Gaussian Classifiers

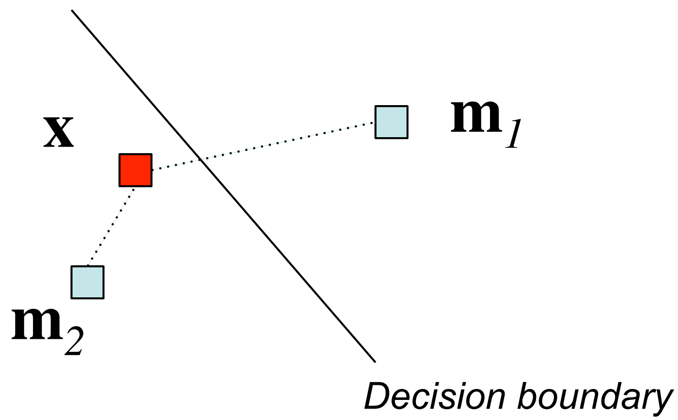
CS498

Today's lecture

- The Gaussian
- Gaussian classifiers
  - A slightly more sophisticated classifier

Nearest Neighbors

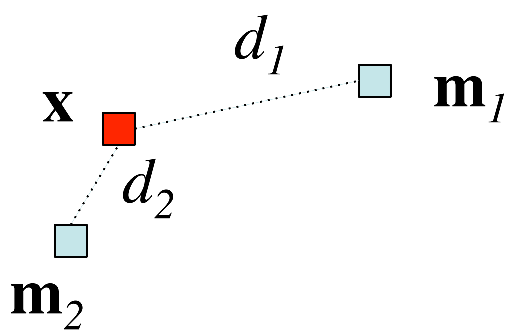
- We can classify with nearest neighbors



- Can we get a probability?

Nearest Neighbors

- Nearest neighbors offers an intuitive distance measure

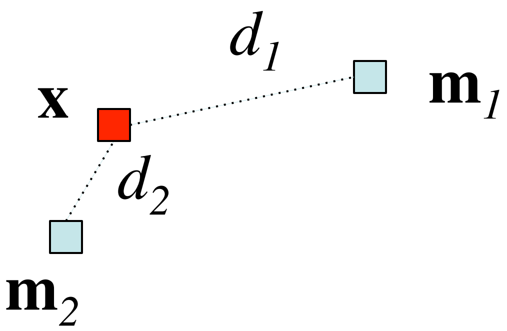


$$d_i \propto (x_1 - m_{i,1})^2 + (x_2 - m_{i,2})^2 = \|\mathbf{x} - \mathbf{m}\|$$

Making a “Soft” Decision

- What if I didn’t want to classify
  - What if I wanted a “degree of belief”

- How would you do that?



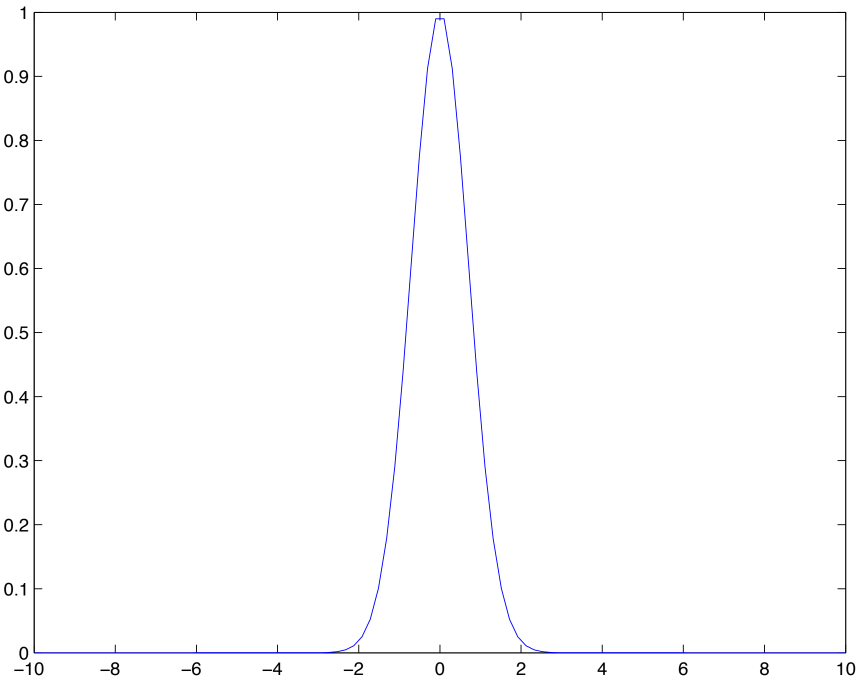
From a Distance to a Probability

- If the distance is 0 the probability is high
- If the distance is ∞ the probability is zero
- How do we make a function like that?

Here's a first crack at it

- Use exponentiation:

