FUNDAMENTALS OF MACHINE LEARNING

MULTIVARIATE METHODS

CSCI3320

Prof. John C.S. Lui, CSE Department, CUHK
Introduction to Machine Learning

Background

- In previous chapter, we discussed parametric approach to "*classification*" and "*regression*"
- Let us now generalize the technique wherein:
  - We have ***multiple features*** (*d* dimensions)
  - Output is either
    - a particular object (***classification***)
    - a function (***regression***)
- Learn the output function from a labeled multivariate sample (or supervised learning)

Parametric Methods

In Chapter 3, we wrote posterior probability for class C_i

$$P(C_i|x) = \frac{p(x|C_i)P(C_i)}{p(x)} = \frac{p(x|C_i)P(C_i)}{\sum_{k=1}^{K} p(x|C_k)P(C_k)}$$

Using discriminant function

$$g_i(x) = p(x|C_i)P(C_i) \quad \text{or} \quad g_i(x) = \log p(x|C_i) + \log P(C_i)$$

In parametric methods, we assume $p(x|C_i)$ follows some probability distributions, so we can use the probability density function of x to represent $p(x|C_i)$. The only question is how to "*estimate*" parameters of that distribution.

Now the probability density function has to be multivariate

Multivariate Data

- Multiple measurements are taken from the targeted inputs.
- *d* columns of inputs/features/attributes: *d*-variate
- *N* rows of instances/observations/examples

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

1st training data

2nd training data

Nth training data

Example of input

Notes on Multivariate Data

- These variables are **correlated** (**example**)
- Several objectives:
  - ***Simplification***: summarize the large data by means of relatively few parameters
  - ***Exploration***: generating hypotheses about the data
  - ***Prediction***: predict value of one variable from values of other variables. If the predicted value is:
    - **Discrete**: Then it is a "multivariate classification"
    - **Continuous**: Then it is a "multivariate regression"

Multivariate Parameters

How to compute this ?

- **mean vector $\boldsymbol{\mu}$:** $E[\boldsymbol{x}] = \boldsymbol{\mu} = [\mu_1, \ldots, \mu_d]^T$
- **Variance of X_i is σ_i^2** *How to compute this ?*
- **Covariance of X_i and X_j is** *How to compute this ?*

$$\sigma_{ij} \equiv \mathrm{Cov}(X_i, X_j) = E[(X_i - \mu_i)(X_j - \mu_j)] = E[X_i X_j] - \mu_i \mu_j$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2d} \\ \vdots & & & \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_d^2 \end{bmatrix}$$

$$\sigma_{ij} = \sigma_{ji},$$

$$\sigma_{ii} = \sigma_i^2$$

physical meaning

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

Parameter Estimation

The sample mean vector :

$$\boldsymbol{m} = \frac{\sum_{t=1}^{N} \boldsymbol{x}^t}{N} \text{ with } m_i = \frac{\sum_{t=1}^{N} x_i^t}{N}, i = 1, \ldots, d$$

The sample covariance matrix S where:

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

biased estimates

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

The sample correlation matrix R where:

$$r_{ij} = \frac{s_{ij}}{s_i s_j}$$

Covariance Matrix Σ

- σ_{ij} relates feature $\boldsymbol{x}_i$ with $\boldsymbol{x}_j$
- To illustrate, consider

$$\boldsymbol{x} = [x_1; x_2], \text{ then } \Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}$$

- What is $\boldsymbol{x}^T \Sigma \boldsymbol{x} = ?$
- We have

$$\begin{aligned} \boldsymbol{x}^T \Sigma \boldsymbol{x} &= [x_1, x_2] \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= [\sigma_1^2 x_1 + \sigma_{12} x_2,\ \sigma_{12} x_1 + \sigma_2^2 x_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= \sigma_1^2 x_1^2 + 2\sigma_{12} x_1 x_2 + \sigma_2^2 x_2^2 \end{aligned}$$

- Generalize to Σ which is a *d x d matrix*

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 or most occurred (if discrete))
  - **Imputation by regression:** Predict based on other attributes

Parametric Methods

Using discriminant function

$$g_i(x) = p(x|C_i)P(C_i) \quad \text{or} \quad g_i(x) = \log p(x|C_i) + \log P(C_i)$$

In parametric methods, we assume $p(x|C_i)$ follows some probability distributions, so we can use the probability density function of x to represent $p(x|C_i)$. The only question is how to "*estimate*" parameters of that distribution.

