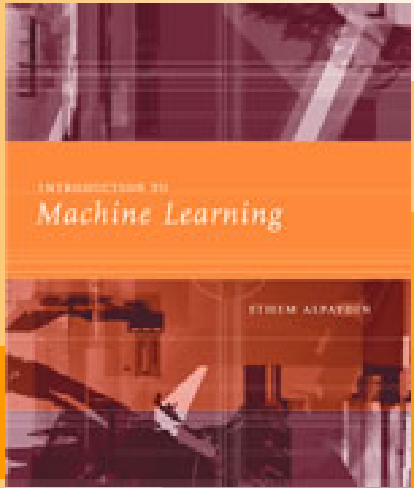


Lecture Slides for

INTRODUCTION TO

Machine Learning

ETHEM ALPAYDIN

alpaydin@boun.edu.tr
http://www.cmpe.boun.edu.tr/~ethem/i2ml

CHAPTER 5:

Multivariate Methods

Multivariate Data

- Multiple measurements (sensors)
- d inputs/features/attributes: d-variate
- N instances/observations/examples

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

Multivariate Parameters

$$\text{Mean}: E[\boldsymbol{x}] = \boldsymbol{\mu} = [\mu_1, \ldots, \mu_d]^T$$

$$\text{Covariance}: \sigma_{ij} \equiv \mathrm{Cov}(X_i, X_j)$$

$$\text{Correlation}: \mathrm{Corr}(X_i, X_j) \equiv \rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$$

$$\Sigma \equiv \mathrm{Cov}(\boldsymbol{X}) = E\left[(\boldsymbol{X} - \mu)(\boldsymbol{X} - \mu)^T\right] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2d} \\ \vdots & & & \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_d^2 \end{bmatrix}$$

Parameter Estimation

Sample mean $\mathbf{m}$: $m_i = \frac{\sum_{t=1}^{N} x_i^t}{N}, i = 1,\ldots,d$

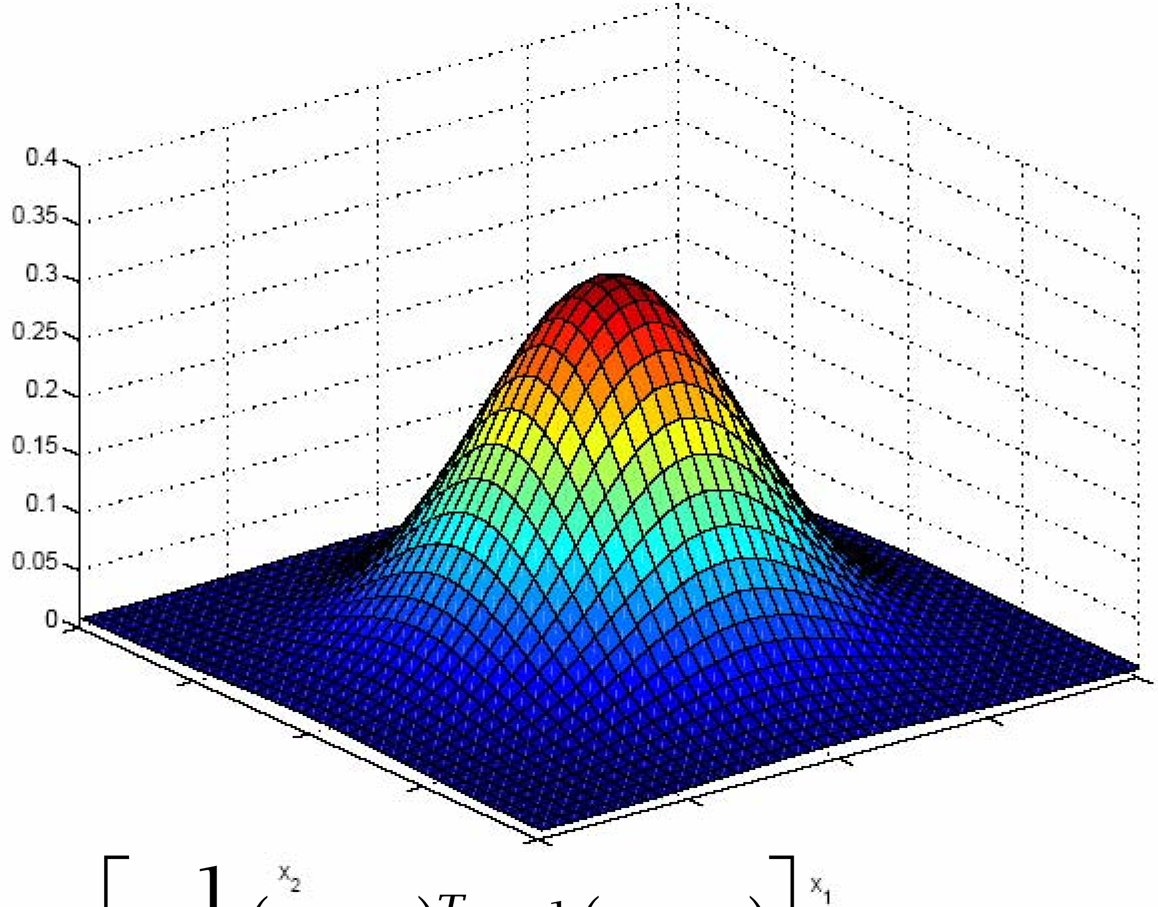
Covariance matrix $\mathbf{S}$: $s_{ij} = \frac{\sum_{t=1}^{N} \left(x_i^t - m_i\right)\left(x_j^t - m_j\right)}{N}$