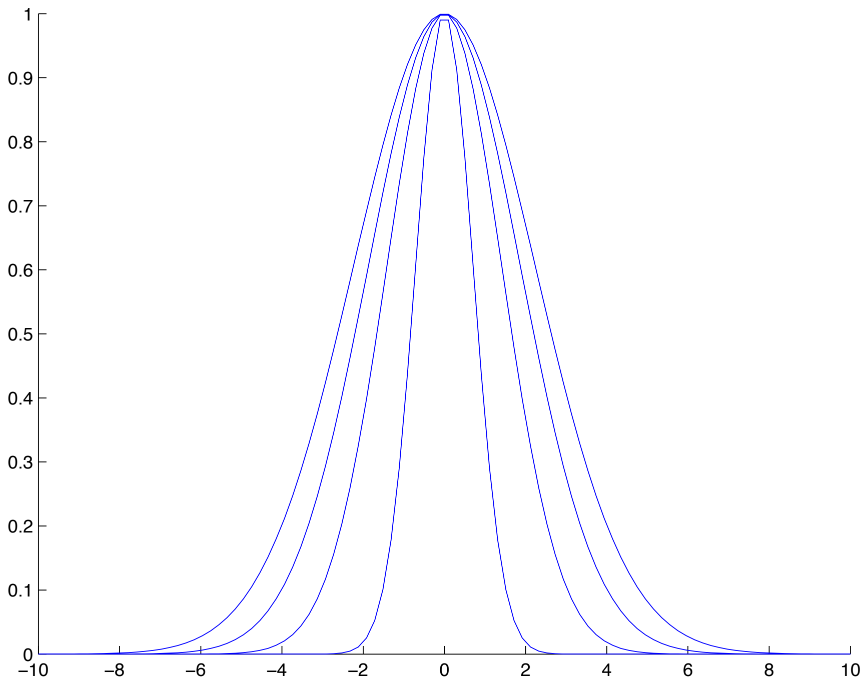
$$e^{-\|\mathbf{x}-\mathbf{m}\|}$$



Adding an "importance" factor

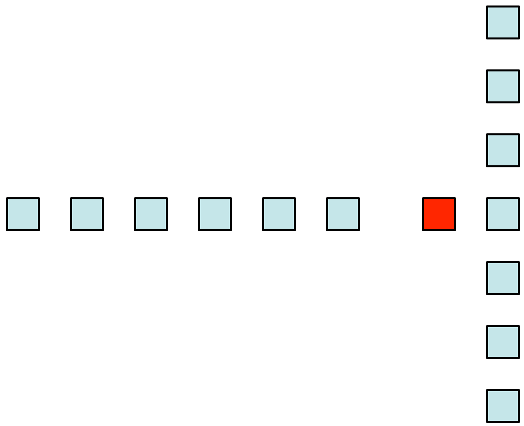
- Let's try to tune the output by adding a factor denoting importance

$$e^{-\frac{\|\mathbf{x}-\mathbf{m}\|}{c}}$$



One more problem

- Not all dimensions are equal



Adding variable "importance" to dimensions

- Somewhat more complicated now:

$$e^{-(\mathbf{x}-\mathbf{m})^T \mathbf{C}^{-1}(\mathbf{x}-\mathbf{m})}$$

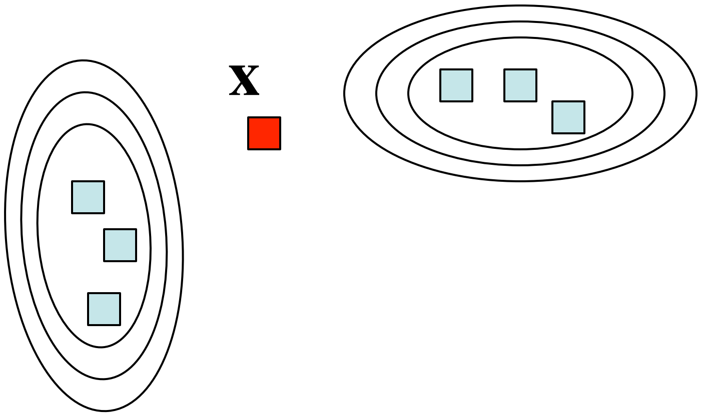
The Gaussian Distribution

- This is the idea behind the Gaussian
  - Adding some normalization we get:

$$P(\mathbf{x};\mathbf{m},\mathbf{C}) = \frac{1}{2\pi^{k/2}\,|\,\mathbf{C}\,|^{1/2}}\, e^{-(\mathbf{x}-\mathbf{m})^T\mathbf{C}^{-1}(\mathbf{x}-\mathbf{m})}$$

Gaussian models

- We can now describe data using Gaussians



- How? That's very easy

Learn Gaussian parameters

- Estimate the mean:

$$\mathbf{m} = \frac{1}{N} \sum \mathbf{x}_i$$

- Estimate the covariance:

$$\mathbf{C} = \frac{1}{N-1} (\mathbf{x} - \mathbf{m})^T \cdot (\mathbf{x} - \mathbf{m})$$

Now we can make classifiers

- We will use probabilities this time
- We'll compute a "belief" of class assignment

The Classification Process

- We provide examples of classes
- We make models of each class
- We assign all new input data to a class

Making an assignment decision

- Face classification example
- Having a probability for each face how do we make a decision?

Motivating example

- Face 1 is more "likely"

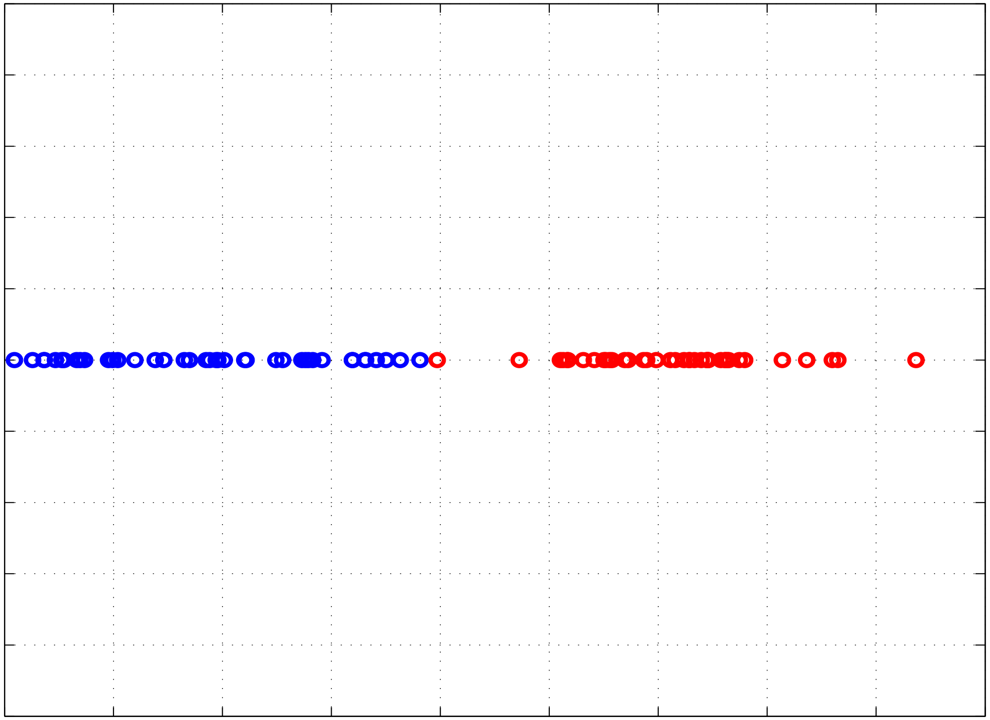
$\mathbf{x}$	$\mathbf{y}$	$P(\mathbf{y} \mid \{face_1, face_2\})$
Template face 1 	Unknown face 	0.93
Template face 2 		0.87

How the decision is made

- In simple cases the answer is intuitive
- To get a complete picture we need to probe a bit deeper

