbol{x}_i - \sum_{i=1}^{N}\boldsymbol{\mu}_x\right)$$

$$= \frac{1}{N}A^T\left(N\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i - N\boldsymbol{\mu}_x\right) = \frac{1}{N}A^T(N\boldsymbol{\mu}_x - N\boldsymbol{\mu}_x) = \boldsymbol{0}$$

$$C_y = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{y}_i - \boldsymbol{\mu}_y)(\boldsymbol{y}_i - \boldsymbol{\mu}_y)^T = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{y}_i\boldsymbol{y}_i^T$$

$$= \frac{1}{N}\sum_{i=1}^{N}A^T(\boldsymbol{x}_i - \boldsymbol{\mu}_x)(\boldsymbol{x}_i - \boldsymbol{\mu}_x)^T A = A^T C_x A$$

$$= \begin{bmatrix}\boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdot\cdot & \boldsymbol{e}_d\end{bmatrix}^T C_x \begin{bmatrix}\boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdot\cdot & \boldsymbol{e}_d\end{bmatrix}$$

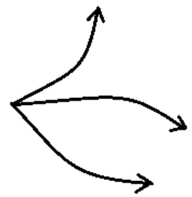
$$= \begin{bmatrix}\boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdot\cdot & \boldsymbol{e}_d\end{bmatrix}^T \begin{bmatrix}C_x\boldsymbol{e}_1 & C_x\boldsymbol{e}_2 & \cdot\cdot & C_x\boldsymbol{e}_d\end{bmatrix}$$

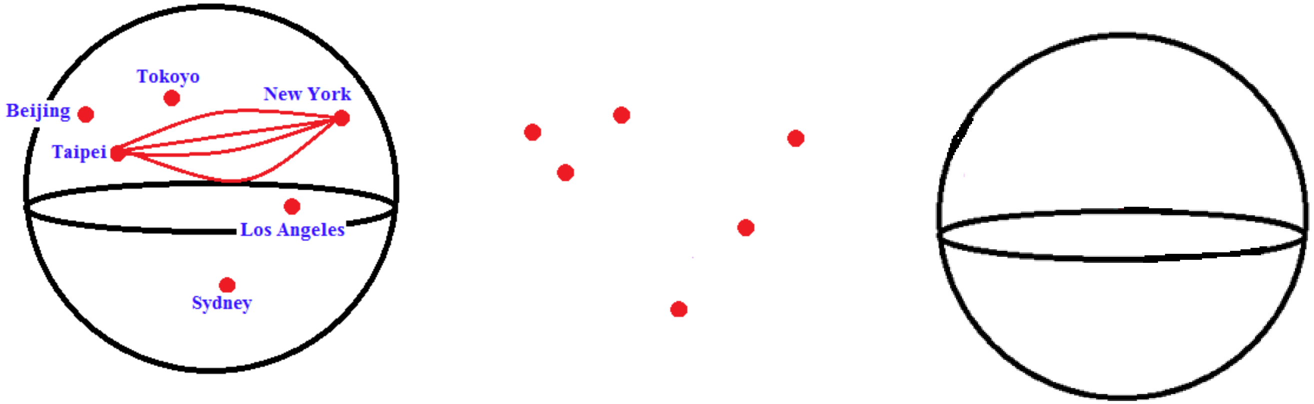
$$= \begin{bmatrix}\boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdot\cdot & \boldsymbol{e}_d\end{bmatrix}^T \begin{bmatrix}\lambda_1\boldsymbol{e}_1 & \lambda_2\boldsymbol{e}_2 & \cdot\cdot & \lambda_d\boldsymbol{e}_d\end{bmatrix}$$

$$= \begin{bmatrix}\lambda_1\boldsymbol{e}_1^T\boldsymbol{e}_1 & 0 & \cdot\cdot & 0 \\ 0 & \lambda_2\boldsymbol{e}_2^T\boldsymbol{e}_2 & 0 & \cdot\cdot \\ \cdot\cdot & 0 & \cdot\cdot & 0 \\ 0 & \cdot\cdot & 0 & \lambda_d\boldsymbol{e}_d^T\boldsymbol{e}_d\end{bmatrix} = \begin{bmatrix}\lambda_1 & 0 & \cdot\cdot & 0 \\ 0 & \lambda_2 & 0 & \cdot\cdot \\ \cdot\cdot & 0 & \cdot\cdot & 0 \\ 0 & \cdot\cdot & 0 & \lambda_d\end{bmatrix}$$

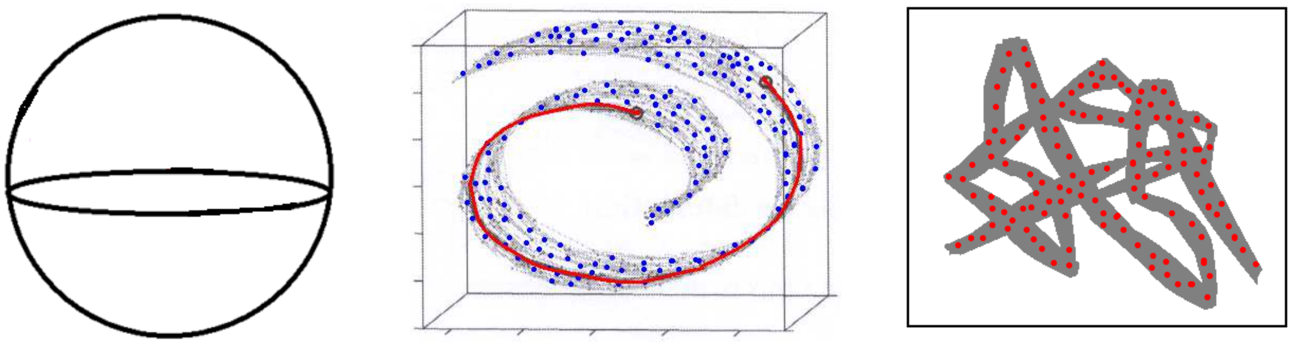
(c) Curvilineal attributes – Manifold Learning

- Euclidean vs Geodesic distance





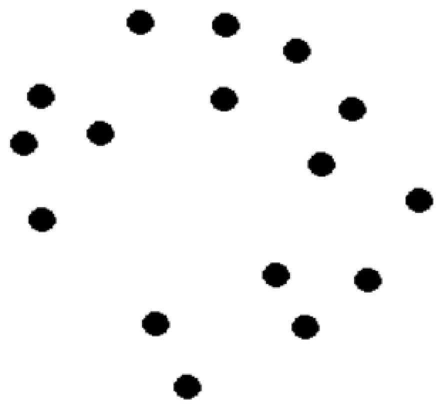
- Manifold

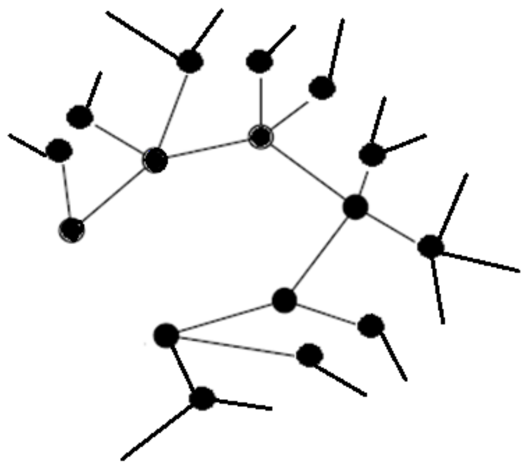


Without knowing the manifold, on which data points lied, their distances are meaningless.