Now the probability density function has to be multivariate

Let's condition ***Multivariate Normal Distribution***

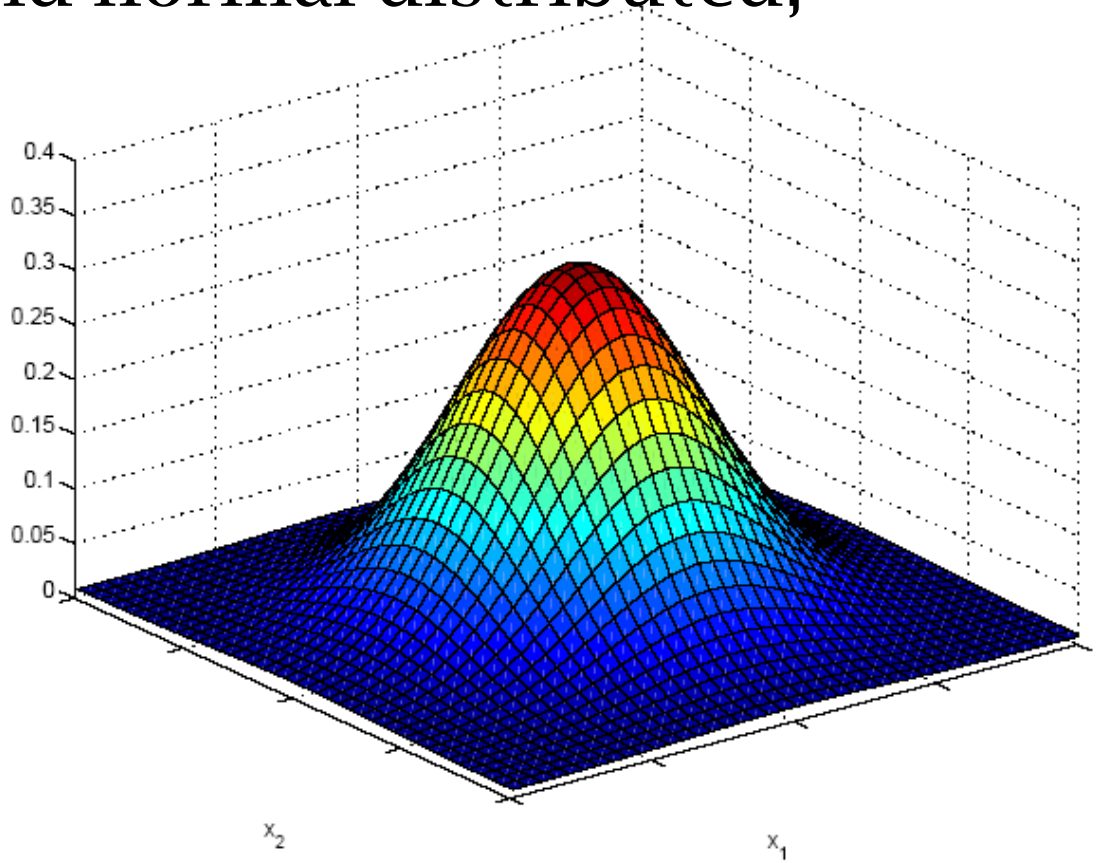
Multivariate Normal Distribution

$\boldsymbol{x}$ is d-dimensional and normal distributed,

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

$\boldsymbol{\mu}$ is the mean vector

$\boldsymbol{\Sigma}$ is the covariance matrix



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

determinant

Matrix inverse

Multivariate Normal Distribution

squared distance from x to μ in standard deviation units

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

- **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 Σ (normalizes for difference in variances and correlations)

- Bivariate: $d = 2$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} \qquad \rho = \frac{\sigma_{12}}{\sigma_1\sigma_2}$$

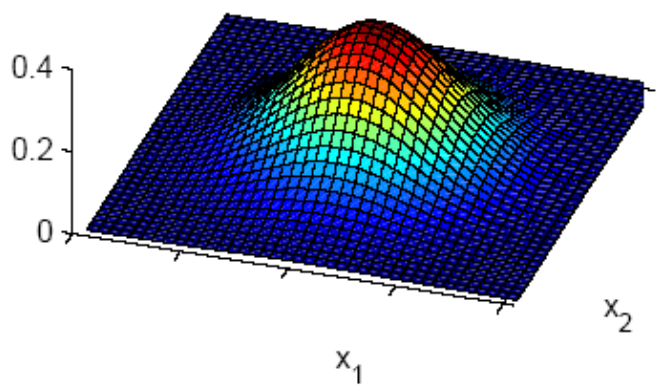
$$p(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left[\frac{-1}{2(1-\rho^2)}\left(z_1^2 - 2\rho z_1 z_2 + z_2^2\right)\right]$$

$$\text{where} \quad z_i = \frac{(x_i-\mu_i)}{\sigma_i} \text{ for } i \in \{1,2\}$$

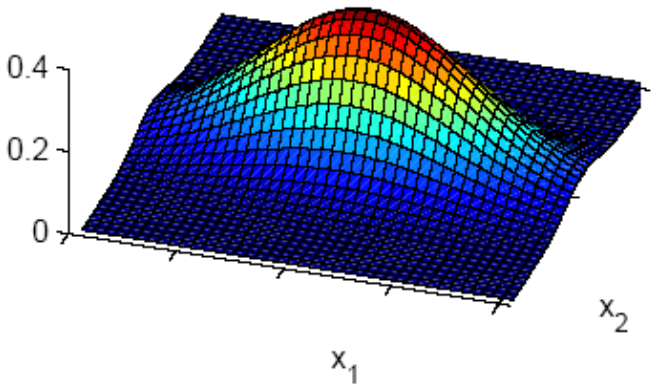
Multivariate Normal Distribution (Supplement)

- **Determinant**
  - If $\boldsymbol{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, then $|\boldsymbol{A}| = ad - bc$
  - If $\boldsymbol{A} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$, then $|\boldsymbol{A}| = aei + bfg + cdh - ceg - bdi - afh$
- **Inverse**
  - If $\boldsymbol{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, $\boldsymbol{A}^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{|\boldsymbol{A}|} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$
- For higher dimension, you can do Google lookup