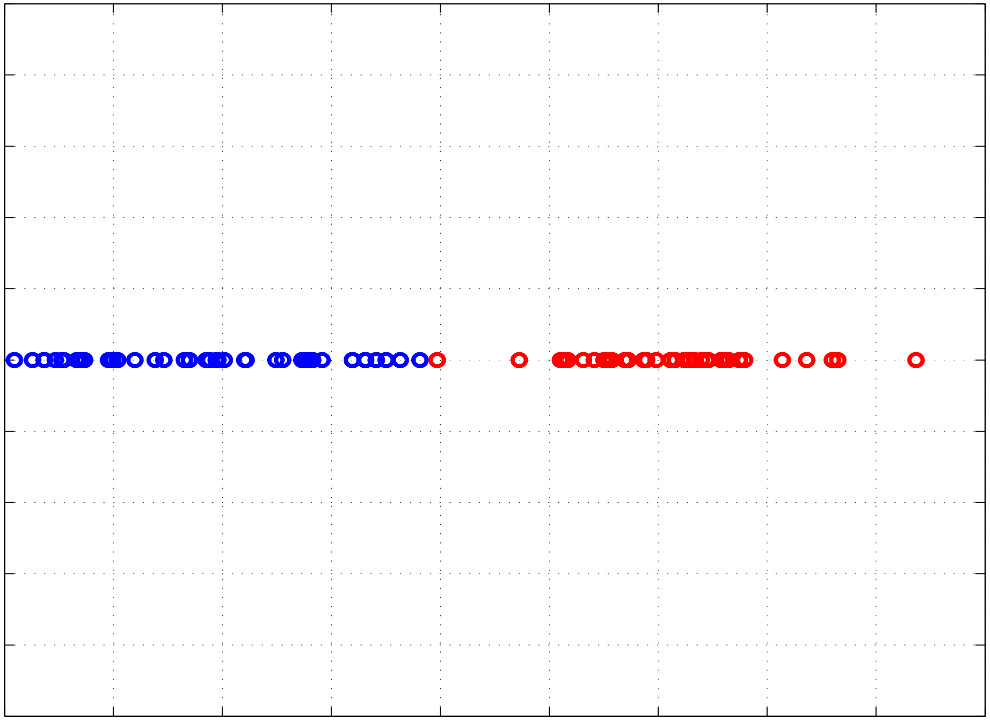
Starting simple

- Two class case, ω_1 and ω_2



Starting simple

- Given a sample x, is it ω_1 or ω_2?
  - i.e. $P(\omega_i \mid x) = ?$



Getting the answer

- The *class posterior probability* is:

Likelihood *Priors*

$$P(\omega_i \mid x) = \frac{P(x \mid \omega_i)P(\omega_i)}{P(x)}$$

Evidence

- To find the answer we need to fill in the terms in the right-hand-side

Filling the unknowns

- Class priors
  - How much of each class?

$$P(\omega_1) \approx N_1 / N$$
$$P(\omega_2) \approx N_2 / N$$

- Class likelihood: $P(x \mid \omega_i)$
  - Requires that we know the distribution of ω_i
    - We'll assume it is the Gaussian

Filling the unknowns

- Evidence:

$$P(x) = P(x \mid \omega_1)P(\omega_1) + P(x \mid \omega_2)P(\omega_2)$$

- We now have $P(\omega_1 \mid x), P(\omega_2 \mid x)$

Making the decision

- Bayes classification rule

$$\text{If } P(\omega_1 \mid x) > P(\omega_2 \mid x) \text{ then } x \text{ belongs to class } \omega_1$$

$$\text{If } P(\omega_1 \mid x) < P(\omega_2 \mid x) \text{ then } x \text{ belongs to class } \omega_2$$

- Easier version

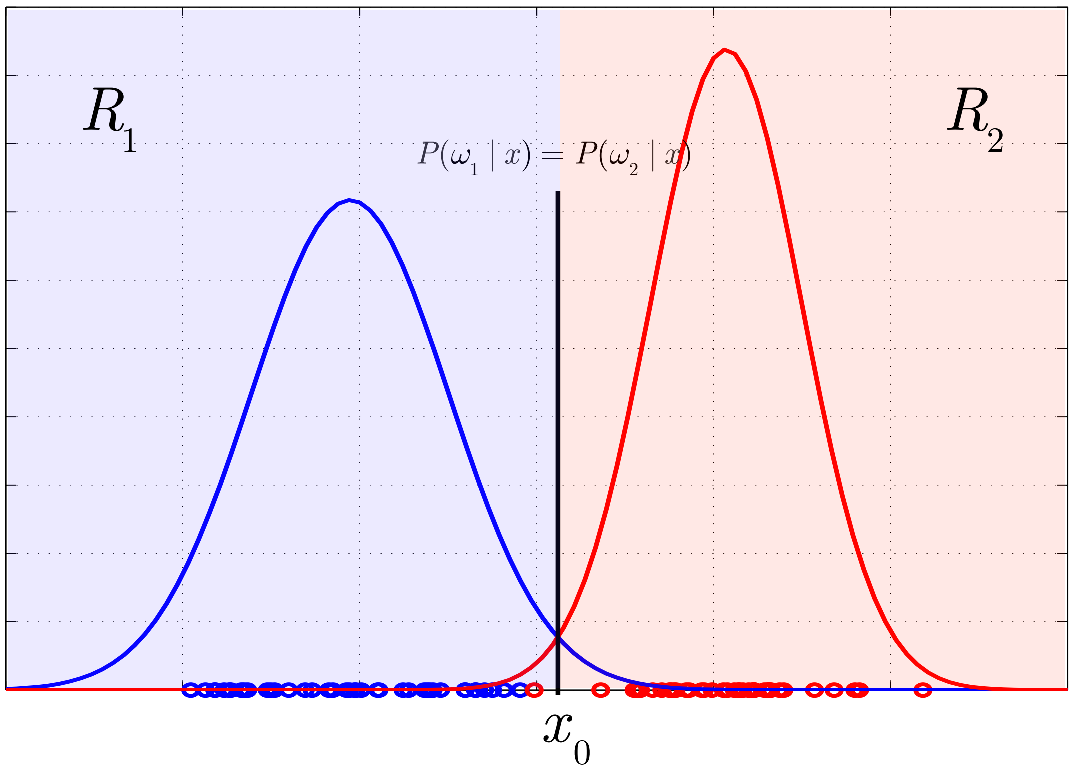
$$P(x \mid \omega_1)P(\omega_1) \gtrless P(x \mid \omega_2)P(\omega_2)$$