Correlation matrix $\mathbf{R}$: $r_{ij} = \frac{s_{ij}}{s_i s_j}$

Estimation of Missing Values

- What to do if certain instances have missing attributes?
- Ignore those instances: not a good idea if the sample is small
- Use ‘missing’ as an attribute: may give information
- Imputation: Fill in the missing value
  - Mean imputation: Use the most likely value (e.g., mean)
  - Imputation by regression: Predict based on other attributes

Multivariate Normal Distribution



$$\mathbf{x} \sim \mathcal{N}_d(\boldsymbol{\mu}, \Sigma)$$

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

Multivariate Normal Distribution

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

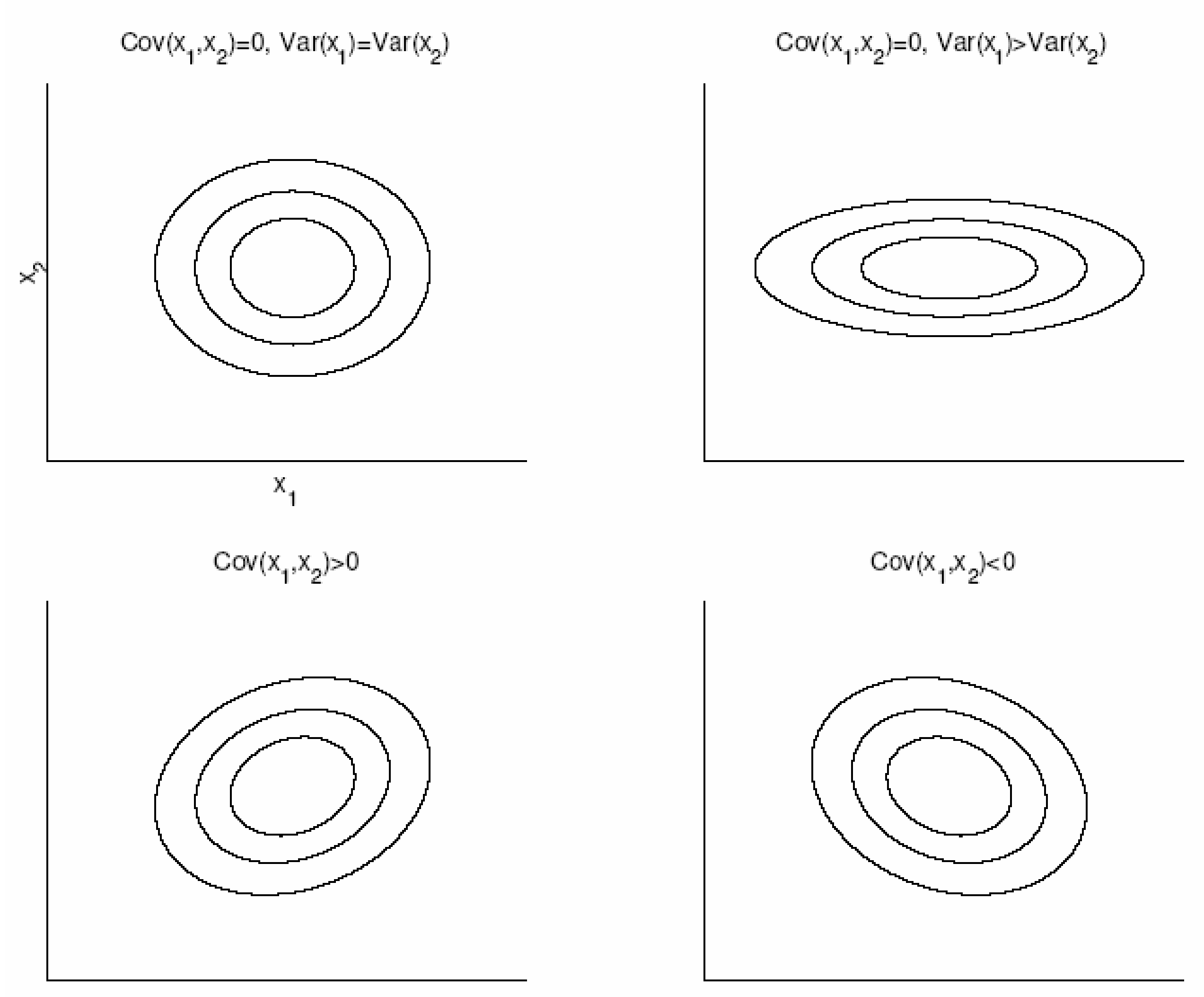
  measures the distance from $\boldsymbol{x}$ to $\boldsymbol{\mu}$ in terms of $\boldsymbol{\Sigma}$ (normalizes for difference in variances and correlations)
- Bivariate: $d = 2$

