Lecture Slides for

INTRODUCTION TO

Machine Learning
2nd Edition

ETHEM ALPAYDIN
© The MIT Press, 2010

alpaydin@boun.edu.tr
http://www.cmpe.boun.edu.tr/~ethem/i2ml2e

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

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

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

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

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

Parameter Estimation

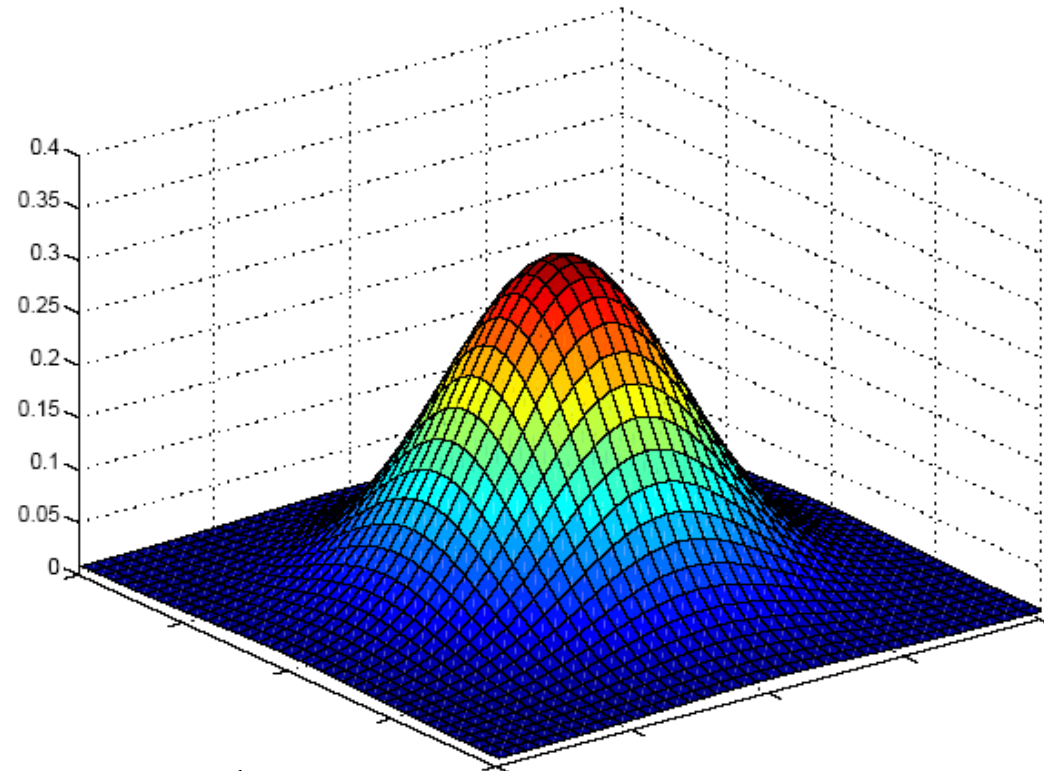Sample mean $\mathbf{m}$: $m_i = \dfrac{\sum_{t=1}^{N} x_i^t}{N}, i = 1, \ldots, d$