- Equiprobable class version

$$P(x \mid \omega_1) \gtrless P(x \mid \omega_2)$$

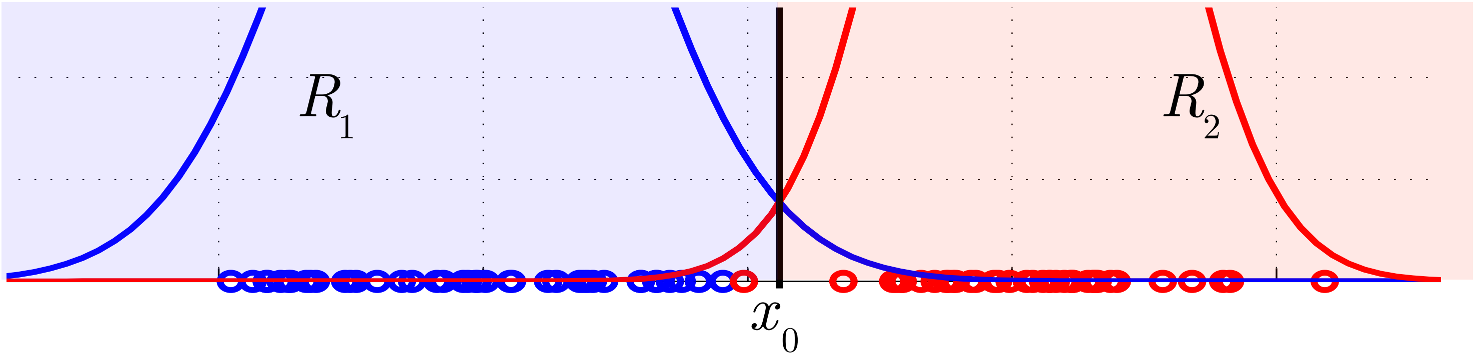
Visualizing the decision

- Assume Gaussian data
  - $P(x \mid \omega_i) = \mathcal{N}(x \mid \mu_i, \sigma_i)$



Errors in classification

- We can't win all the time though
  - Some inputs will be misclassified

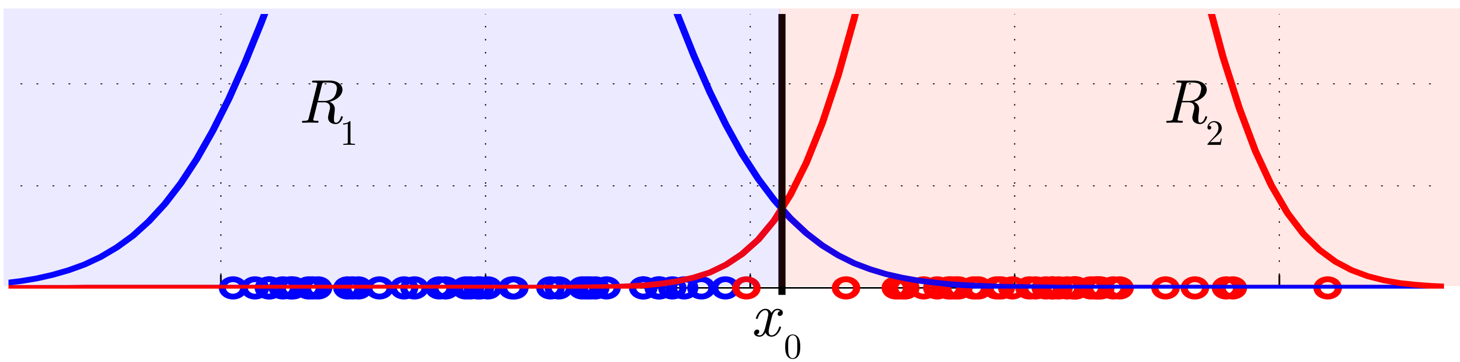


- Probability of errors

$$P_{error} = \frac{1}{2}\int_{-\infty}^{x_0} P(x \mid \omega_2)\,dx + \frac{1}{2}\int_{x_0}^{\infty} P(x \mid \omega_1)\,dx$$

Minimizing misclassifications

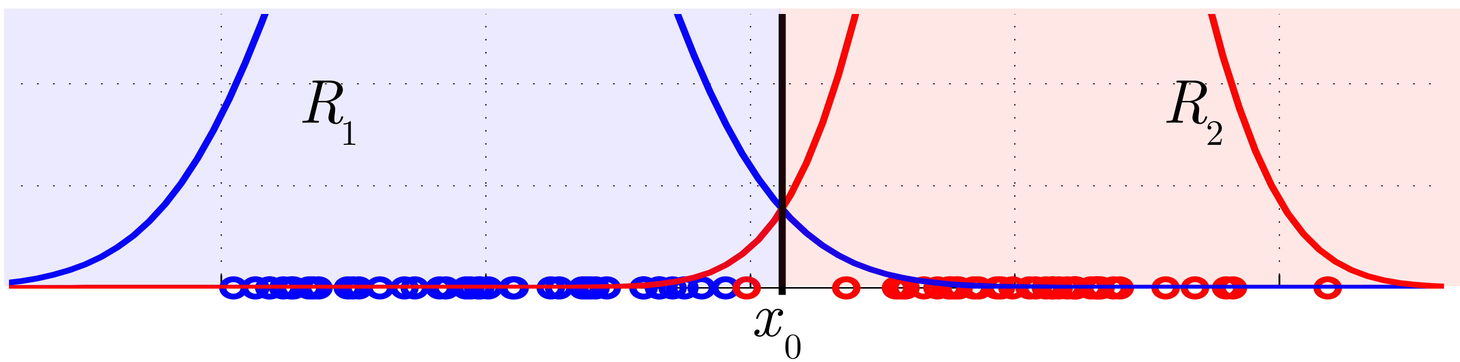
- The Bayes classification rule minimizes any potential misclassifications



- Can you do any better moving the line?

Minimizing *risk*

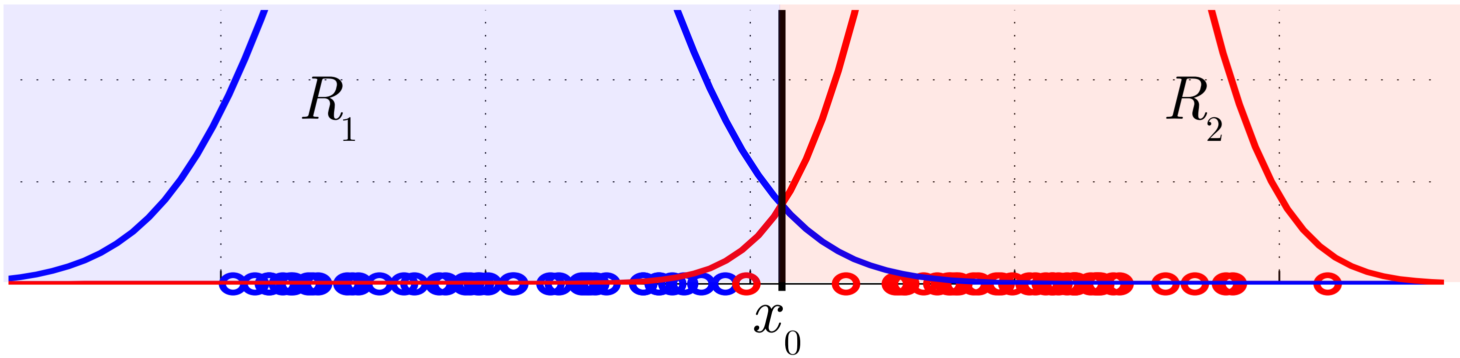
- Not all errors are equal!
  - e.g. medical diagnoses



