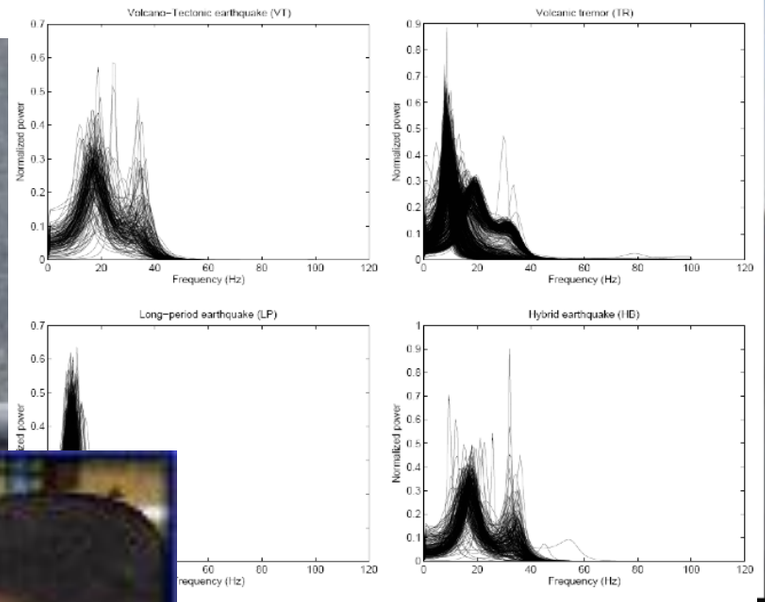
Machine Learning: the basics

- Learning from examples



TU Delft

Outline of this lecture

- Objects, features, measurements,
- ... datasets and feature space
- Traditional pattern recognition: classification
- Class posterior probabilities and Bayes' rule
- Bayes' classifier and Bayes' error
- Misclassification costs

Learning from Examples

- Given some **examples**, we may perform:
  - clustering
  - outlier detection
  - classification
  - regression
  - …

  on new objects.

- We assume that no complete physical model is known!

TU Delft

Generalization

We don't want to just describe the data...

We want to predict for new, unseen data!

TU Delft

Generalization

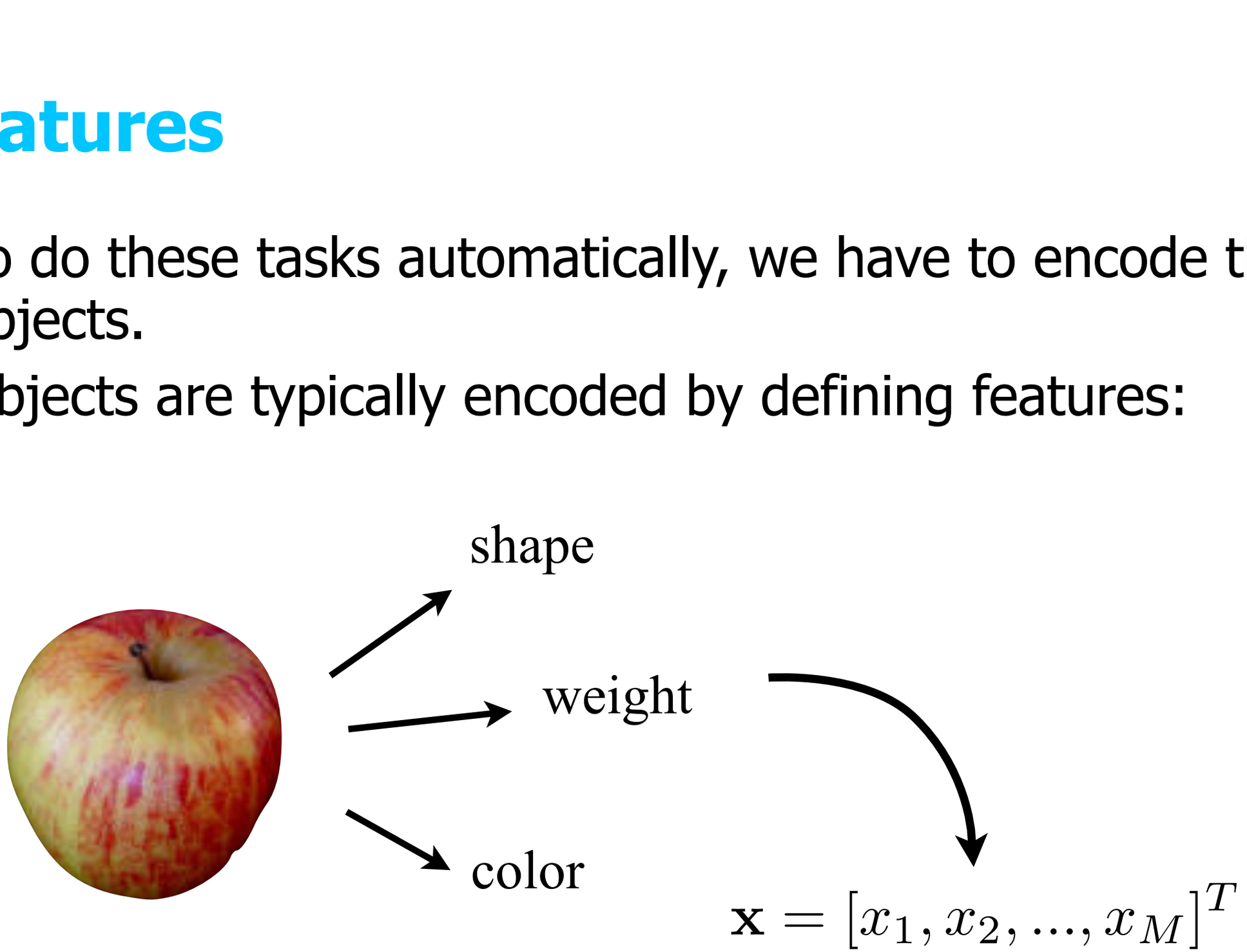
- **Training set**:
  All examples are labeled
  This set is used to train/develop our system