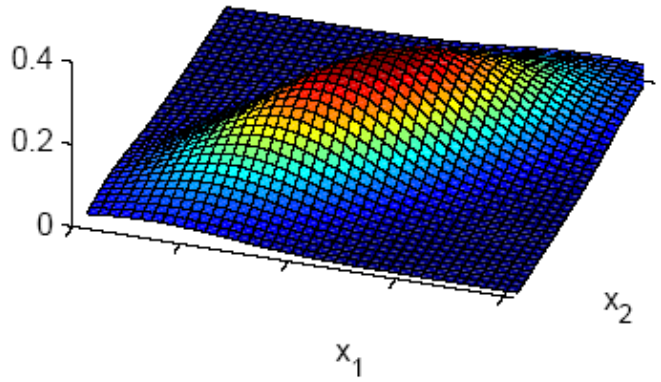
$Cov(x_1,x_2)=0$, $Var(x_1)=Var(x_2)$



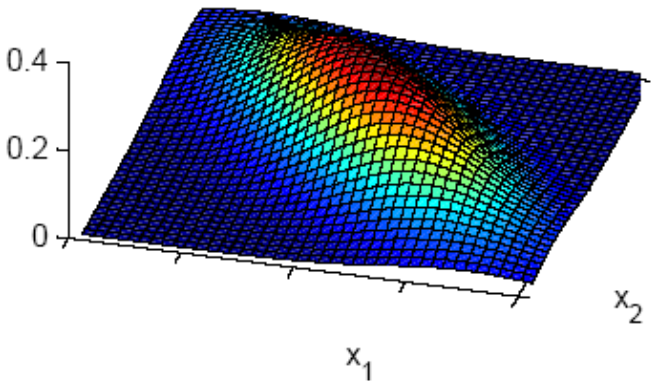
$Cov(x_1,x_2)=0$, $Var(x_1)>Var(x_2)$



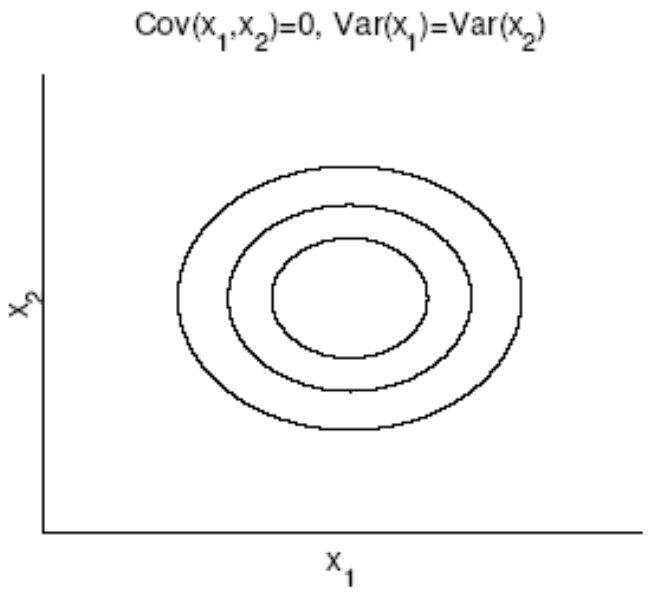
$Cov(x_1,x_2)>0$

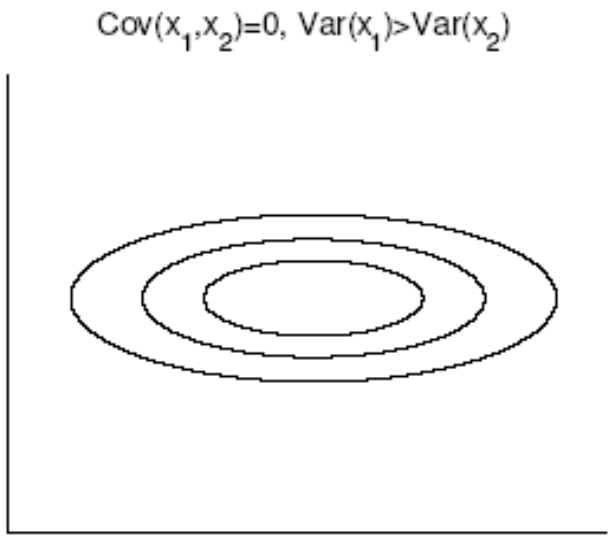


$Cov(x_1,x_2)<0$

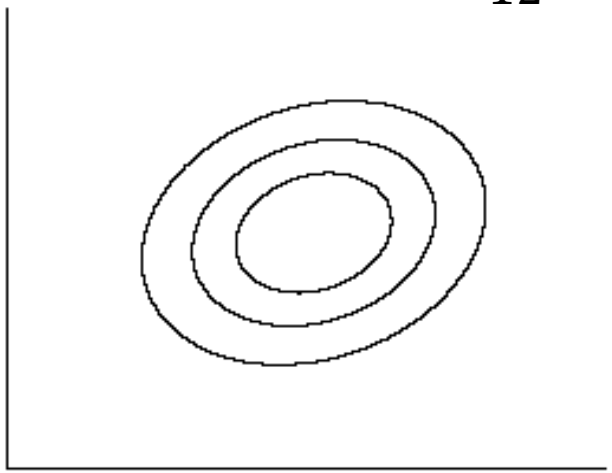


Bivariate Normal, d=2, contour plot

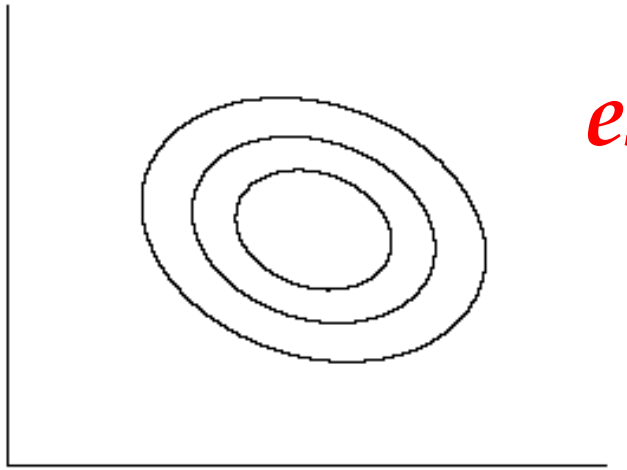




Cov(x_1,x_2)>0 $\sigma_{12} > 0$



Cov(x_1,x_2)<0 $\sigma_{12} < 0$



explain

Multivariate Normal Distribution

- In the multivariate case, small $|\Sigma|$ means samples are close to the mean vector $\boldsymbol{\mu}$
- Σ is a symmetric positive definite matrix, or the multivariate way of saying that $Var(X) > 0$
- If Σ is not a symmetric positive definite matrix means
  - There is linear dependency between the dimensions
  - One of the dimension has variance 0
  - In such a case, we need to reduce the "dimension" (say via PCA or other techniques, which we will go over in the next chapter)

Multivariate Normal Distribution: Naïve case

- If x_i are independent (or $\text{COV}(X_i, X_j) = 0, \text{Var}(X_i) = \sigma_i^2, \forall i$
- Off diagonals of Σ are 0.

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

- Mahalanobis distance reduces to weighted (by $1/\sigma_i$)
- If variances are also equal, reduces to Euclidean distance
- We can put this $p(\boldsymbol{x})$ in our discriminant function

Multivariate Normal Distribution: Property

- Let: $\boldsymbol{x} \sim \mathcal{N}_d(\boldsymbol{\mu}, \Sigma)$ and $\boldsymbol{w} \in \Re^d$
- We have: $\boldsymbol{w}^T\boldsymbol{x} = w_1x_1 + w_2x_2 + \cdots + w_dx_d \sim \mathcal{N}(\boldsymbol{w}^T\boldsymbol{\mu}, \boldsymbol{w}^T\Sigma\boldsymbol{w})$
- The mean and variance can be shown as:

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

Show derivation

- **Result**: *the projection of a d-dimensional normal on a vector* ***w*** *is univariate normal*