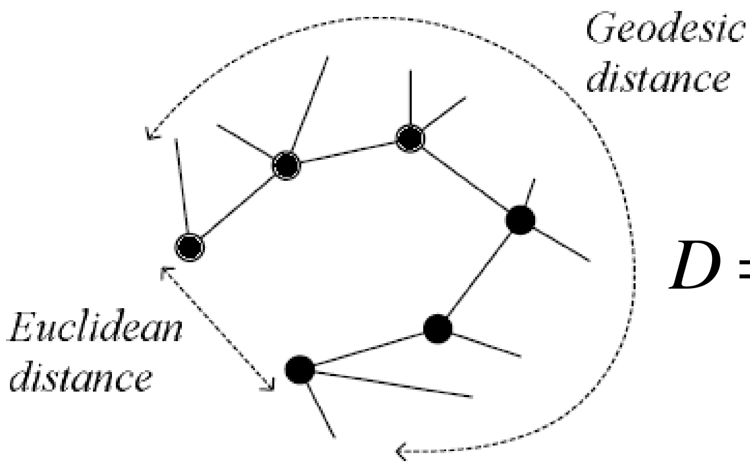
Manifold Learning – Isometric Feature Mapping

-- Approximates the manifold by defining a graph whose nodes correspond to data points and whose edges connect neighboring points.





For neighboring points that are close in the input space (those with i, distance less than some ε or ii, one of the n nearest), Euclidean distance is used. For faraway points, geodesic distance is calculated by summing the distances between the points along the way over the manifold.



$$D = \begin{array}{c c} & \begin{array}{cccc} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdot\cdot & \boldsymbol{x}_N \end{array} \\ \begin{array}{c} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \cdot\cdot \\ \boldsymbol{x}_N \end{array} & \begin{pmatrix} d_{11} & d_{12} & \cdot\cdot & d_{1N} \\ d_{21} & d_{22} & \cdot\cdot & d_{2N} \\ \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \\ d_{N1} & d_{N2} & \cdot\cdot & d_{NN} \end{pmatrix} \end{array}$$

- Once the distance matrix $D = \left[d_{ij} \right]_{N \times N}$ is formed, use multidimensional scaling (**MDS**) technique to place the N points in any selected space s.t. the Euclidean distances between them is as close possible to D.

A Global Geometric Framework for Nonlinear Dimensionality Reduction, J.B. Tenenbaum, V. de Silva and J.C. Langford, Science, Vol 290, 2000.

MDS Algorithm:

Given matrix $D = [d_{rs}]$, where d_{rs} is the distance between data points r and s in the p-D space.

Suppose data points have been centered at the origin.

1. Calculate $B = \left[b_{rs}\right]$, where

$$b_{rs} = \frac{1}{2}(d_{r\bullet}^2 + d_{\bullet s}^2 - d_{\bullet\bullet}^2 - d_{rs}^2)$$

$$d_{r\bullet}^2 = \frac{1}{N}\sum_s d_{rs}^2, \quad d_{\bullet s}^2 = \frac{1}{N}\sum_r d_{rs}^2, \quad d_{\bullet\bullet}^2 = \frac{1}{N^2}\sum_r \sum_s d_{rs}^2$$

2. Find the spectral decomposition of B, $B = E\Lambda E^T$.

3. Discard from Λ the $N - p$ small eigenvalues and from E the corresponding eigenvectors to form Λ' and E', respectively.

4. Find $Z = E'\Lambda'^{1/2}$.

The coordinates of the points are the rows of Z.

See Appendix for the detail of MDS.

Multidimensional Scaling, Michael A.A. Cox and T.E. Cox, 2006.

Spectral Decomposition

- Let E be a $n \times n$ matrix whose ith column is the unit eigenvector $\boldsymbol{e_i}$ of square real matrix $A_{n\times n}$.

$$A = AEE^T = A(\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_n)E^T = (A\boldsymbol{e}_1, A\boldsymbol{e}_2, \cdots, A\boldsymbol{e}_n)E^T$$

$$= (\lambda_1 \boldsymbol{e}_1, \lambda_2 \boldsymbol{e}_2, \cdots, \lambda_n \boldsymbol{e}_n)E^T = \lambda_1 \boldsymbol{e}_1 \boldsymbol{e}_1^T + \lambda_2 \boldsymbol{e}_2 \boldsymbol{e}_2^T + \cdots + \lambda_n \boldsymbol{e}_n \boldsymbol{e}_n^T$$

$$= \boldsymbol{e}_1 \lambda_1 \boldsymbol{e}_1^T + \boldsymbol{e}_2 \lambda_2 \boldsymbol{e}_2^T + \cdots + \boldsymbol{e}_n \lambda_n \boldsymbol{e}_n^T$$

$$= E\Lambda E^T$$

$$\text{where} \quad \Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & \ddots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

$A = E\Lambda E^T$:

spectral decomposition of matrix A .

Singular Value Decomposition

- $A_{m\times n}$: $m\times n$ real matrix ($m > n$)

  Let $U_{m\times m}$ contain eigenvectors of $(AA^T)_{m\times m}$

  Let $V_{n\times n}$ contain eigenvectors of $(A^T A)_{n\times n}$

  s.t. $A = UWV^T$

  where $W_{m\times n} = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_n)$

  $\sigma_1, \sigma_2, \cdots, \sigma_n$: **singular values** of A

- $A^T A = (UWV^T)^T UWV^T = VW^T (U^T U) WV^T$

$$= VW^T IWV^T = VW^T WV^T = V\mathrm{diag}(\sigma_1^2, \sigma_2^2, \cdots, \sigma_n^2) V^T$$

- Let the spectral decomposition of $(A^T A)_{n \times n}$

$$A^T A = [\boldsymbol{e}_1 \cdots\cdot \boldsymbol{e}_n]\mathrm{diag}(\lambda_1, \cdots, \lambda_n)[\boldsymbol{e}_1 \cdots\cdot \boldsymbol{e}_n]^T$$

$$= V\mathrm{diag}(\lambda_1, \cdots, \lambda_n) V^T$$

$$A^T A = V\mathrm{diag}(\sigma_1^2, \sigma_2^2, \cdots, \sigma_n^2) V^T = V\mathrm{diag}(\lambda_1, \cdots\cdots, \lambda_n) V^T$$

The eigenvalues λ_i of $A^T A$ correspond to the singular values σ_i of A.