- **Test set**:
  These examples cannot be used to train our system
  The examples do not have to be labeled
  When labels are available, we can objectively evaluate our system

TU Delft

Features

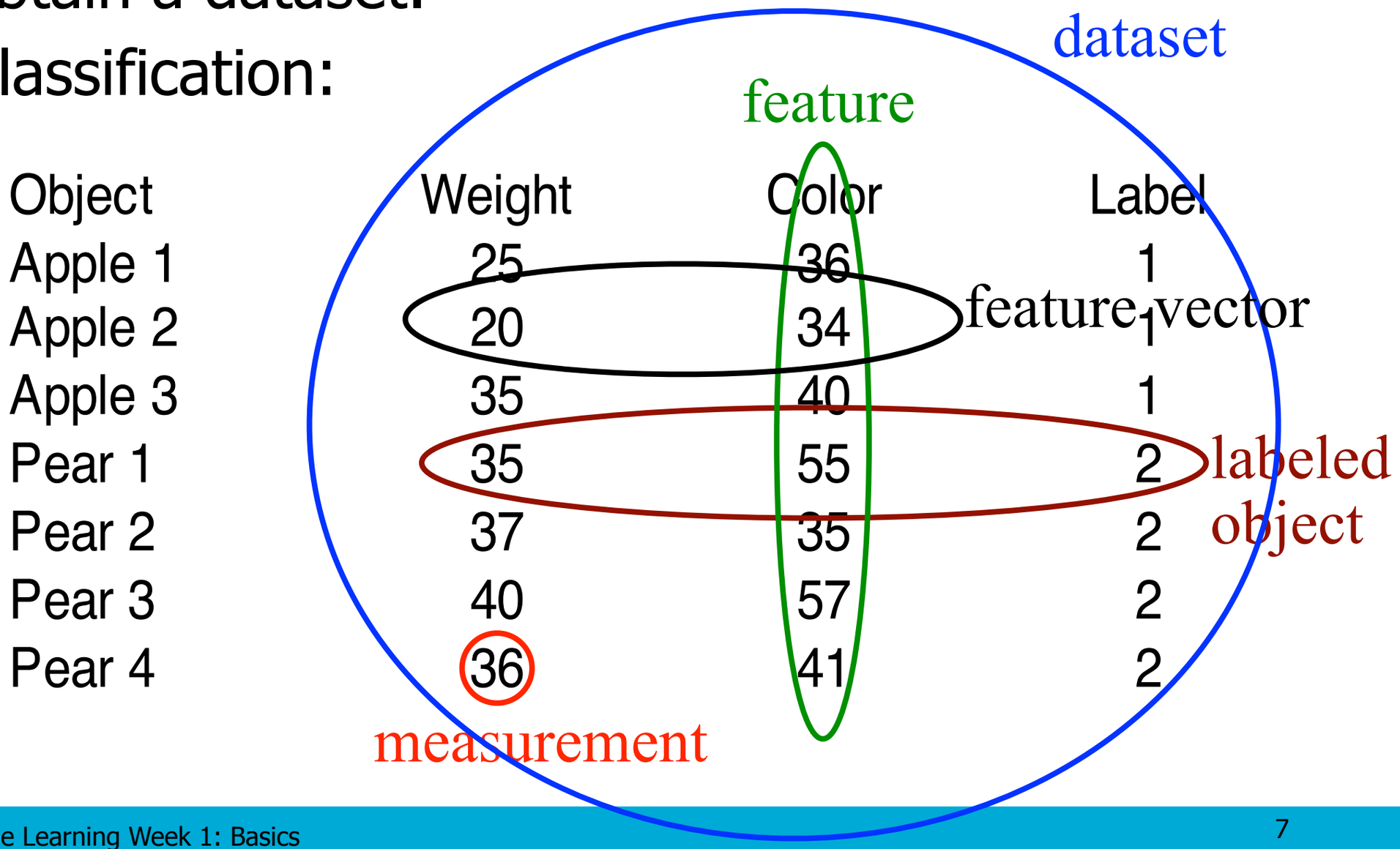
- To do these tasks automatically, we have to encode the objects.
- Objects are typically encoded by defining features:



TU Delft

Datasets

- When we measure the features of many objects: we obtain a dataset.
- For classification:



TU Delft

How to define features?

- Note that features reduce, and give a specific view of the objects: YOU (the user) is responsible for it
- Good features allow for pattern recognition, bad features allow for nothing
- Other (than feature vector) approaches of defining objects are:
  - Dissimilarity approach
  - Structural pattern recognition (graphs)

- Feature approach is very well developed, other approaches are still more research.

TU Delft

Noise in the measurements

- The measurements will never be perfect
- Objects within a class will vary

We need to apply some statistics to cover all the variations

TU Delft

Measurements

- Task: distinguish between 3 types of Iris flowers:

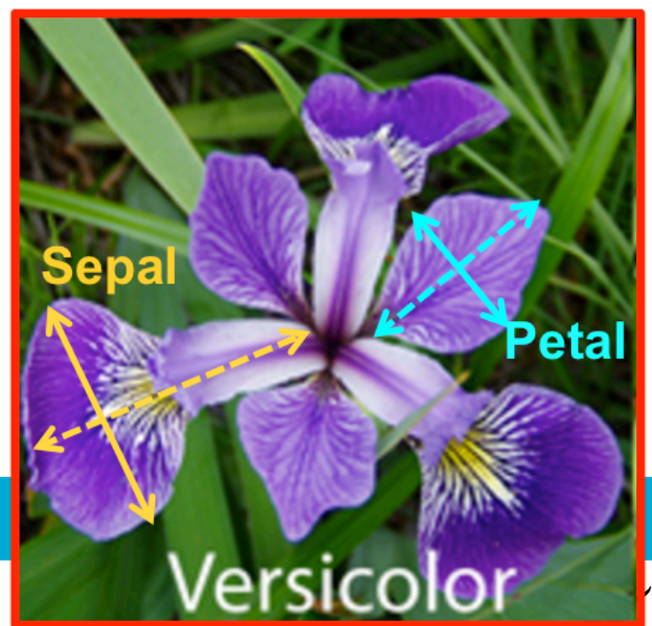
(Images from Wikipedia)

Iris Setosa

Iris Versicolor

Iris Virginica

- Measurements:
  sepal width, sepal length,
  petal width, petal length.