- **Generalization**: project of a *d-dimensional normal* on ***W*** which is a *d x k* matrix with rank $k < d$, then

$\mathbf{W}^T\boldsymbol{x}$ is k-variate normal

Multivariate Parametric Classification

- When $\boldsymbol{x} \in \Re^d$
- If conditional densities are $p(\boldsymbol{x}|C_i) \sim N_d(\mu_i, \sum_i)$ *for class i*

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

- **Application**: predict the type of car a customer is interested in
- Different cars = classes
- Observable data of customers: $\boldsymbol{x}$
- Vector of mean age/income of customers who buy car type *i*: $\boldsymbol{\mu}_i$
- Σ_i Covariance *matrix*:
  - σ_{i1}^2 : *variance of age of type i*
  - σ_{i2}^2 : *variance of income of type i*
  - σ_{i12} : *covariance of age & income in the group of customers who buy type i*

Multivariate Parametric Classification

- Discriminant functions and assuming $p(\boldsymbol{x}|C_i) \sim \mathcal{N}_d(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

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

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

- Training sample for $K >= 2$ classes:

$$\mathcal{X} = \{\boldsymbol{x}^t, \boldsymbol{r}^t\}, \text{ where } r_i^t = 1 \text{ if } \boldsymbol{x}^t \in C_i \text{ and } 0 \text{ otherwise}$$

- How can we do the parametric estimate?

Estimation of Parameters

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

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

Note: *this is a (dx1) vector*

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

Note: *this is a (dxd) matrix*

Plug the above expressions in

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

Expand it

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

Different S_i

- **The above discriminant is a "*Quadratic discriminant*"**

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

Why quadratic?

where

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

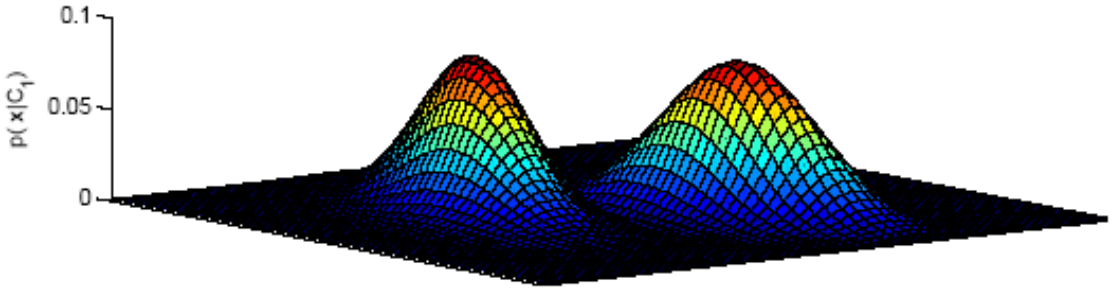
$$\boldsymbol{w}_i = \mathbf{S}_i^{-1}\boldsymbol{m}_i$$