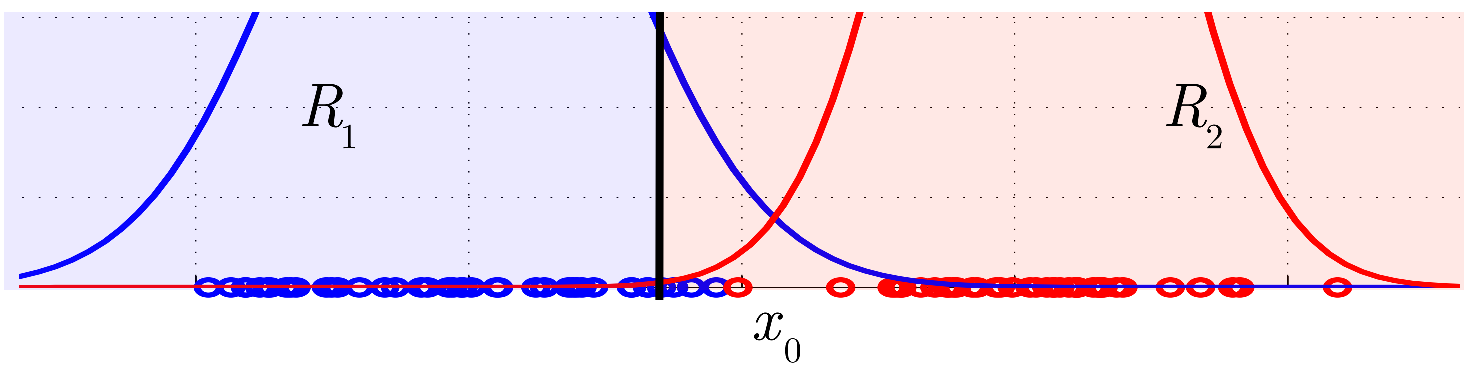
- Misclassification is often very tolerable depending on the assumed risks

Example

- Minimum classification error



- Minimum risk with $\lambda_{21} > \lambda_{12}$
  - i.e. class 2 is more important



ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

True/False – Positives/Negatives

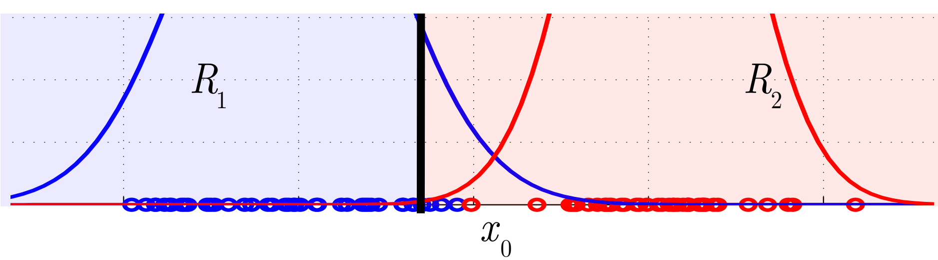
- Naming the outcomes

classifying for ω_1	x is ω_1	x is ω_2
x classified as ω_1	True positive	False positive
x classified as ω_2	False negative	True negative

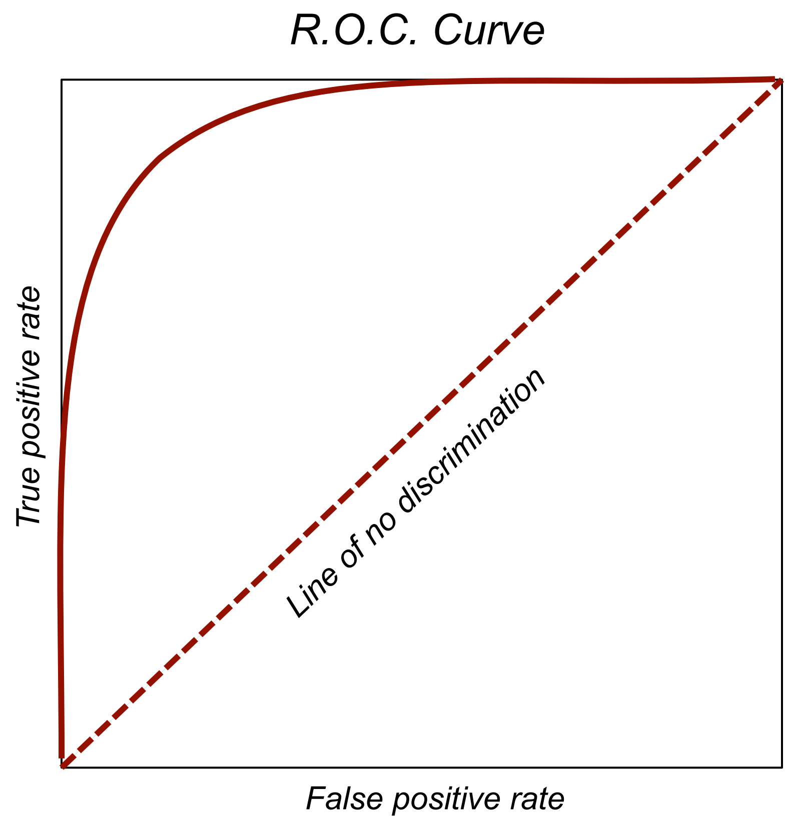
- False positive/false alarm/Type I error
- False negative/miss/Type II error

