Multivariate Methods

Slides from Machine Learning by Ethem Alpaydin
Expanded by some slides from Gutierrez-Osuna

Overview

- We learned how to use the Bayesian approach for classification *if* we had the probability distribution of the underlying classes ($p(x|C_i)$).

- Now going to look into how to estimate those densities from given samples.

Expectations

The average value of a function f(x) under a probability distribution p(x) is called the expectation of f(x).

Average is weighted by the relative probabilities of different values of x.

$$\mathbb{E}[f] = \sum_x p(x) f(x) \qquad \mathbb{E}[f] = \int p(x) f(x)\, \mathrm{d}x$$

$$\mathbb{E}[f] \simeq \frac{1}{N} \sum_{n=1}^{N} f(x_n)$$

Approximate Expectation
(discrete and continuous)

Now we are going to look at concepts: variance and co-variance , of 1 or more random variables, using the concept of expectation.

Variance and Covariance

The variance of f(x) provides **a measure for how much f(x) varies around its mean** E[f(x)].

$$\mathrm{var}[f] = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])^2\right] = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2$$

Given a set of N points $\{x_i\}$ in the1D-space, **the variance of the corresponding random variable x** is var[x] = $E[(x-\mu)^2]$ where μ=E[x].

You can estimate the expected value as

$$\mathrm{var}(x) = E[(x-\mu)^2] \approx 1/N \sum_{x_i} (x_i-\mu)^2$$

Remember the definition of expectation: $\mathbb{E}[f] \simeq \frac{1}{N}\sum_{n=1}^{N} f(x_n)$

Variance and Covariance

The variance of **x** provides **a measure for how much x varies around its mean** μ=E[x].

$$\text{var}(x) = E[(x-\mu)^2]$$

$$\text{var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2$$

Co-variance of two random variables x and y measures the extent to which they vary together.

$$\begin{aligned} \text{cov}[x, y] &= \mathbb{E}_{x,y}\left[\{x - \mathbb{E}[x]\}\{y - \mathbb{E}[y]\}\right] \\ &= \mathbb{E}_{x,y}[xy] - \mathbb{E}[x]\mathbb{E}[y] \end{aligned}$$

Variance and Covariance

Co-variance of two random variables x and y measures the extent to which they vary together.

$$\begin{aligned} \mathrm{cov}[x, y] &= \mathbb{E}_{x,y}\left[\{x - \mathbb{E}[x]\}\{y - \mathbb{E}[y]\}\right] \\ &= \mathbb{E}_{x,y}[xy] - \mathbb{E}[x]\mathbb{E}[y] \end{aligned}$$

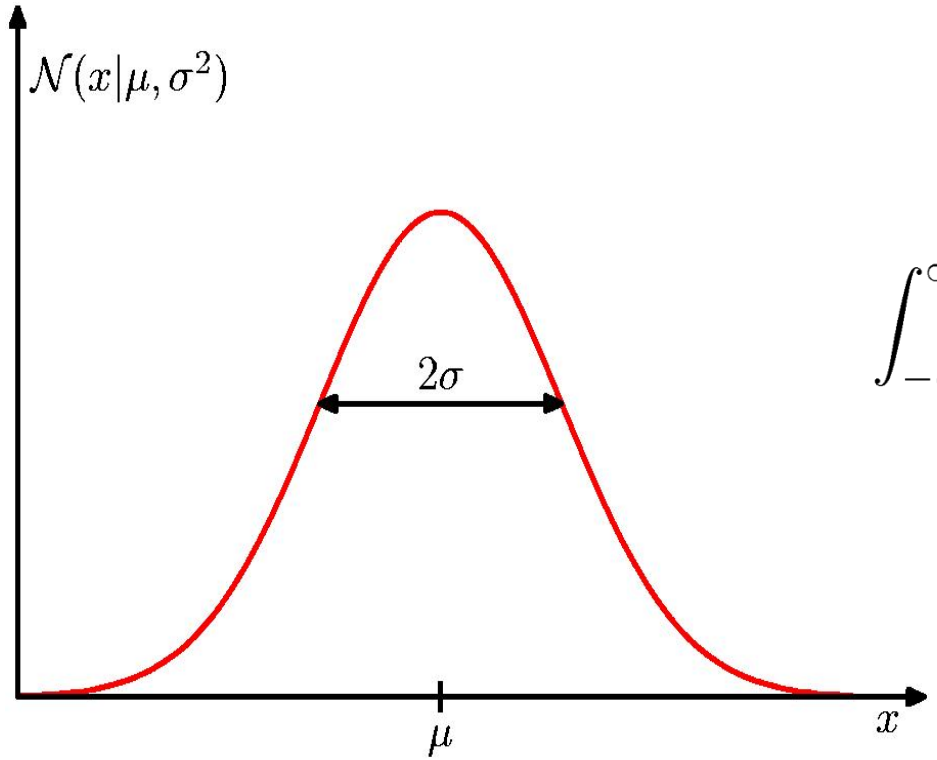
: two random variables x, y

: two vector random variables **x, y** – covariance is a matrix

Multivariate Normal Distribution

The Gaussian Distribution

$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



$$\mathcal{N}(x|\mu,\sigma^2) > 0$$

$$\int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu,\sigma^2\right)\,\mathrm{d}x = 1$$

Expectations

$$\mathbb{E}[f] = \sum_x p(x) f(x) \qquad \mathbb{E}[f] = \int p(x) f(x)\, \mathrm{d}x \qquad \mathbb{E}[f] \simeq \frac{1}{N} \sum_{n=1}^{N} f(x_n)$$

$$\mathrm{var}[f] = \mathbb{E}\left[\left(f(x) - \mathbb{E}[f(x)] \right)^2 \right] = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2$$

For normally distributed x:

$$\mathbb{E}[x] = \int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu, \sigma^2\right) x\, \mathrm{d}x = \mu$$

$$\mathbb{E}[x^2] = \int_{-\infty}^{\infty} \mathcal{N}\left(x|\mu, \sigma^2\right) x^2\, \mathrm{d}x = \mu^2 + \sigma^2$$

$$\mathrm{var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \sigma^2$$

- Assume we have a d-dimensional input (e.g. 2D), **x**.

- We will see how can we characterize p(**x**), assuming **x** is normally distributed.

  - For 1-dimension it was the **mean** (μ) and **variance** (σ^2)
    - $\text{Mean}=E[x]$
    - $\text{Variance}=E[(x-\mu)^2]$

  - For d-dimensions, we need
    - the d-dimensional **mean** vector
    - dxd dimensional **covariance** matrix

  - If $\mathbf{x} \sim N_d(\boldsymbol{\mu},\boldsymbol{\Sigma})$ then each dimension of x is univariate normal
    - Converse is not true

Normal Distribution & Multivariate Normal Distribution

- For a single variable, the normal density function is:

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

- For variables in higher dimensions, this generalizes to:

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

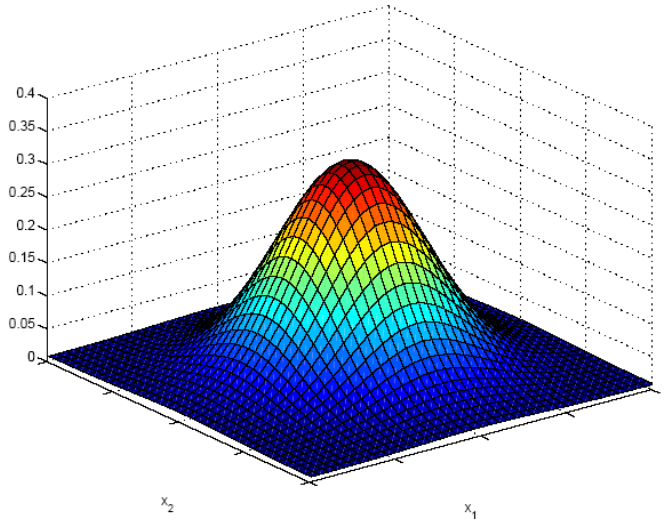
- where the mean μ is now a d-dimensional vector,

  Σ is a *d* x *d covariance matrix*

  $|\Sigma|$ is the determinant of Σ:

$$\mu = \mathcal{E}[\mathbf{x}]$$

$$\Sigma = \mathcal{E}[(\mathbf{x}-\mu)(\mathbf{x}-\mu)^{\mathrm{T}}].$$



Multivariate Parameters: Mean, Covariance

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

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

$$\sigma_{ij} \equiv \text{Cov}(x_i, x_j) \equiv E[(x_i - \mu_i)(x_j - \mu_j)]$$

Matlab code

- close all;
- rand('twister', 1987) % seed

%Define the parameters of two 2d-Normal distribution

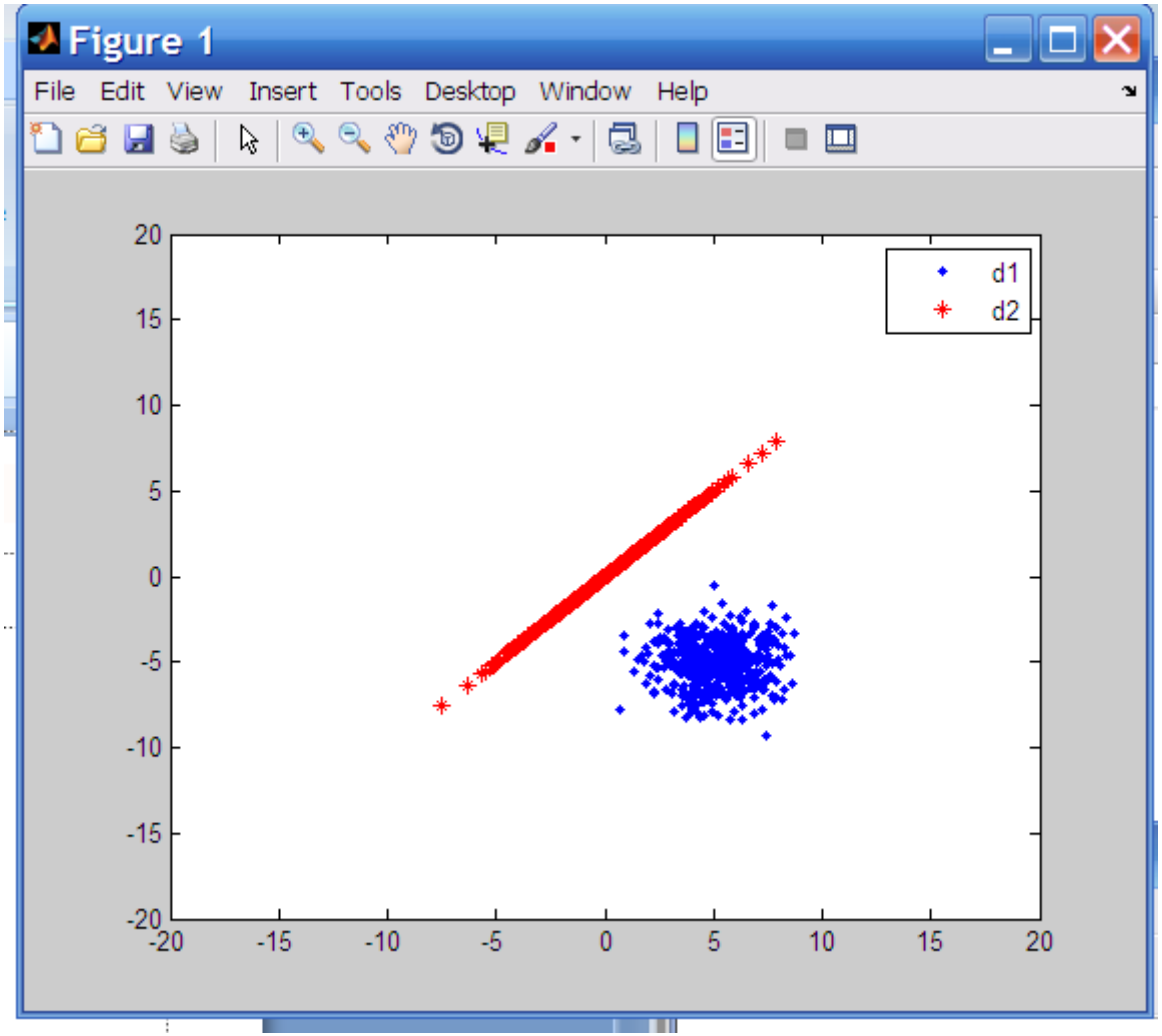
- **mu1 = [5 -5];**
- **mu2 = [0 0];**
- **sigma1 = [2 0; 0 2];**
- **sigma2 = [5 5; 5 5];**

- N=500; %Number of samples we want to generate from this distribution

- **samp1 = mvnrnd(mu1,sigma1, N);**
- **samp2 = mvnrnd(mu2, sigma2, N);**
-
- figure; clf;
- plot(samp1(:,1), samp1(:,2),'.', 'MarkerEdgeColor', 'b');
- hold on;
- plot(samp2(:,1), samp2(:,2),'*', 'MarkerEdgeColor', 'r');
- axis([-20 20 -20 20]); legend('d1', 'd2');

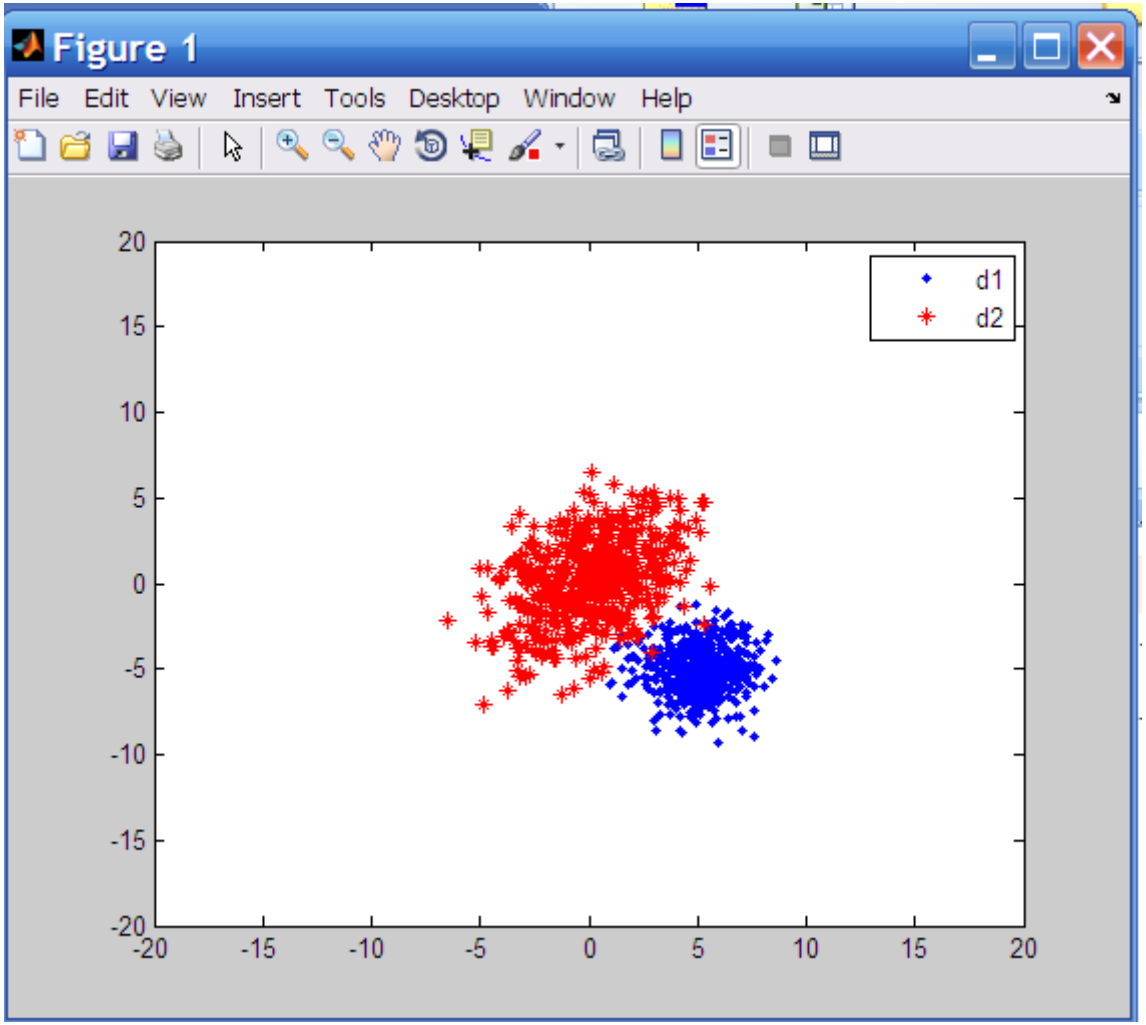
mu1 = [5 -5];
mu2 = [0 0];

sigma1 = [2 0; 0 2];
sigma2 = [5 5; 5 5];



mu1 = [5 -5];
mu2 = [0 0];

sigma1 = [2 0; 0 2];
sigma2 = [5 2; 2 5];



Matlab sample cont.

- % Lets compute the mean and covariance as if we are given this data
- **sampmu1 = sum(samp1)/N;**
- **sampmu2 = sum(samp2)/N;**
- sampcov1 = zeros(2,2);
- sampcov2 = zeros(2,2);
- **for i =1:N**
- **sampcov1 = sampcov1 + (samp1(i,:)-sampmu1)' * (samp1(i,:)-sampmu1);**
- **sampcov2 = sampcov2 + (samp2(i,:)-sampmu2)' * (samp2(i,:)-sampmu2);**
- **End**
- sampcov1 = sampcov1 /N;
- sampcov2 = sampcov2 /N;
- %%
- % Lets compute the mean and covariance as if we are given this data USING MATRIX OPERATIONS
- % Notice that in samp1, samples are given in ROWS – but for this multiplication, columns * rows is req.
- **sampcov1 = (samp1'*samp1)/N - sampmu1'*sampmu1;**
- %Or simply
- **mu=mean(samp1);**
- **cov=cov(samp1);**

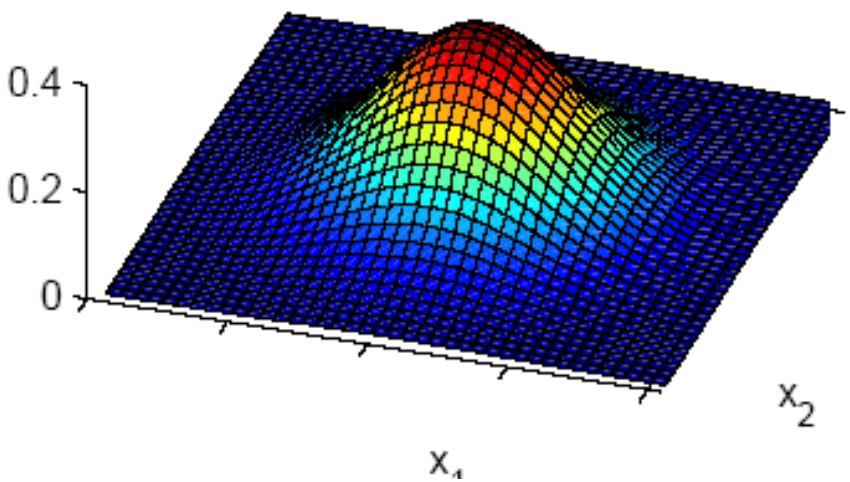
- **Variance**: How much X varies around the expected value
- **Covariance** is the measure the strength of the **linear relationship** between two random variables
  - covariance becomes more positive for each pair of values which differ from their mean in the same direction
  - covariance becomes more negative with each pair of values which differ from their mean in opposite directions.
  - **if two variables are independent, then their covariance/correlation is zero (converse is not true).**

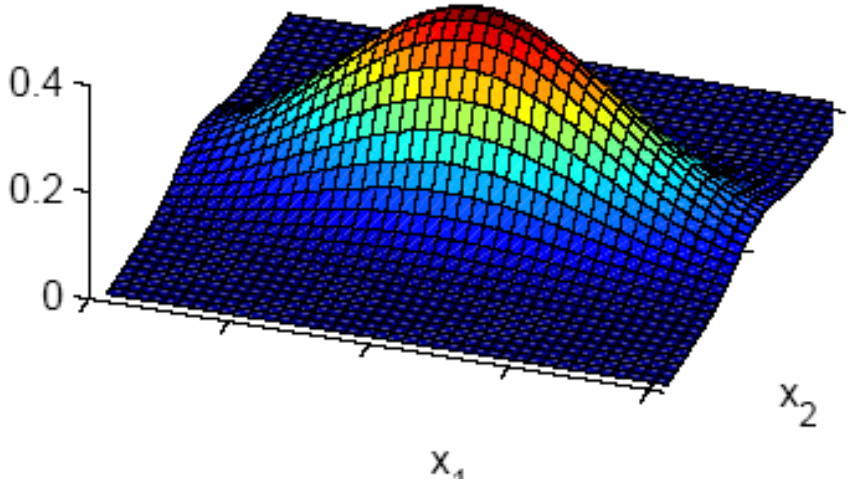
- **Correlation** is a dimensionless measure of linear dependence.
  - **range between –1 and +1**

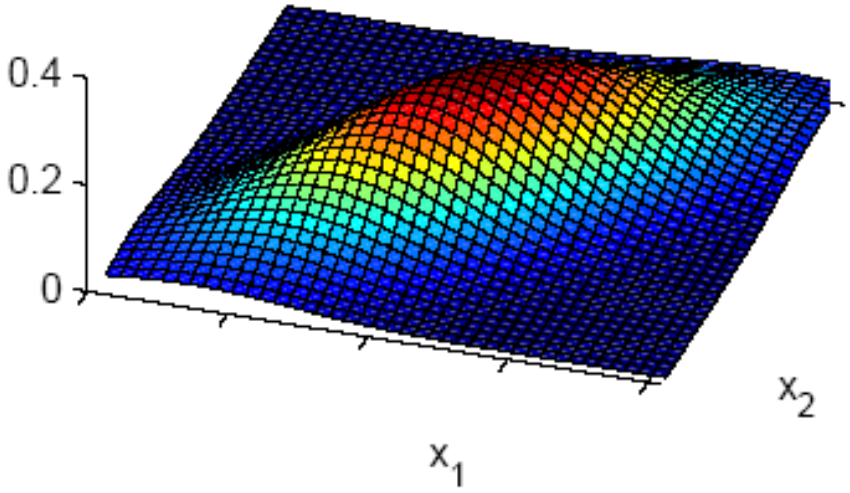
$$\text{Covariance}: \sigma_{ij} \equiv \text{Cov}(x_i, x_j) \equiv E[(x_i - \mu_i)(x_j - \mu_j)]$$

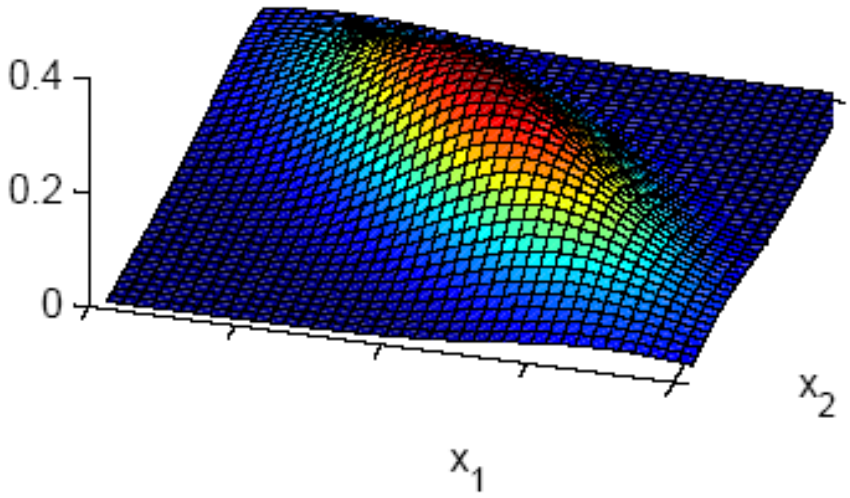
$$\text{Correlation}: \text{Corr}(x_i, x_j) \equiv \rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$$