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

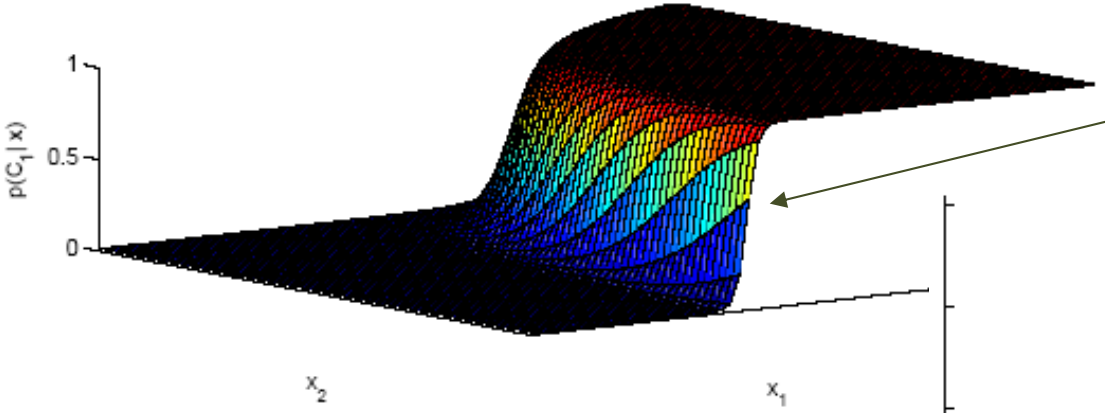
Number of parameters to estimate:

- ***K*d* for means**
- ***K*d(d+1)/2* for covariance matrices**

- S_i may be singular and inverse may not exist
- $|S_i|$ may be nonzero but very small, makes the process unstable
- If small # of samples, need to reduce dimension d ***(CHAPTER 6)***

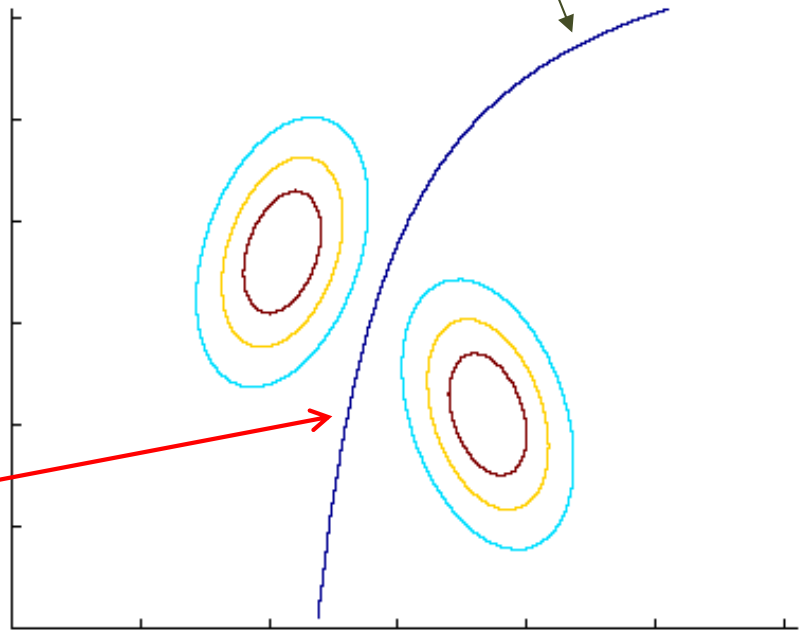


likelihoods



posterior for C_1

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



Quadratic distriminant

Case 1: Common Covariance Matrix S

- Shared common sample covariance **S**

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

- Discriminant reduces to

Common $\boldsymbol{x}^T \boldsymbol{S}^{-1} \boldsymbol{x}$ $\forall C_i$, can be ignored

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

which becomes a linear discriminant

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

where

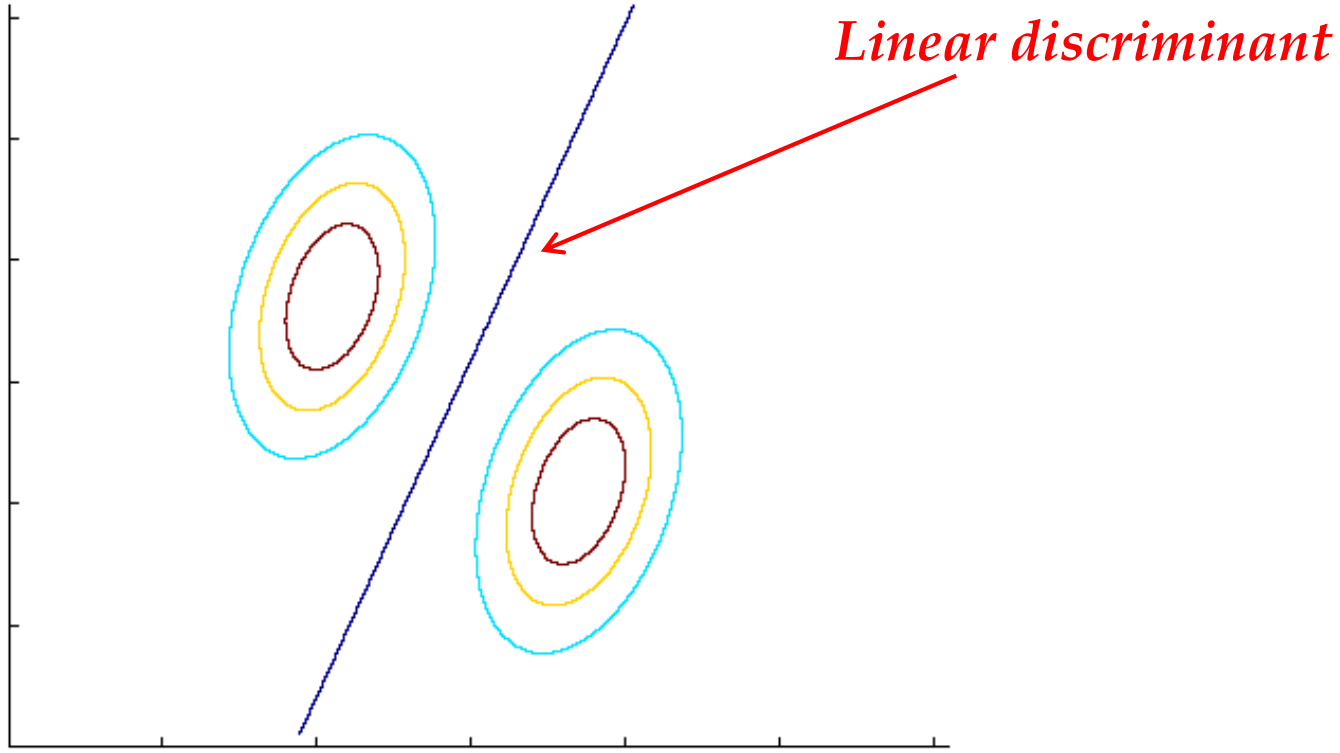
$$\boldsymbol{w}_i = \mathbf{S}^{-1}\boldsymbol{m}_i$$