Receiver Operating Characteristic

- Visualize classification balance



http://www.anaesthetist.com/mnm/stats/roc/Findex.htm



Classifying Gaussian data

- Remember that we need the class likelihood to make a decision
  - For now we'll assume that:

$$P(x \mid \omega_i) = \mathcal{N}(x \mid \mu_i, \sigma_i)$$

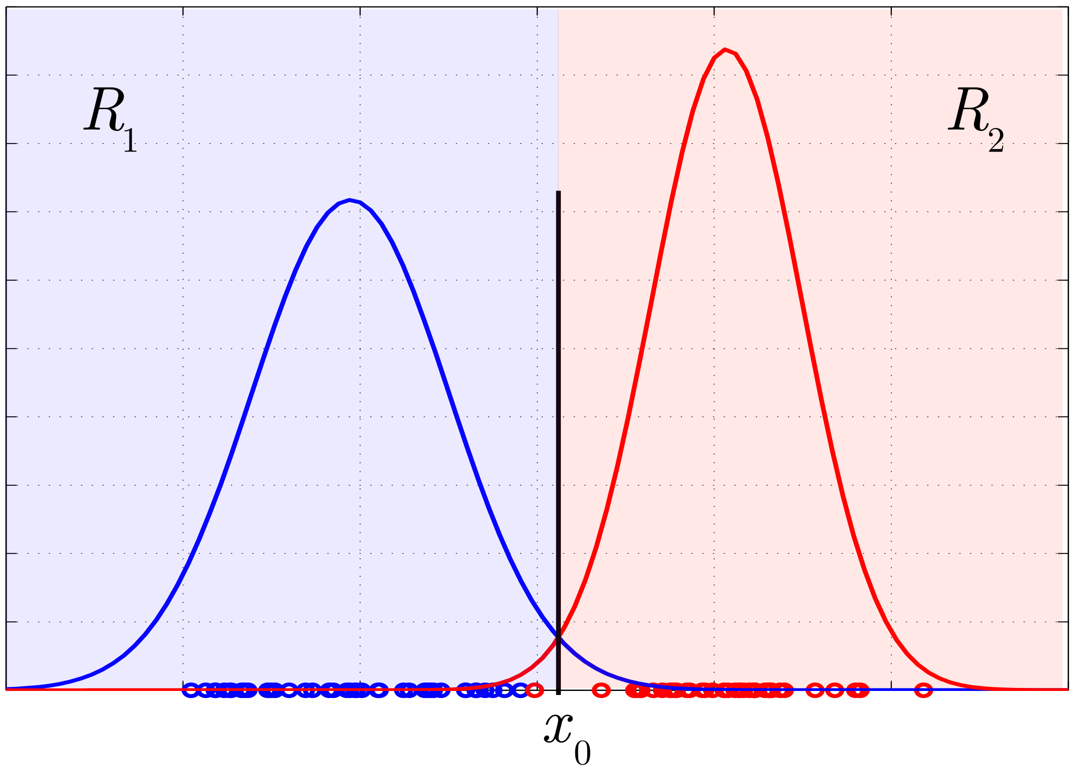
  - i.e. that the input data is Gaussian distributed

Overall methodology

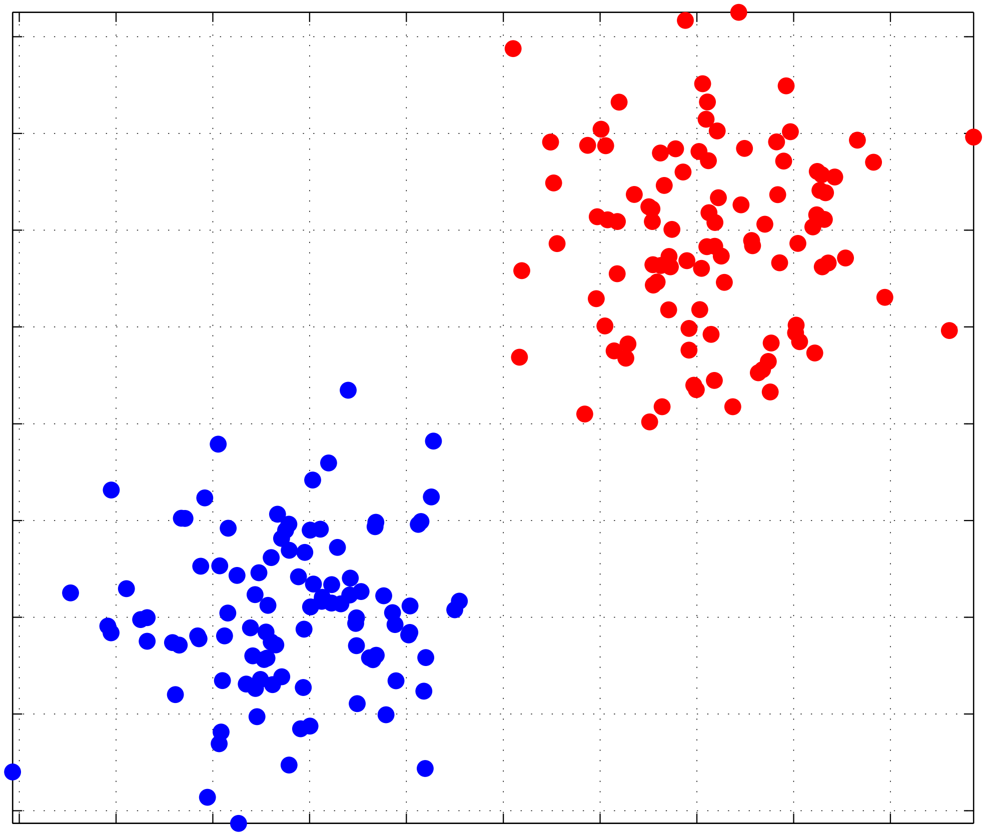
- Obtain training data
- Fit a Gaussian model to each class
  - Perform parameter estimation for mean, variance and class priors
- Define decision regions based on models and any given constraints