How to characterize differences between these distributions

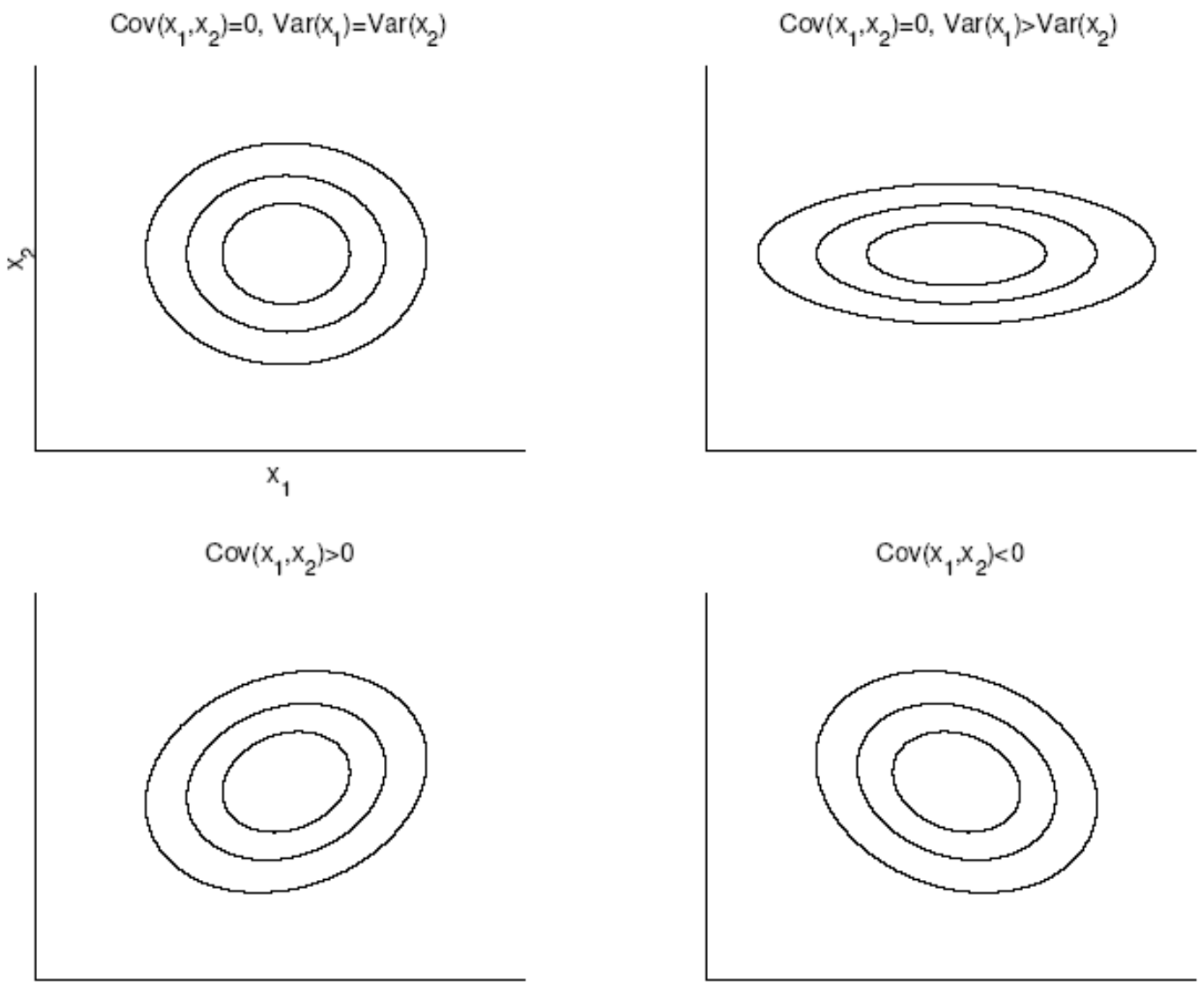


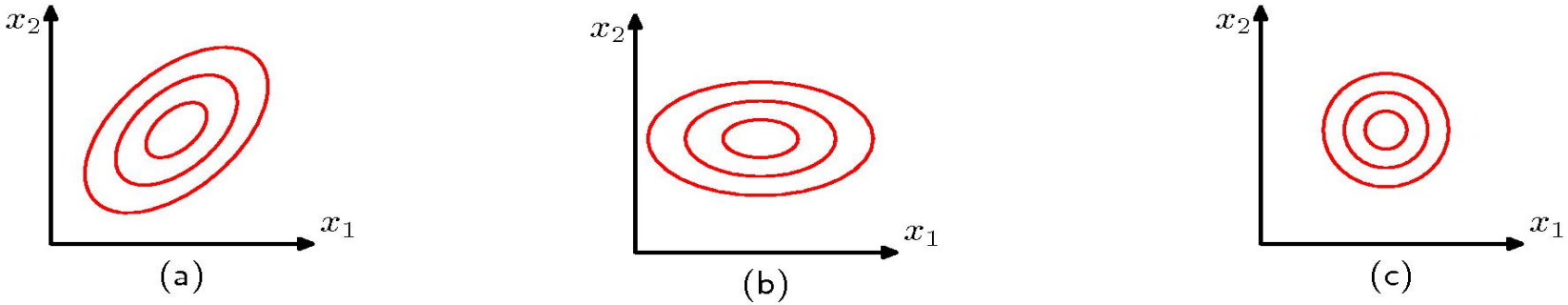






Covariance Matrices





Contours of constant probability density for a 2D Gaussian distributions with
a) general covariance matrix
b) diagonal covariance matrix (covariance of x1,x2 is 0)
c) Σ proportional to the identity matrix (covariances are 0, variances of each dimension are the same)

Shape and orientation of the hyper-ellipsoid centered at μ is defined by Σ

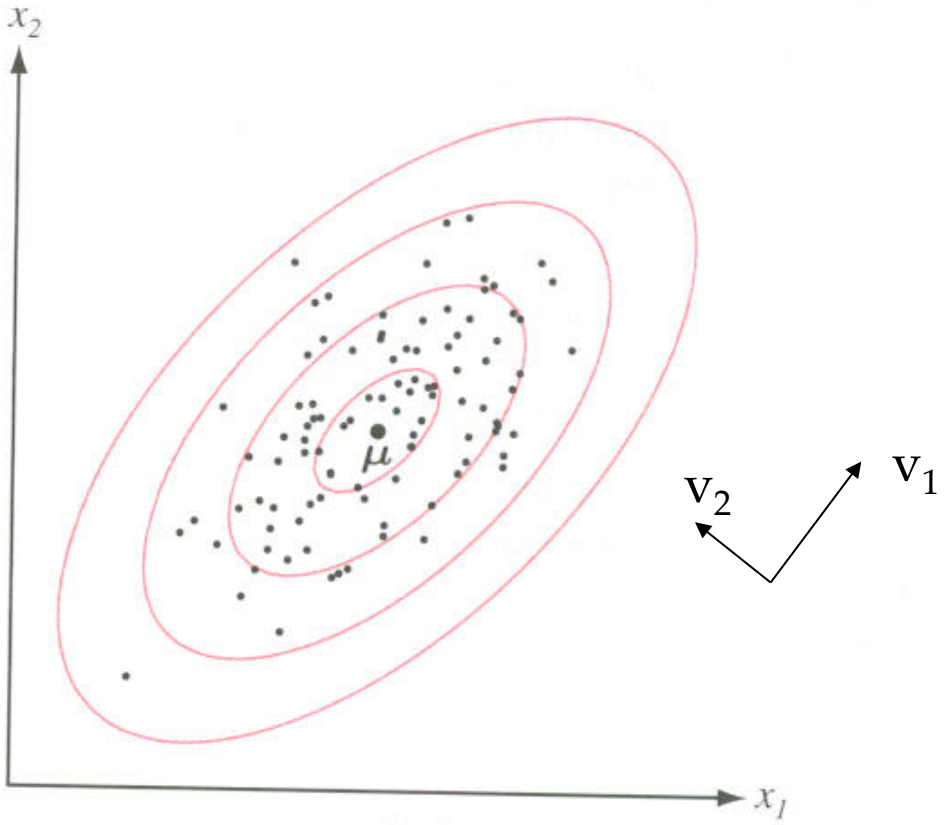


FIGURE 2.9. Samples drawn from a two-dimensional Gaussian lie in a cloud centered on the mean $\boldsymbol{\mu}$. The ellipses show lines of equal probability density of the Gaussian.

Properties of Σ

- A small value of $|\Sigma|$ (determinant of the covariance matrix) indicates that samples are close to μ

- Small $|\Sigma|$ may also indicate that there is a **high correlation** between variables

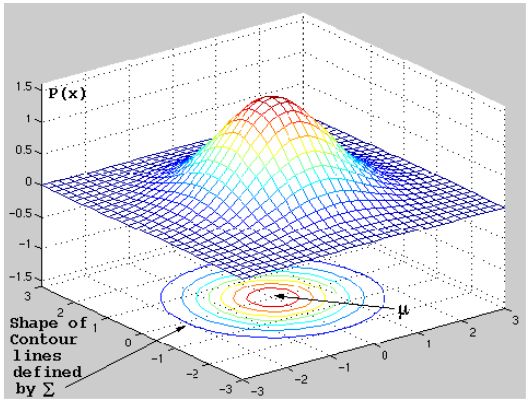
- If some of the variables are linearly dependent, or

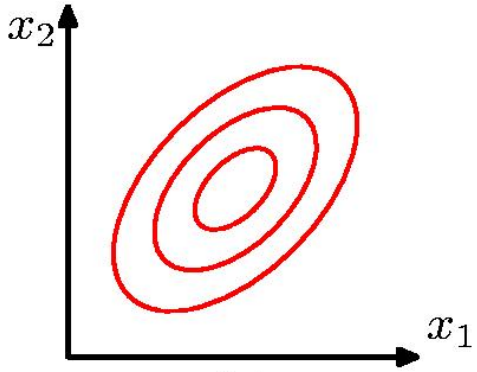
  if the variance of one variable is 0, then Σ is singular and $|\Sigma|$ is 0.

  - **Dimensionality should be reduced** to get a positive definite matrix

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

Mahalanobis Distance





From the equation for the normal density, it is apparent that points which have the **same density** must have the same constant term:

$$(\mathbf{x} - \boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$$

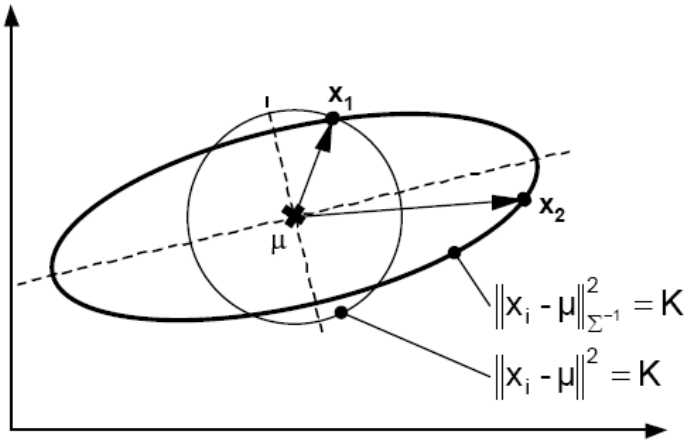
Mahalanobis distance measures the distance from $\boldsymbol{x}$ to $\boldsymbol{\mu}$ in terms of $\boldsymbol{\Sigma}$

Points that are the same distance from μ

- The quadratic term is called the Mahalanobis distance, a very important distance in Statistical PR

Mahalanobis Distance

$$\|x - y\|^2_{\Sigma^{-1}} = (x-y)^T \Sigma^{-1}(x-y)$$

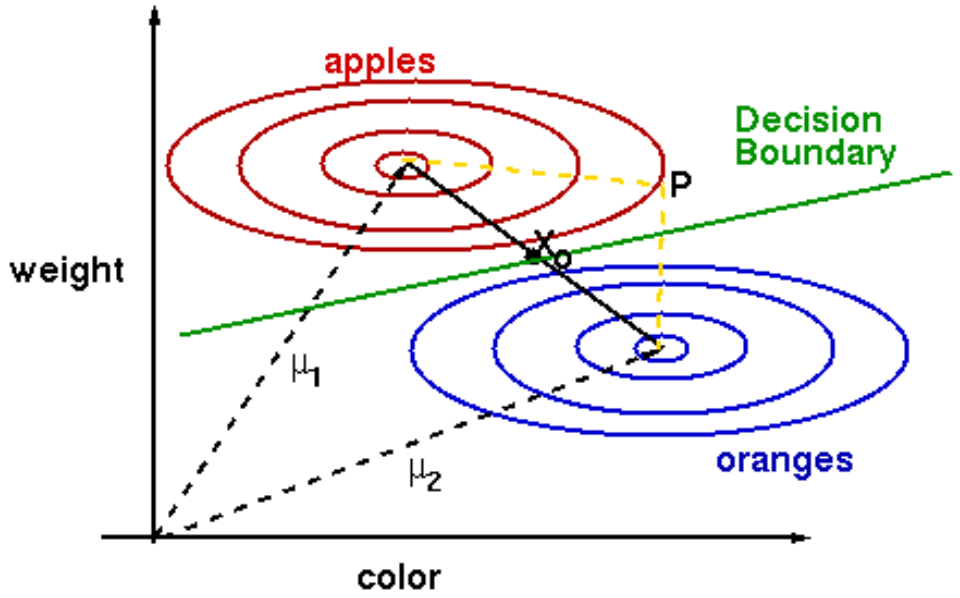


- **The Mahalanobis distance is a vector distance that uses a Σ^{-1} norm**
  - Σ^{-1} can be thought of as a stretching factor on the space
  - Note that for an identity covariance matrix (Σ=I), the Mahalanobis distance becomes the familiar Euclidean distance

•The ellipse consists of points that are equi-distant to the center w.r.t. Mahalanobis distance.

•The circle consists of points that are equi-distant to the center w.r.t. The Euclidian distance.

Why Mahalanobis Distance



It takes into account the covariance of the data.

- Point P is at closer (Euclidean) to the mean for the orange class, but using the Mahalanobis distance, it is found to be closer to 'apple' class.

Positive Semi-Definite-Advanced

- The covariance matrix Σ is ***symmetric*** and ***positive semi-definite***
- An $n \times n$ real symmetric matrix M is ***positive definite***
  if $z^{\mathrm{T}}Mz > 0$ for all non-zero vectors z with real entries.
- An $n \times n$ real symmetric matrix M is ***positive semi-definite***
  if $z^{\mathrm{T}}Mz >= 0$ for all non-zero vectors z with real entries.
- Comes from the requirement that the variance of each dimension is >= 0 and that the matrix is symmetric.
- When you randomly generate a covariance matrix, it may violate this rule
  - Test to see if all the eigenvalues are >= 0
  - Higham 2002 – how to find nearest valid covariance matrix
  - Set negative eigenvalues to small positive values

Parameter Estimation

Covered only ML estimator

- You have some samples coming from an unknown distribution and you want to characterize that distribution; i.e. find the necessary parameters.
- For instance, assuming you are given the samples in Slides 14 or 15, and that you assume that they are normally distributed, you will try to find the parameters μ and Σ of the corresponding normal distribution.
- We have referred to the sample mean (mean of the samples, m) and sample variance S before, but why use those instead of μ and Σ?

- Given an i.i.d data set X, sampled from a normal distribution with parameters μ and Σ, we would like to determine these parameters, given the samples.

- Once we have those, we can estimate p(x) for any given x (little x), given the known distribution.

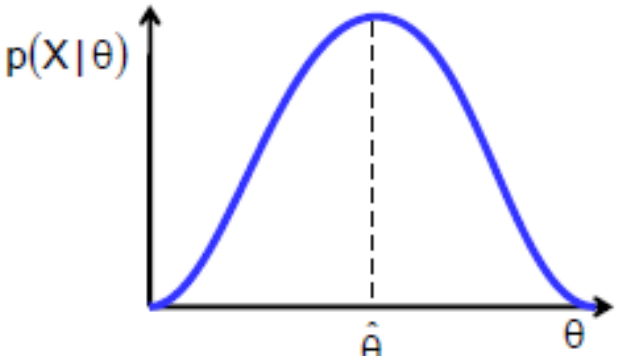
- Two approaches in parameter estimation:
  - Maximum Likelihood approach
  - Bayesian approach

Maximum Likelihood vs. Bayesian Parameter Estimation

- **Maximum Likelihood**
  - The parameters are assumed to be FIXED but unknown
  - The ML solution seeks the solution that "best" explains the dataset X

$$\hat{\theta} = \text{argmax}[p(X \mid \theta)]$$

MAXIMUM LIKELIHOOD



Maximum Likelihood vs. Bayesian Parameter Estimation

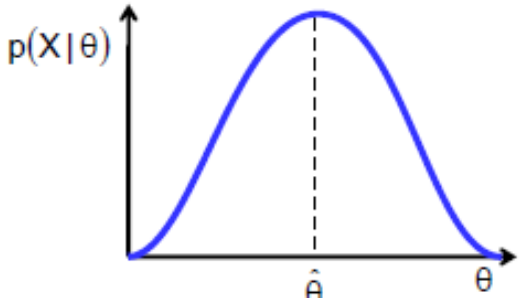
- **Maximum Likelihood**
  - The parameters are assumed to be FIXED but unknown
  - The ML solution seeks the solution that "best" explains the dataset X

$$\hat{\theta} = \text{argmax}[p(X \mid \theta)]$$

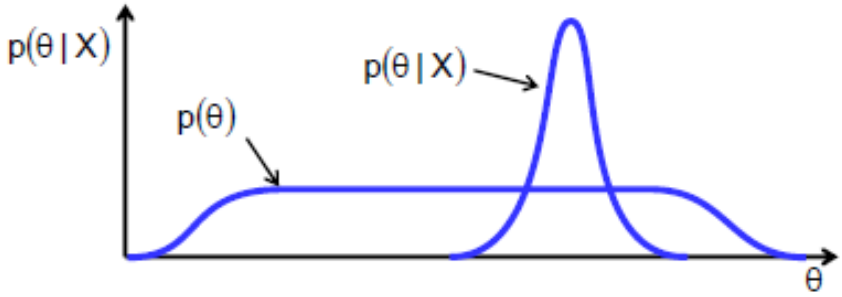
- **Bayesian Estimation**
  - The parameters are assumed to be random variables with some (assumed) known a priori distribution
  - Bayesian methods seeks to estimate the posterior density $p(\theta|X)$
  - The final density $p(x|X)$ is obtained by integrating out the parameters:

$$p(x \mid X) = \int p(x \mid \theta) p(\theta \mid X) d\theta$$

MAXIMUM LIKELIHOOD



BAYESIAN



Maximum Likelihood (1)

- **Suppose we consider estimating a density function p(x) which depends on a number of parameters $\theta=[\theta_1, \theta_2,\ldots\theta_M]^T$**
  - For a Gaussian pdf $\theta_1=\mu$, $\theta_2=\sigma$ and $p(x)=N(\mu, \sigma)$
  - To make the dependence on the parameters θ explicit we write $p(x|\theta)$
- **Assume that we have a number of examples $X=\{x^{(1}, x^{(2},\ldots x^{(N}\}$ drawn independently from the distribution $p(x|\theta)$ (an i.i.d. set)**
  - Then we can write

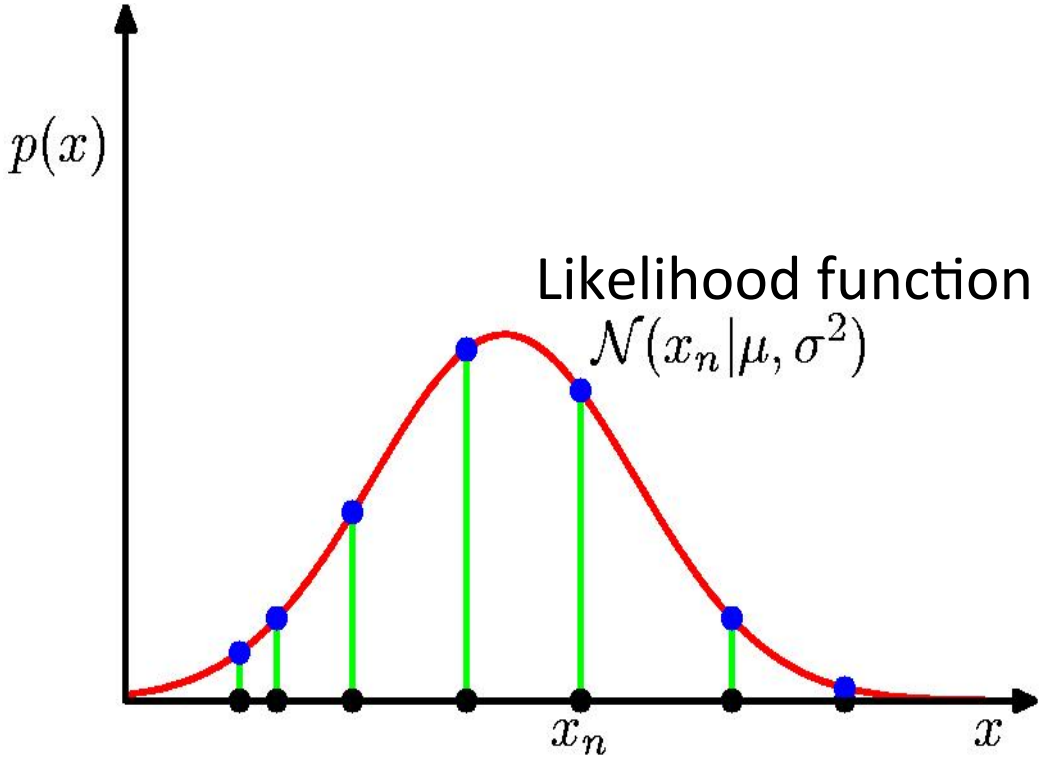
$$p(X|\theta)=\prod_{k=1}^{N} p(x^{(k}|\theta)$$

  - The ML estimate of θ is the value that maximizes the likelihood $p(X|\theta)$

$$\hat{\theta}=\text{argmax}[p(X|\theta)]$$

    - This corresponds to the intuitively pleasing idea of choosing the value of θ that is most likely to give rise to the data!

Gaussian Parameter Estimation



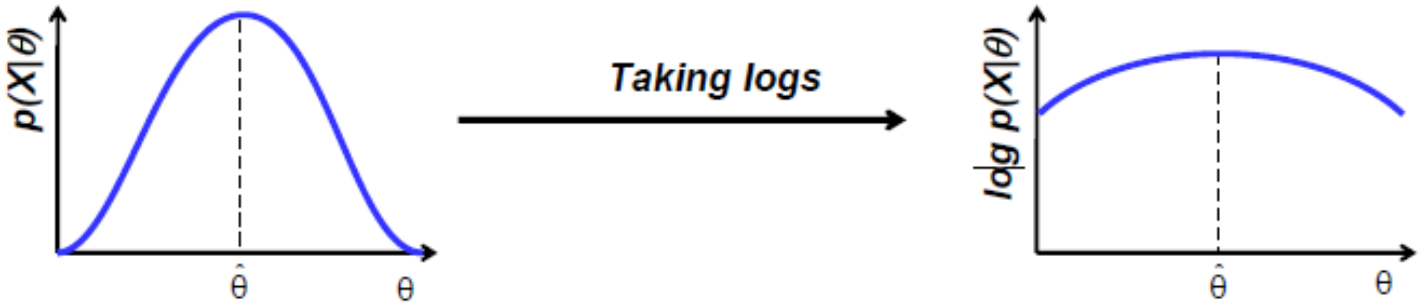
$$p(\mathbf{x}|\mu,\sigma^2) = \prod_{n=1}^{N} \mathcal{N}\left(x_n|\mu,\sigma^2\right)$$

Assuming iid data

Maximum Likelihood (2)

- **For analytical purposes it is convenient to work with the log of the likelihood**
  - Since the log is a monotonic function

$$\hat{\theta} = \text{argmax}\left[p(X \mid \theta)\right] = \text{argmax}\left[\log p(X \mid \theta)\right]$$



- **Then the Maximum Likelihood estimate of the parameter θ can be written as**

$$\hat{\theta} = \text{argmax}\left[\log \prod_{k=1}^{N} p\left(x^{(k} \mid \theta\right)\right] = \text{argmax}\left[\sum_{k=1}^{N} \log p\left(x^{(k} \mid \theta\right)\right]$$

  - Maximizing a sum of terms is always an easier task than maximizing a product
    - To convince yourself, think of computing the derivative of a long product of terms!
  - An added advantage of taking logs will become very clear when the distribution is Gaussian

Example: Gaussian case, μ unknown

- Assume a dataset $X=\{x^{(1}, x^{(2},\ldots x^{(N}\}$ and a density of the form $p(x)=N(\mu,\sigma)$ where the standard deviation σ is known
- What is the Maximum Likelihood estimate of the mean?

$$\theta=\mu \Rightarrow \hat{\theta}=\operatorname{argmax}\sum_{k=1}^{N}\log p\left(x^{(k}\mid\theta\right)$$

$$=\operatorname{argmax}\sum_{k=1}^{N}\log\left(\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2\sigma^2}\left(x^{(k}-\mu\right)^2\right)\right)$$

$$=\operatorname{argmax}\sum_{k=1}^{N}\left\{\log\left(\frac{1}{\sqrt{2\pi}\sigma}\right)-\frac{1}{2\sigma^2}\left(x^{(k}-\mu\right)^2\right\}$$

- The maxima (or minima) of a function are defined by the zeros of its derivative:

$$\frac{\partial\sum_{k=1}^{N}\log p\left(x^{(k}\mid\theta\right)}{\partial\theta}=\frac{\partial}{\partial\mu}\sum_{k=1}^{N}\{\bullet\}=0\Rightarrow\mu=\frac{1}{n}\sum_{k=1}^{N}x^{(k}$$

- So the ML estimate of the mean is the average value of the training data, a very intuitive result!