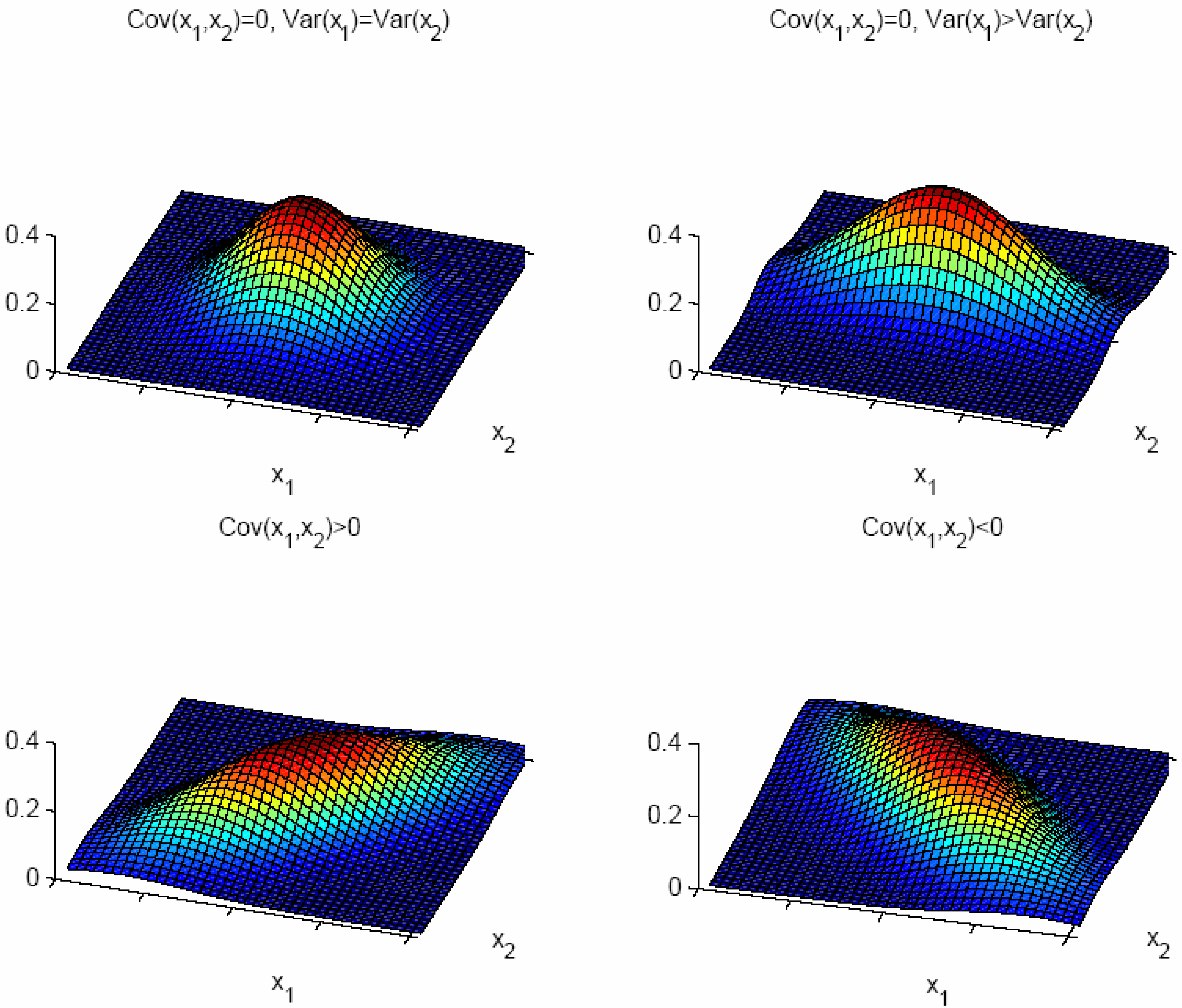
$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$$