1D example

- The *decision boundary* will always be a line separating the two class regions

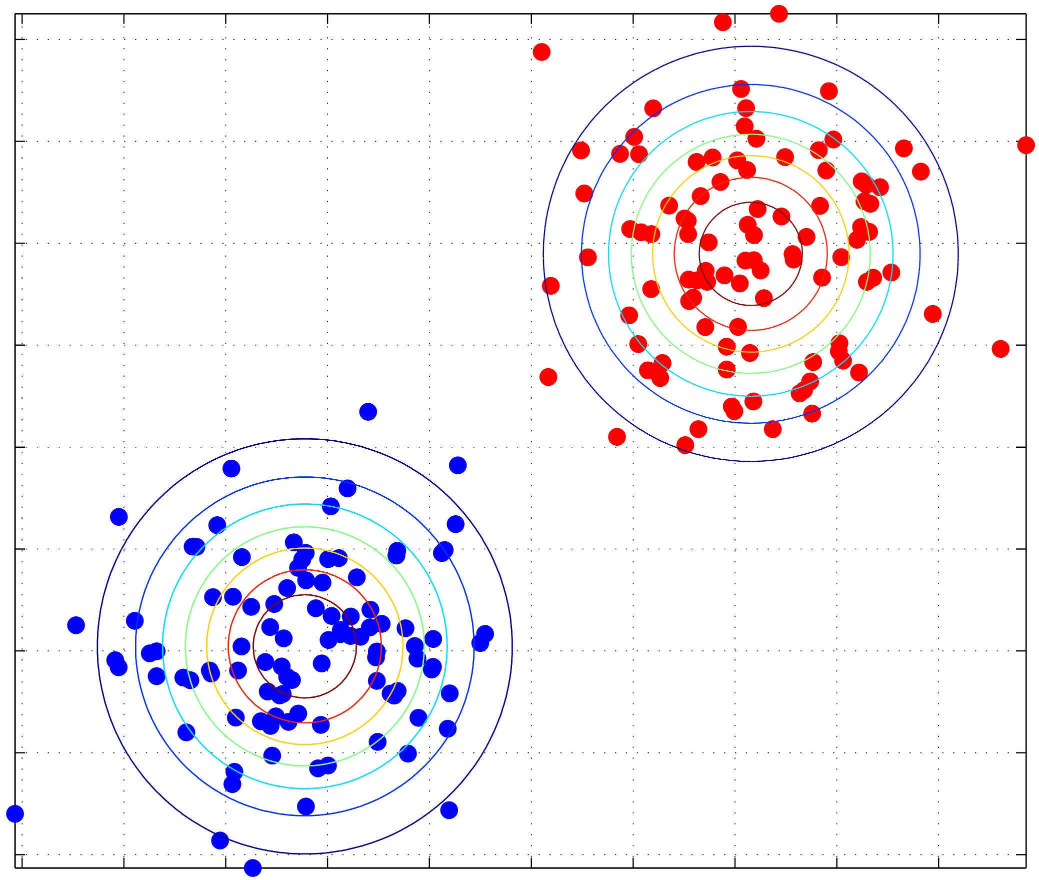


2D example



ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

2D example fitted Gaussians



Gaussian decision boundaries

- The decision boundary is defined as:

$$P(\mathbf{x} \mid \omega_1)P(\omega_1) = P(\mathbf{x} \mid \omega_2)P(\omega_2)$$

- We can substitute Gaussians and solve to find what the boundary looks like

Discriminant functions

- Define a function so that:

$$\text{classify } \mathbf{x} \text{ in } \omega_i \text{ if } g_i(\mathbf{x}) > g_j(\mathbf{x}), \forall i \neq j$$

- Decision boundaries are now defined as:

$$g_{ij}(\mathbf{x}) \equiv \left(g_i(\mathbf{x}) = g_j(\mathbf{x}) \right)$$

Back to the data

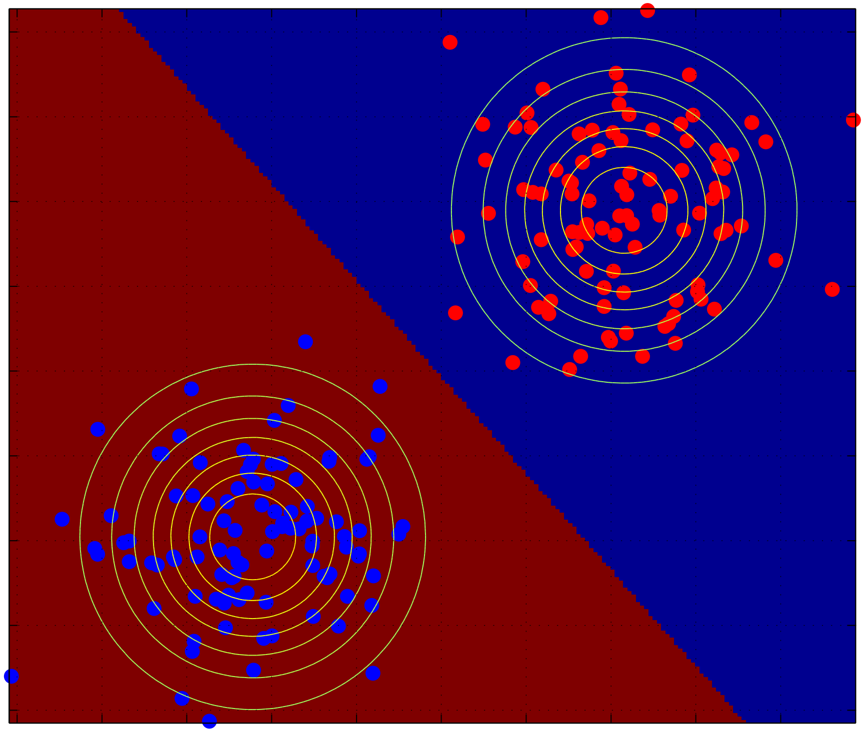
- $\mathbf{\Sigma}_i = \sigma_i^2 \mathbf{I}$ produces line boundaries

- Discriminant:

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

$$\mathbf{w}_i = \boldsymbol{\mu}_i / \sigma^2$$

$$b = -\frac{\boldsymbol{\mu}_i^T \boldsymbol{\mu}_i}{2\sigma^2} + \log P(\omega_i)$$



Quadratic boundaries

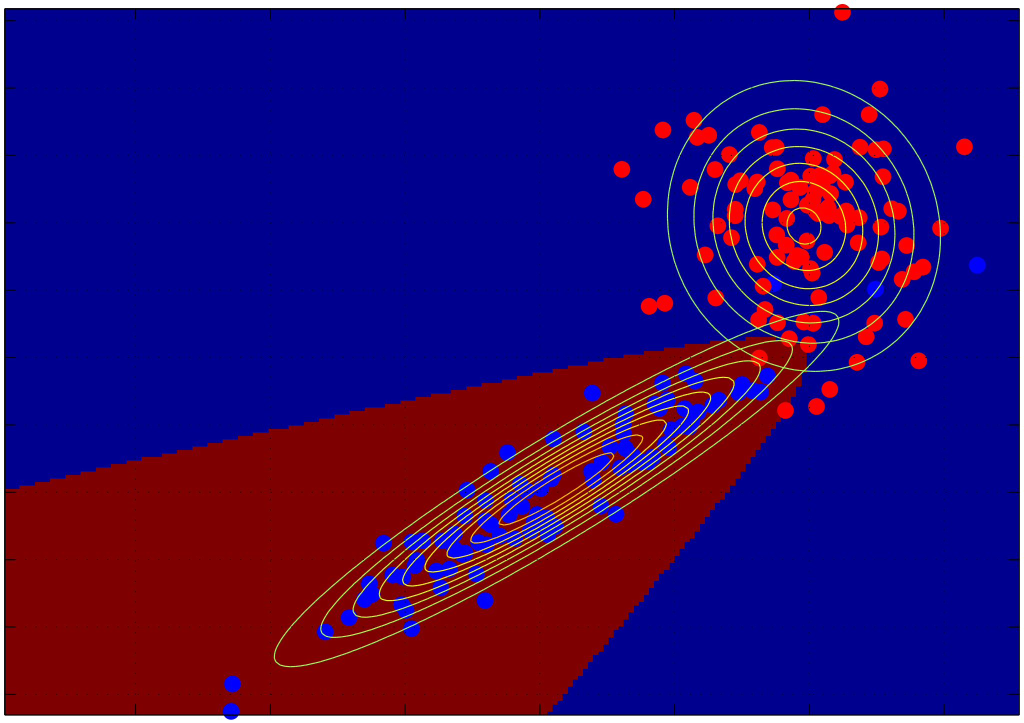
- Arbitrary covariance produces more elaborate patterns in the boundary