Reminder

- In order to maximize or minimize a function f(x) w.r.to x, we compute the derivative df(x)/dx and set to 0, since a necessary condition for extrema is that the derivative is 0.
- Commonly used derivative rules are given on the right.

$$\frac{d}{dx}\left(x^n\right) = nx^{n-1}$$

$$\frac{d}{dx}\left(a^x\right) = \ln a \cdot a^x$$

$$\frac{d}{dx}\left(f(x)\cdot g(x)\right) = f(x)\cdot g'(x) + g(x)\cdot f'(x)$$

$$\frac{d}{dx}\left(\frac{f(x)}{g(x)}\right) = \frac{g(x)\cdot f'(x) - f(x)\cdot g'(x)}{\left(g(x)\right)^2}$$

$$\frac{d}{dx}\left(f(g(x))\right) = f'(g(x))\cdot g'(x)$$

$$\frac{d}{dx}\left(\ln x\right) = \frac{1}{x}$$

Derivation – general case-ADVANCED

- **This is a more general case when neither the mean nor the standard deviation are known**
  - Fortunately, the problem can be solved in the same fashion
  - In this case, the derivative becomes a gradient since we have two variables

$$\hat{\theta} = \begin{bmatrix} \theta_1 = \mu \\ \theta_2 = \sigma^2 \end{bmatrix} \Rightarrow \nabla_\theta = \begin{bmatrix} \frac{\partial}{\partial \theta_1} \sum_{k=1}^{N} \log p\left(x^{(k} \mid \theta\right) \\ \frac{\partial}{\partial \theta_2} \sum_{k=1}^{N} \log p\left(x^{(k} \mid \theta\right) \end{bmatrix} = \sum_{k=1}^{N} \begin{bmatrix} \frac{1}{\theta_2}\left(x^{(k} - \theta_1\right) \\ -\frac{1}{2\theta_2} + \frac{\left(x^{(k} - \theta_1\right)^2}{2\theta_2^2} \end{bmatrix} = 0$$

  - Solving for θ_1 and θ_2 yields

$$\hat{\theta}_1 = \frac{1}{N}\sum_{k=1}^{N} x^{(k}; \quad \hat{\theta}_2 = \frac{1}{N}\sum_{k=1}^{N}\left(x^{(k} - \hat{\theta}_1\right)^2$$

    - Therefore, the ML of the variance is the sample variance of the dataset, again a very pleasing result

Maximum (Log) Likelihood for Multivarate Case

Similarly, it can be shown that the Maximum Likelihood parameter estimates for the multivariate Gaussian are also the sample mean vector and sample covariance matrix

$$\hat{\mu} = \frac{1}{N}\sum_{k=1}^{N} x^{(k}; \; \hat{\Sigma} = \frac{1}{N}\sum_{k=1}^{N} \left(x^{(k} - \hat{\mu}\right)\left(x^{(k} - \hat{\mu}\right)^{T}$$

$\mu =$

Maximum (Log) Likelihood for a 1D-Gaussian

$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}}\exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

$$\ln p\left(\mathbf{x}|\mu,\sigma^2\right) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi)$$

$$\mu_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}x_n \qquad \sigma^2_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}(x_n-\mu_{\mathrm{ML}})^2$$

In other words, **maximum likelihood estimates of mean and variance are the same as sample mean and variance.**

Sample Mean and Variance for Multivariate Case

Where N is the number of data points x^t:

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

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

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

ML estimates for the mean and variance in Bernouilli/ Multinomial distributions

Not covered in class – but you should be able to do as a take home etc. Same idea as before, starting from the likelihood, you find the value that maximizes the likelihood.

Examples: Bernoulli/Multinomial

- **Bernouilli:** Binary random variable x may take one of the two values:
  - success/failure, 0/1 with probabilities:
  - $P(x=1)= p_o$
  - $P(x=0)=1-p_o$
  - *Unifying the above, we get:* $P(x) = p_o^{x}(1 - p_o)^{(1 - x)}$

- Given a sample set X={x^1,x^2,…}, we can estimate *p* using the ML estimate by maximizing the log-likelihood of the sample set:

Log likelihood: $\log \mathcal{P}(\mathrm{X}|p_o) = \log \Pi_t \, p_o^{x^t}(1 - p_o)^{(1 - x^t)}$

Examples: Bernoulli

$$\mathcal{L}=\mathcal{P}(\mathrm{X}|p_o) = \log \Pi_t \, p_o^{x^t} (1 - p_o)^{(1 - x^t)} \qquad x^t \text{ in } \{0,1\}$$

Solving for the necessary condition for extrema, (we must have dL/dp = 0)

…

MLE: $p_o = \Sigma_t \, x^t / N$

ratio of the number of occurences of the event to the number of experiments

Examples: Multinomial

- **Multinomial:** $K>2$ states, x_i in $\{0,1\}$
- Instead of two states, we now have K mutually exclusive and exhaustive events, with probability of occurence of
  p_i where $\Sigma p_i = 1$.
- Ex. A dice with 6 outcomes.

$$P(x_1,x_2,...,x_K) = \Pi_i \, p_i^{x_i}$$ where x_i is 1 if the outcome is state i
0 otherwise

$$\mathcal{L}(p_1,p_2,...,p_K|\mathcal{X}) = \log \Pi_t \Pi_i \, p_i^{x_i^t}$$

MLE: $p_i = \Sigma_t \, x_i^t / N$

Ratio of experiments with outcome of state i
(e.g. 60 dice throws, 15 of them were 6 ⟶ p_6 = 15/60

Discrete Features

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

If x_j are independent (Naive Bayes' assumption)

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

r_i=1 if x^t in C_i
0 otherwise

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

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: $x_j \in \{v_1, v_2, ..., v_{n_j}\}$

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

if x_j are independent

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

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

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

Parametric Classification

We will use the Bayesian decision criteria applied to normally distributed classes, whose parameters are either known or estimated from the sample.

Parametric Classification

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

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

- Discriminant functions are:

$$\begin{aligned} g_i(\mathbf{x}) &= \log P(C_i \mid \mathbf{x}) \\ &= \log p(\mathbf{x} \mid C_i) + \log P(C_i) \\ &= -\frac{d}{2} \log 2\pi - \frac{1}{2} \log |\Sigma_i| - \frac{1}{2} (\mathbf{x} - \mu_i)^T \Sigma_i^{-1} (\mathbf{x} - \mu_i) + \log P(C_i) \end{aligned}$$

Estimation of Parameters

If we estimate the unknown parameters from the sample, the discriminant function becomes:

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

Case 1) Different $\mathbf{S}_i$ (each class has a separate variance)

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

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

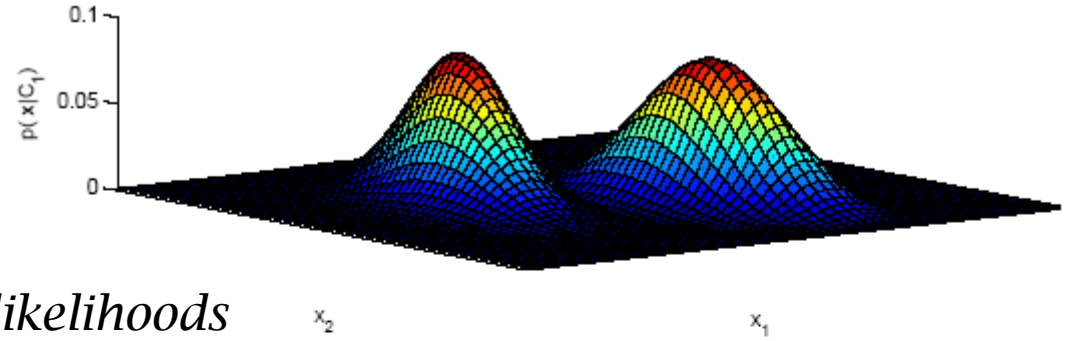
where

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

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

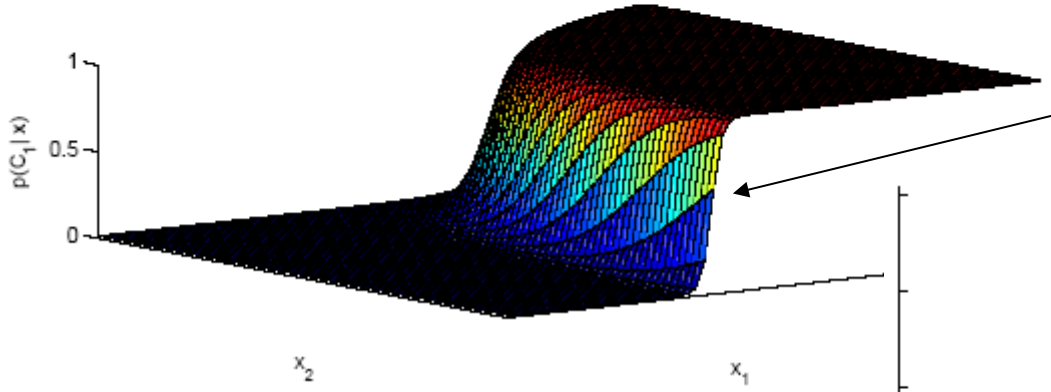
$$w_{i0} = -\frac{1}{2}\boldsymbol{m}_i^T\mathbf{S}_i^{-1}\boldsymbol{m}_i - \frac{1}{2}\log|\mathbf{S}_i| + \log\hat{P}(C_i)$$

if we group the terms, we see that there are second order terms, which means the **discriminant is quadratic.**

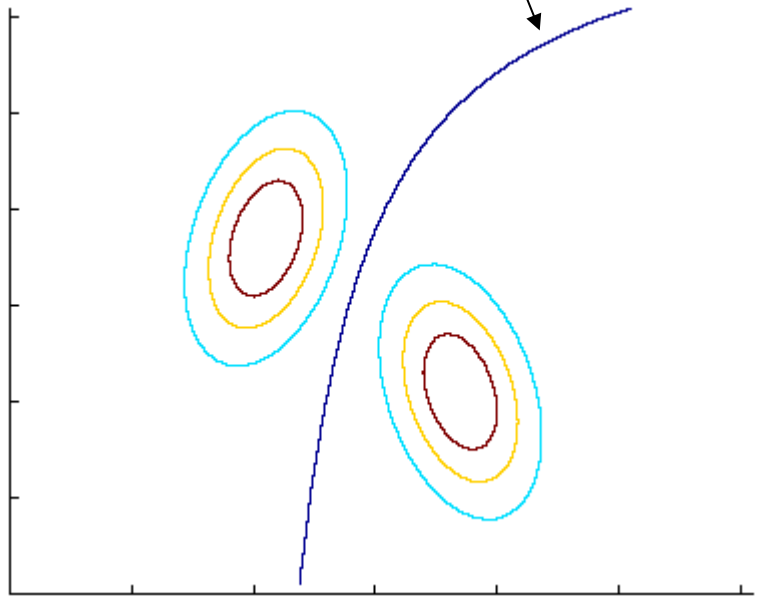


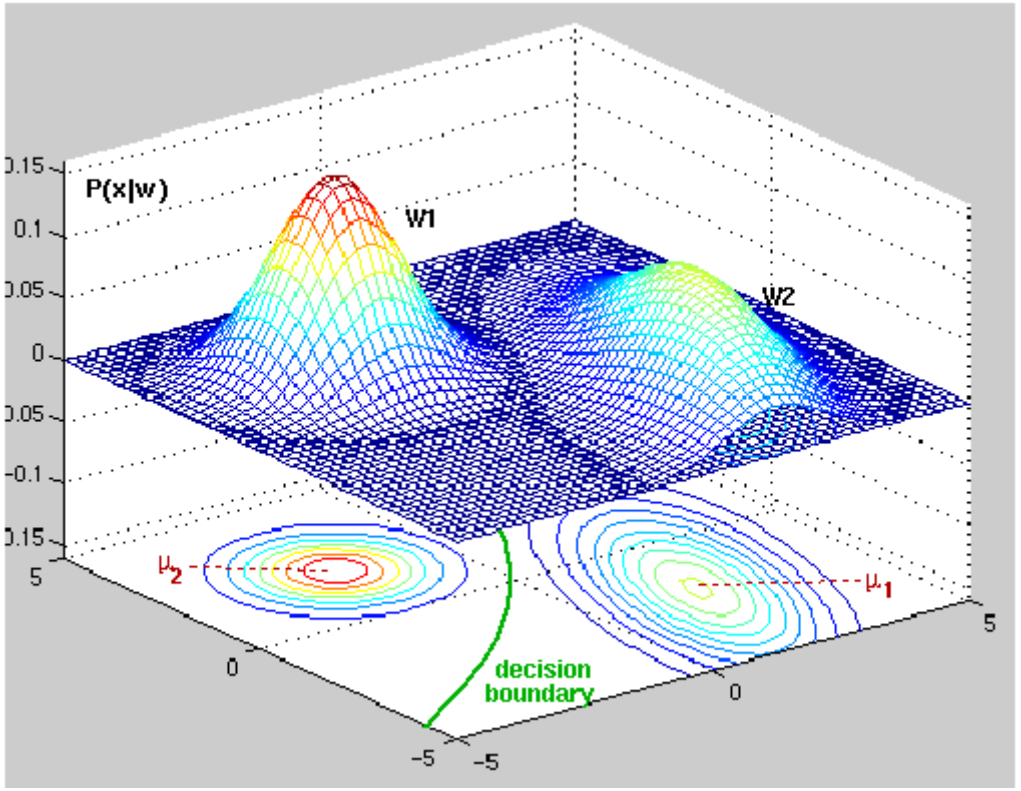
likelihoods

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

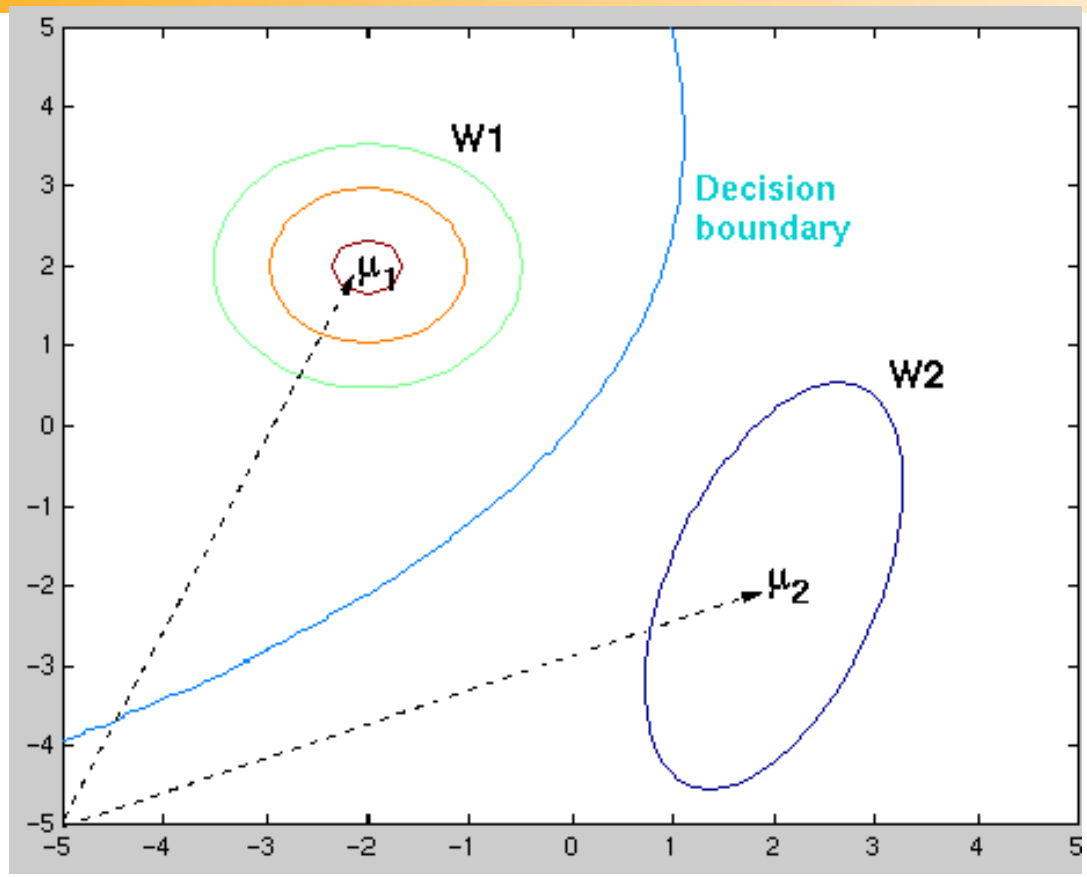


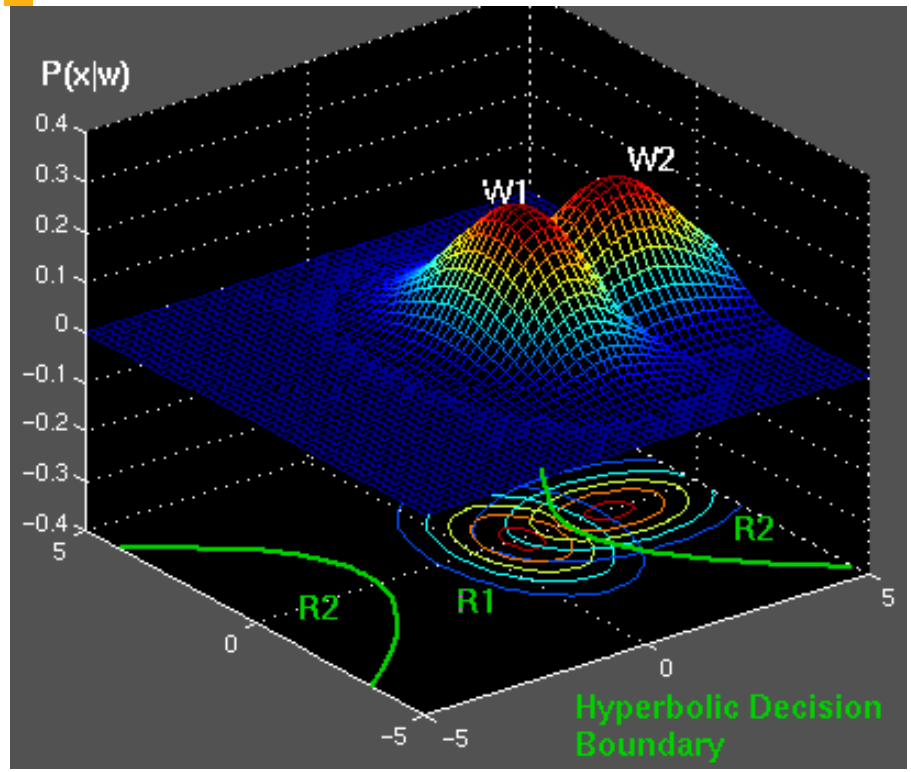
posterior for C_1

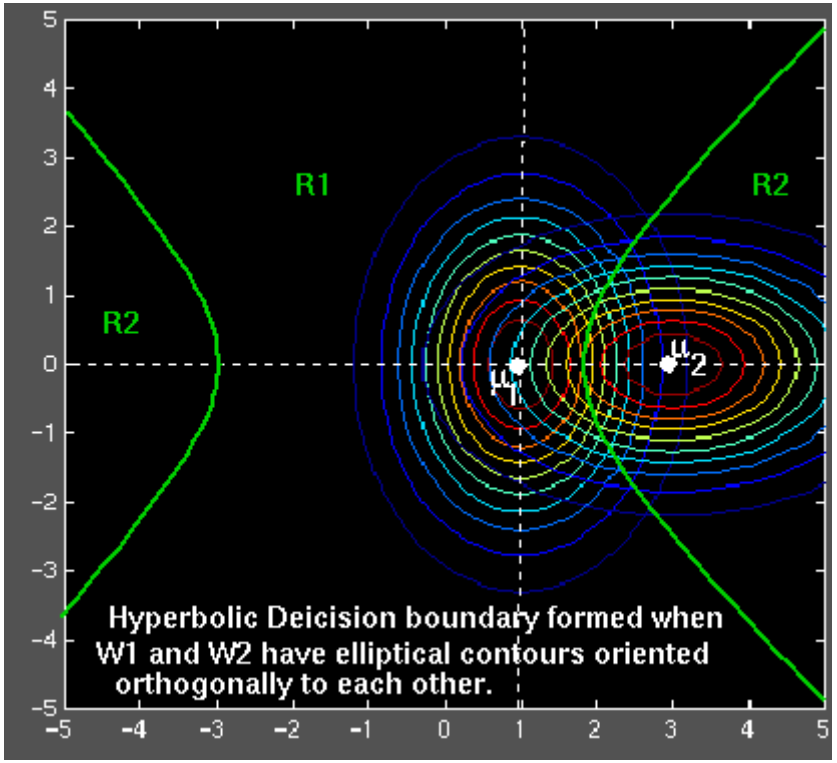




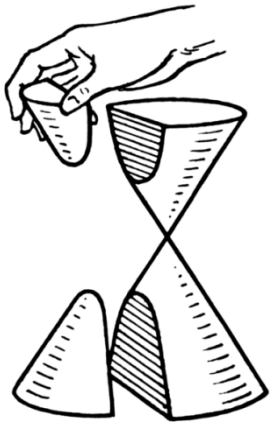
Two bi-variate normals, with completely different covariance matrix, showing a hyper-quadratic decision boundary.