5.4 Multivariate Normal Distribution

Suppose data vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_d)^T \sim N(\boldsymbol{\mu}, \Sigma)$

i.e., $p(\boldsymbol{x}) = \dfrac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\dfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]$

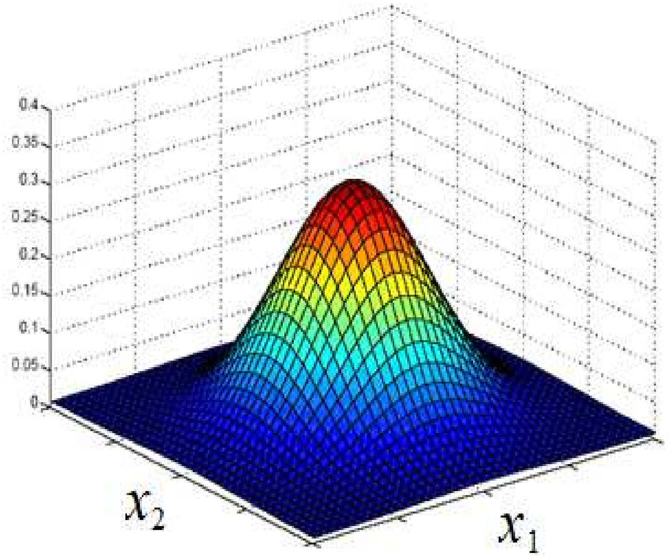
where $(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$: Mahalanobis distance

measures the distance

from $\boldsymbol{x}$ to $\boldsymbol{\mu}$ and $\sum$

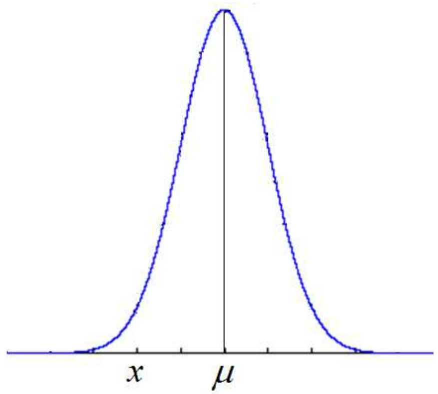
normalizes for different

variances.



e.g., $d = 1$,

$$p(x) = \frac{1}{(2\pi)^{1/2}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2} \right]$$



$$\frac{(x-\mu)^2}{\sigma^2} = \left(\frac{x-\mu}{\sigma} \right)^2 = (x-\mu)(\sigma^2)^{-1}(x-\mu):$$

the square distance from x to μ in σ unit.

- Σ^{-1} normalizes all variables to unit variance
- If a variable has a larger variance than another, it contributes less weight in the Mahalanobis distance.

- $(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu}) = c^2$: hyperellipsoid centered at $\boldsymbol{\mu}$. Both its shape and orientation are governed by $\boldsymbol{\Sigma}$.

e.g., $d = 2, \quad \Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$

$$\Sigma^{-1} = \begin{bmatrix} \sigma_2^2 & -\rho\sigma_1\sigma_2 \\ -\rho\sigma_1\sigma_2 & \sigma_1^2 \end{bmatrix} / \begin{vmatrix} \sigma_2^2 & -\rho\sigma_1\sigma_2 \\ -\rho\sigma_1\sigma_2 & \sigma_1^2 \end{vmatrix}$$

$$\begin{vmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{vmatrix} = \sigma_1^2\sigma_2^2 - \rho^2\sigma_1^2\sigma_2^2 = (1-\rho^2)\sigma_1^2\sigma_2^2$$

$$\Sigma^{-1} = \frac{\begin{bmatrix} \sigma_2^2 & -\rho\sigma_1\sigma_2 \\ -\rho\sigma_1\sigma_2 & \sigma_1^2 \end{bmatrix}}{(1-\rho^2)\sigma_1^2\sigma_2^2} = \frac{1}{1-\rho^2}\begin{bmatrix} \frac{1}{\sigma_1^2} & -\frac{\rho}{\sigma_1\sigma_2} \\ -\frac{\rho}{\sigma_1\sigma_2} & \frac{1}{\sigma_2^2} \end{bmatrix}$$

$$\left(\boldsymbol{x}-\boldsymbol{\mu}\right)^T \Sigma^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)$$

$$= \frac{1}{1-\rho^2}(x_1-\mu_1 \quad x_2-\mu_2)\begin{bmatrix} \frac{1}{\sigma_1^2} & -\frac{\rho}{\sigma_1\sigma_2} \\ -\frac{\rho}{\sigma_1\sigma_2} & \frac{1}{\sigma_2^2} \end{bmatrix}\begin{pmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{pmatrix}$$

$$= \frac{1}{1-\rho^2}\left(\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2 - 2\rho\frac{x_1-\mu_1}{\sigma_1}\frac{x_2-\mu_2}{\sigma_2} + \left(\frac{x_2-\mu_2}{\sigma_2}\right)^2\right)$$

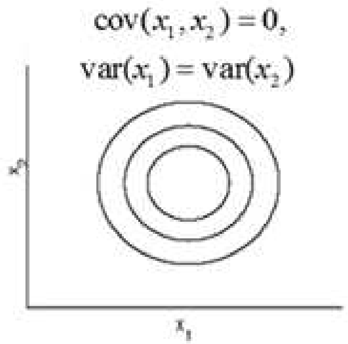
Let $z_i = \dfrac{x_i - \mu_i}{\sigma_i} \sim N(0,1),\ i = 1,2$ (z- normalization)

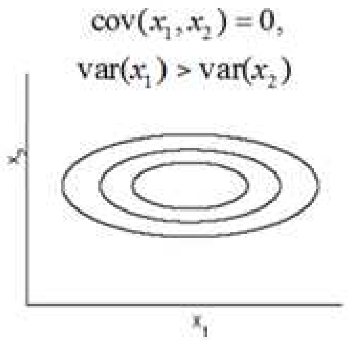
$$\Rightarrow\ (\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu}) = -\frac{1}{1-\rho^2}\left(z_1^2 - 2\rho z_1 z_2 + z_2^2\right)$$

Consider $z_1^2 - 2\rho z_1 z_2 + z_2^2 = c^2,\quad -1 \le \rho \le 1$

which expresses an ellipse.

- When z_1 and z_2 are independent, the major axes of the density are parallel to the input axes.
- The density becomes an ellipse if the variances of z_1 and z_2 are different.