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

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

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

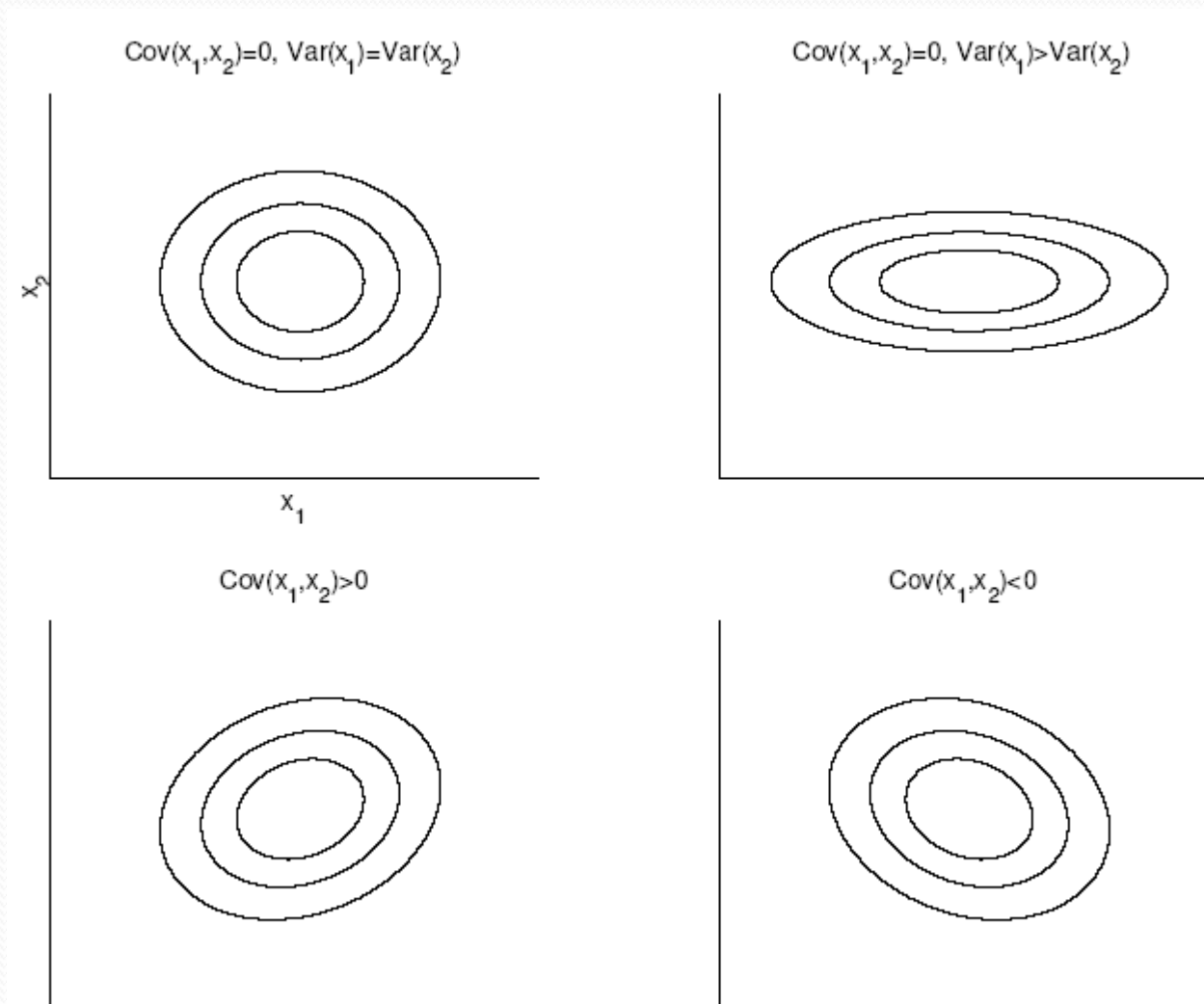- Mahalanobis distance: $(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})$