W1
Decision
boundary
W2
μ_1
μ_2
5
4
3
2
1
0
−1
−2
−3
−4
−5
−5
−4
−3
−2
−1
0
1
2
3
4
5





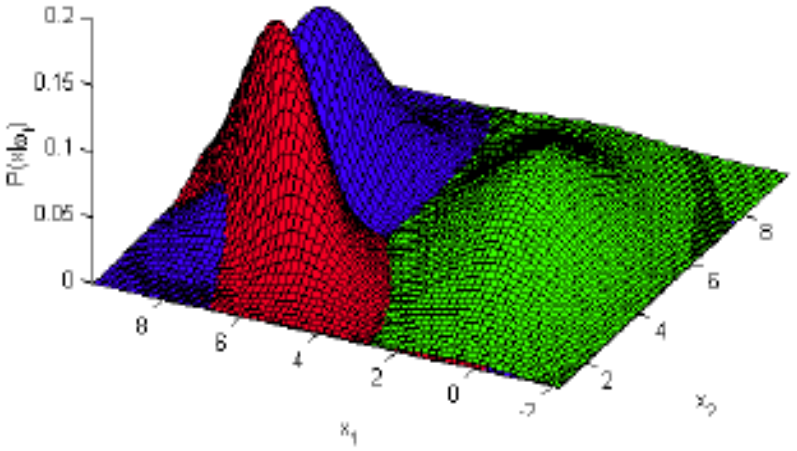
Hyperbola: A hyperbola is an open curve with two branches, the intersection of a plane with both halves of a double cone. The plane may or may not be parallel to the axis of the cone. *Definition from Wikipedia.*

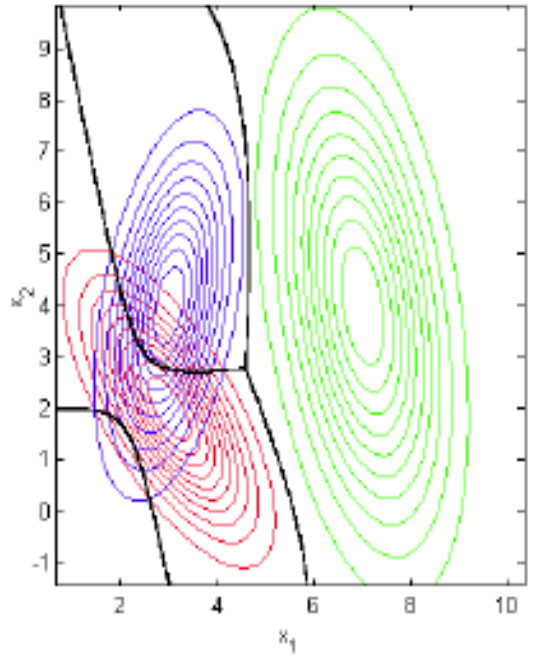


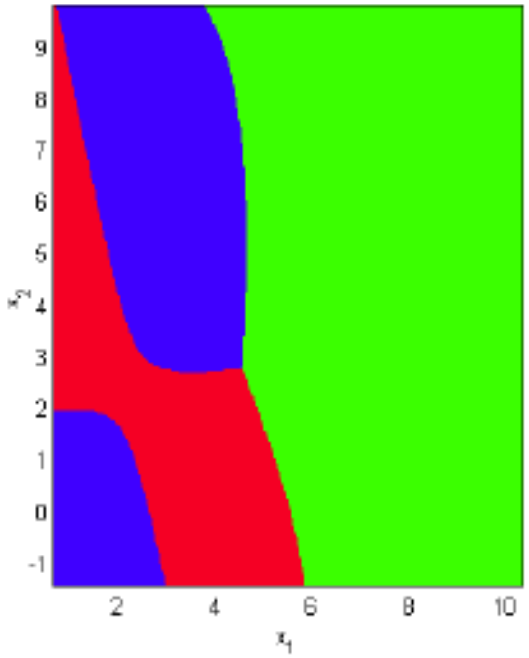
- To illustrate the previous result, we will compute the decision boundaries for a 3-class, 2-dimensional problem with the following class mean vectors and covariance matrices and equal priors

$$\mu_1 = [3 \quad 2]^T \qquad \mu_2 = [5 \quad 4]^T \qquad \mu_3 = [2 \quad 5]^T$$

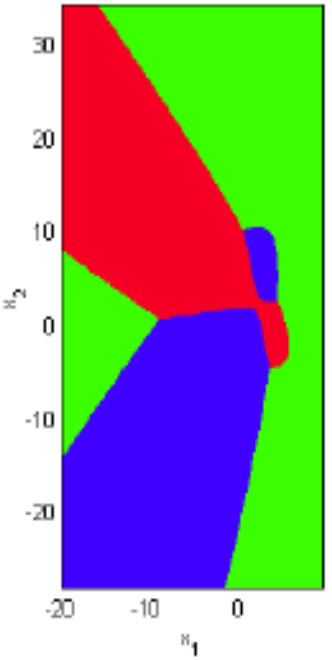
$$\Sigma_1 = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix} 1 & -1 \\ -1 & 7 \end{bmatrix} \qquad \Sigma_3 = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 3 \end{bmatrix}$$

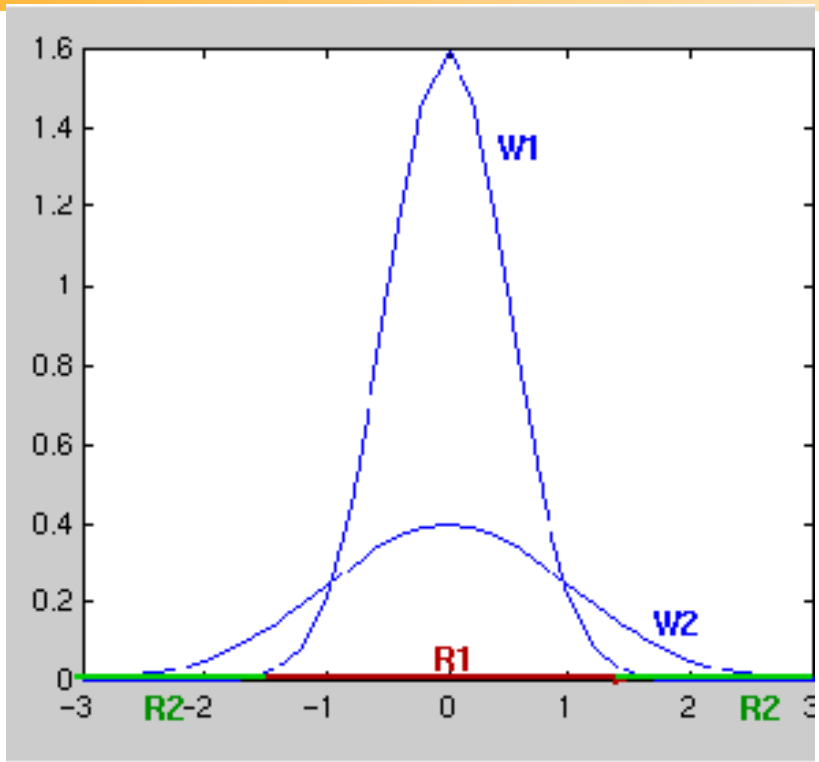






Zoom out





Typical single-variable normal distributions showing a disconnected decision region R_2

Notation: Ethem book

Using the notation in Ethem book, the sample mean and sample covariance… can be estimated as follows:

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

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

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

where r_i^t is 1 if the t_{th} sample belongs to class i

- If d (dimension) is large with respect to N (number of samples), we may have a problem with this approach:
  - $|\Sigma|$ may be zero, thus Σ will be singular (inverse does not exist)
  - $|\Sigma|$ may be non-zero, but very small, instability would result
    - Small changes in Σ would cause large changes in Σ^{-1}

- Solutions:
  - Reduce the dimensionality
    - Feature selection
    - Feature extraction: PCA
  - Pool the data and estimate a common covariance matrix for all classes

$$\Sigma = \Sigma_i \, P(C_i) * \Sigma_i$$

Case 2) Common Covariance Matrix $S=S_i$

- Shared common sample covariance **S**
  - An arbitrary covariance matrix – but shared between the classes

- We had this full discriminant function:

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

which now reduces to:

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

which is a linear discriminant

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

where

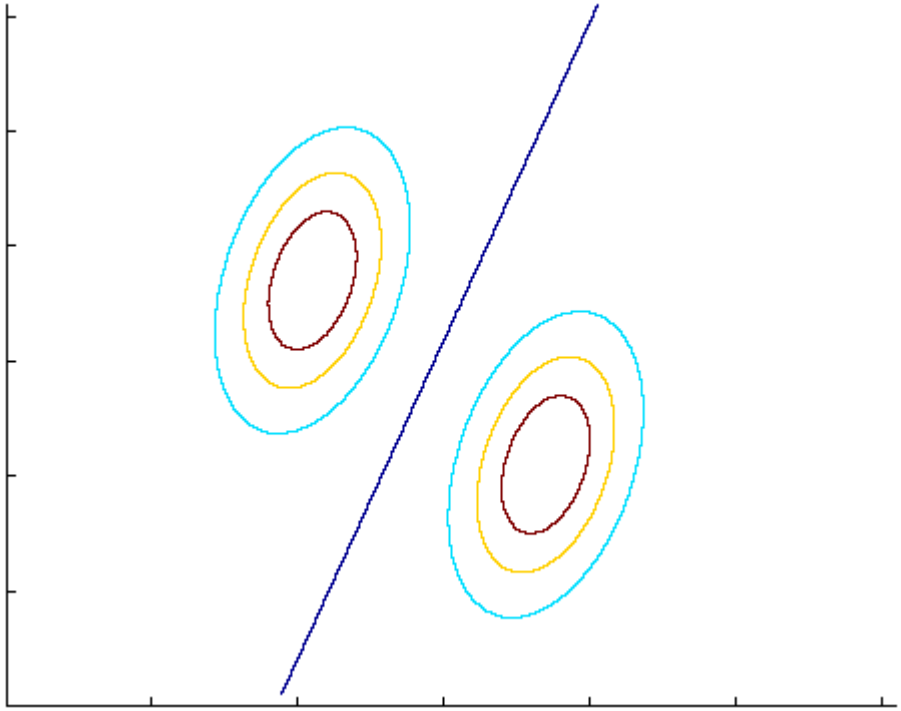
$$\mathbf{w}_i = \mathbf{S}^{-1}\mathbf{m}_i \quad w_{i0} = -\frac{1}{2}\mathbf{m}_i^T\mathbf{S}^{-1}\mathbf{m}_i + \log\hat{P}(C_i)$$

- Linear discriminant:
  - **Decision boundaries are hyper-planes**
  - **Convex** decision regions:
    - All points between two arbitrary points chosen from one decision region belongs to the same decision region

- **If we also assume equal class priors**, the classifier becomes a minimum Mahalanobis classifier

Case 2) Common Covariance Matrix S

Linear discriminant



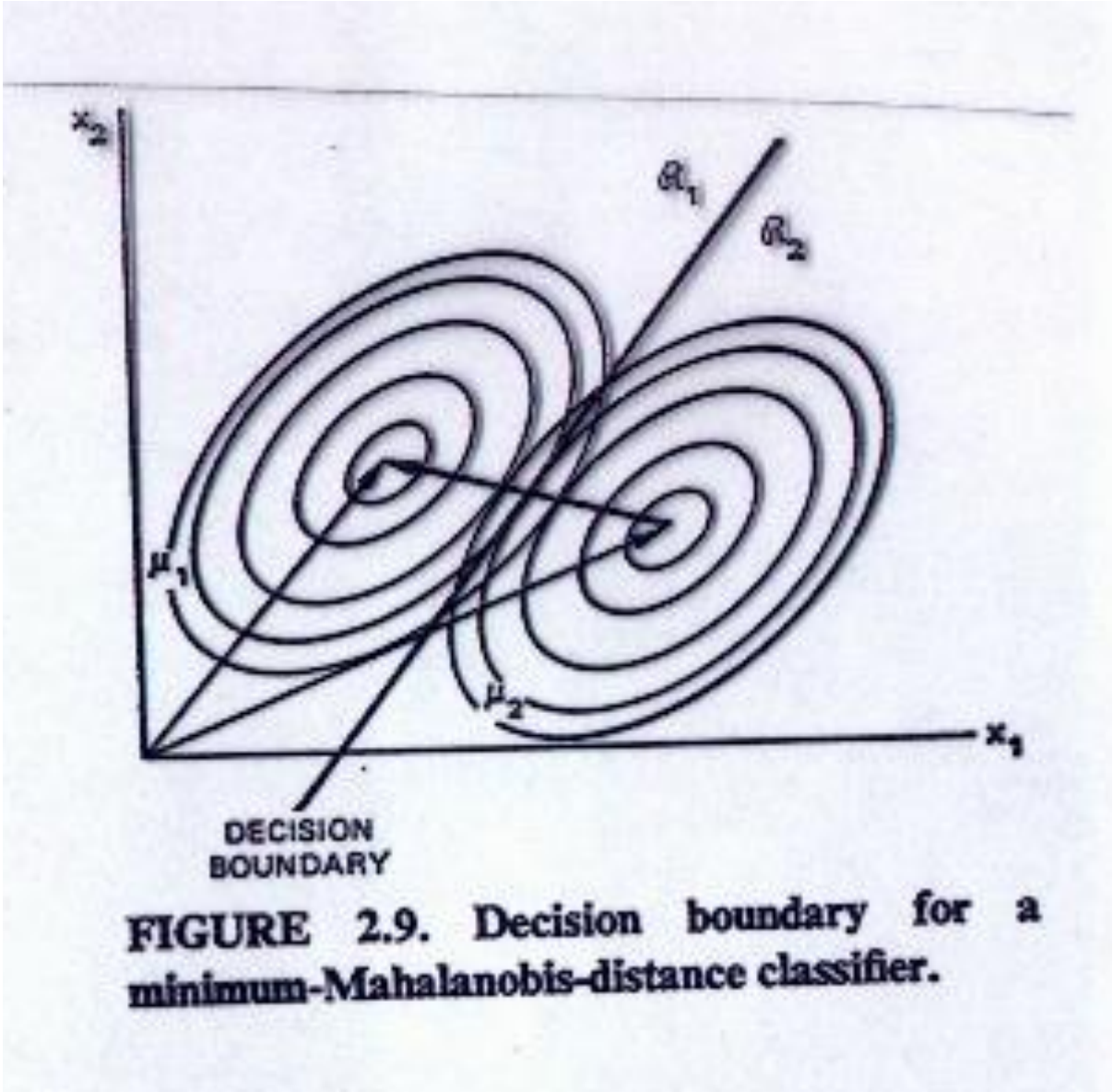
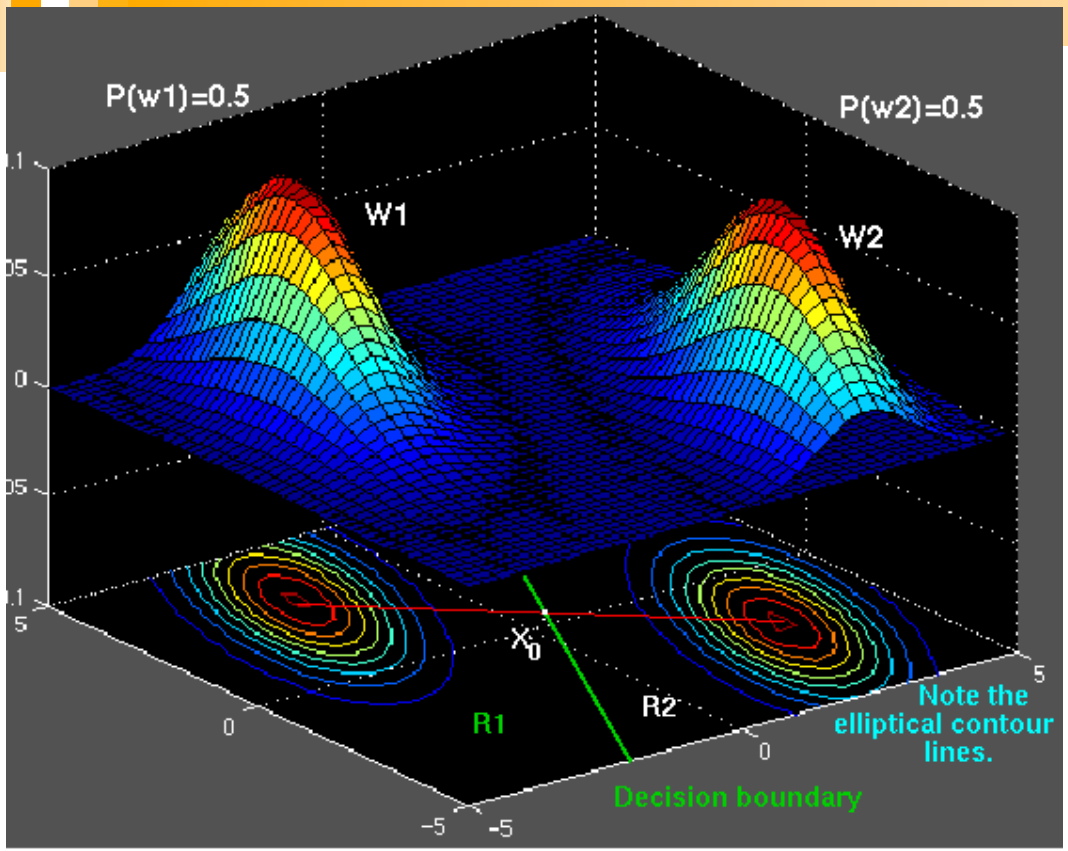
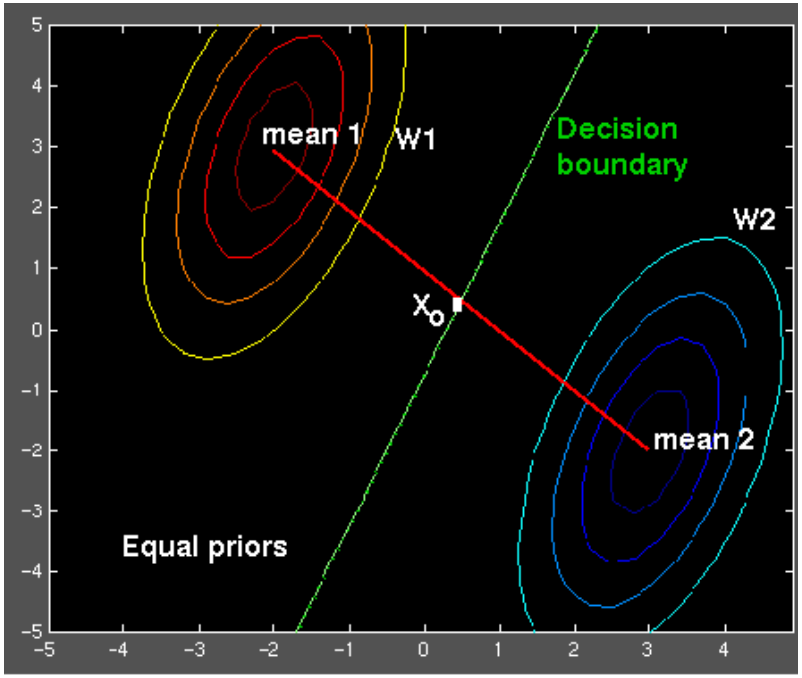
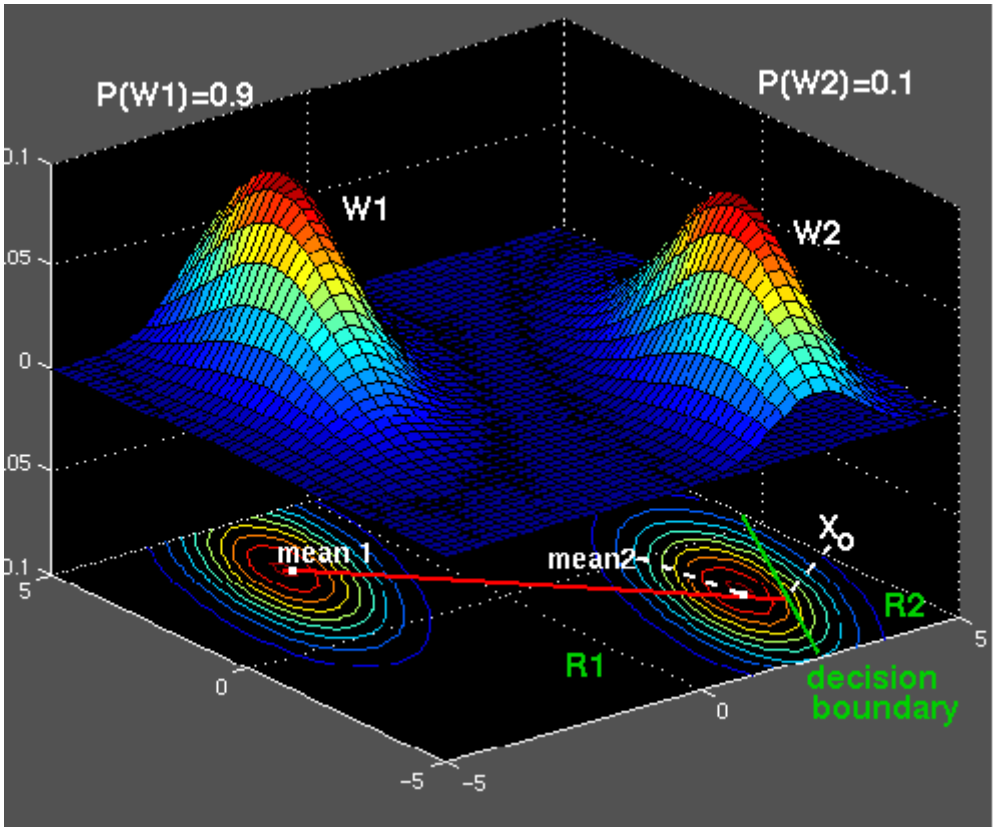


FIGURE 2.9. Decision boundary for a minimum-Mahalanobis-distance classifier.


P(w1)=0.5
P(w2)=0.5
W1
W2
X_0
R1
R2
Note the elliptical contour lines.
Decision boundary


mean 1
W1
Decision boundary
W2
X_0
mean 2
Equal priors

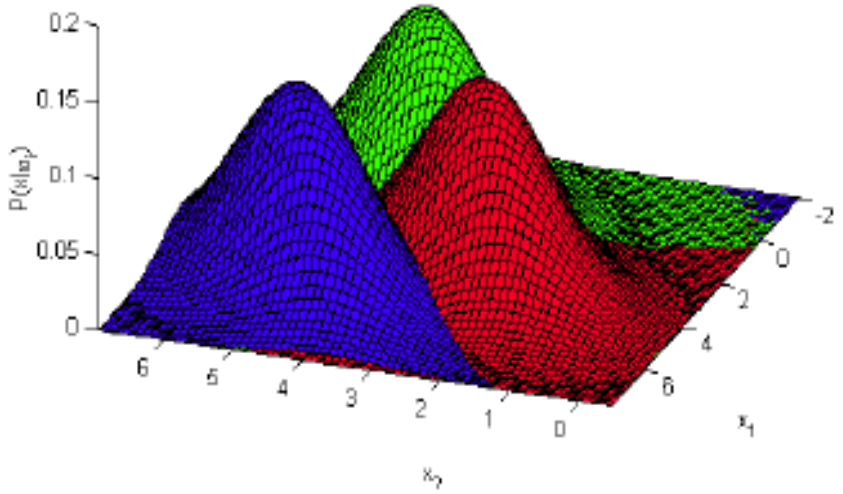


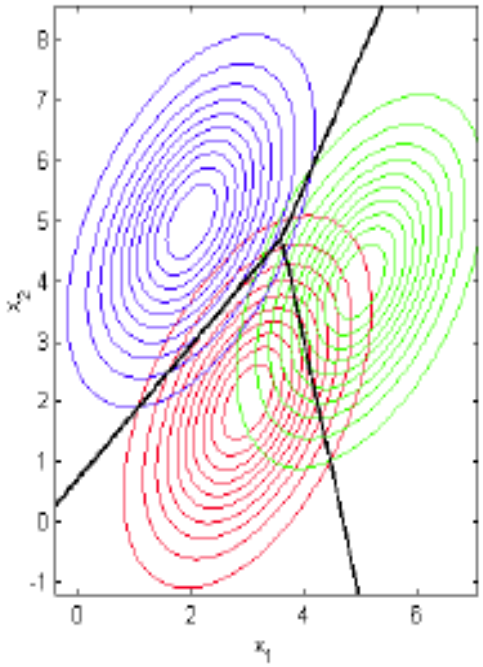
Unequal priors shift the decision boundary towards the less likely class, as before.