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

Number of parameters to estimate:

- ***K*d* for means**
- ***d(d+1)/2* for covariance matrices**

Common Covariance Matrix S



If the priors are equal, the "optimal decision" is to assign input to the class whose mean's Mahalanobis distance to the input is the smallest.

Case 2: Diagonal S

- If we assume all off-diagonals of the covariance matrix are zeros (or when $x_i, i = 1, \ldots, d$ x_i are independent)
- This is the ***"Naïve Bayes' classifier"***: Σ is diagonal, $p(\boldsymbol{x}_i|C_i)$ are univariate Gaussian, S and its inverse are diagonal

$$p(\boldsymbol{x}|C_i) = \prod_d p(x_d|C_i), \text{ or Naive Bayes' assumption}$$

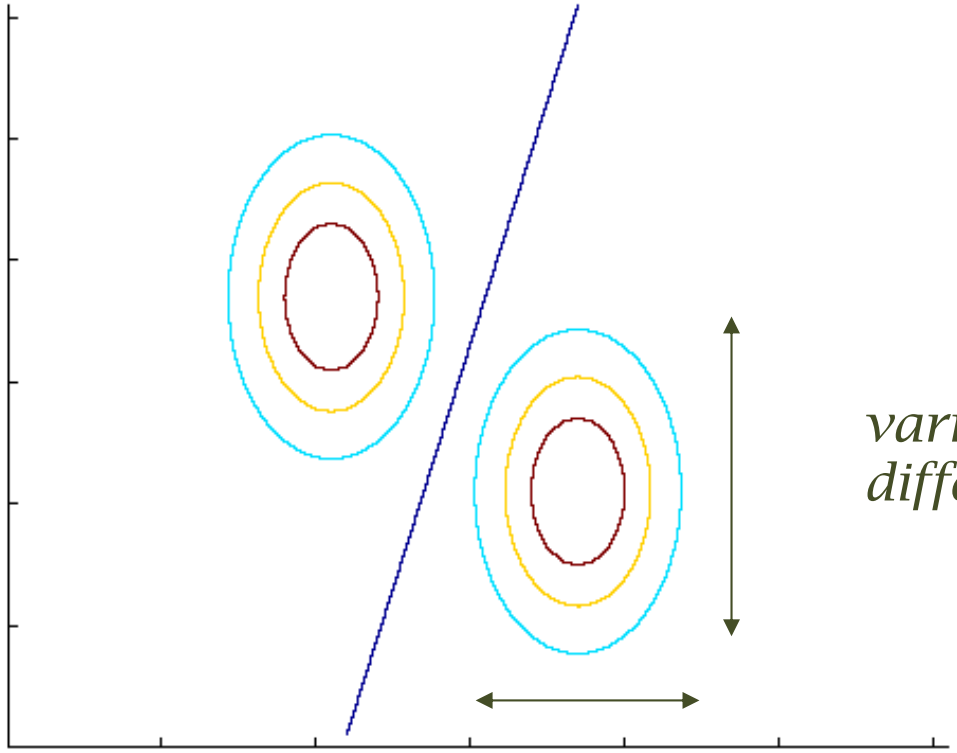
$$g_i(\boldsymbol{x}) = -\frac{1}{2}\sum_{j=1}^{d}\left(\frac{x_j^t - m_{ij}}{s_j}\right)^2 + \log \hat{P}(C_i)$$

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

Number of parameters to estimate:

- ***K*d* for means**
- ***d* for covariance matrices**

Diagonal S



variances may be different

Diagonal S, equal variances

- When all variances are equal, $|S|=s^{2d}$, $S^{-1}=(1/s^2)$, Mahalanobis distance reduces to the *Euclidean distance*
- **Discriminant function**:

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

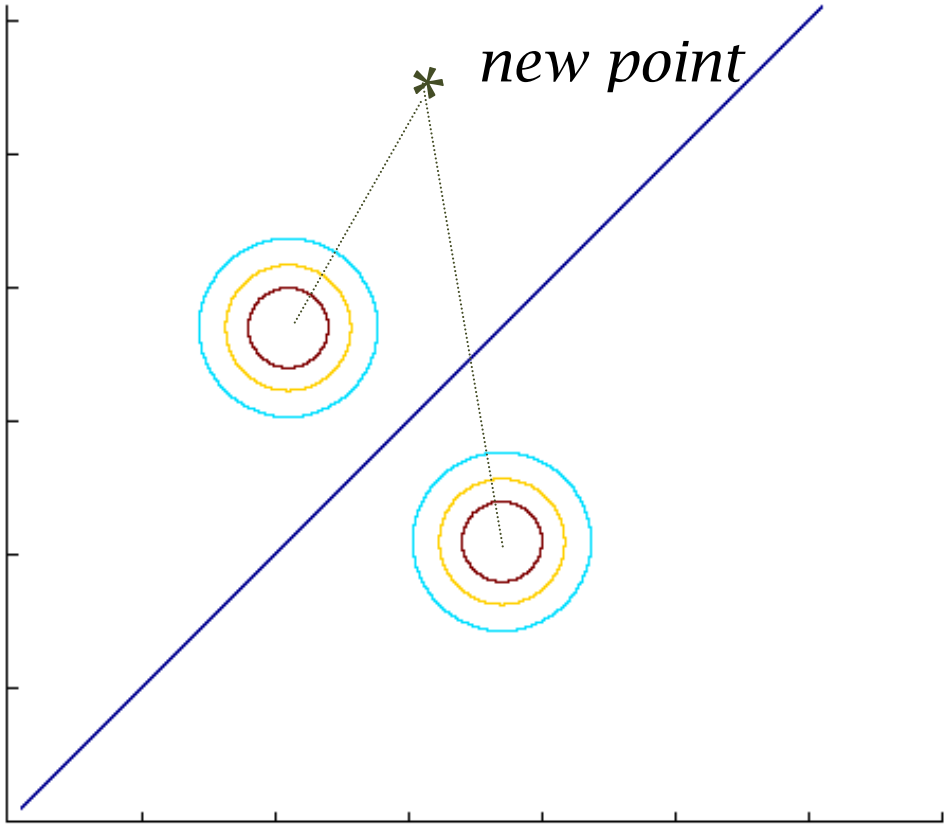
- When the priors are the same ($P(C_i)$), it becomes the nearest mean classifier: **Classify based on Euclidean distance to the nearest mean**
- Each mean can be considered a prototype or template and this is template matching

$$\begin{aligned} g_i(\boldsymbol{x}) &= -\|\boldsymbol{x}-\boldsymbol{m}_i\|^2 = -(\boldsymbol{x}-\boldsymbol{m}_i)^T(\boldsymbol{x}-\boldsymbol{m}_i) \\ &= -(\boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{m}_i^T\boldsymbol{x} + \boldsymbol{m}_i^T\boldsymbol{m}_i) \end{aligned}$$

Number of parameters to estimate:

- ***K*d* for means**
- ***1* for covariance matrices**

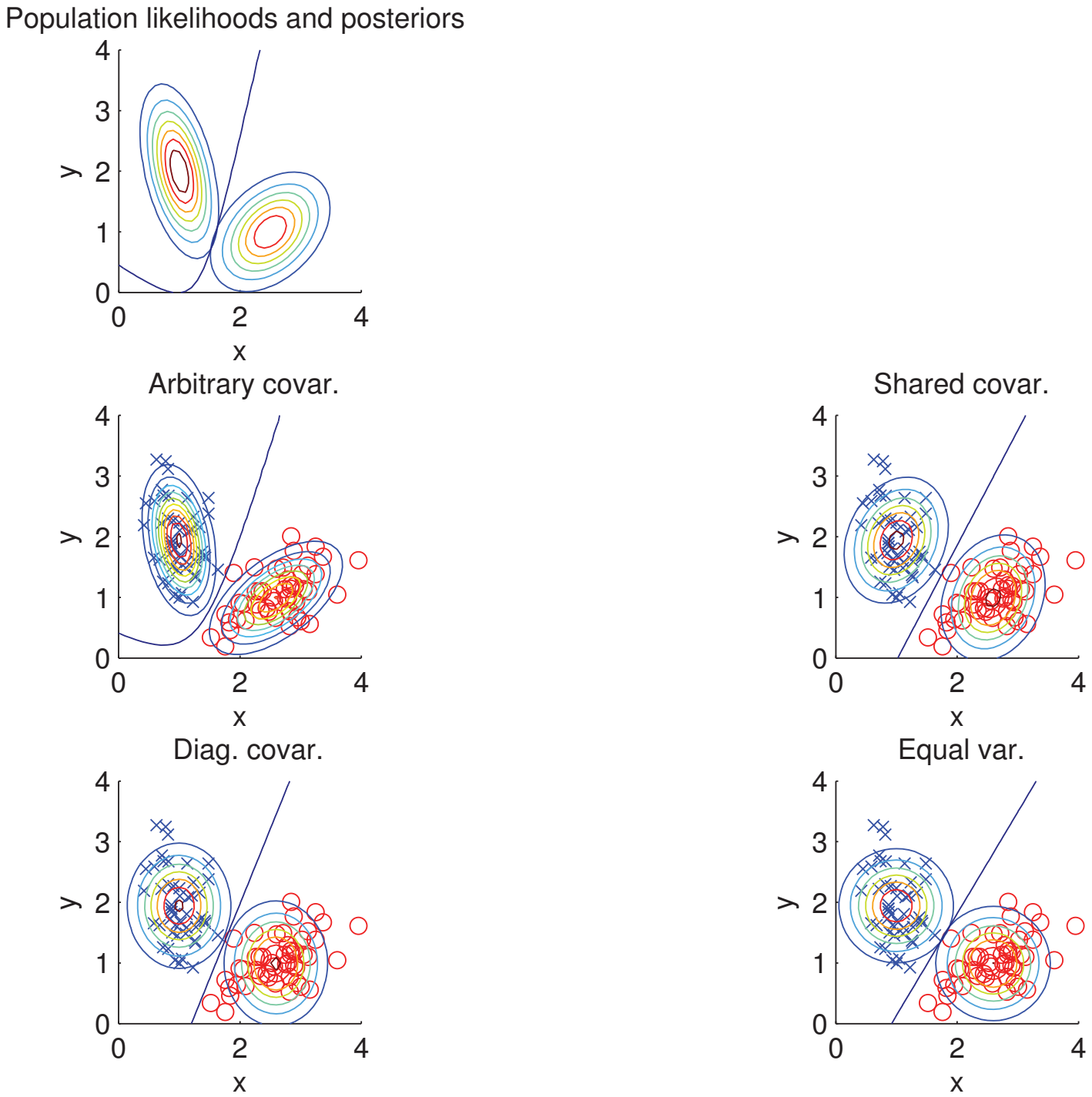
Diagonal S, equal variances



Tuning Complexity: Model Selection

Assumption	*Covariance matrix*	*No of parameters in* $\mathbf{S}_i$
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 (bias/variance dilemma)
- Assume simple models (allow some bias) to control variance (regularization)


Population likelihoods and posteriors
Arbitrary covar.
Shared covar.
Diag. covar.
Equal var.
x
y
0
1
2
3
4

Discrete Features

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

  Read book on application

  if x_i are **independent** (Naive Bayes')

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

the discriminant is linear

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

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: $x_j \in \{v_1, v_2, \ldots, v_{n_j}\}$
- Define new variables:

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

- Let p_{ijk} denote the probability that x_j belonging to C_i takes value v_k:

$$p_{ijk} \equiv p(z_{jk} = 1 | C_i) = p(x_j = v_k | C_i) \qquad \sum_{k=1}^{n_j} p_{ijk} = 1$$

- If all attributes are *independent*, we have

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

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

- Maximum likelihood estimator: $\hat{p}_{ijk} = \frac{\sum_t z_{jk}^t r_i^t}{\sum_t r_i^t}$

Multivariate Regression

- **Multivariate linear model**: numeric output r is a weighted sum of all d variables, $x_1, \ldots, x_d$, and noise.

$$r^t = g(\mathbf{x}^t | w_0, w_1, \ldots, w_d) + \epsilon = w_0 + w_1 x_1^t + w_2 x_2^t + \cdots + w_d x_d^t + \epsilon$$

- When noise is normal with 0 mean and constant variance, maximize the likelihood is to minimize the sum square error (*page 33 for slide in Chapter 4*):

$$E(w_0, w_1, \ldots, w_d | \mathcal{X}) = \frac{1}{2} \sum_t (r^t - w_0 - w_1 x_1^t - w_2 x_2^t - \cdots - w_d x_d^t)^2$$

- Taking the derivative with respect to $w_j, j = 0, \ldots, d$

$$\begin{aligned}
\sum_t r^t &= N w_0 + w_1 \sum_t x_1^t + w_2 \sum_t x_2^t + \cdots + w_d \sum_t x_d^t \\
\sum_t x_1^t r^t &= w_0 \sum_t x_1^t + w_1 \sum_t (x_1^t)^2 + w_2 \sum_t x_1^t x_2^t + \cdots + w_d \sum_t x_1^t x_d^t \\
\sum_t x_2^t r^t &= w_0 \sum_t x_2^t + w_1 \sum_t x_1^t x_2^t + w_2 \sum_t (x_2^t)^2 + \cdots + w_d \sum_t x_2^t x_d^t \\
&\vdots \\
\sum_t x_d^t r^t &= w_0 \sum_t x_d^t + w_1 \sum_t x_d^t x_1^t + w_2 \sum_t x_d^t x_2^t + \cdots + w_d \sum_t (x_d^t)^2
\end{aligned}$$

Multivariate Regression

- Define vectors and matrix:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1^1 & x_2^1 & \cdots & x_d^1 \\ 1 & x_1^2 & x_2^2 & \cdots & x_d^2 \\ \vdots & & & & \\ 1 & x_1^N & x_2^N & \cdots & x_d^N \end{bmatrix}, \boldsymbol{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d, \end{bmatrix}, \boldsymbol{r} = \begin{bmatrix} r^1 \\ r^2 \\ \vdots \\ r^N \end{bmatrix}$$

- ***Normal equation***: $\mathbf{X}^T\mathbf{X}\boldsymbol{w} = \mathbf{X}^T\boldsymbol{r}$, or solve for: $\boldsymbol{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{r}$
- Same as we did for polynomial regression using one input when we define: $x_1=x$, $x_2=x^2$,….,$x_k=x^k$
- This is known as the ***multivariate polynomial regression***
- We can use this method if d is small
- If w_i is + (-), it has a positive (negative) effect on the output
- If all x_i are in the same range, w_i indicates which features are important

Conclusion

- We extend the parametric method in Chapter 4 (single feature) to general *d*-features
- Consider multivariate distributions: Gaussian and multinomial *d*-feature distribution
- Multivariate Gaussian: features not necessary independent.
  - Depending on my assumptions, we have different type of classifier (from quadratic to linear to nearest distance)
- Multinomial distribution: assume independence in features
- **Question**: *What are the short-comings of this method?*