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

- Bivariate: $d = 2$

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

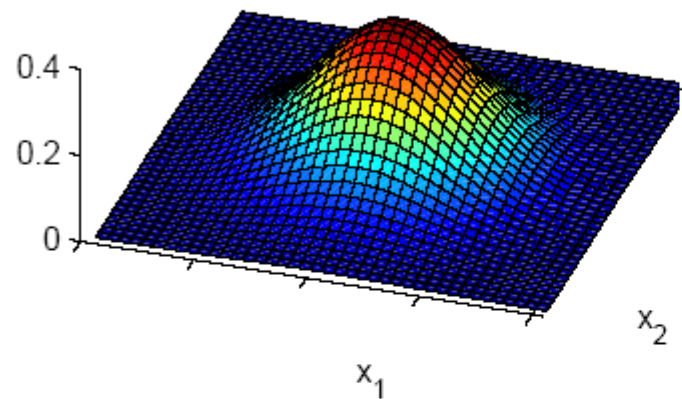$$p(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left[-\frac{1}{2(1-\rho^2)}\left(z_1^2 - 2\rho z_1 z_2 + z_2^2\right)\right]$$
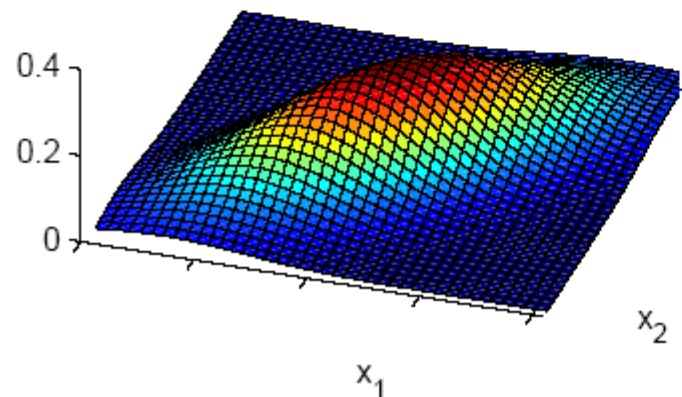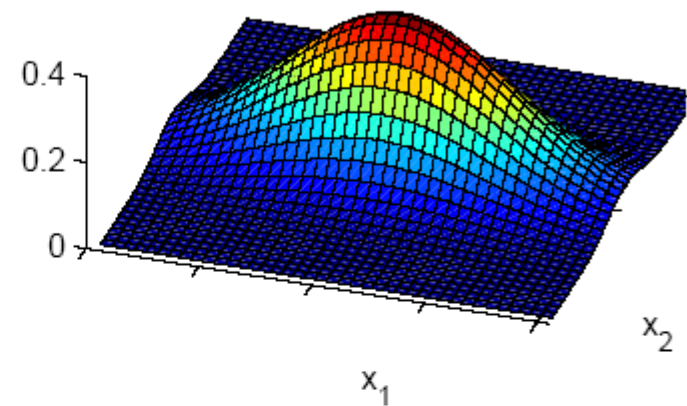
$$z_i = (x_i - \mu_i)/\sigma_i$$

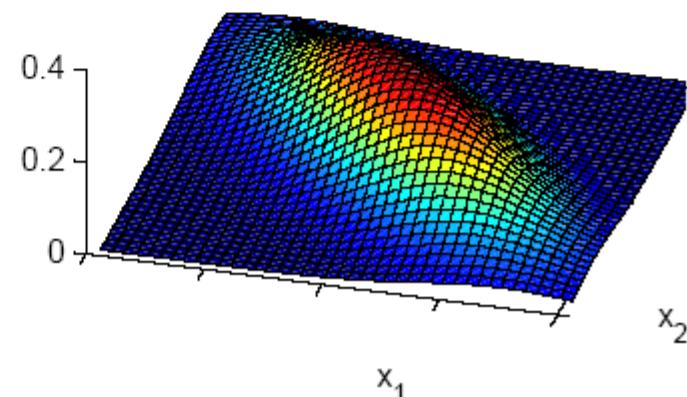# Bivariate Normal

$Cov(x_1, x_2) = 0, \ Var(x_1) = Var(x_2)$

$Cov(x_1, x_2) = 0, \ Var(x_1) > Var(x_2)$

$Cov(x_1, x_2) > 0$

$Cov(x_1, x_2) < 0$

Cov$(x_1,x_2)=0$, Var$(x_1)=$Var(x_2)

Cov$(x_1,x_2)=0$, Var$(x_1)>$Var(x_2)