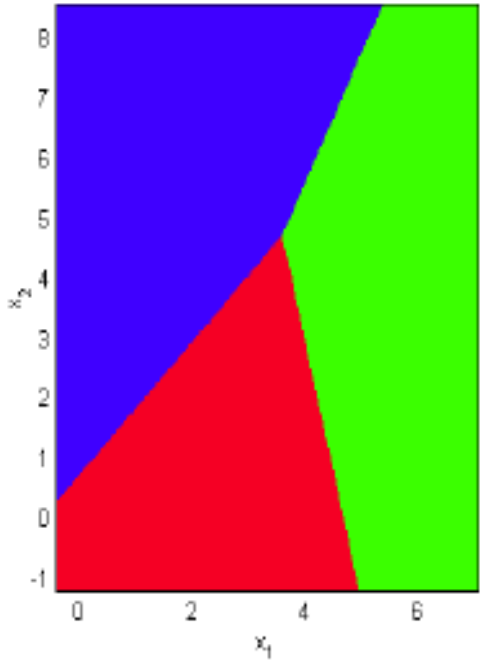
- **To illustrate the previous result, we will compute the decision boundaries for a 3-class, 2-dimensional problem with the following class mean vectors and covariance matrices and equal priors**

$$\mu_1 = \begin{bmatrix} 3 & 2 \end{bmatrix}^T \quad \mu_2 = \begin{bmatrix} 5 & 4 \end{bmatrix}^T \quad \mu_3 = \begin{bmatrix} 2 & 5 \end{bmatrix}^T$$

$$\Sigma_1 = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 2 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 2 \end{bmatrix} \quad \Sigma_3 = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 2 \end{bmatrix}$$







Case 3) Common Covariance Matrix S which is Diagonal

- In the previous case, we had a common, general covariance matrix, resulting in these discriminant functions:

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

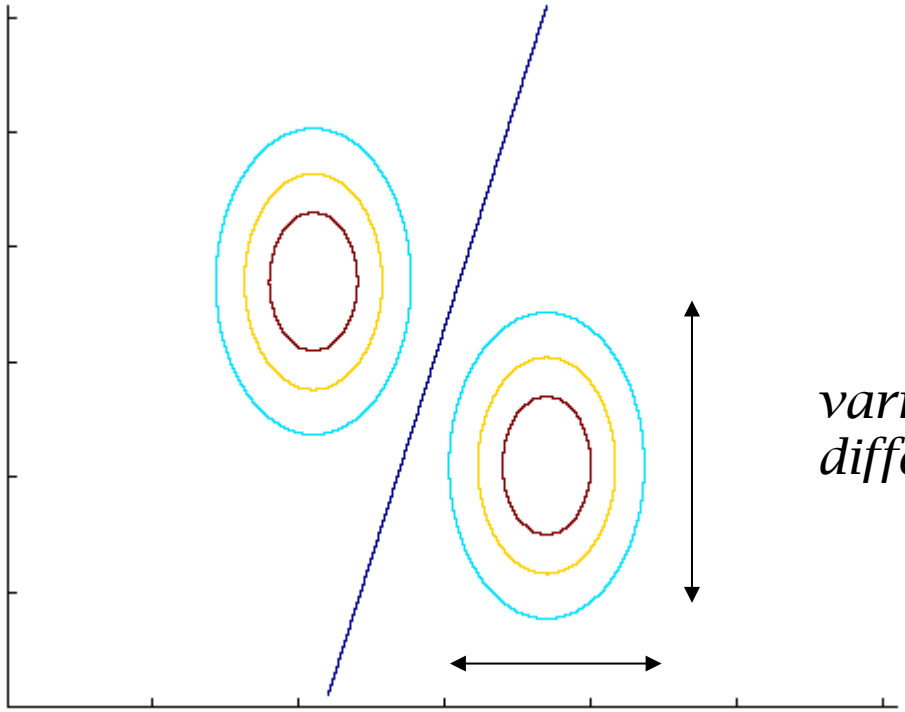
- When x_j $(j = 1,..d)$ are independent, Σ is diagonal
- Classification is done based on weighted Euclidean distance (in s_j units) to the nearest mean.

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

Naive Bayes classifier where $p(x_j|C_i)$ are univariate Gaussian

$p(\mathbf{x}|C_i) = \Pi_j\, p(x_j|C_i)$ (Naive Bayes' assumption)

Case 3) Common Covariance Matrix S which is Diagonal

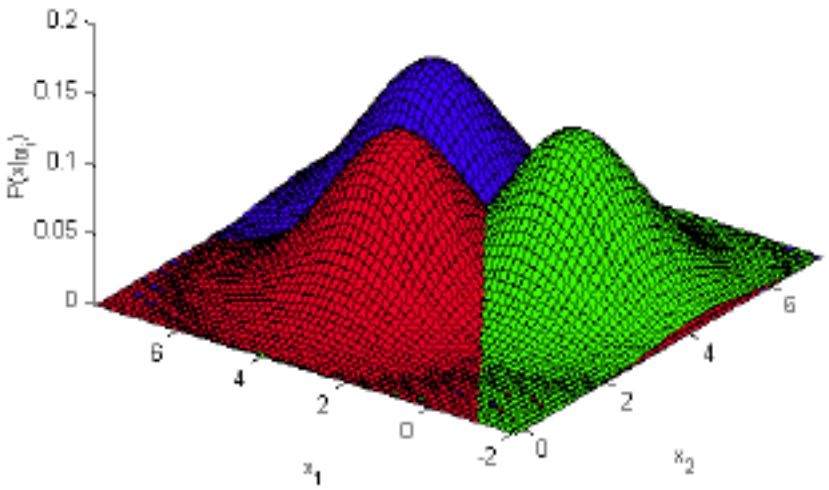


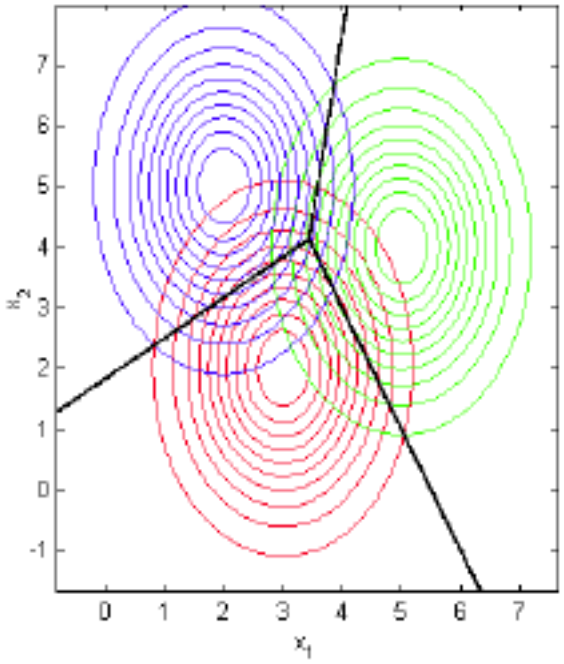
variances may be different

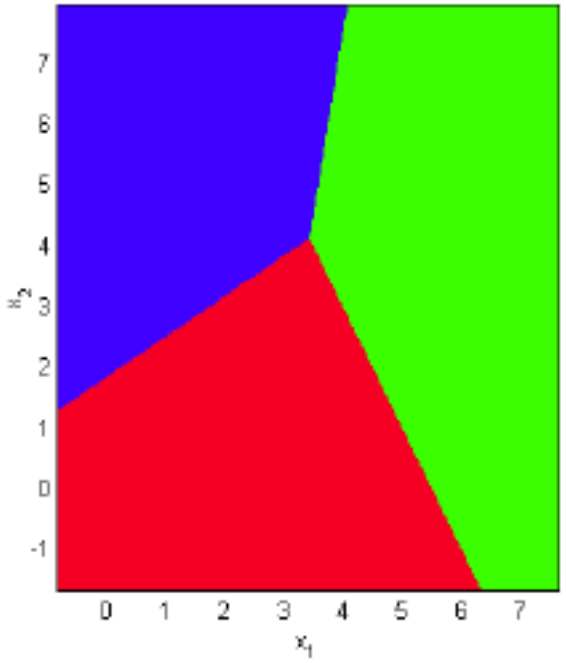
- **To illustrate the previous result, we will compute the decision boundaries for a 3-class, 2-dimensional problem with the following class mean vectors and covariance matrices and equal priors**

$$\mu_1 = [3 \quad 2]^T \quad \mu_2 = [5 \quad 4]^T \quad \mu_3 = [2 \quad 5]^T$$

$$\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \quad \Sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$







Case 4) Common Covariance Matrix S which is Diagonal + equal variances

- We had this before (S which is diagonal):

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

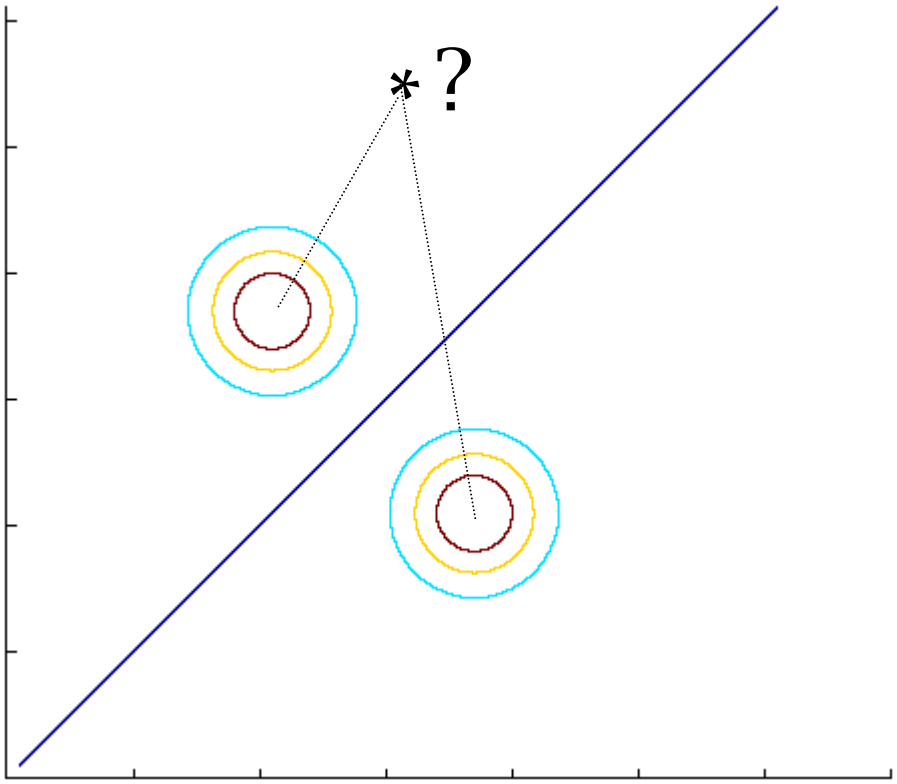
- **If the priors are also equal,** we have the **Nearest Mean classifier:**
  Classify based on Euclidean distance to the nearest mean!

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

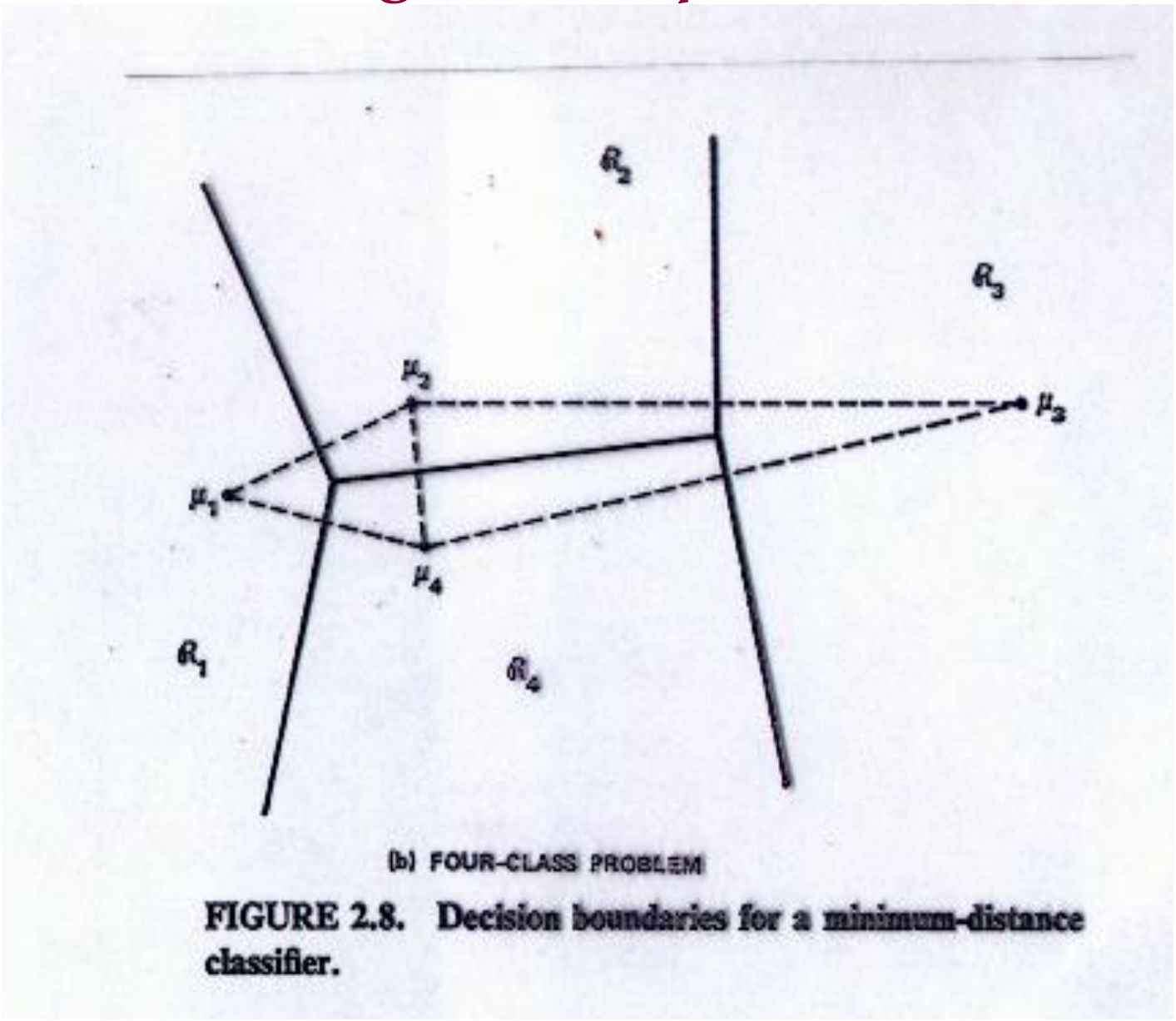
$$= -\frac{1}{2s^2}\sum_{j=1}^{d}\left(x_j^t - m_{ij}\right)^2 + \log \hat{P}(C_i)$$

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

Case 4) Common Covariance Matrix S which is Diagonal + equal variances

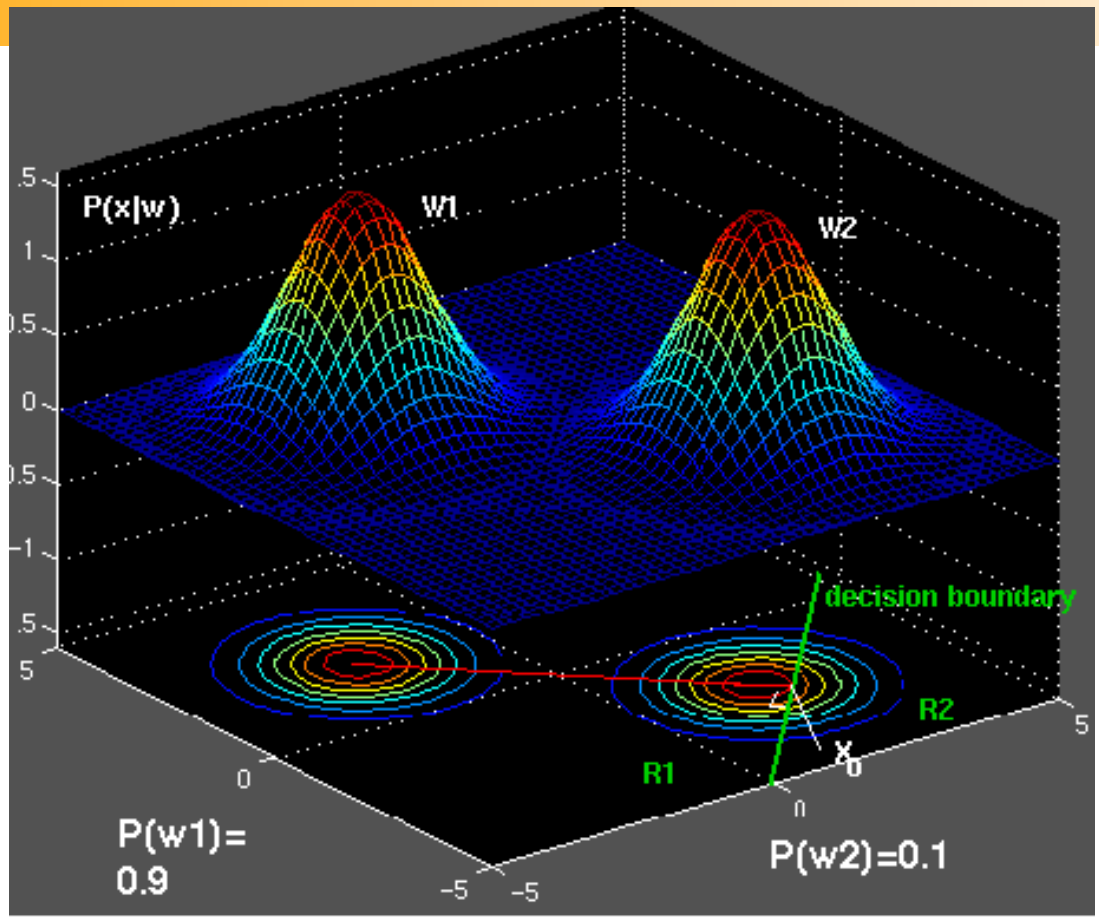


Case 4) Common Covariance Matrix S which is Diagonal + equal variances



(b) FOUR-CLASS PROBLEM

FIGURE 2.8. Decision boundaries for a minimum-distance classifier.

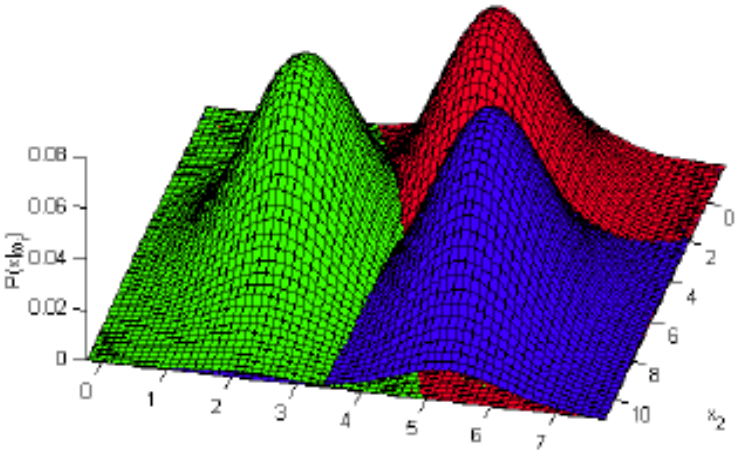


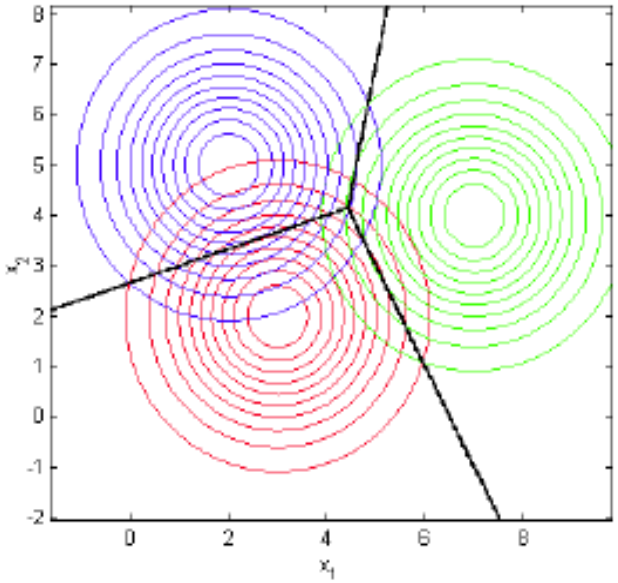
A second example where the priors have been changed:
The decision boundary has shifted away from the more likely class, although it is still orthogonal to the line joining the 2 means.

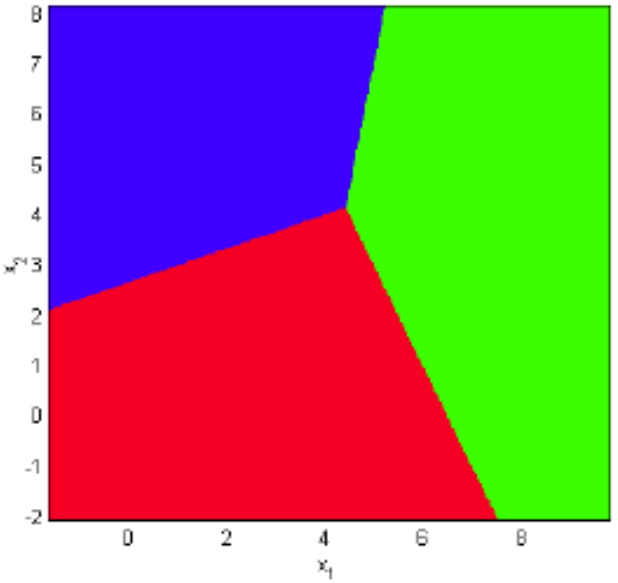
- **To illustrate the previous result, we will compute the decision boundaries for a 3-class, 2-dimensional problem with the following class mean vectors and covariance matrices and equal priors**

$$\mu_1 = \begin{bmatrix} 3 & 2 \end{bmatrix}^T \quad \mu_2 = \begin{bmatrix} 7 & 4 \end{bmatrix}^T \quad \mu_3 = \begin{bmatrix} 2 & 5 \end{bmatrix}^T$$

$$\Sigma_1 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad \Sigma_3 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$







Case 5: $S_i=\sigma_i^2 I$

- **In this case, each class has a different covariance matrix, which is proportional to the identity matrix**
  - The quadratic discriminant becomes

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i) - \frac{1}{2}\log\left(\left|\Sigma_i\right|\right) + \log(P(\omega_i)) =$$
$$= -\frac{1}{2}(x-\mu_i)^T \sigma_i^{-2}(x-\mu_i) - \frac{1}{2}N\log\left(\sigma_i^2\right) + \log(P(\omega_i))$$

- **This expression cannot be reduced further so**
  - The decision boundaries are quadratic: hyper-ellipses
  - The loci of constant probability are hyper-spheres aligned with the feature axis

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)

Estimation of Missing Values

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

- Another important problem is sensitivity due to outliers

Tuning Complexity

- When we use Euclidian distance to measure similarity, we are assuming that all variables have the same variance and that they are independent
  - E.g. Two variables age and yearly income

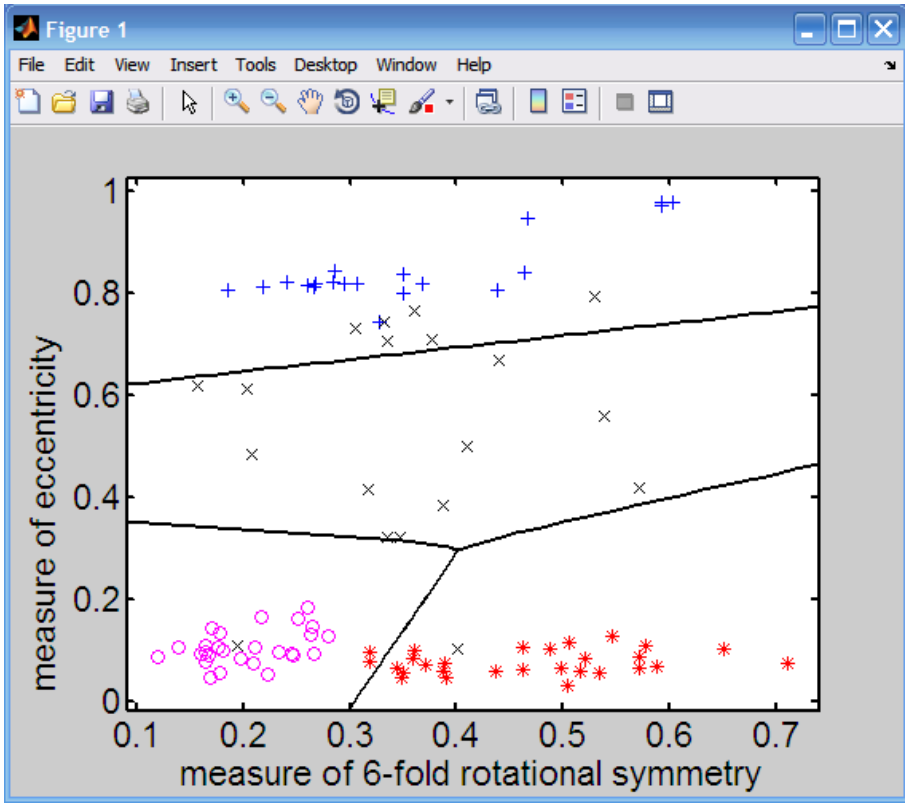
- When these assumptions don't hold,
  - Normalization may be used (use PCA, whitening, make each dimension zero mean and unit variance…) to use Euclidian distance
  - We may still want to use simpler models in order to estimate the related parameters more accurately