$$p(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left[-\frac{1}{2(1-\rho^2)}\left(z_1^2 - 2\rho z_1 z_2 + z_2^2\right)\right]$$

$$z_i = (x_i - \mu_i)/\sigma_i$$

Bivariate Normal




Cov(x1,x2)=0, Var(x1)=Var(x2)
Cov(x1,x2)=0, Var(x1)>Var(x2)
0.4
0.2
0
x1
x2
Cov(x1,x2)>0
Cov(x1,x2)<0

Independent Inputs: Naive Bayes

- If x_i are independent, offdiagonals of $\mathbf{\Sigma}$ are 0, Mahalanobis distance reduces to weighted (by $1/\sigma_i$) Euclidean distance:

$$p(\boldsymbol{x}) = \prod_{i=1}^{d} p_i(x_i) = \frac{1}{(2\pi)^{d/2}\prod_{i=1}^{d}\sigma_i}\exp\left[-\frac{1}{2}\sum_{i=1}^{d}\left(\frac{x_i-\mu_i}{\sigma_i}\right)^2\right]$$

- If variances are also equal, reduces to Euclidean distance

Parametric Classification

- If $p(\boldsymbol{x} \mid C_i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

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

- Discriminant functions are

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

$$= -\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_i| - \frac{1}{2}(\boldsymbol{x} - \mu_i)^T \Sigma_i^{-1} (\boldsymbol{x} - \mu_i) + \log P(C_i)$$

Estimation of Parameters

$$\hat{P}(C_i) = \frac{\sum_t r_i^t}{N}$$

$$\boldsymbol{m}_i = \frac{\sum_t r_i^t \boldsymbol{x}^t}{\sum_t r_i^t}$$

$$\mathbf{S}_i = \frac{\sum_t r_i^t (\boldsymbol{x}^t - \boldsymbol{m}_i)(\boldsymbol{x}^t - \boldsymbol{m}_i)^T}{\sum_t r_i^t}$$

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

Different $\mathbf{S}_i$

- Quadratic discriminant

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

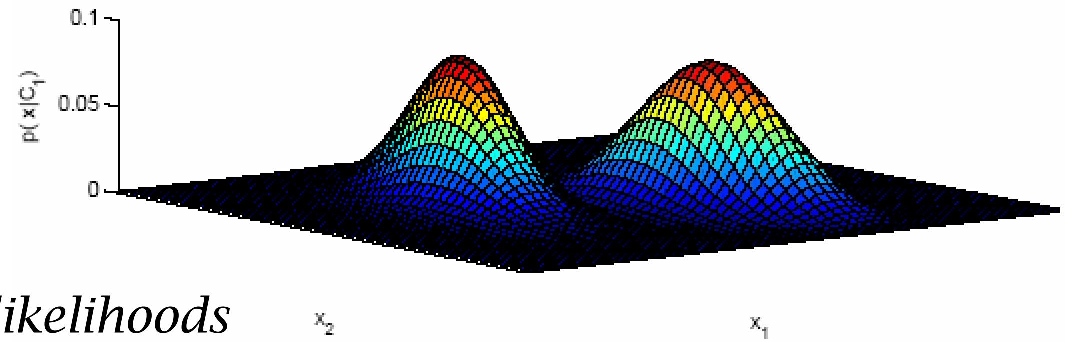
$$= \boldsymbol{x}^T\mathbf{W}_i\boldsymbol{x} + \boldsymbol{w}_i^T\boldsymbol{x} + w_{i0}$$