Cov$(x_1,x_2)>0$

Cov$(x_1,x_2)<0$

Independent Inputs: Naive Bayes

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

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

- If variances are also equal, reduces to Euclidean distance

Parametric Classification

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

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

- Discriminant functions

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

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

Estimation of Parameters

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

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

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

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

Different $\mathbf{S}_i$

- Quadratic discriminant

$$g_i(\mathbf{x}) = -\frac{1}{2}\log|\mathbf{S}_i| - \frac{1}{2}\left(\mathbf{x}^T\mathbf{S}_i^{-1}\mathbf{x} - 2\mathbf{x}^T\mathbf{S}_i^{-1}\mathbf{m}_i + \mathbf{m}_i^T\mathbf{S}_i^{-1}\mathbf{m}_i\right) + \log\hat{P}(C_i)$$
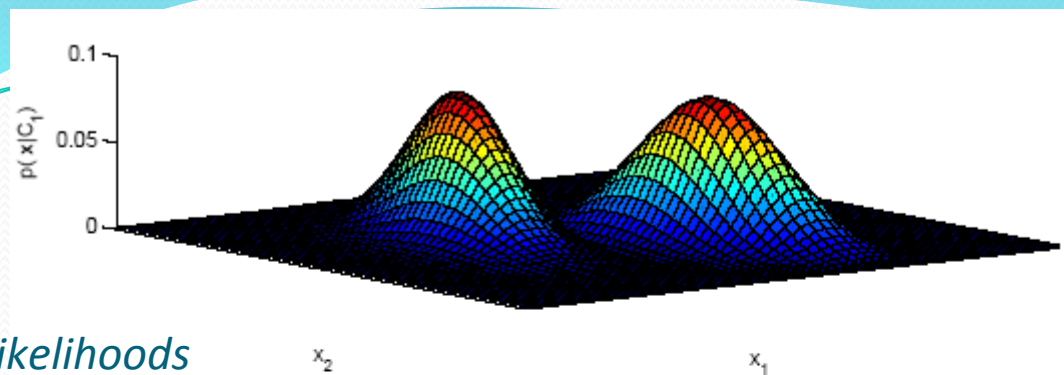
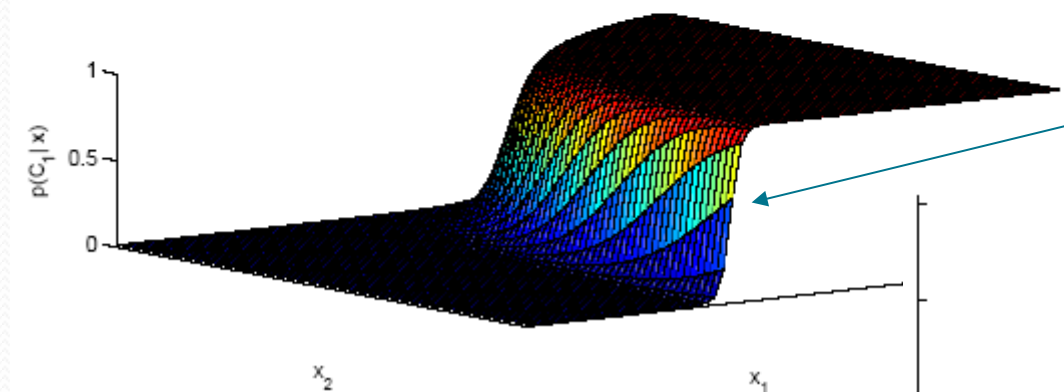$$= \mathbf{x}^T\mathbf{W}_i\mathbf{x} + \mathbf{w}_i^T\mathbf{x} + w_{i0}$$

where

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

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

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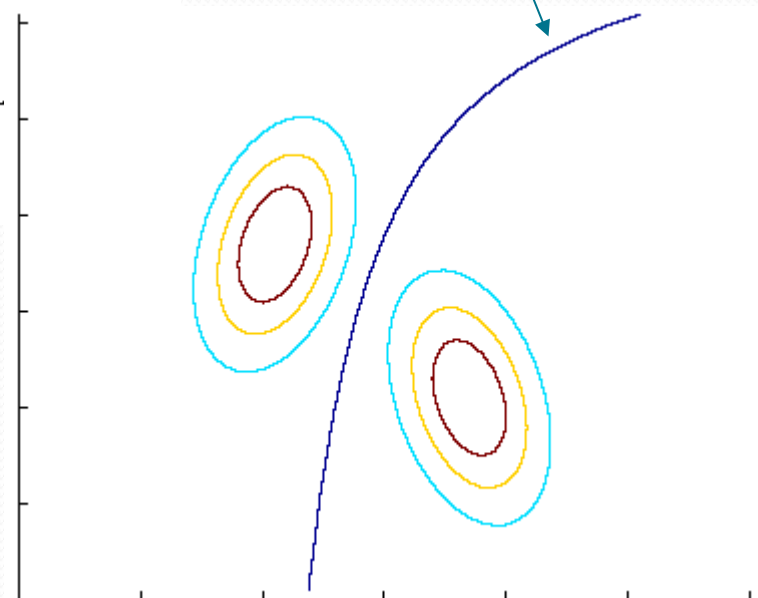
likelihoods