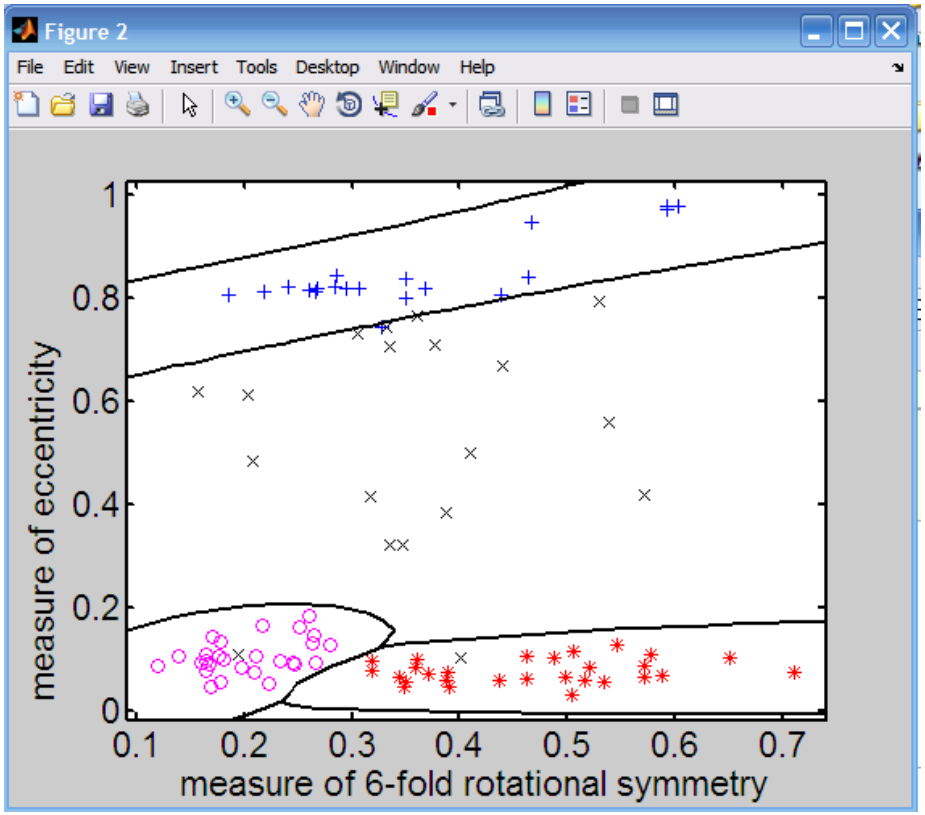
Conclusions

- The Bayes classifier for normally distributed classes is a quadratic classifier
- The Bayes classifier for normally distributed classes with equal covariance matrices is a linear classifier
- The minimum Mahalanobis distance is Bayes-optimal for
  - Normally distributed classes, having
  - Equal covariance matrices and
  - Equal priors
- The minimum Euclidian distance is Bayes-optimal for
  - Normally distributed classes -and-
  - Equal covariance matrices proportional to the identity matrix–and-
  - Equal priors
- Both Euclidian and Mahalanobis distance classifiers are linear classifiers

PRTOOLS

- load nutsbolts; //an existing Prtools dataset with 4 classes
- w1=ldc(z); //linear Bayes Normal classifier
- w2=qdc(z); //quadratic Bayes Normal classifier
- figure(1); scatterd(z); hold on; plotc(w1);
- figure(2); scatterd(z); hold on; plotc(w2);





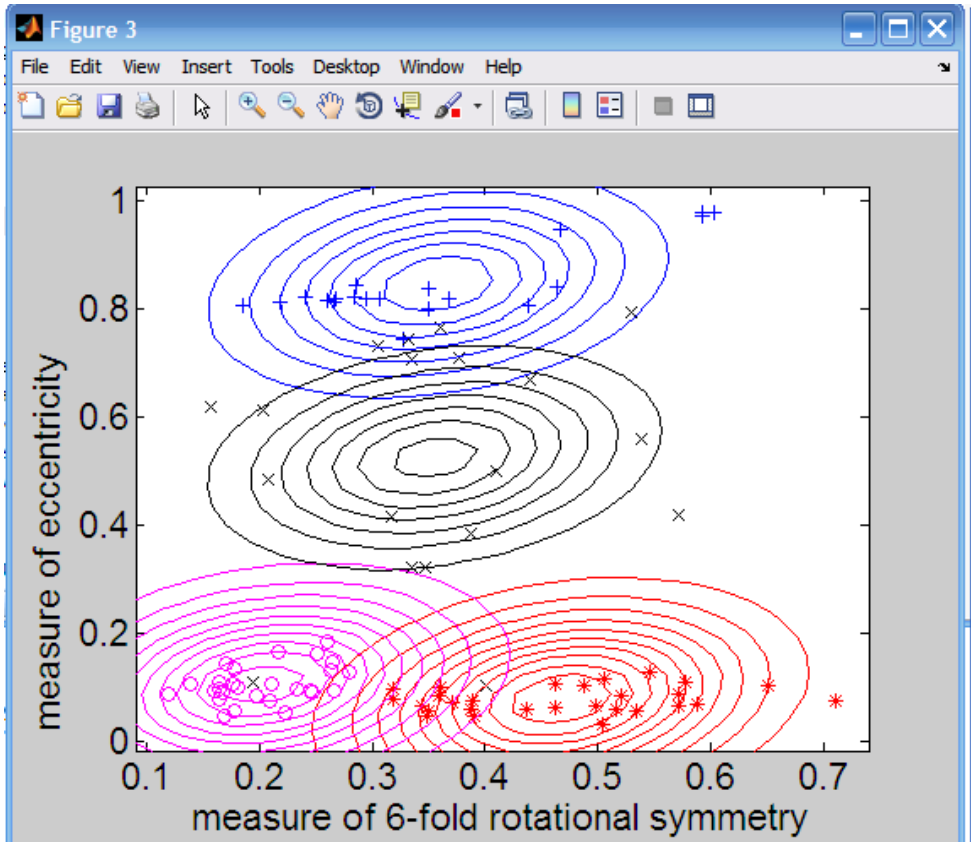


Figure 3
File Edit View Insert Tools Desktop Window Help
measure of eccentricity
measure of 6-fold rotational symmetry
0
0.2
0.4
0.6
0.8
1
0.1
0.2
0.3
0.4
0.5
0.6
0.7

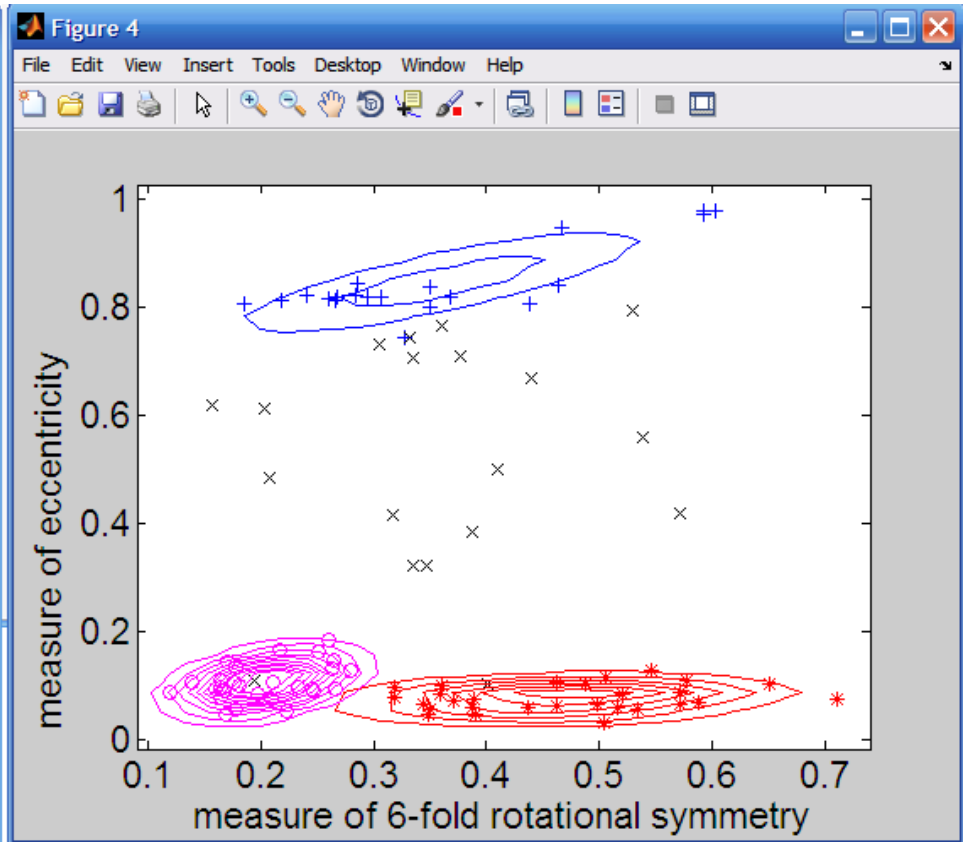
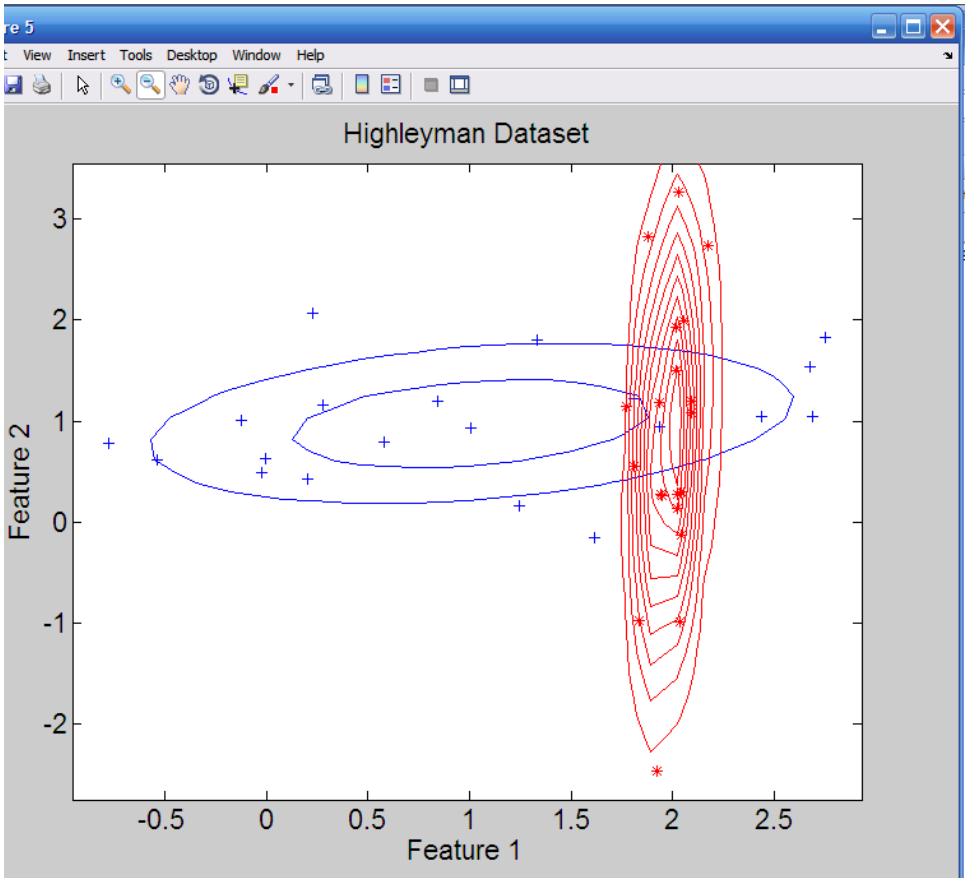
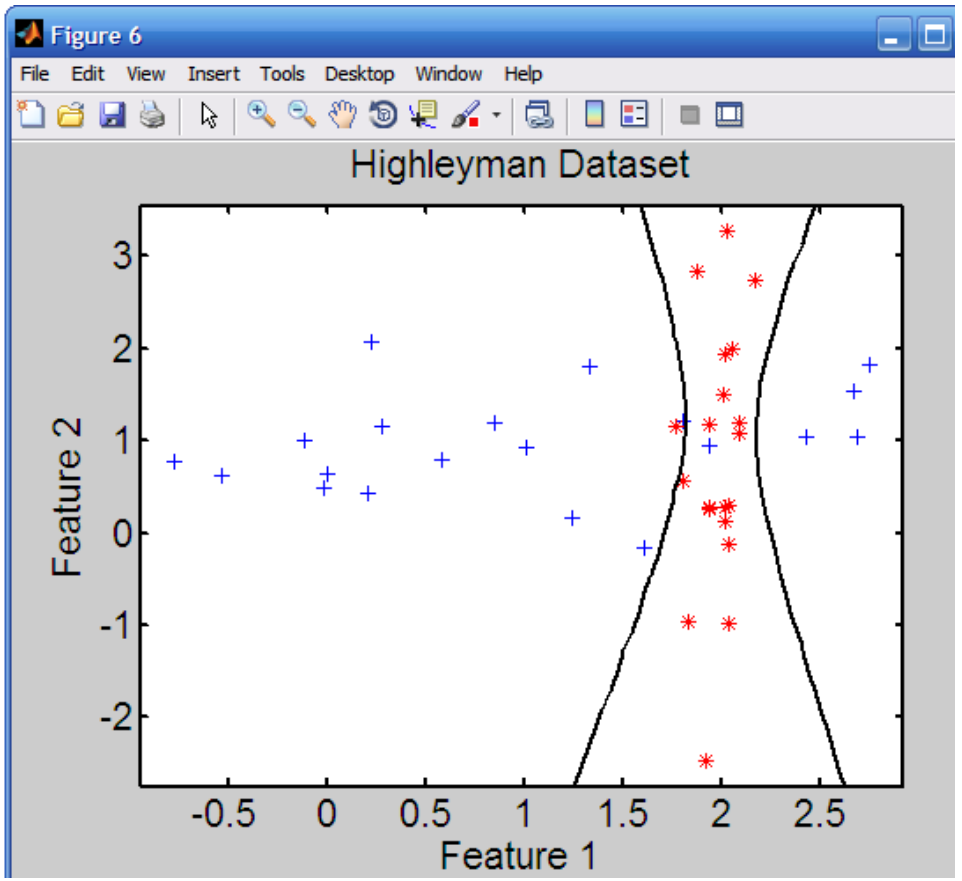


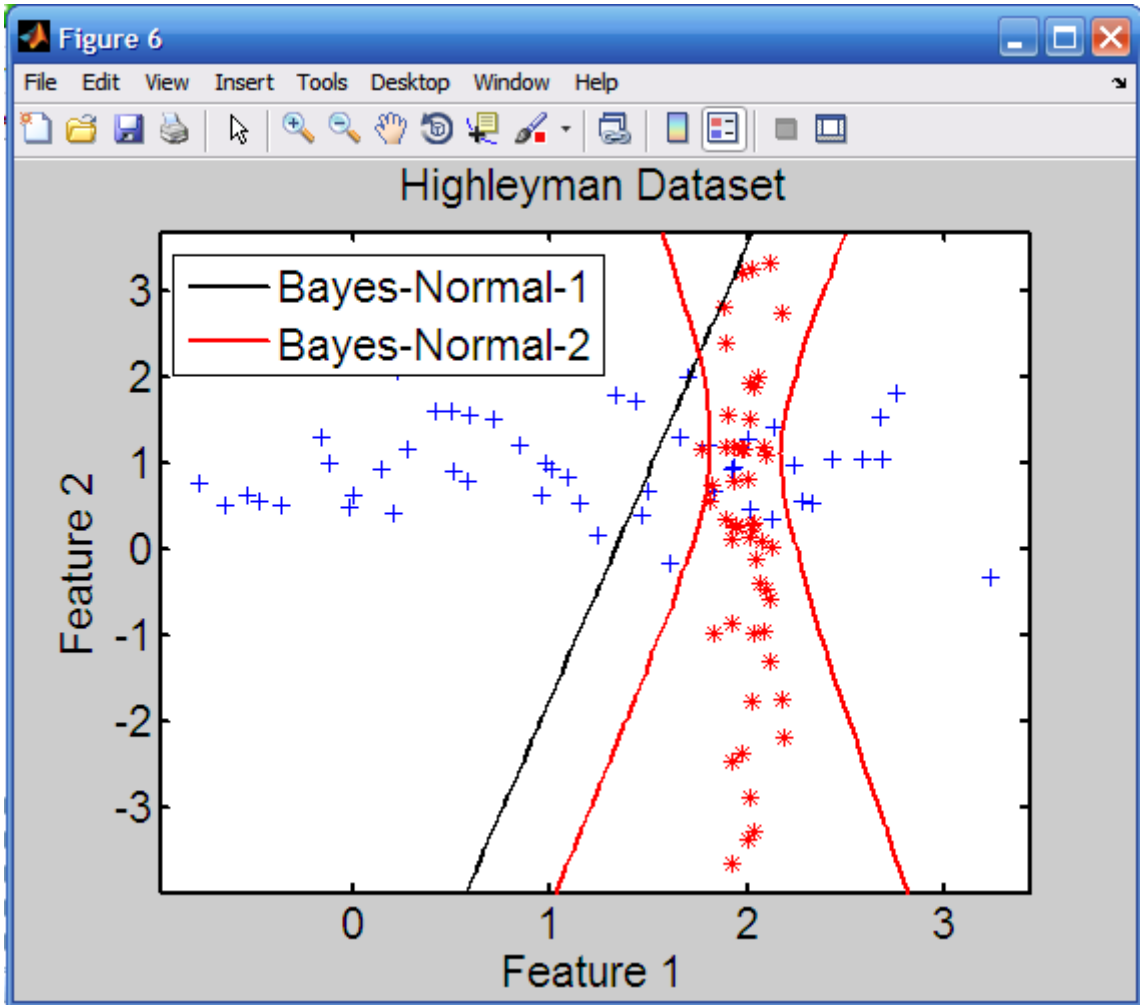
Figure 4
File Edit View Insert Tools Desktop Window Help
measure of eccentricity
measure of 6-fold rotational symmetry
0
0.2
0.4
0.6
0.8
1
0.1
0.2
0.3
0.4
0.5
0.6
0.7

- >> A = gendath([50 50]); //Generate data with 2 classes x 50 samples each
- >> [C,D] = gendat(A,[20 20]); //Split 20 as C=train; rest becomes D=test
- >> W1 = ldc(C); //linear
- >> W2 = qdc(C); //quadratic
- >> figure(5); scatterd(C); hold on; plotm(W2);





- scatterd(A); //scatter plot
- plotc({W1,W2});



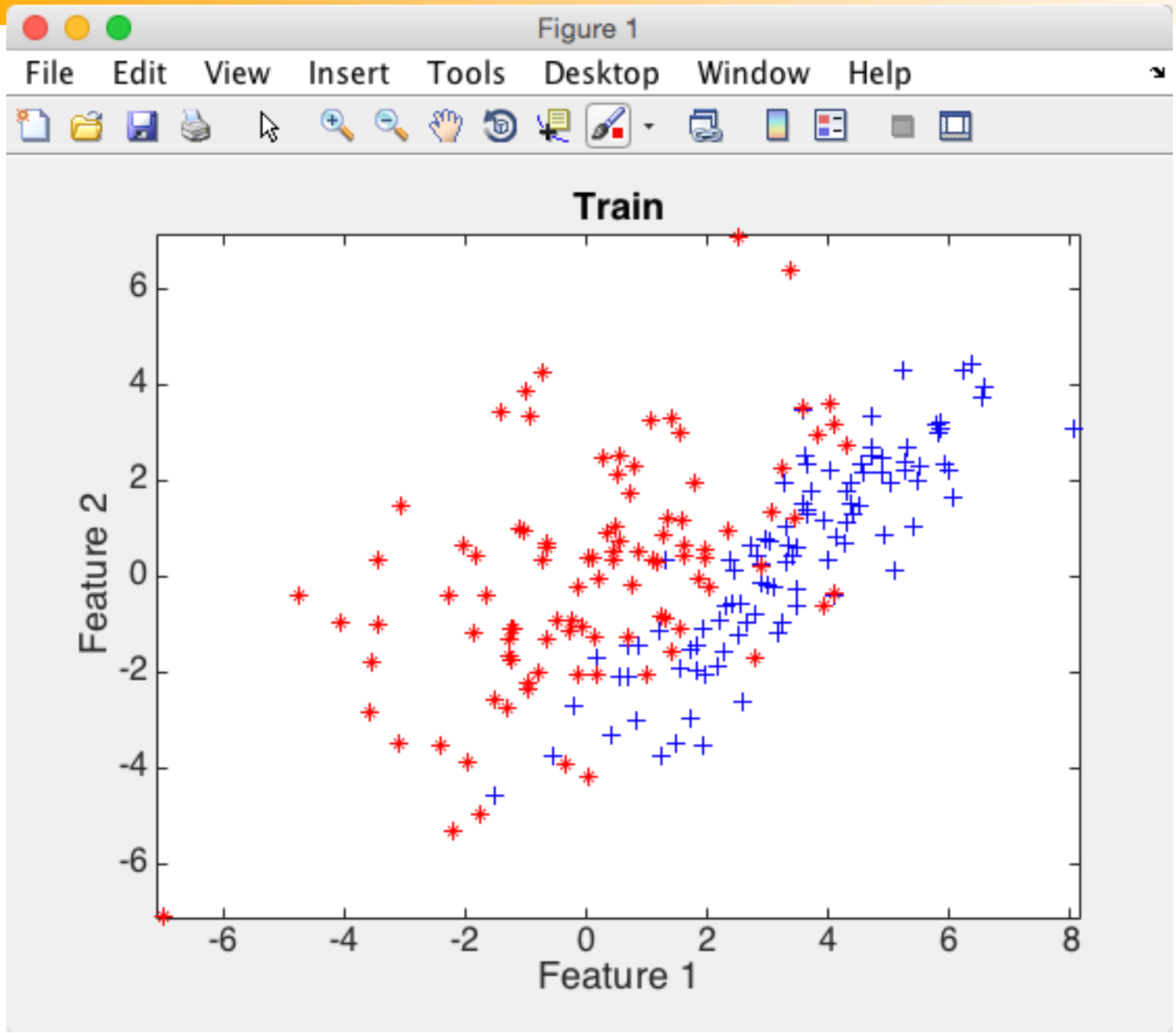


Figure 1
File Edit View Insert Tools Desktop Window Help
Train
Feature 2
Feature 1
6
4
2
0
-2
-4
-6
-6 -4 -2 0 2 4 6 8