TU Delft

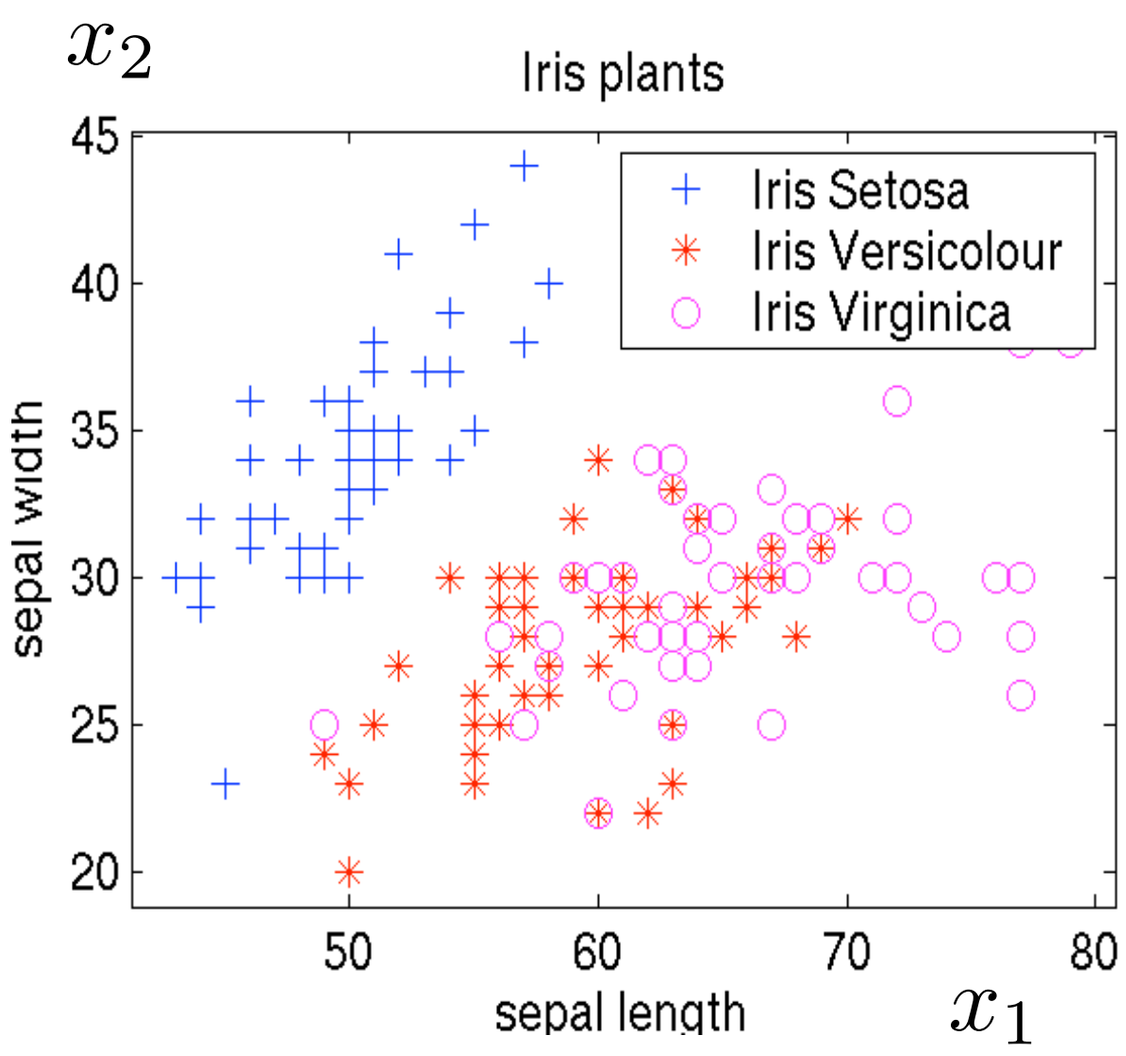
Objects in feature space

- We can interpret the measurements as a vector in a vector space:

$$\mathbf{x} = [x_1, x_2, ..., x_M]^T$$

- This originates in principle from a probability density over the whole feature space