posterior for C_1

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

Common Covariance Matrix S

- Shared common sample covariance **S**

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

- Discriminant reduces to

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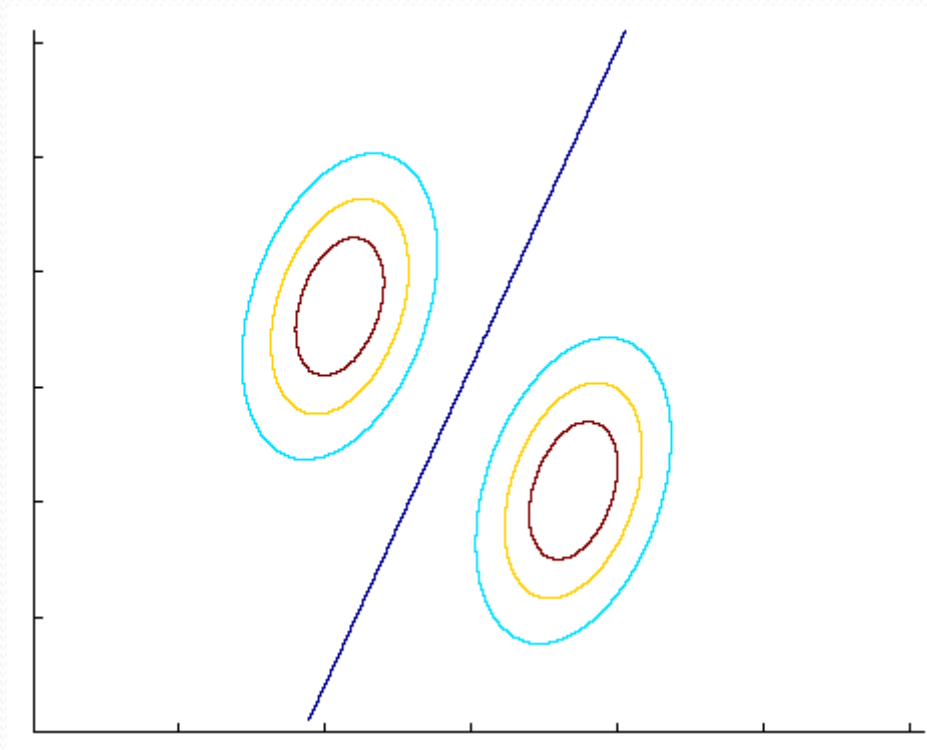
which is a linear discriminant