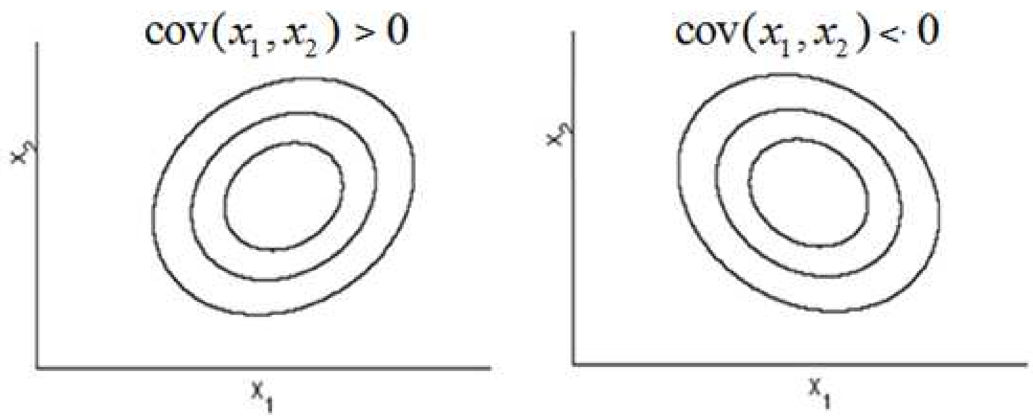




- The density rotates depending on the sign of the correlation.

  i) When $\rho > 0$, the major axis of the ellipse has a positive slop.

  ii) When $\rho < 0$, the major axis of the ellipse has a negative slop.



- Let $\boldsymbol{x} \sim N(\boldsymbol{\mu}, \Sigma)$, $\boldsymbol{w} \in R^d \Rightarrow \boldsymbol{w}^T \boldsymbol{x} \sim N\left(\boldsymbol{w}^T \boldsymbol{\mu},\ \boldsymbol{w}^T \Sigma \boldsymbol{w}\right)$

$\because\ \boldsymbol{x} \sim N(\boldsymbol{\mu}, \Sigma)$,

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right]$$

$$p\left(\boldsymbol{w}^T \boldsymbol{x}\right) = \frac{1}{\sqrt{2\pi}(\mathrm{Var}(\boldsymbol{w}^T \boldsymbol{x}))^{1/2}} \cdot \exp\left[-\frac{\left(\boldsymbol{w}^T \boldsymbol{x} - E[\boldsymbol{w}^T \boldsymbol{x}]\right)^T \left(\boldsymbol{w}^T \boldsymbol{x} - E[\boldsymbol{w}^T \boldsymbol{x}]\right)}{2\mathrm{Var}(\boldsymbol{w}^T \boldsymbol{x})}\right]$$

$$E[\boldsymbol{w}^T \boldsymbol{x}] = \boldsymbol{w}^T E[\boldsymbol{x}] = \boldsymbol{w}^T \boldsymbol{\mu}.$$

$$\begin{aligned}\mathrm{Var}(\boldsymbol{w}^T\boldsymbol{x}) &= E[(\boldsymbol{w}^T\boldsymbol{x}-E[\boldsymbol{w}^T\boldsymbol{x}])^2] \\ &= E[(\boldsymbol{w}^T\boldsymbol{x}-\boldsymbol{w}^T\boldsymbol{\mu})^2] = E[(\boldsymbol{w}^T(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{w}] \\ &= \boldsymbol{w}^T E[(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T]\boldsymbol{w} = \boldsymbol{w}^T\Sigma\boldsymbol{w}\end{aligned}$$

$$p\left(\boldsymbol{w}^T\boldsymbol{x}\right) = \frac{1}{\sqrt{2\pi}\sqrt{\boldsymbol{w}^T\Sigma\boldsymbol{w}}}\cdot$$

$$\exp\left[-\frac{\left(\boldsymbol{w}^T\boldsymbol{x}-\boldsymbol{w}^T\boldsymbol{\mu}\right)^T\left(\boldsymbol{w}^T\boldsymbol{x}-\boldsymbol{w}^T\boldsymbol{\mu}\right)}{2\boldsymbol{w}^T\Sigma\boldsymbol{w}}\right]$$

i.e., $\boldsymbol{w}^T\boldsymbol{x} \sim N\left(\boldsymbol{w}^T\boldsymbol{\mu},\ \boldsymbol{w}^T\Sigma\boldsymbol{w}\right)$

The projection of a d-D normal on a vector $\boldsymbol{w}$ is univariate (i.e., 1-D) normal.

- Let W be a $d \times k$ matrix.

  Then $W^T \boldsymbol{x} \sim N\left(W^T \boldsymbol{\mu},\ W^T \Sigma W\right)$

5.5 Multivariate Classification

From the Bayes' rule, the posterior probability of C_i

$$P(C_i \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid C_i)\, p(C_i)}{p(\boldsymbol{x})}, \quad i = 1, \cdots, K$$

Define discriminant function of C_i as

$$g_i\left(\boldsymbol{x}\right) = \log p\left(\boldsymbol{x} | C_i\right) + \log P\left(C_i\right).$$

Assume

$$p\left(\boldsymbol{x} | C_i\right) = \frac{1}{\left(2\pi\right)^{d/2} \left|\Sigma_i\right|^{1/2}} \exp\left[-\frac{1}{2}\left(\boldsymbol{x} - \boldsymbol{\mu}_i\right)^T \Sigma_i^{-1}\left(\boldsymbol{x} - \boldsymbol{\mu}_i\right)\right]$$

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

Ignore $-\frac{d}{2}\log 2\pi$

$$g_i(\boldsymbol{x}) = -\frac{1}{2}\log\left|\Sigma_i\right| - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T \Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)$$
$$+\log P(C_i) \quad ------ (\mathrm{A})$$

Given a sample $X = \{\boldsymbol{x}^t, \boldsymbol{r}^t\}_{t=1}^N$, where

$$\boldsymbol{x} = \{x_1, \cdots, x_d\}, \quad \boldsymbol{r} = \{r_1, \cdots, r_K\}, \quad r_i = \begin{cases} 1 & \boldsymbol{x} \in C_i \\ 0 & \text{otherwise} \end{cases}$$

Let $\hat{P}(C_i)$, $\boldsymbol{m}_i$, S_i be the estimators of $P(C_i)$, $\boldsymbol{\mu}_i$, Σ_i from the sample.

$$\hat{P}(C_i) = \frac{\sum_t r_i^t}{N}, \quad \boldsymbol{m}_i = \frac{\sum_t r_i^t \boldsymbol{x}^t}{\sum_t r_i^t}, \quad S_i = \frac{\sum_t r_i^t \left(\boldsymbol{x}^t - \boldsymbol{m}_i\right)\left(\boldsymbol{x}^t - \boldsymbol{m}_i\right)^T}{\sum_t r_i^t}$$

Substituting into (A)