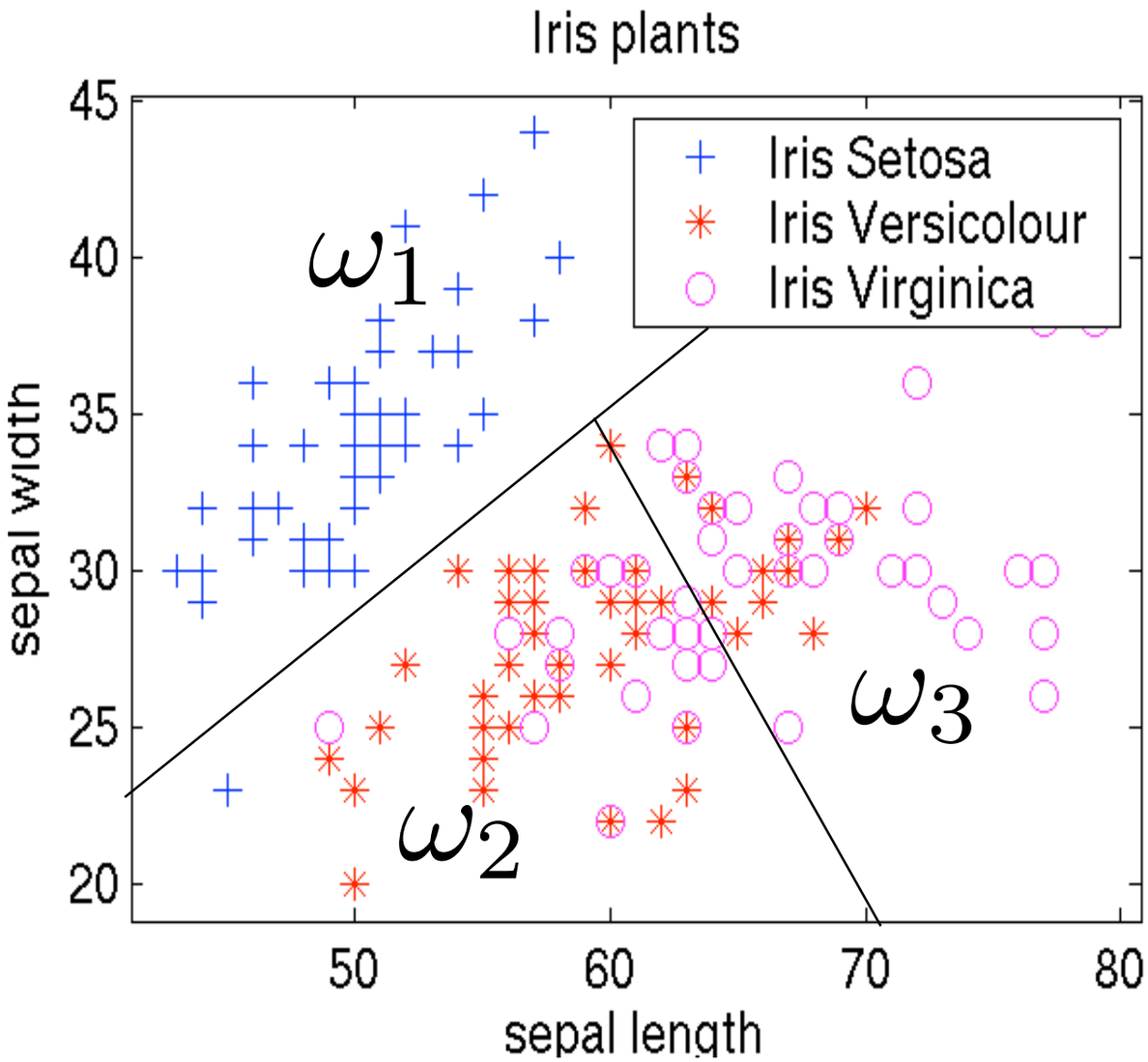
$$p(\mathbf{x}, y)$$



TU Delft

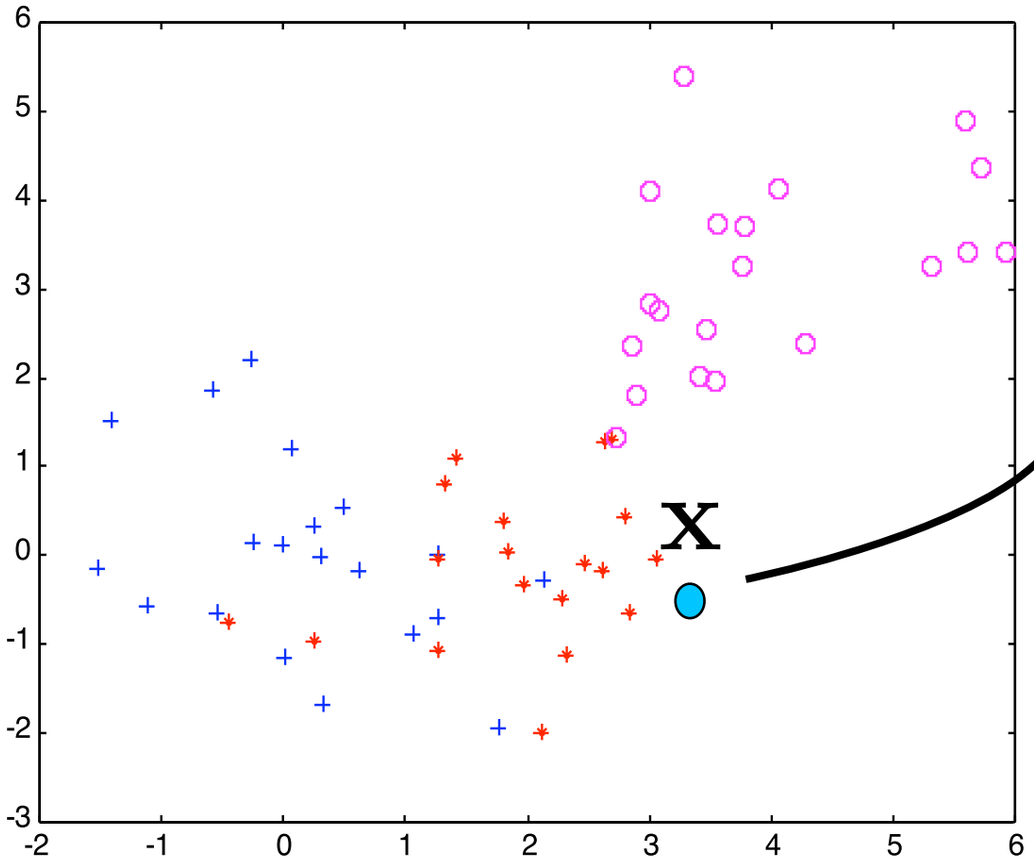
Classification

- Given labeled data: $\mathbf{x}$
- Assign to each object a class label ω
- In effect splits the feature space in separate regions



TU Delft

The general model



Function f should give the predicted output.