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

where

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

Common Covariance Matrix **S**
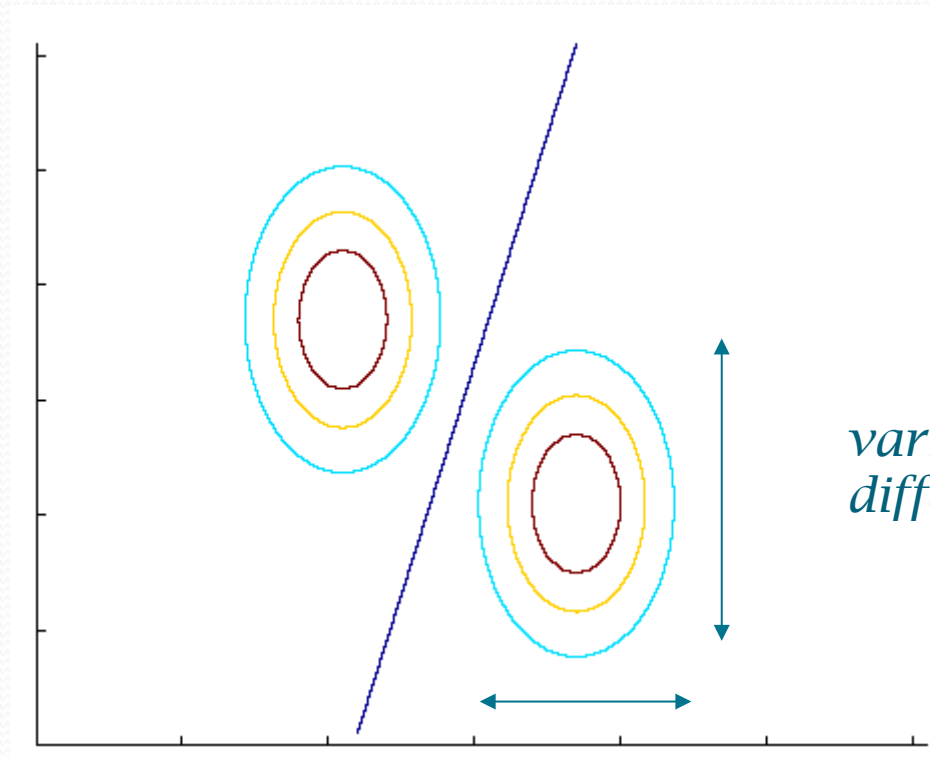
Diagonal **S**

- When x_j $j = 1,..d$, are independent, $\sum$ is diagonal

  $p\,(\mathbf{x}\,|\,C_i) = \prod_j p\,(x_j\,|\,C_i)$ \qquad (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
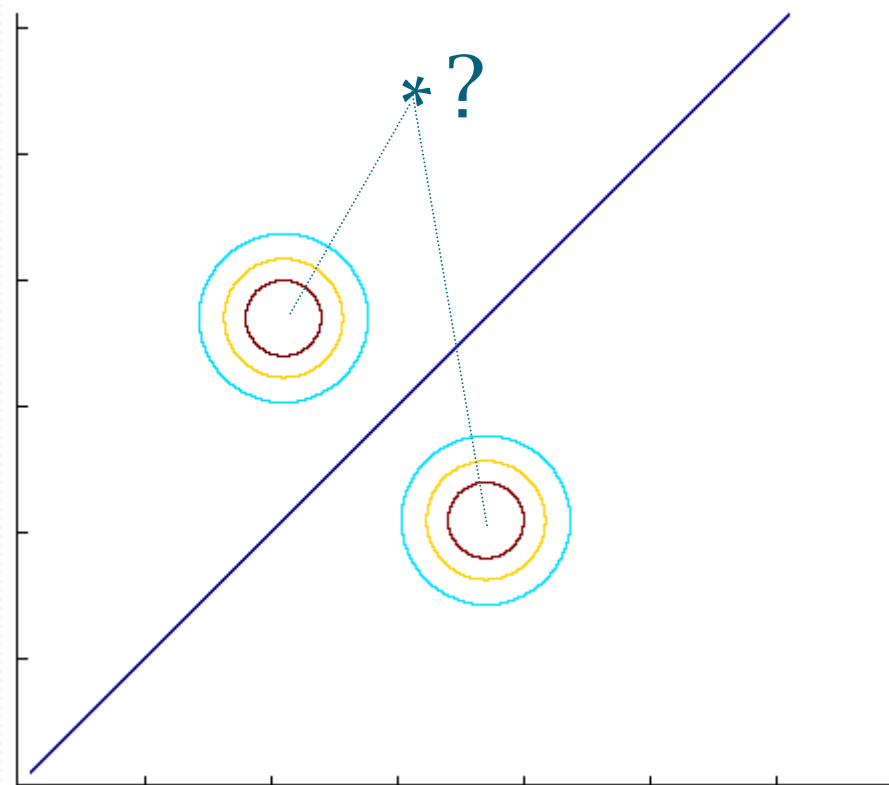
Diagonal **S**