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

$$= -\frac{1}{2}\log |S_i| - \frac{1}{2}\left(\boldsymbol{x}^T S_i^{-1} \boldsymbol{x} - 2\boldsymbol{x}^T S_i^{-1} \boldsymbol{m}_i + \boldsymbol{m}_i^T S_i^{-1} \boldsymbol{m}_i\right)$$

$$+ \log \hat{P}(C_i)$$

$$= -\frac{1}{2}\boldsymbol{x}^T S_i^{-1}\boldsymbol{x} + (S_i^{-1}\boldsymbol{m}_i)^T\boldsymbol{x} - \frac{1}{2}\boldsymbol{m}_i^T S_i^{-1}\boldsymbol{m}_i - \frac{1}{2}\log|S_i| + \log\hat{P}(C_i) \qquad \text{-----} \ \text{(B)}$$

i) Quadratic discriminant:

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

where $W_i = -\frac{1}{2}S_i^{-1}, \quad \boldsymbol{w}_i = S_i^{-1}\boldsymbol{m}_i$

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

The number of parameters to be estimated is $K \cdot d$

for means $\boldsymbol{m}_i = (m_1, m_2, \cdots, m_d)$ and $Kd(d+1)/2$

for covariance matrices $S_i = [s_{mn}]_{d\times d}, \ i = 1, \cdots, K.$

- Share common sample covariance, i.e., $\forall i \;\; S_i = S$, and ignore $-\frac{1}{2}\log |S|$. (B) is reduced to

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

$$= -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{m}_i)^T S^{-1}(\boldsymbol{x} - \boldsymbol{m}_i) + \log \hat{P}(C_i) \quad \text{-- (C)}$$

Ignoring quadratic term $\boldsymbol{x}^T S^{-1}\boldsymbol{x}$, (C) reduces to

$$g_i(\boldsymbol{x}) = \boldsymbol{x}^T S^{-1}\boldsymbol{m}_i - \frac{1}{2}\boldsymbol{m}_i^T S^{-1}\boldsymbol{m}_i + \log \hat{P}(C_i) \quad \text{---- (D)}$$

ii) Linear discriminant: $g_i(\boldsymbol{x}) = \boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}$, where

$$\boldsymbol{w}_i = S^{-1}\boldsymbol{m}_i,\ \ w_{i0} = -\frac{1}{2}\boldsymbol{m}_i^T S^{-1}\boldsymbol{m}_i + \log \hat{P}(C_i)$$

The numbers of parameters to be estimated: $K \cdot d$

for means and $d(d+1)/2$ for covariance matrices.

- Assuming off-diagonals of S to be 0,

$$S = \begin{bmatrix} s_1^2 & 0 & \cdot & 0 \\ 0 & s_2^2 & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & 0 & s_d^2 \end{bmatrix} \text{ and } S^{-1} = \begin{bmatrix} 1/s_1^2 & 0 & \cdot & 0 \\ 0 & 1/s_2^2 & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & 0 & 1/s_d^2 \end{bmatrix}$$

Substitute into $-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^T S^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)$

$$=-\frac{1}{2}(x_1-m_{1i},x_2-m_{2i},\cdots,x_d-m_{di}) \cdot \begin{bmatrix} 1/s_1^2 & 0 & \cdot & 0 \\ 0 & 1/s_2^2 & \cdot & \cdot \\ \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & 0 & 1/s_d^2 \end{bmatrix}$$

$$\cdot \begin{pmatrix} x_1-m_{1i} \\ x_2-m_{2i} \\ \cdot \\ x_d-m_{di} \end{pmatrix} = -\frac{1}{2}\sum_{j=1}^{d}\left(\frac{x_j-m_{ij}}{s_j}\right)^2$$ Substitute this into (C)

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

The number of parameters to be estimated is $K \cdot d$ for means and d for covariance matrices.

iii) Naive Bayes' classifier

- Assuming all variances to be equal, i.e., $\forall j \;\; s_j = s$

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

Assuming equal priors $\hat{P}(C_i)$ and ignore s,

$$g_i(\boldsymbol{x}) = -\sum_{j=1}^{d}\left(x_j^t - m_{ij}\right)^2 = -\left\|\boldsymbol{x} - \boldsymbol{m}_i\right\|^2$$

The number of parameters to be estimated is $K \cdot d$ for means

iv) Nearest mean classifier

- $$g_i(\boldsymbol{x}) = -\|\boldsymbol{x} - \boldsymbol{m}_i\|^2 = -(\boldsymbol{x} - \boldsymbol{m}_i)^T(\boldsymbol{x} - \boldsymbol{m}_i)$$

  $$= -(\boldsymbol{x}\boldsymbol{x}^T - 2\boldsymbol{m}_i^T\boldsymbol{x} + \boldsymbol{m}_i^T\boldsymbol{m}_i)$$

  Ignore the common term $\boldsymbol{x}\boldsymbol{x}^T$

  $$g_i(\boldsymbol{x}) = \boldsymbol{m}_i^T\boldsymbol{x} - \frac{1}{2}\boldsymbol{m}_i^T\boldsymbol{m}_i = \boldsymbol{m}_i^T\boldsymbol{x} - \frac{1}{2}\|\boldsymbol{m}_i\|^2$$

  Assuming equal $\|\boldsymbol{m}_i\|$,

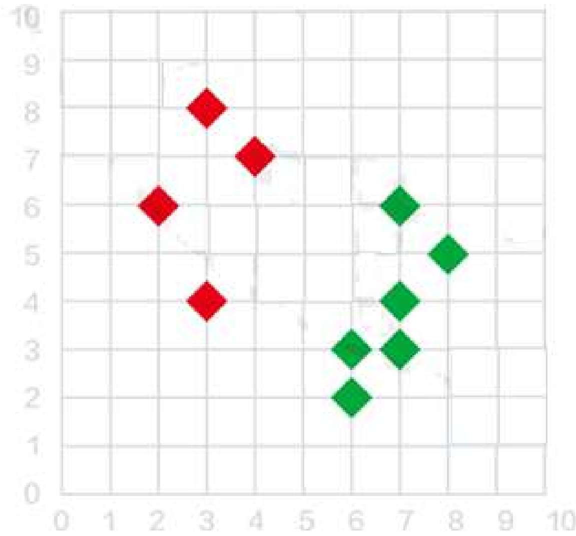
  The number of parameters to be estimated is $K \cdot d$ for means

Example: 2 classes

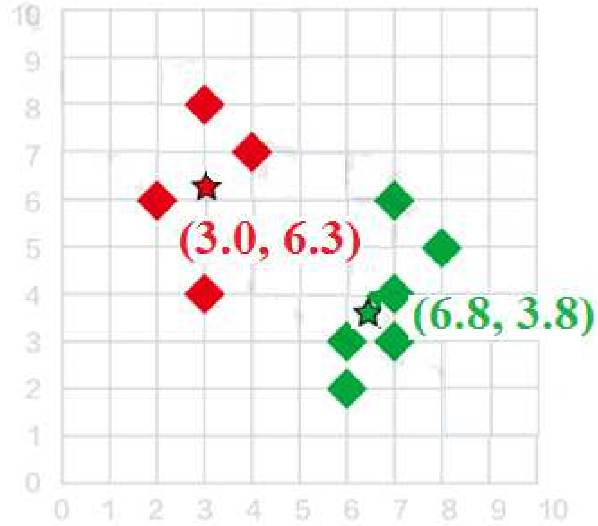
Given two classes of data points marked by red and green colors, respectively:

(2,6), (3,4), (3,8), (4,7)

(6,2), (6,3), (7,3), (7,4), (7,6), (8,5)



The mean points of these two groups of data points are (3.0, 6.3) and (6.8, 3.8), respectively.