where

$$\mathbf{W}_i = -\frac{1}{2}\mathbf{S}_i^{-1}$$

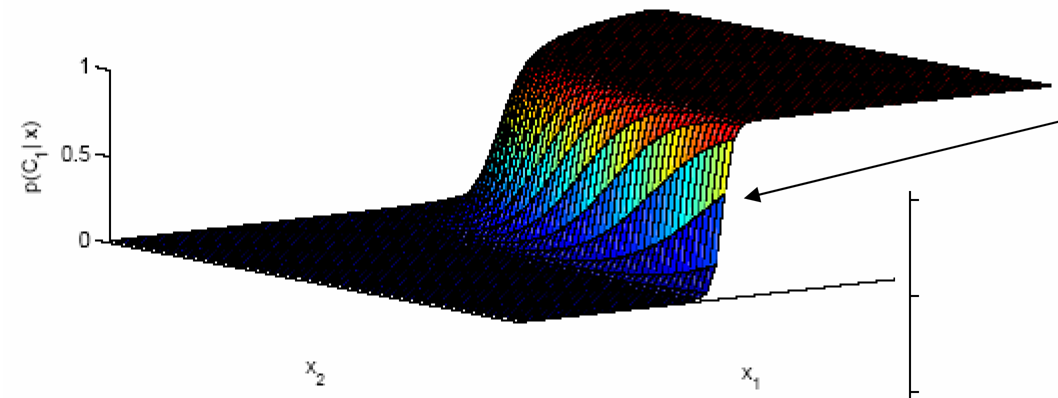
$$\boldsymbol{w}_i = \mathbf{S}_i^{-1}\boldsymbol{m}_i$$

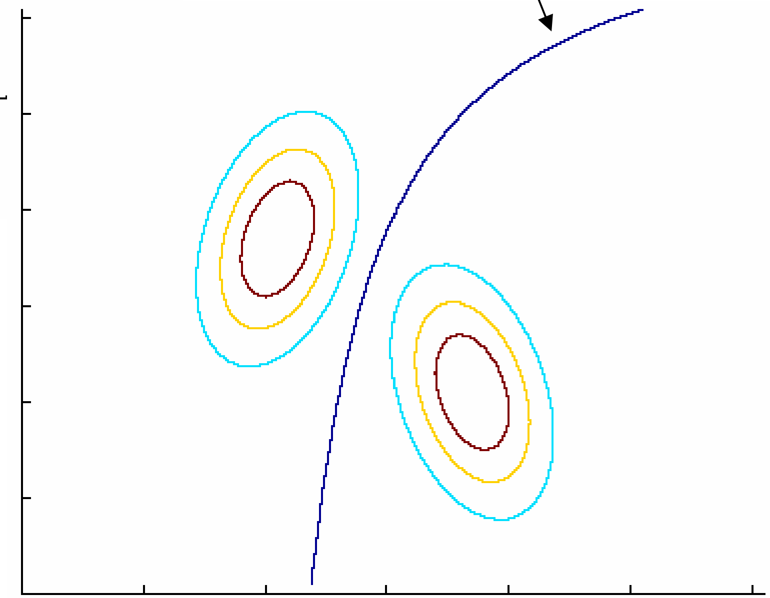
$$w_{i0} = -\frac{1}{2}\boldsymbol{m}_i^T\mathbf{S}_i^{-1}\boldsymbol{m}_i - \frac{1}{2}\log|\mathbf{S}_i| + \log\hat{P}(C_i)$$