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

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

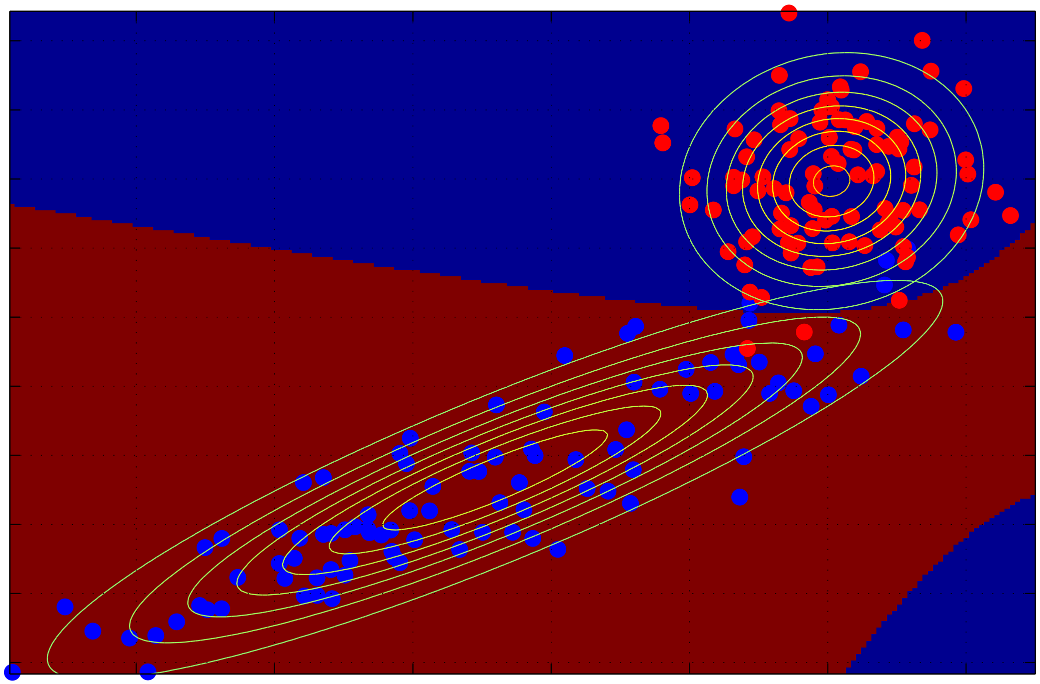
$$\mathbf{w}_i = \boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i$$

$$w = -\frac{1}{2}\boldsymbol{\mu}_i^T \boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i - \frac{1}{2}\log\left|\boldsymbol{\Sigma}_i\right| + \log P(\omega_i)$$



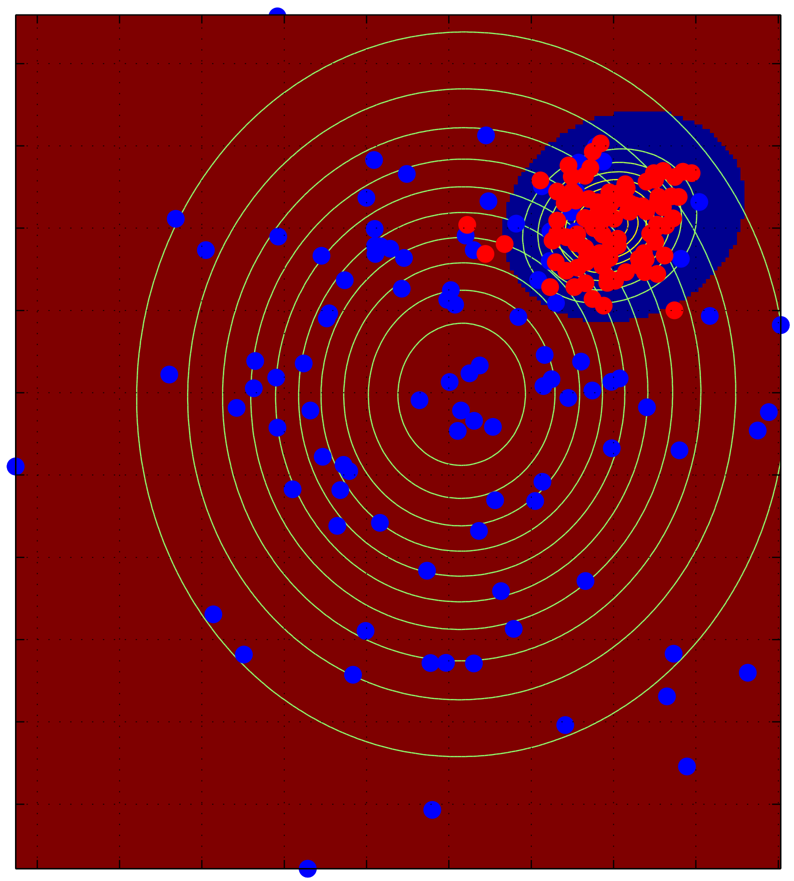
Quadratic boundaries

- Arbitrary covariance produces more elaborate patterns in the boundary



Quadratic boundaries

- Arbitrary covariance produces more elaborate patterns in the boundary



Example classification run

- Learning to recognize two handwritten digits

- Let's try this in MATLAB

Recap

- The Gaussian
- Bayesian Decision Theory
  - Risk, decision regions
  - Gaussian classifiers