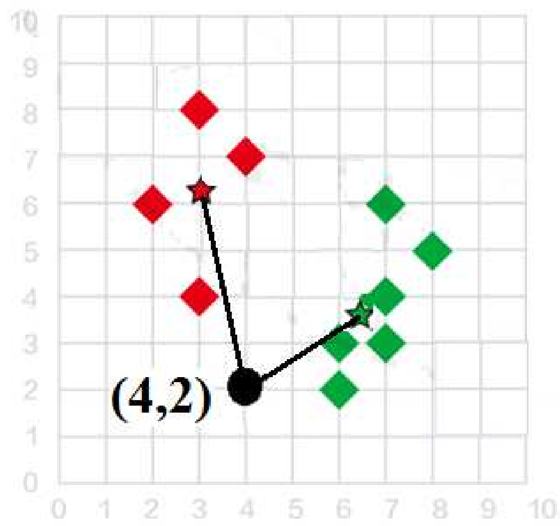


The new point (4,2) has distances of

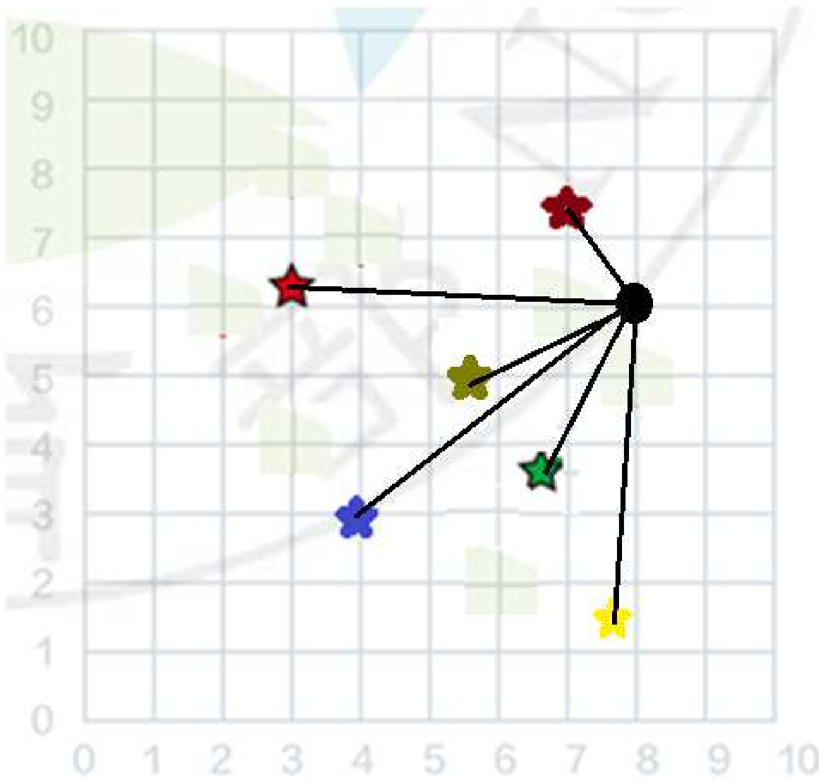
$$\sqrt{(4.0-3.0)^2+(2.0-6.3)^2}=4.4 \text{ (to the red center)}$$

$$\sqrt{(4.0-6.8)^2+(2.0-3.8)^2}=3.4 \text{ (to the green center).}$$

(4,2) is classified as belonging to the green class.



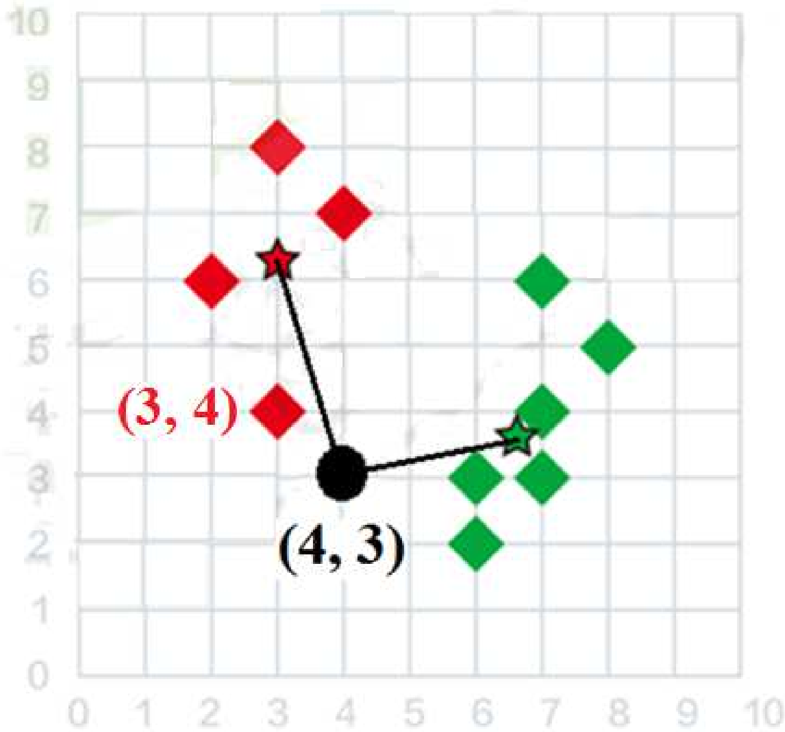
- N ($N > 2$) Classes



The black point is classified as belonging to the brown class.

K Nearest Neighbor (K-NN) Classifier

According to SD classifier, point (4,3) will be classified as the **green** class. However, **(4,3)** has the nearest red neighbor **(3,4).** It is desirable to be classified as the **red** class.



The K-NN classifier assigns a point $\boldsymbol{x}$ the label most frequently present among its K nearest neighbors.