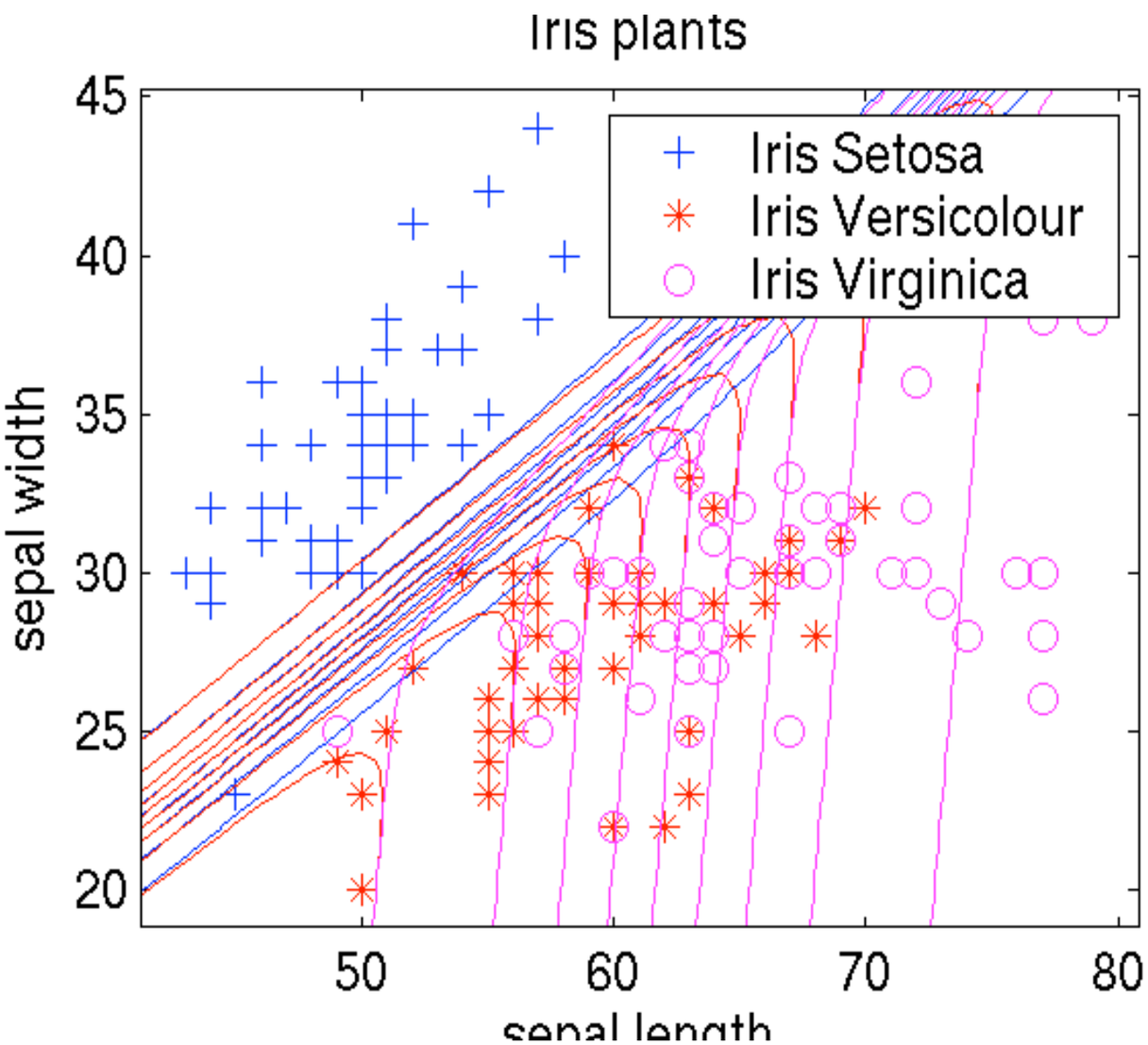
$f(\mathbf{x})$ → ω

$p(\omega|\mathbf{x})$

and... model should be fitted to the data!

TU Delft

Output of the model