likelihoods

discriminant:
$P\,(C_1|\boldsymbol{x}\,) = 0.5$





posterior for C_1

Common Covariance Matrix **S**

- Shared common sample covariance **S**

$$\mathbf{S} = \sum_i \hat{P}(C_i)\mathbf{S}_i$$

- Discriminant reduces to

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

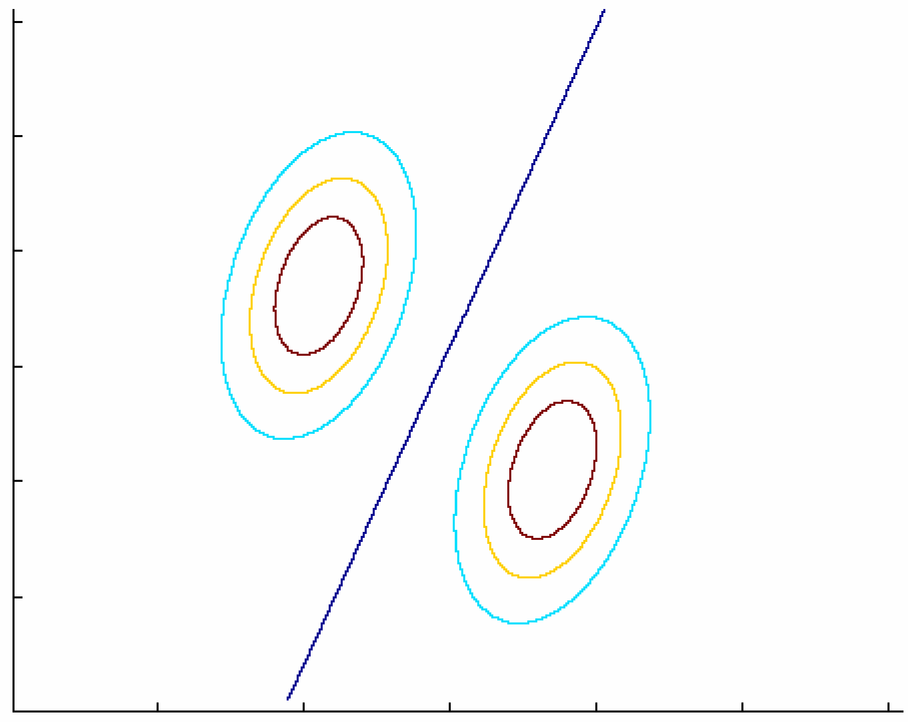
which is a linear discriminant

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

where

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

Common Covariance Matrix S



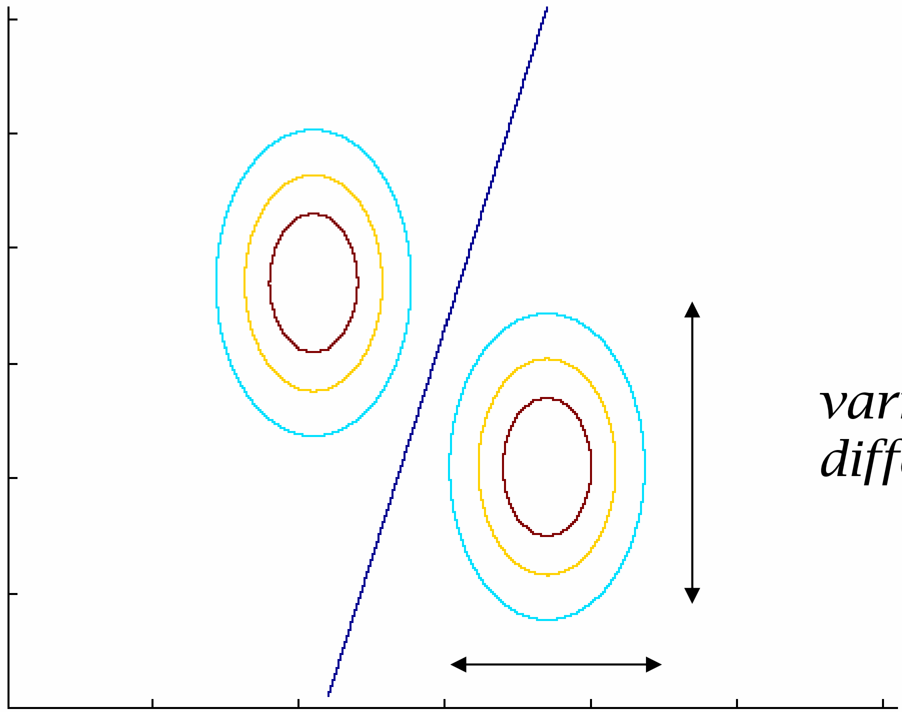
Diagonal S

- When x_j $j = 1,..d$, are independent, Σ is diagonal
  $p(\mathbf{x}|C_i) = \Pi_j\, p(x_j|C_i)$ (Naive Bayes' assumption)