Let $N_{\boldsymbol{x}} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_K\}$: the K nearest neighbors of point $\boldsymbol{x}$

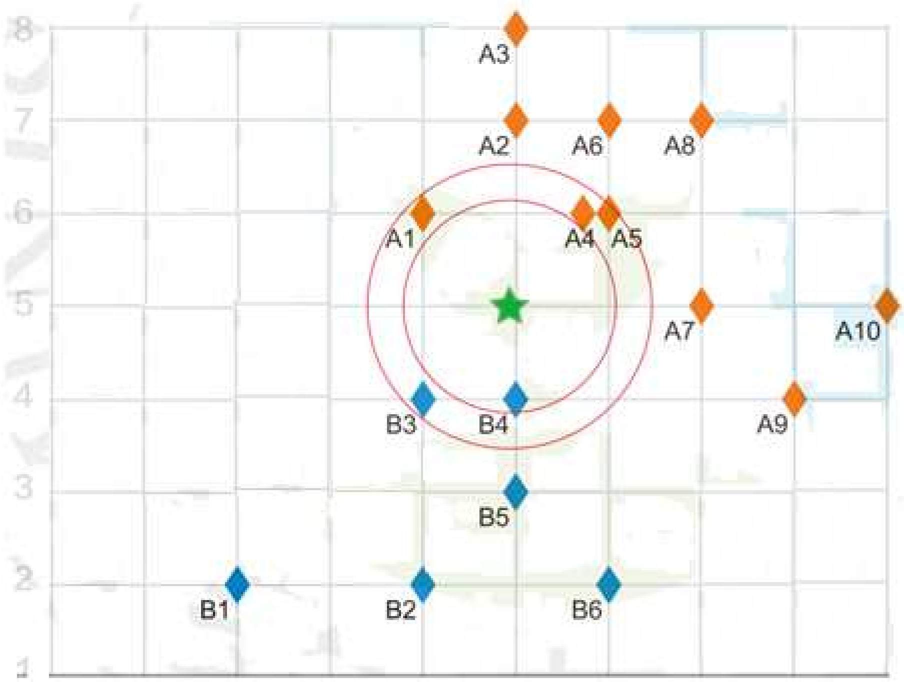
n_i: the number of points in $N_{\boldsymbol{x}}$, which belong to class C_i, $\sum_{i=1}^{c} n_i = K$.

Decision rule:

Assign $\boldsymbol{x}$ to C_k if $\max_{1 \le i \le c} n_i = n_k$

Example: 5-NN classifier

The **green** point has two **blue** neighbors and three **orange** neighbors. The **green** point is classified as belonging to the **orange** class.



v) Inner product classifier: $g_i(\boldsymbol{x}) = \boldsymbol{m}_i^T \boldsymbol{x}$

Tuning Complexity

Assumption	Covariance matrix	No of parameters
Different, Hyperellipsoidal	$\boldsymbol{S}_i$	$K\,d(d+1)/2$
Shared, Hyperellipsoidal	$\boldsymbol{S}_i = \boldsymbol{S}$	$d(d+1)/2$
Shared, Axis-aligned	$\boldsymbol{S}_i = \boldsymbol{S}$, with $s_{ij} = 0$	d
Shared, Hyperspheric	$\boldsymbol{S}_i = \boldsymbol{S} = s^2\boldsymbol{I}$	1

5.6 Discrete Attributes

- Binary pattern: $\boldsymbol{x}=(x_1, x_2, \cdots, x_d)$, $x_j \in \{0, 1\}$

  Let $p_{ij} = p\left(x_j = 1 | C_i\right)$. $p\left(x_j | C_i\right) = p_{ij}^{x_j}\left(1-p_{ij}\right)^{\left(1-x_j\right)}$

  If x_j 's are independent,

  $$p\left(\boldsymbol{x}|C_i\right)=\prod_{j=1}^{d} p\left(x_j|C_i\right)=\prod_{j=1}^{d} p_{ij}^{x_j}\left(1-p_{ij}\right)^{\left(1-x_j\right)}$$

  The discriminant function:

  $$\begin{aligned} g_i(\boldsymbol{x}) &= \log p\left(\boldsymbol{x}|C_i\right)+\log P\left(C_i\right) \\ &= \log \prod_{j=1}^{d} p_{ij}^{x_j}\left(1-p_{ij}\right)^{\left(1-x_j\right)}+\log P\left(C_i\right) \\ &= \sum_j \left[x_j \log p_{ij}+\left(1-x_j\right) \log\left(1-p_{ij}\right)\right]+\log P\left(C_i\right) \end{aligned}$$

- Multinomial pattern: $\boldsymbol{x} = (x_1, x_2, \cdots, x_d)$,

Define $z_{jk} = \begin{cases} 1 & \text{if } x_j = v_k \\ 0 & \text{otherwise} \end{cases}$ $\quad x_j \in \{v_1, v_2, \cdots, v_{n_j}\}$

Let p_{ijk} : the probability that $x_j \in C_i$ takes value v_k,

i.e., $p_{ijk} = p\left(z_{jk} = 1 | C_i\right)$. $p\left(x_j | C_i\right) = \prod_{k=1}^{n_j} p_{ijk}^{z_{jk}}$

If x_j 's are independent, $p(\boldsymbol{x}|C_i) = \prod_{j=1}^{d} p\left(x_j|C_i\right) = \prod_{j=1}^{d}\prod_{k=1}^{n_j} p_{ijk}^{z_{jk}}$

The discriminant function:

$$g_i(\boldsymbol{x}) = \log p(\boldsymbol{x}|C_i) + \log P(C_i) = \log \prod_{j=1}^{d}\prod_{k=1}^{n_j} p_{ijk}^{z_{jk}} + \log P(C_i)$$

$$= \sum_j \sum_k z_{jk} \log p_{ijk} + \log P(C_i)$$

5.7 Generalizing the linear model:

i) A polynomial model

$$f(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_n x^n$$

can lead to a linear multivariate model by

letting $x = x_1,\ x^2 = x_2,\ \cdots,\ x^n = x_n$

$$f(x) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

Example: Find a quadratic model of two variables x_1 and x_2, i.e.,

$$f(x_1, x_2) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2$$

Let $z_1 = x_1,\ z_2 = x_2,\ z_3 = x_1 x_2,\ z_4 = x_1^2,\ z_5 = x_2^2$

$\Rightarrow$

$$f(x_1, x_2) = w_0 + w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_4 + w_5 z_5$$

Use linear regression to learn $w_i,\ i = 1, \cdots, 5.$

ii) Let $\sin x = x_1,\ \exp x^2 = x_2,\ \cdots\cdots$

A nonlinear model can lead to a linear multivariate model.

Appendix A - Multidimensional Scaling (MDS)

Given pairwise distances d_{ij} of a set of points, MDS places these points in a low space s.t. the Euclidean distances between them is as close as possible to d_{ij}.