variances may be different

Diagonal **S**, equal variances

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

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

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

- Each mean can be considered a prototype or template and this is template matching
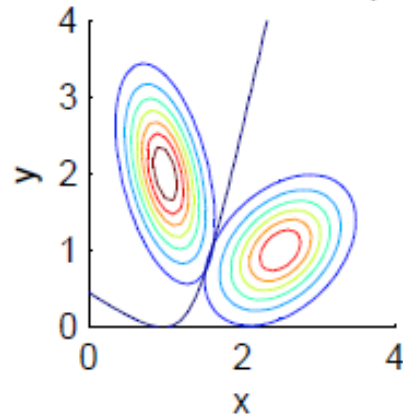
Diagonal **S**, equal variances

Model Selection

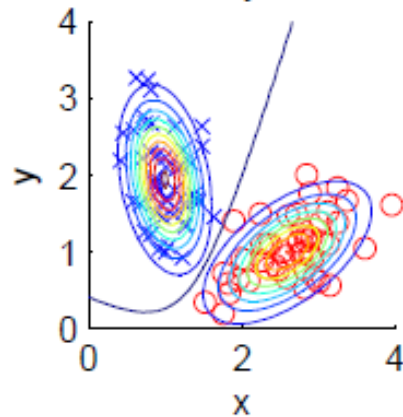| Assumption	Covariance matrix	No of parameters
Shared, Hyperspheric	$\mathbf{S}_i=\mathbf{S}=s^2\mathbf{I}$	1
Shared, Axis-aligned	$\mathbf{S}_i=\mathbf{S}$, with $s_{ij}=0$	d
Shared, Hyperellipsoidal	$\mathbf{S}_i=\mathbf{S}$	$d(d+1)/2$
Different, Hyperellipsoidal	$\mathbf{S}_i$	$K\,d(d+1)/2$

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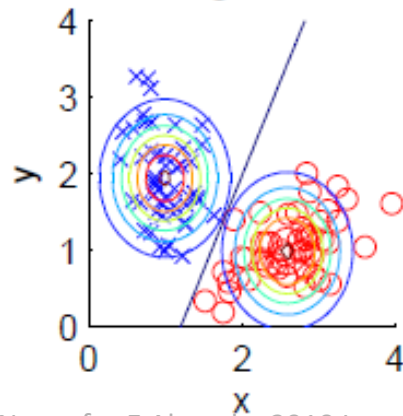- Assume simple models (allow some bias) to control variance (regularization)
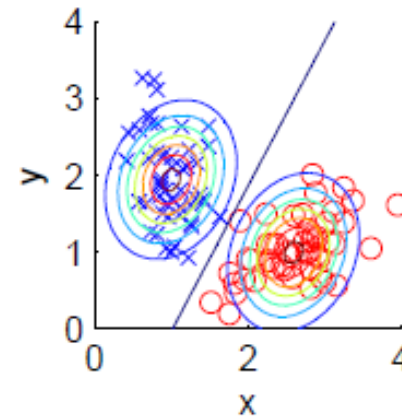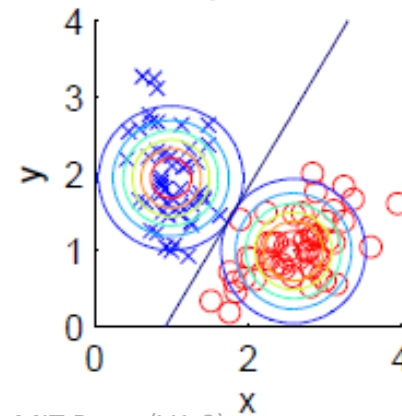
Population likelihoods and posteriors

Arbitrary covar.

Shared covar.

Diag. covar.

Equal var.

Discrete Features

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

  if x_j are independent (Naive Bayes')

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

  the discriminant is linear

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

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

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

Discrete Features

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

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

if x_j are independent

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

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

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

Multivariate Regression

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

- 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 **z** space

    (basis functions, kernel trick: Chapter 13)