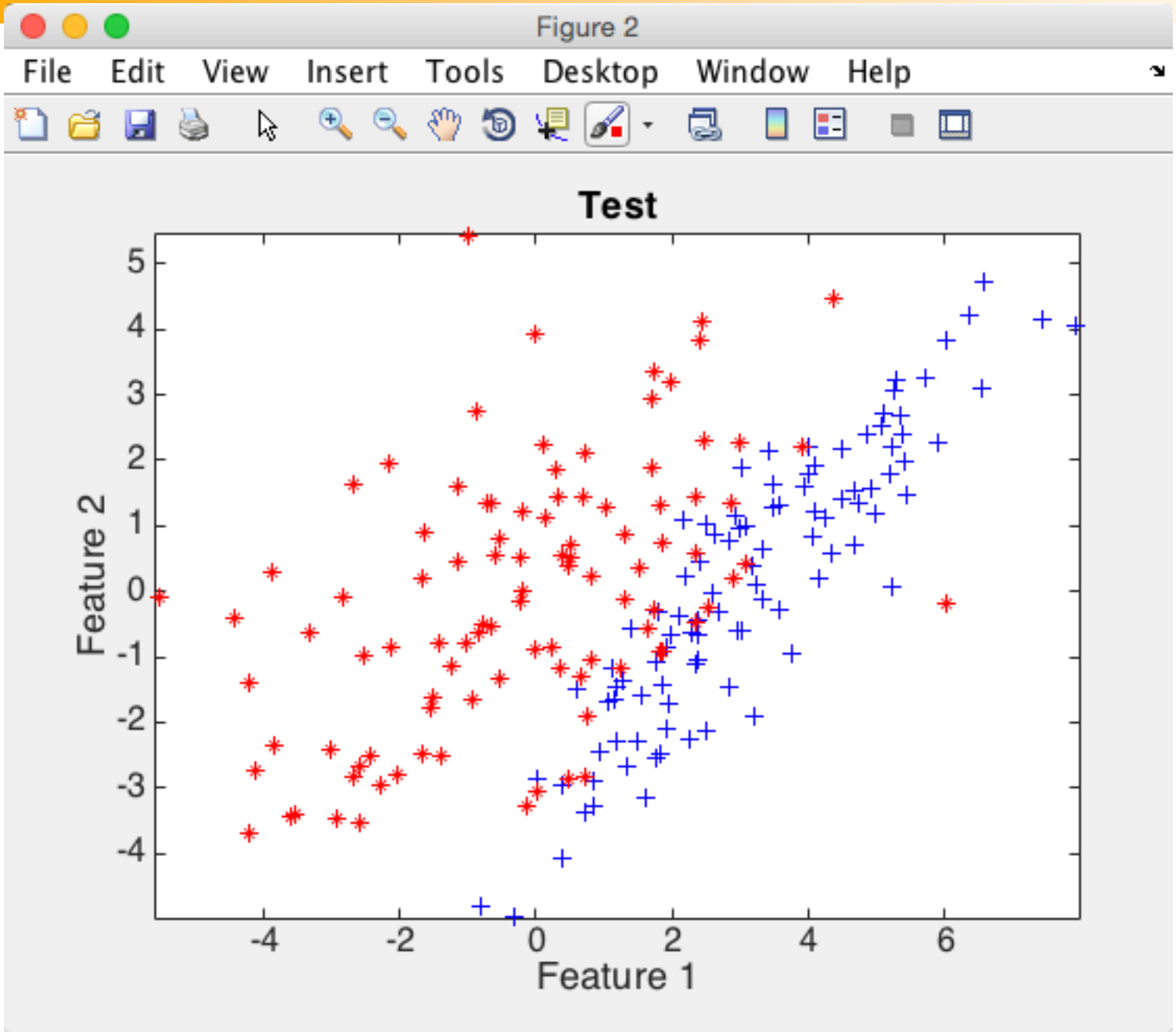
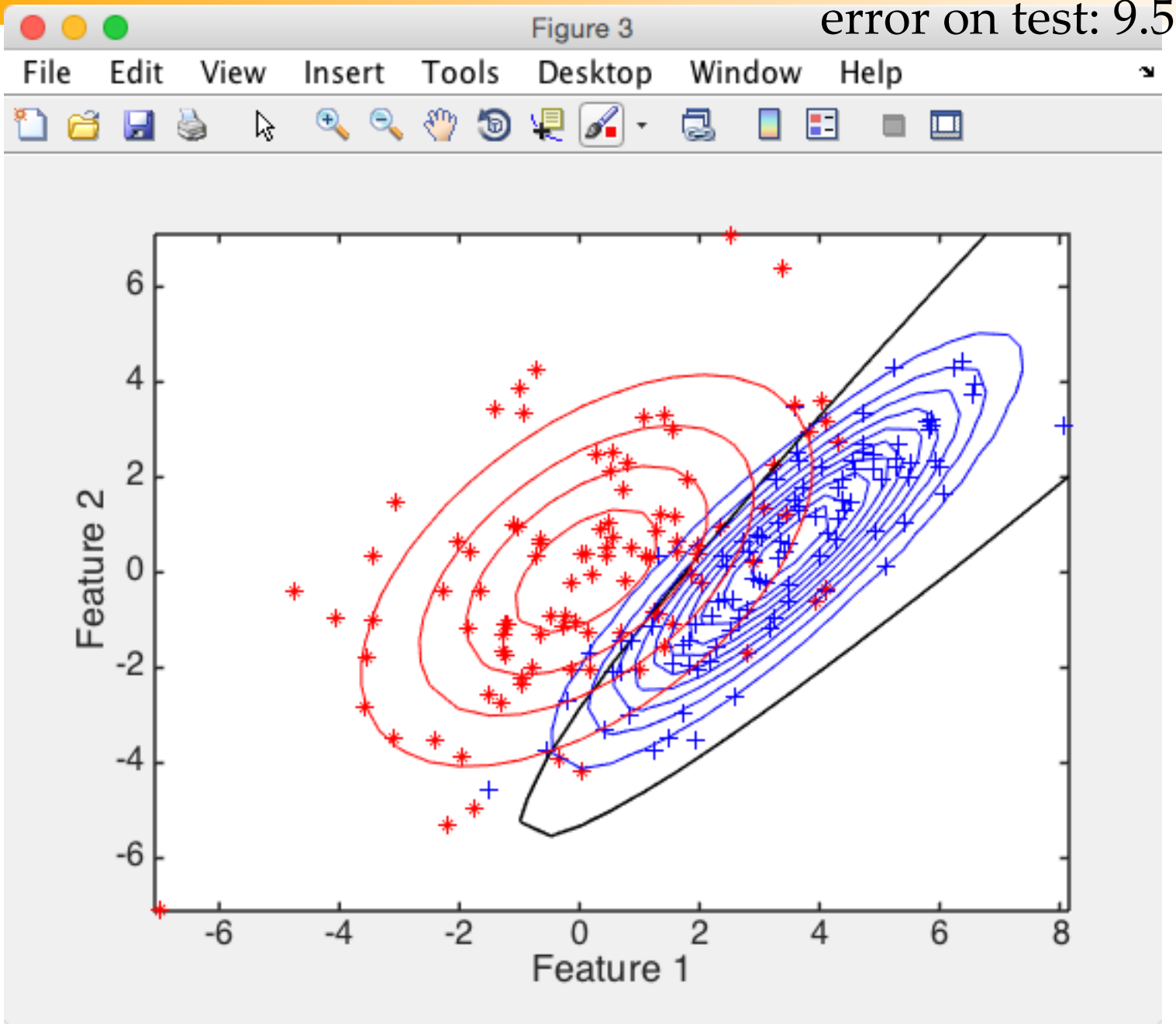
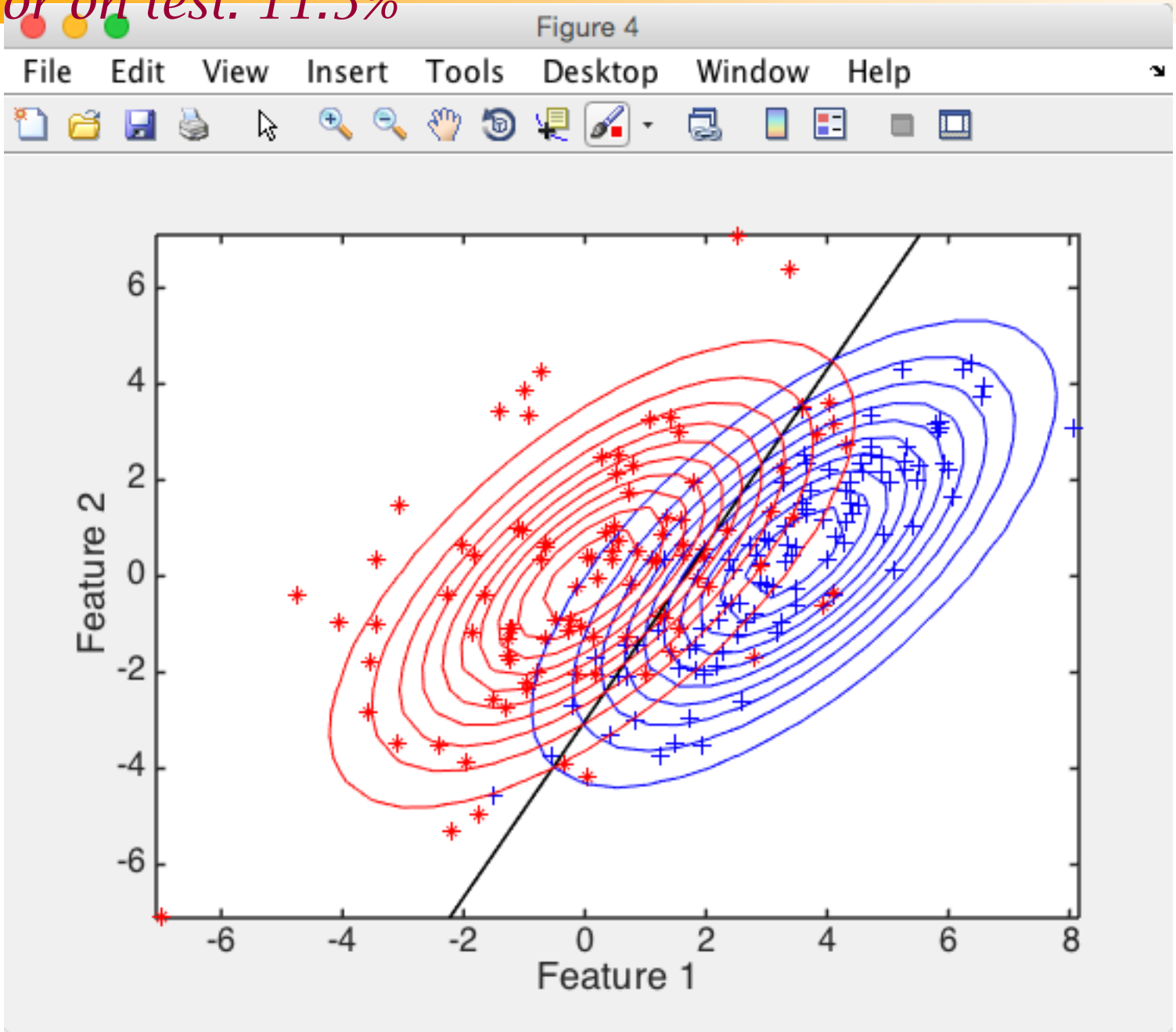


Figure 2
File Edit View Insert Tools Desktop Window Help
Test
Feature 2
Feature 1
5
4
3
2
1
0
-1
-2
-3
-4
-4
-2
0
2
4
6

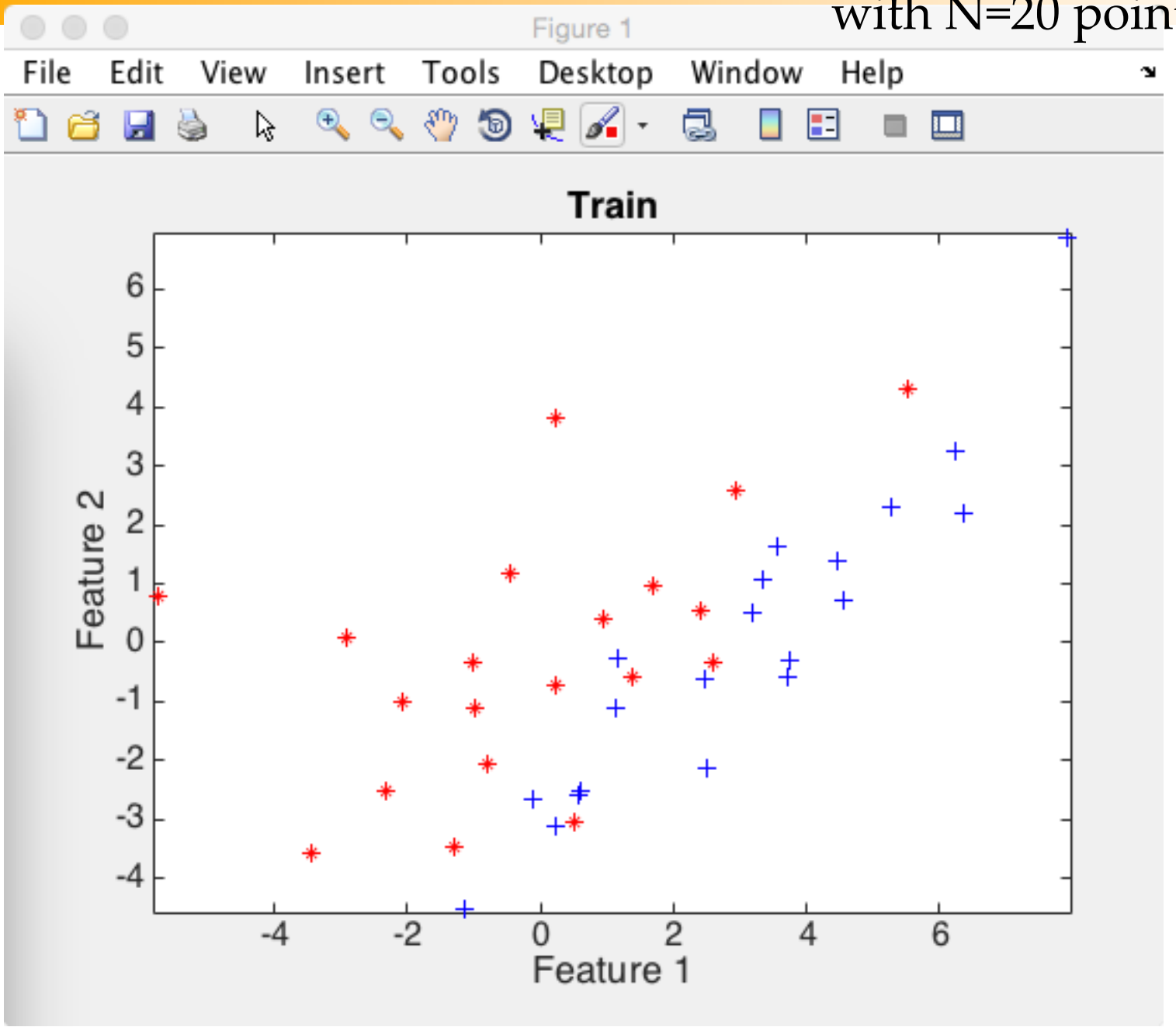
error on test: 9.5%



error on test: 11.5%



with N=20 points



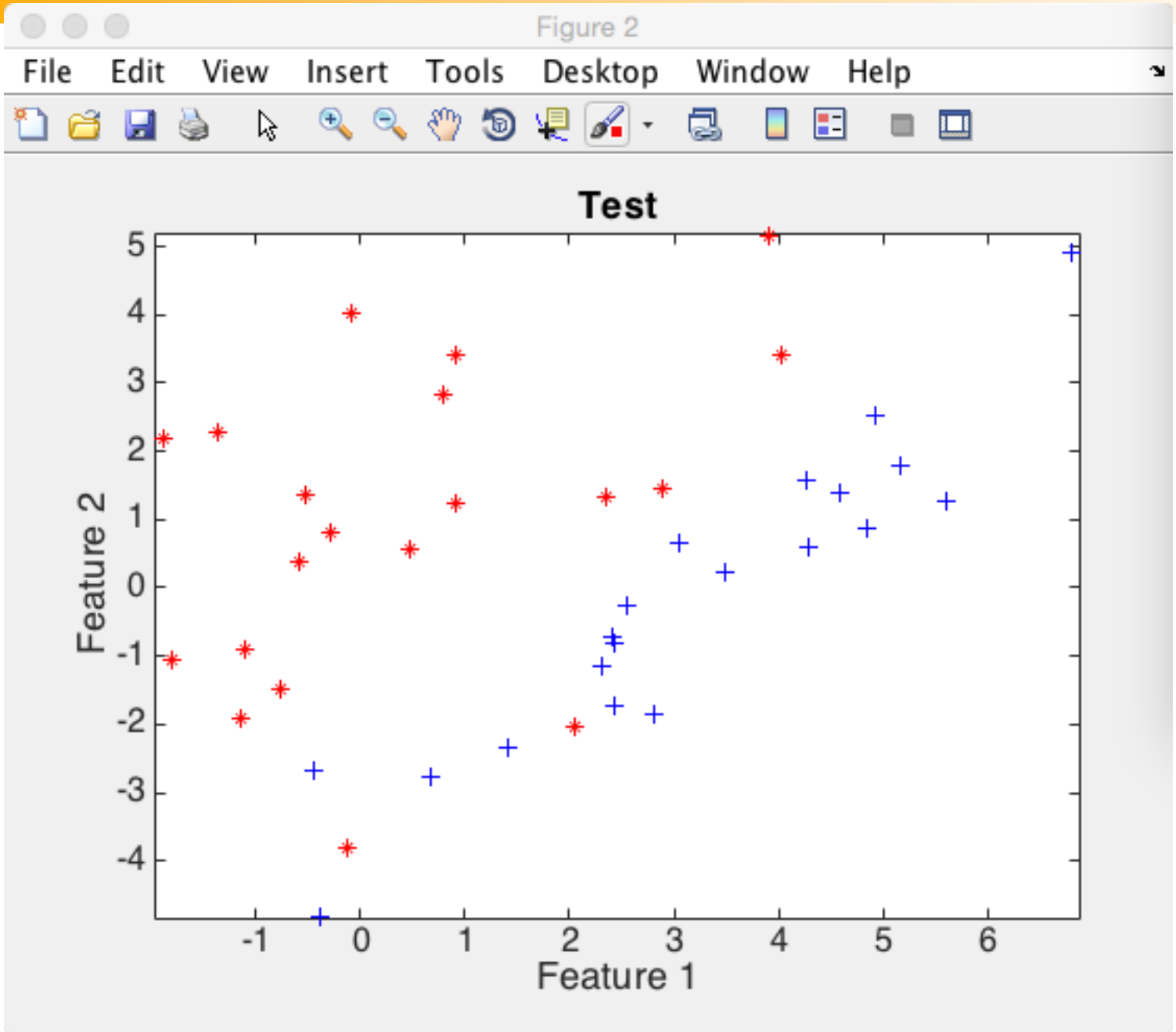


Figure 2
File Edit View Insert Tools Desktop Window Help
Test
Feature 2
Feature 1
5
4
3
2
1
0
-1
-2
-3
-4
-1
0
1
2
3
4
5
6

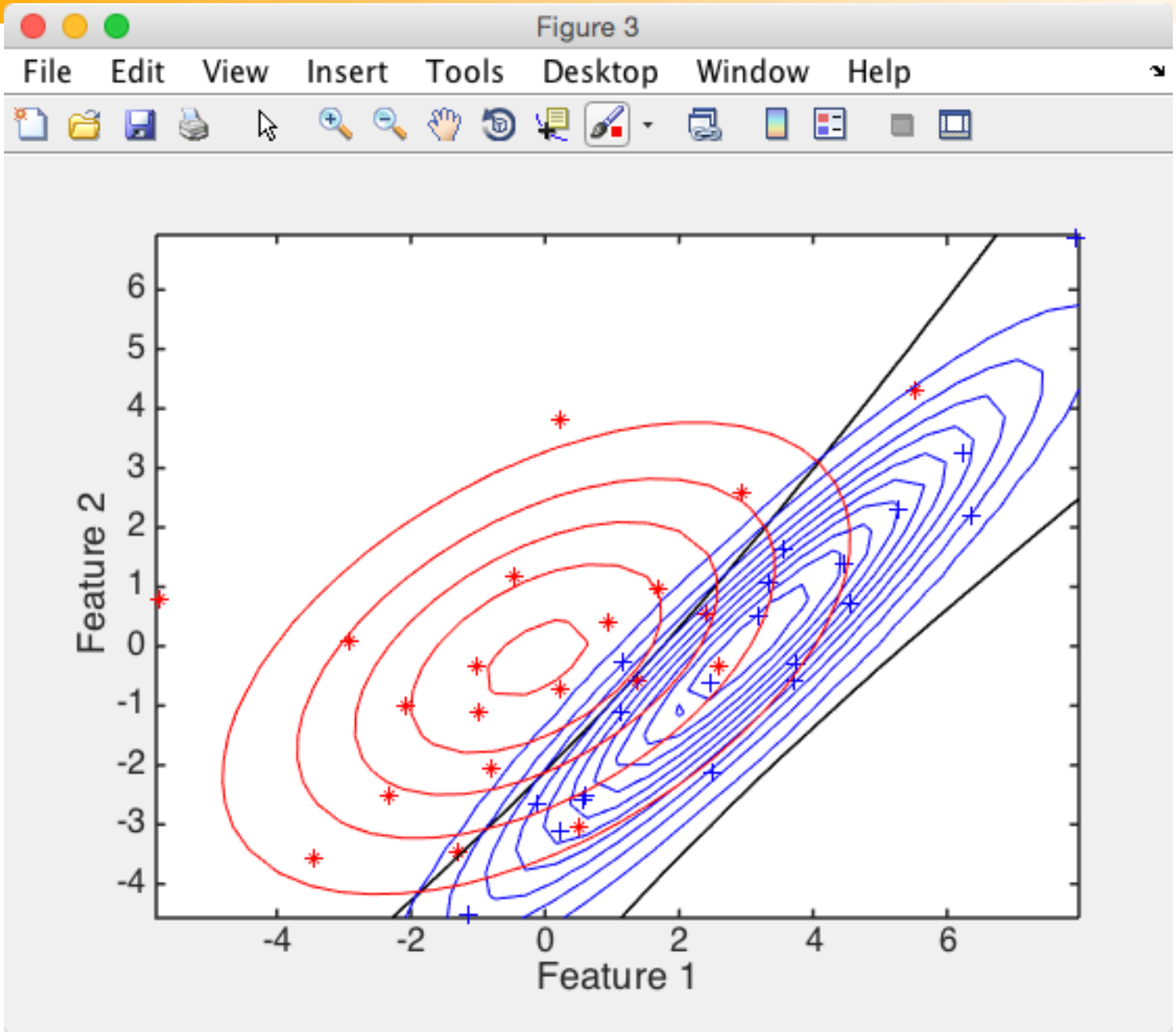


Figure 3
File Edit View Insert Tools Desktop Window Help
Feature 2
Feature 1
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-2
0
2
4
6

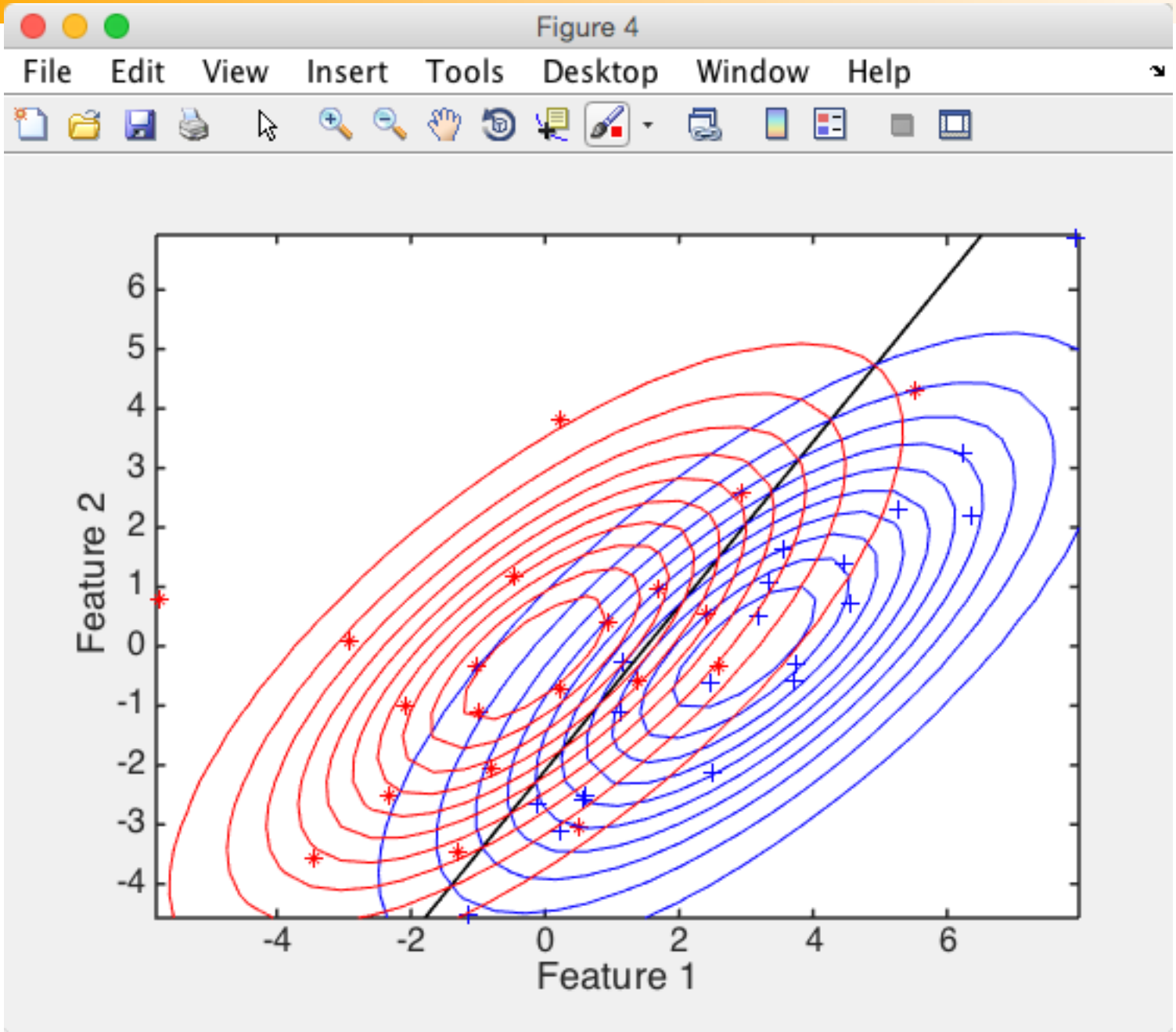


Figure 4
File Edit View Insert Tools Desktop Window Help
Feature 2
Feature 1
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-2
0
2
4
6