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

Classify based on weighted Euclidean distance (in s_j units) to the nearest mean

Diagonal S



variances may be different

Diagonal **S**, *equal variances*

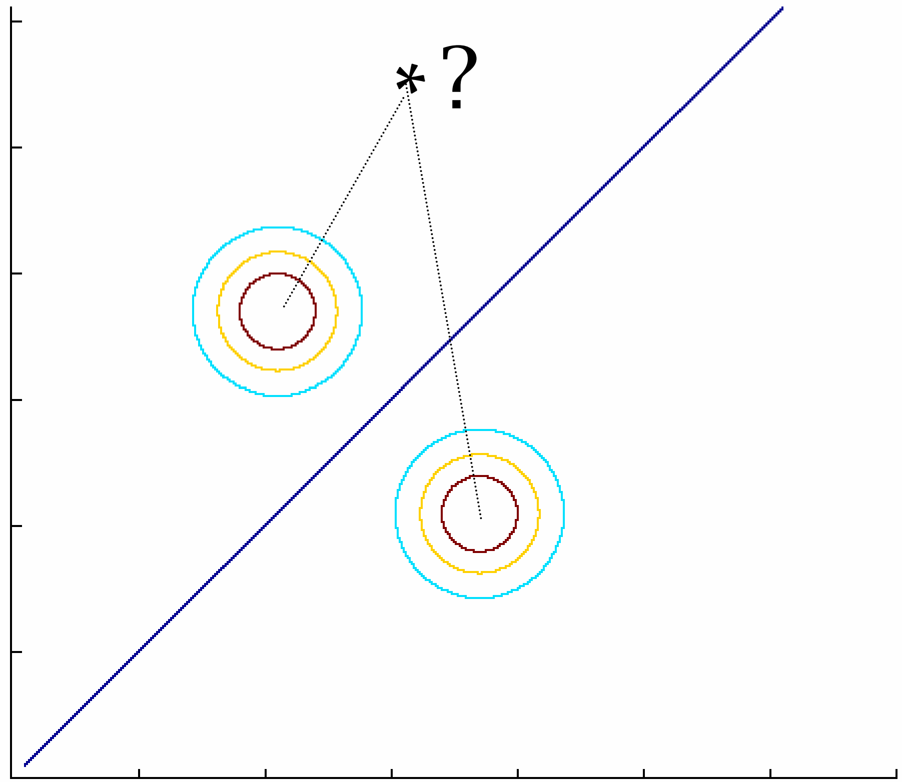
- Nearest mean classifier: Classify based on Euclidean distance to the nearest mean

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

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

- Each mean can be considered a prototype or template and this is template matching

Diagonal **S**, equal variances



Model Selection

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

- As we increase complexity (less restricted **S**), bias decreases and variance increases
- Assume simple models (allow some bias) to control variance (regularization)

Discrete Features

- Binary features: $p_{ij} \equiv p(x_j = 1 \mid C_i)$

  if x_j are independent (Naive Bayes')

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

  the discriminant is linear

$$g_i(\boldsymbol{x}) = \log p(\boldsymbol{x} \mid C_i) + \log P(C_i)$$
$$= \sum_j \left[x_j \log p_{ij} + (1 - x_j) \log (1 - p_{ij}) \right] + \log P(C_i)$$

  Estimated parameters $\hat{p}_{ij} = \frac{\sum_t x_j^t r_i^t}{\sum_t r_i^t}$

Discrete Features

- Multinomial (1-of-n_j) features: $x_j \in \{v_1, v_2,\ldots, v_{n_j}\}$

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

if x_j are independent

$$p(\boldsymbol{x} \mid C_i) = \prod_{j=1}^{d} \prod_{k=1}^{n_j} p_{ijk}^{z_{jk}}$$

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

$$\hat{p}_{ijk} = \frac{\sum_t z_{jk}^t r_i^t}{\sum_t r_i^t}$$

Multivariate Regression

$$r^t = g(x^t \mid w_0, w_1, \ldots, w_d) + \varepsilon$$

- Multivariate linear model

$$w_0 + w_1 x_1^t + w_2 x_2^t + \cdots + w_d x_d^t$$

$$E(w_0, w_1, \ldots, w_d \mid \mathcal{X}) = \frac{1}{2} \sum_t \left[r^t - w_0 - w_1 x_1^t - \cdots - w_d x_d^t \right]^2$$

- Multivariate polynomial model:

  Define new higher-order variables

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

  and use the linear model in this new $\boldsymbol{z}$ space

  (basis functions, kernel trick, SVM: Chapter 10)