Let $X = \{\boldsymbol{x}^t\}_{t=1}^N$: a sample, where $\boldsymbol{x}^t \in R^d$

Two points: *r* and *s*, their squared Euclidean distance

$$d_{rs}^2 = \left\|\boldsymbol{x}^r - \boldsymbol{x}^s\right\|^2 = \sum_{j=1}^{d}(x_j^r - x_j^s)^2$$

$$= \sum_{j=1}^{d}(x_j^r)^2 - 2\sum_{j=1}^{d} x_j^r x_j^s + \sum_{j=1}^{d}(x_j^s)^2 = b_{rr} - 2b_{rs} + b_{ss} \text{ -- (A)}$$

where $b_{rr} = \sum_{j=1}^{d}(x_j^r)^2,\ b_{rs} = \sum_{j=1}^{d} x_j^r x_j^s,\quad b_{ss} = \sum_{j=1}^{d}(x_j^s)^2$

$$\sum_{r=1}^{N} d_{rs}^2 = \sum_{r=1}^{N} b_{rr} + \sum_{r=1}^{N} b_{ss} - 2\sum_{r=1}^{N} b_{rs} = \sum_{r=1}^{N} b_{rr} + Nb_{ss} - 2\sum_{r=1}^{N} b_{rs}$$

Let $T = \sum_{r=1}^{N} b_{rr}$. $\sum_{r=1}^{N} d_{rs}^2 = T + Nb_{ss} - 2\sum_{r=1}^{N} b_{rs}$ --- (B)

$$\because \sum_{r=1}^{N} b_{rs} = \sum_{r=1}^{N}\sum_{j=1}^{d} x_j^r x_j^s = \sum_{j=1}^{d} x_j^1 x_j^s + \sum_{j=1}^{d} x_j^2 x_j^s + \cdots + \sum_{j=1}^{d} x_j^N x_j^s$$

$$= x_1^1 x_1^s + x_2^1 x_2^s + \cdots + x_d^1 x_d^s + x_1^2 x_1^s + x_2^2 x_2^s + \cdots + x_d^2 x_d^s$$

$$+ \cdots\cdots\cdots\cdots\cdots\cdots\cdots + x_1^N x_1^s + x_2^N x_2^s + \cdots + x_d^N x_d^s$$

$$= x_1^s \sum_{r=1}^{N} x_1^r + x_2^s \sum_{r=1}^{N} x_2^r + \cdots + x_d^s \sum_{r=1}^{N} x_d^r$$

Suppose data have been centered at the origin so that

$$\sum_{r=1}^{N} x_j^r = 0, \ \ j = 1, \cdots, d. \ \ \therefore \sum_{r=1}^{N} b_{rs} = 0, \ \ \sum_{r=1}^{N} d_{rs}^2 = T + Nb_{ss} \text{ -- (C)}$$

Likewise, $\sum_{s=1}^{N} d_{rs}^2 = T + Nb_{rr}$ --- (D)

$$\sum_{r=1}^{N}\sum_{s=1}^{N} d_{rs}^2 = \sum_{r=1}^{N}(T + Nb_{rr}) = NT + N\sum_{r=1}^{N} b_{rr} = NT + NT = 2NT \text{ - (E)}$$

From (C) $\sum_{r} d_{rs}^2 = T + Nb_{ss}, \ \ b_{ss} = \frac{1}{N}\left(\sum_{r} d_{rs}^2 - T\right)$

From (D) $\sum_{s} d_{rs}^2 = T + Nb_{rr}, \ \ b_{rr} = \frac{1}{N}\left(\sum_{s} d_{rs}^2 - T\right)$

From (A) $d_{rs}^2 = b_{rr} - 2b_{rs} + b_{ss}$,

$$b_{rs} = \frac{1}{2}\left(b_{rr} + b_{ss} - d_{rs}^2\right) = \frac{1}{2}\left(\frac{1}{N}\sum_s d_{rs}^2 + \frac{1}{N}\sum_r d_{rs}^2 - \frac{2T}{N} - d_{rs}^2\right) \quad \text{---- (F)}$$

From (E) $\displaystyle \frac{1}{N^2}\sum_r\sum_s d_{rs}^2 = \frac{2NT}{N^2} = \frac{2T}{N}$

$$b_{rs} = \frac{1}{2}\left(\frac{1}{N}\sum_s d_{rs}^2 + \frac{1}{N}\sum_r d_{rs}^2 - \frac{1}{N^2}\sum_r\sum_s d_{rs}^2 - d_{rs}^2\right)$$

Let $\displaystyle d_{r\bullet}^2 = \frac{1}{N}\sum_s d_{rs}^2,\ \ d_{\bullet s}^2 = \frac{1}{N}\sum_r d_{rs}^2,\ \ d_{\bullet\bullet}^2 = \frac{1}{N^2}\sum_r\sum_s d_{rs}^2$

$$b_{rs} = \frac{1}{2}(d_{r\bullet}^2 + d_{\bullet s}^2 - d_{\bullet\bullet}^2 - d_{rs}^2)$$

$d_{r\bullet}^2,\ d_{\bullet s}^2,\ d_{\bullet\bullet}^2,\ d_{rs}^2$ can all be calculated from the given

$D=[d_{rs}]$. $b_{rs}=\frac{1}{2}(d_{r\bullet}^2+d_{\bullet s}^2-d_{\bullet\bullet}^2-d_{rs}^2)$ is known.

From (A), $b_{rs}=\sum_{j=1}^{d} x_j^r x_j^s = \left(\boldsymbol{x}^r\right)^T \boldsymbol{x}^s,\ \ r,s=1,\cdots,N.$

In matrix form, $B=\left[b_{rs}\right]=XX^T$

Spectral decomposition of $B=E\Lambda E^T=E\Lambda^{1/2}(E\Lambda^{1/2})^T$

where $E=[\boldsymbol{e}_1\cdot\ \cdot\ \cdot\ \boldsymbol{e}_N]^T,\ \Lambda=\mathrm{diag}\left(\lambda_1\cdot\ \cdot\ \cdot\lambda_N\right)$

$\lambda_i,\ \boldsymbol{e}_i:$ eigenvalues and eigenvectors of B

$$B=XX^T=E\Lambda^{1/2}(E\Lambda^{1/2})^T$$

Decide a dimensionality $k\ (< d)$ based on λ_i.

Let $E_k = [\boldsymbol{e}_1 \cdot \ \cdot\ \boldsymbol{e}_k]^T,\ \Lambda_k = \mathrm{diag}\left(\lambda_1 \cdot \ \cdot \lambda_k\right).$

$$E_k \Lambda_k^{1/2} (E_k \Lambda_k^{1/2})^T = ZZ^T.$$

The new coordinates $\boldsymbol{z}^T = (z_1, z_2, \cdot\cdot, z_k)^T$ of point

$\boldsymbol{x}^T = (x_1, x_2, \cdot\cdot, x_d)^T$ are given by

$$z_j^t = \sqrt{\lambda_j} e_j^t,\ \ j = 1, \cdots, k;\ t = 1, \cdots, N.$$