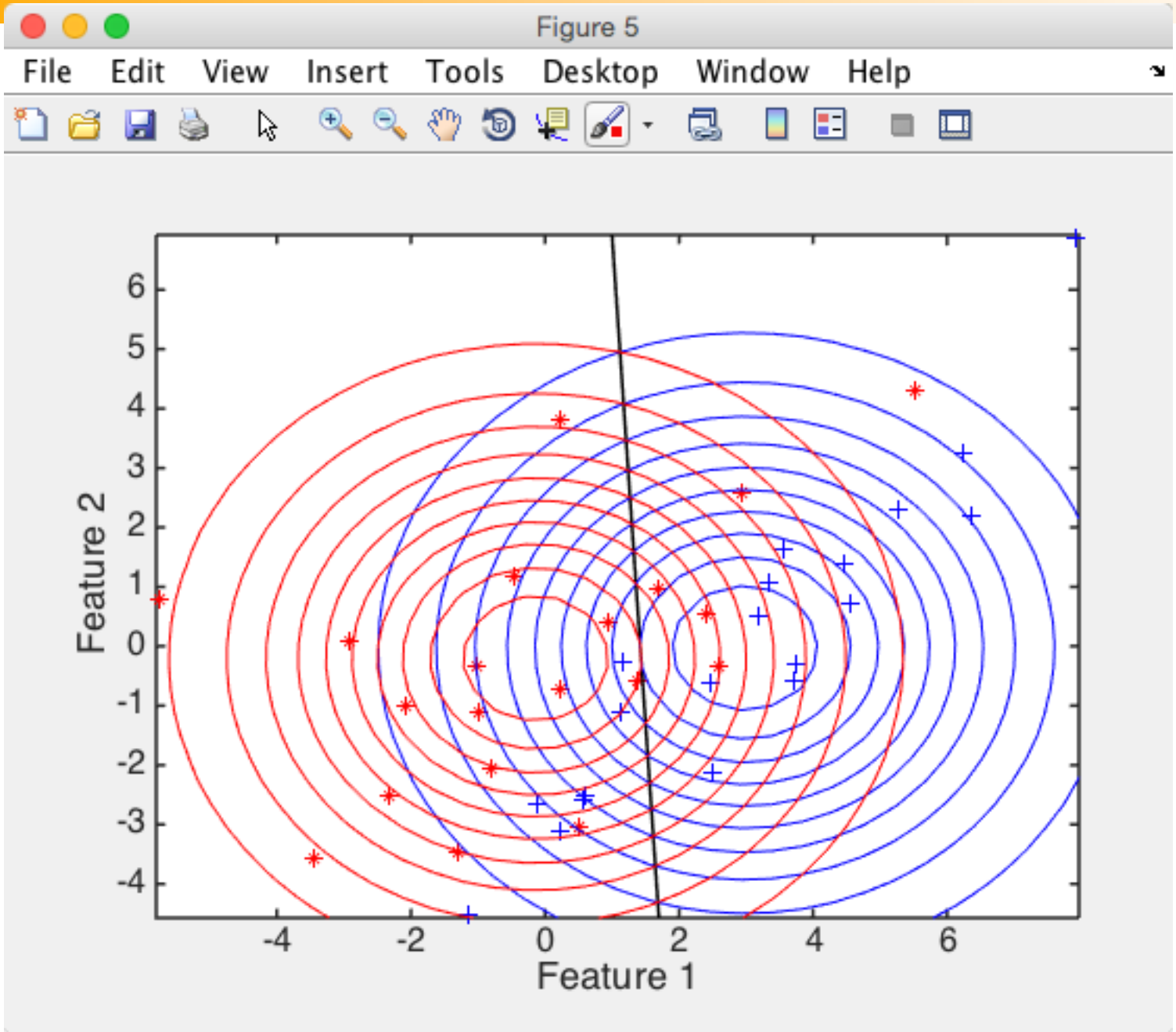


Figure 5
File Edit View Insert Tools Desktop Window Help
Feature 2
Feature 1
6
5
4
3
2
1
0
-1
-2
-3
-4
-4
-2
0
2
4
6

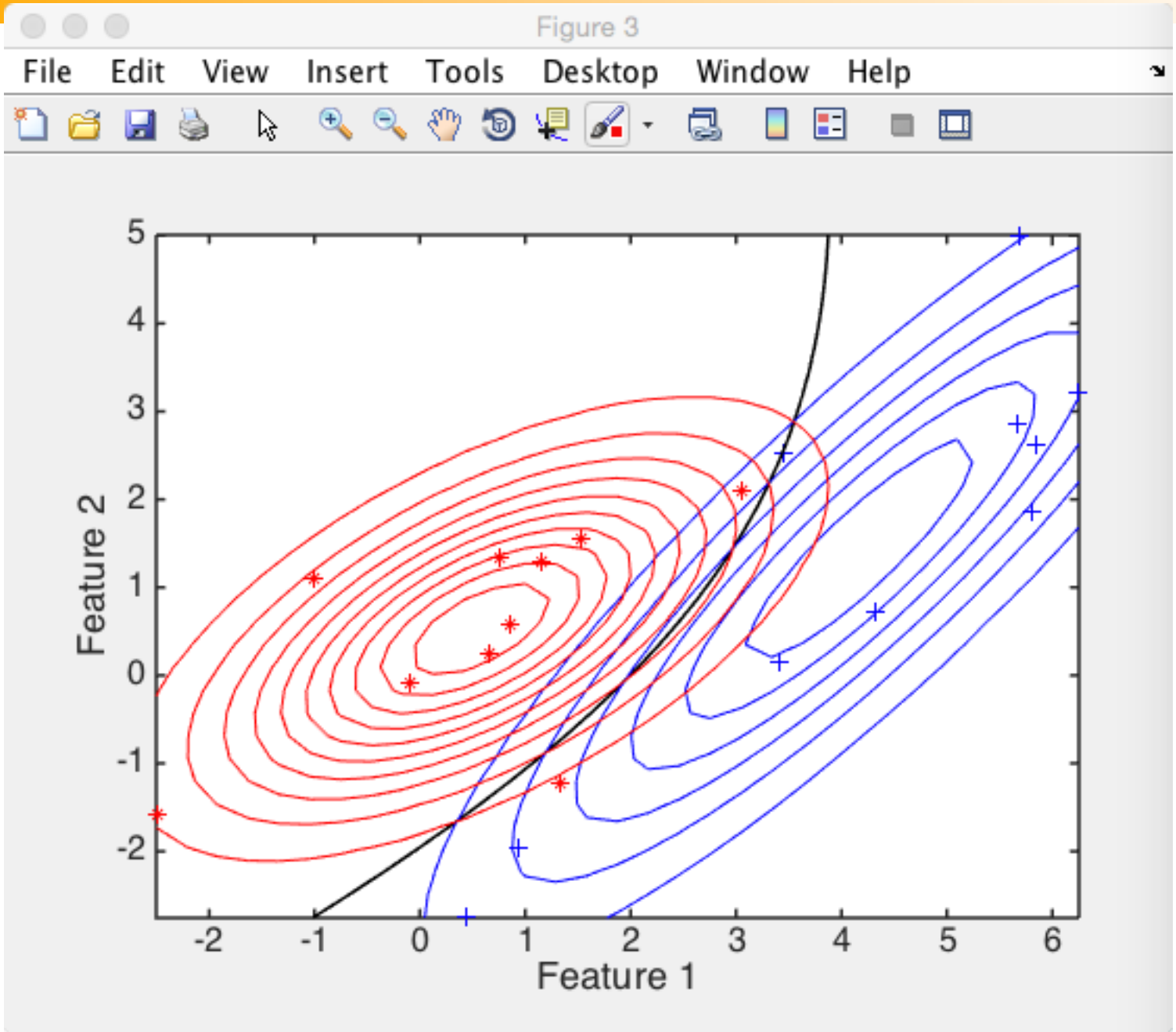


Figure 3
File Edit View Insert Tools Desktop Window Help
Feature 2
Feature 1
5
4
3
2
1
0
-1
-2
-2 -1 0 1 2 3 4 5 6

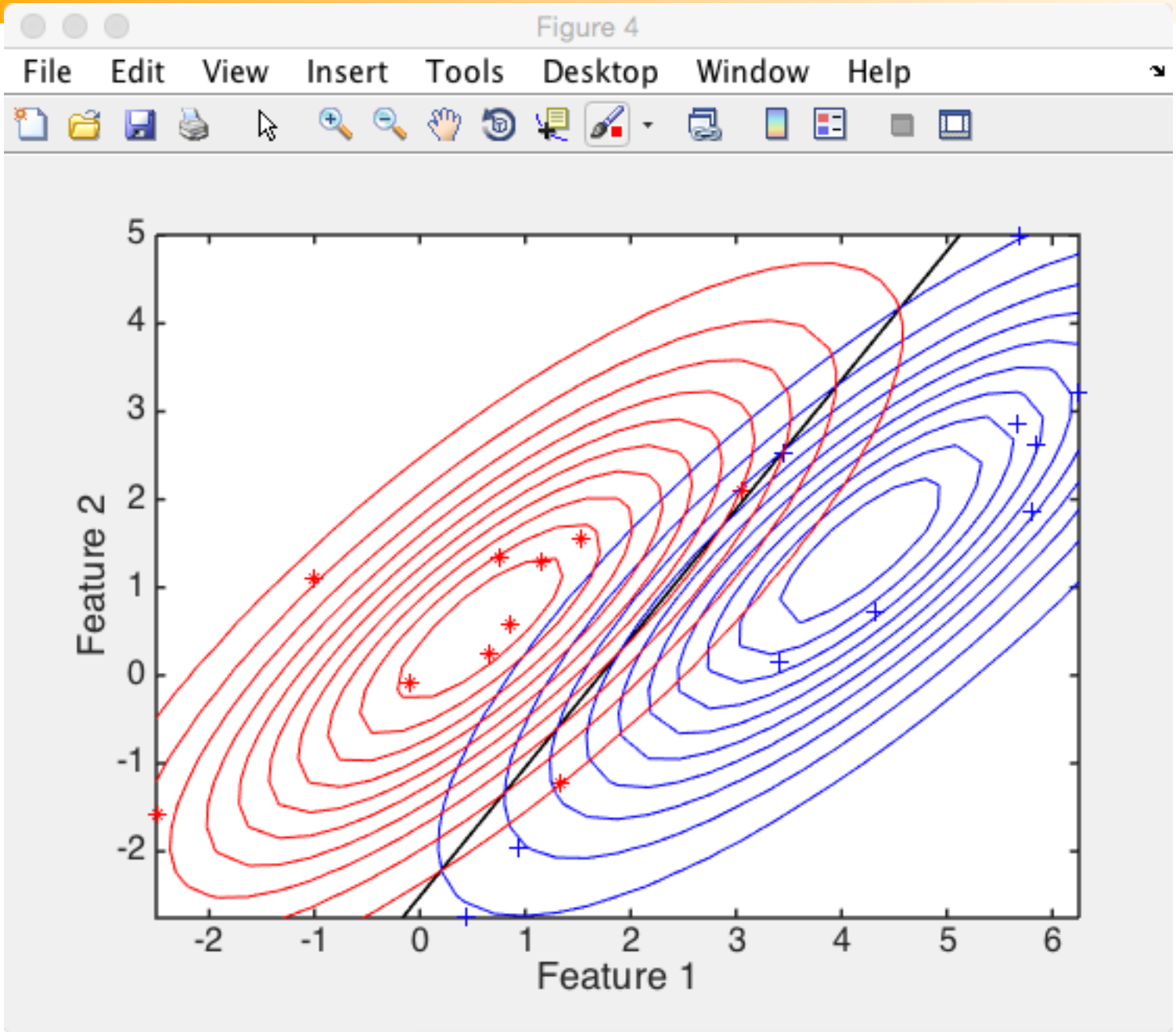


Figure 4
File Edit View Insert Tools Desktop Window Help
Feature 2
Feature 1
5
4
3
2
1
0
-1
-2
-2 -1 0 1 2 3 4 5 6

Eigenvalue Decomposition of the Covariance Matrix

Skip Until Dimensionality Reduction Topic

Eigenvector and Eigenvalue

- Given a linear transformation **A**, a non-zero vector **x** is defined to be an eigenvector of the transformation if it satisfies the eigenvalue equation

$$\mathbf{Ax} = \lambda\ \mathbf{x} \qquad \text{for some scalar } \lambda.$$

- In this situation, the scalar λ is called an eigenvalue of **A** corresponding to the eigenvector **x**.

- $(A - \lambda I)x = 0 \Rightarrow \det(A - \lambda I)$ must be 0.
  - Gives the characteristic polynomial whore roots are the eigenvalues of A

Eigenvalues of the covariance matrix

The covariance matrix is symmetrical and it can always be diagonalized as:

$$\Sigma = \Phi \Lambda \Phi^T$$

where

$\Phi = [\upsilon_1, \upsilon_2, \ldots, \upsilon_l]$ is the column matrix consisting of the eigenvectors of Σ,

$\Phi^T = \Phi^{-1}$ and

Λ is the diagonal matrix whose elements are the eigenvalues of Σ.

Contours of equal Mahalanobis distance:

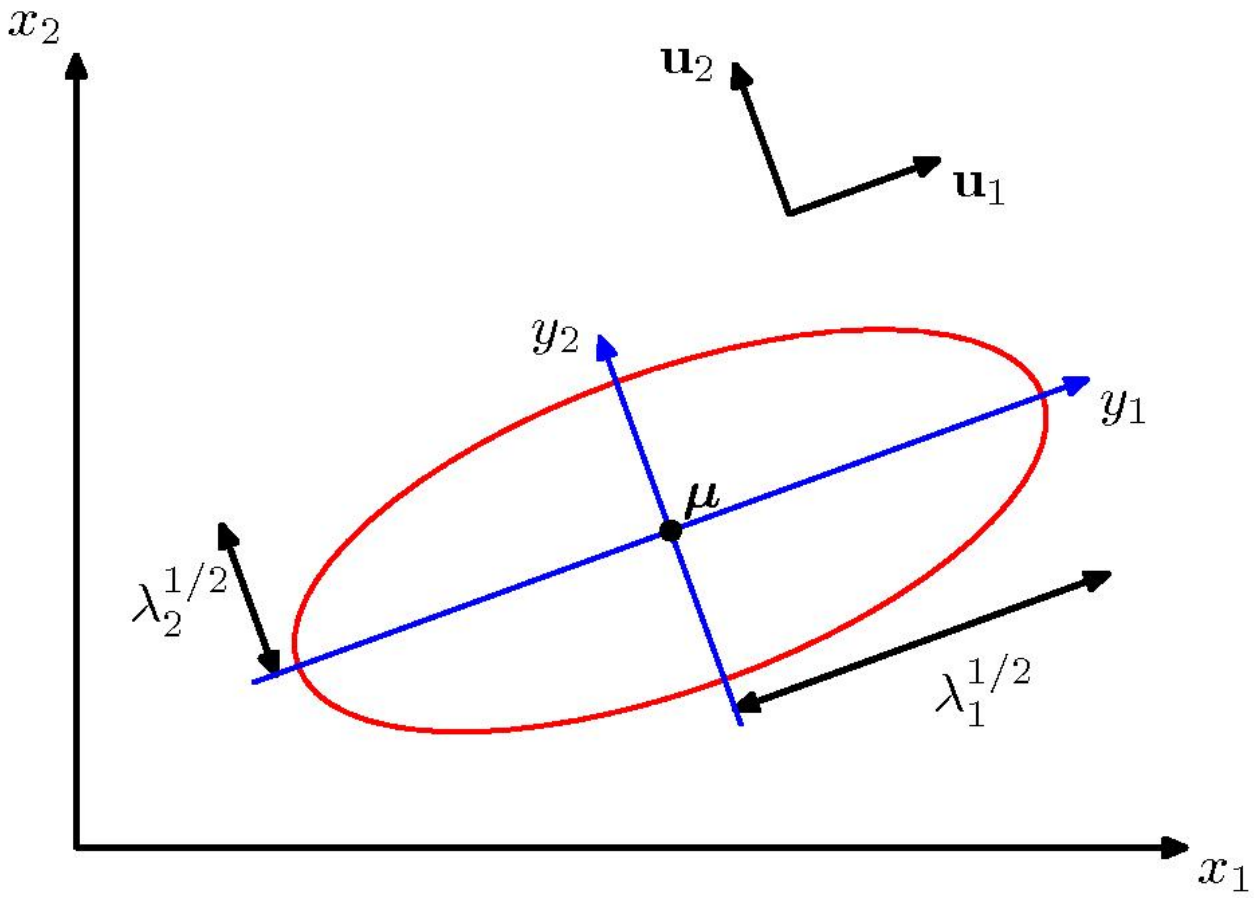
$$d_m = \left((x-\mu_i)^T \Sigma^{-1}(x-\mu_i)\right)^{1/2} = c$$

$$(x-\mu_i)^T \Phi \Lambda^{-1} \Phi^T (x-\mu_i) = c^2$$

Define $\mathbf{x}' = \Phi^T \mathbf{x}$. The coordinates of $\mathbf{x}'$ are equal to $\nu_k^T \mathbf{x}$, $k=1,2,\ldots,l$ that is, the projections of $\mathbf{x}$ onto the eigenvectors.

$$\frac{(x_1' - \mu_{i1}')^2}{\lambda_1} + \ldots + \frac{(x_l' - \mu_{il}')^2}{\lambda_l} = c^2$$

Thus **all points having the same Mahalabonis distance from a specific point are located on an ellipse with center of mass at μ_i, and the principle axes are aligned with the corresponding eigenvectors** and have lengths $2\sqrt{\lambda_k}c$


x_2
$\mathbf{u}_2$
$\mathbf{u}_1$
y_2
y_1
$\boldsymbol{\mu}$
$\lambda_2^{1/2}$
$\lambda_1^{1/2}$
x_1

- Eigenvalue decomposition of the covariance matrix is very useful:
  - PCA
  - Decorrelate data (Whitening transform)
  - …

Matrix Transpose

- Given the matrix A, the **transpose** of A is the , denoted A^T, whose columns are formed from the corresponding rows of A.

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad B = \begin{bmatrix} 3 & 5 \\ 2 & 7 \\ 6 & 9 \\ 1 & 0 \\ 5 & 2 \end{bmatrix}$$

$$A^T = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \qquad B^T = \begin{bmatrix} 3 & 2 & 6 & 1 & 5 \\ 5 & 7 & 9 & 0 & 2 \end{bmatrix}$$

- Some properties of transpose
  1. $(A^T)^T = A$
  2. $(A + B)^T = A^T + B^T$
  3. $(rA)^T = rA^T$ where r is any scalar.
  4. $(AB)^T = B^T A^T$
  5. $A^T B = B^T A$ where A and B are vectors
  6. …

Matrix Derivation

- If $y = \mathbf{x}^T\mathbf{A}\mathbf{x}$ where $\mathbf{A}$ is a square matrix
  - $\partial y/\partial \mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{A}^T\mathbf{x}$
- If $y = \mathbf{x}^T\mathbf{A}\mathbf{x}$ and $\mathbf{A}$ is symmetric
  - $\partial y/\partial \mathbf{x} = 2\mathbf{A}\mathbf{x}$.
- If $y = \mathbf{x}^T\mathbf{x}$
  - $\partial y/\partial \mathbf{x} = 2\mathbf{x}$.

- If $\mathbf{x} \sim N_d(\boldsymbol{\mu},\boldsymbol{\Sigma})$ then each dimension of $\mathbf{x}$ is univariate normal
  - Converse is not true – counterexample?

- The projection of a d-dimensional normal distribution onto the vector $\mathbf{w}$ is univariate normal

$$\mathbf{w}^T\mathbf{x} \sim N(\mathbf{w}^T\boldsymbol{\mu}, \mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w})$$

- More generally, when $\mathbf{W}$ is a dxk matrix with k < d, then the k-dimensional $\mathbf{W}\mathbf{x}$ is k-variate normal with:

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

Univariate normal – transformed x

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

- $\mathrm{Var}[\mathbf{w}^T\mathbf{x}] = E[\,(\mathbf{w}^T\mathbf{x} - \mathbf{w}^T \boldsymbol{\mu})\,(\mathbf{w}^T\mathbf{x} - \mathbf{w}^T \boldsymbol{\mu})^T]$

  $= E[\,(\mathbf{w}^T\mathbf{x} - \mathbf{w}^T \boldsymbol{\mu})^2]$ since $(\mathbf{w}^T\mathbf{x} - \mathbf{w}^T \boldsymbol{\mu})$ is a scalar

  $= E[\,(\mathbf{w}^T\mathbf{x} - \mathbf{w}^T \boldsymbol{\mu})\,(\mathbf{w}^T\mathbf{x} - \mathbf{w}^T \boldsymbol{\mu})\,]$

  $= E[\;\mathbf{w}^T(\mathbf{x}-\boldsymbol{\mu})\,(\mathbf{x}-\boldsymbol{\mu})^T\,\mathbf{w}\;]$ based on slide 27.rule 5

  $= \mathbf{w}^T\boldsymbol{\Sigma}\,\mathbf{w}$ move out and use defin. of Σ

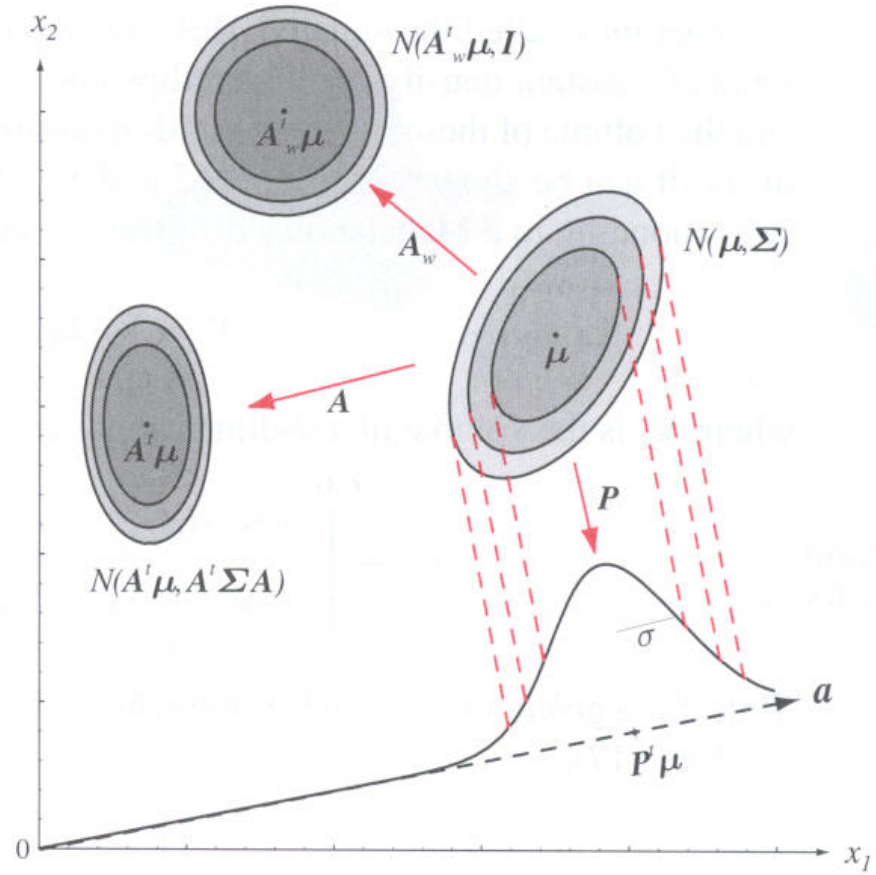


FIGURE 2.8. The action of a linear transformation on the feature space will convert an arbitrary normal distribution into another normal distribution. One transformation, **A**, takes the source distribution into distribution $N(\mathbf{A}^t\boldsymbol{\mu}, \mathbf{A}^t\boldsymbol{\Sigma}\mathbf{A})$. Another linear transformation—a projection **P** onto a line defined by vector **a**—leads to $N(\mu, \sigma^2)$ measured along that line. While the transforms yield distributions in a different space, we show them superimposed on the original x_1x_2-space. A whitening transform, $\mathbf{A}_w$, leads to a circularly symmetric Gaussian, here shown displaced.

Whitening Transform

- We can decorrelate variables and obtain $\Sigma=I$ by using a whitening transform $A_w=\Lambda^{-1/2}\Phi^T$ where
  - Λ is the diagonal matrix of the eigenvalues of the original distribution and
  - Φ is the matrix composed of eigenvectors as its columns