- For each object in the feature space, we should estimate:

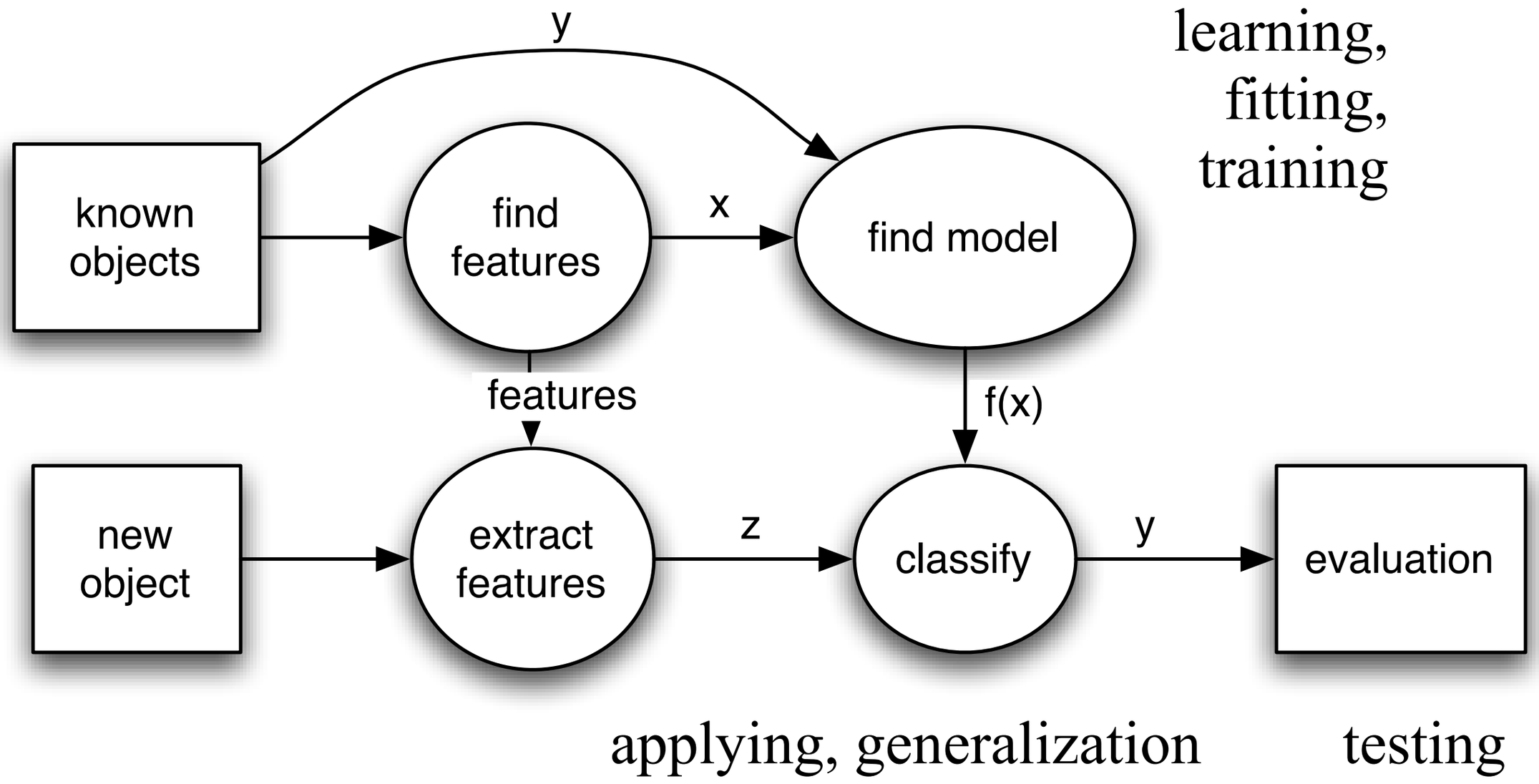
$$p(\omega|\mathbf{x})$$

- In practice we fit a function:

$$f(\mathbf{x})$$

TU Delft

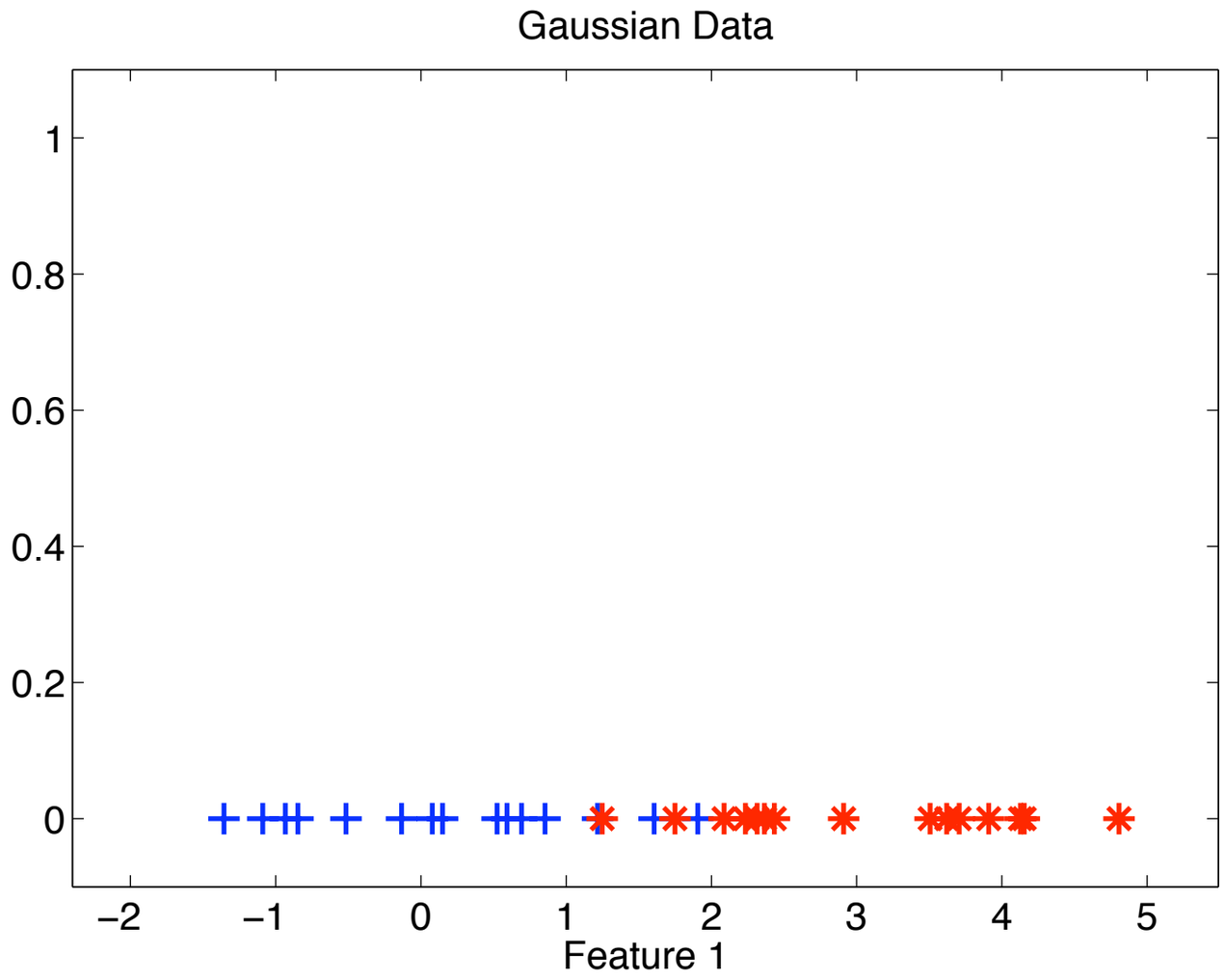
Pattern Recognition pipeline



TU Delft

Classification, how to do it?

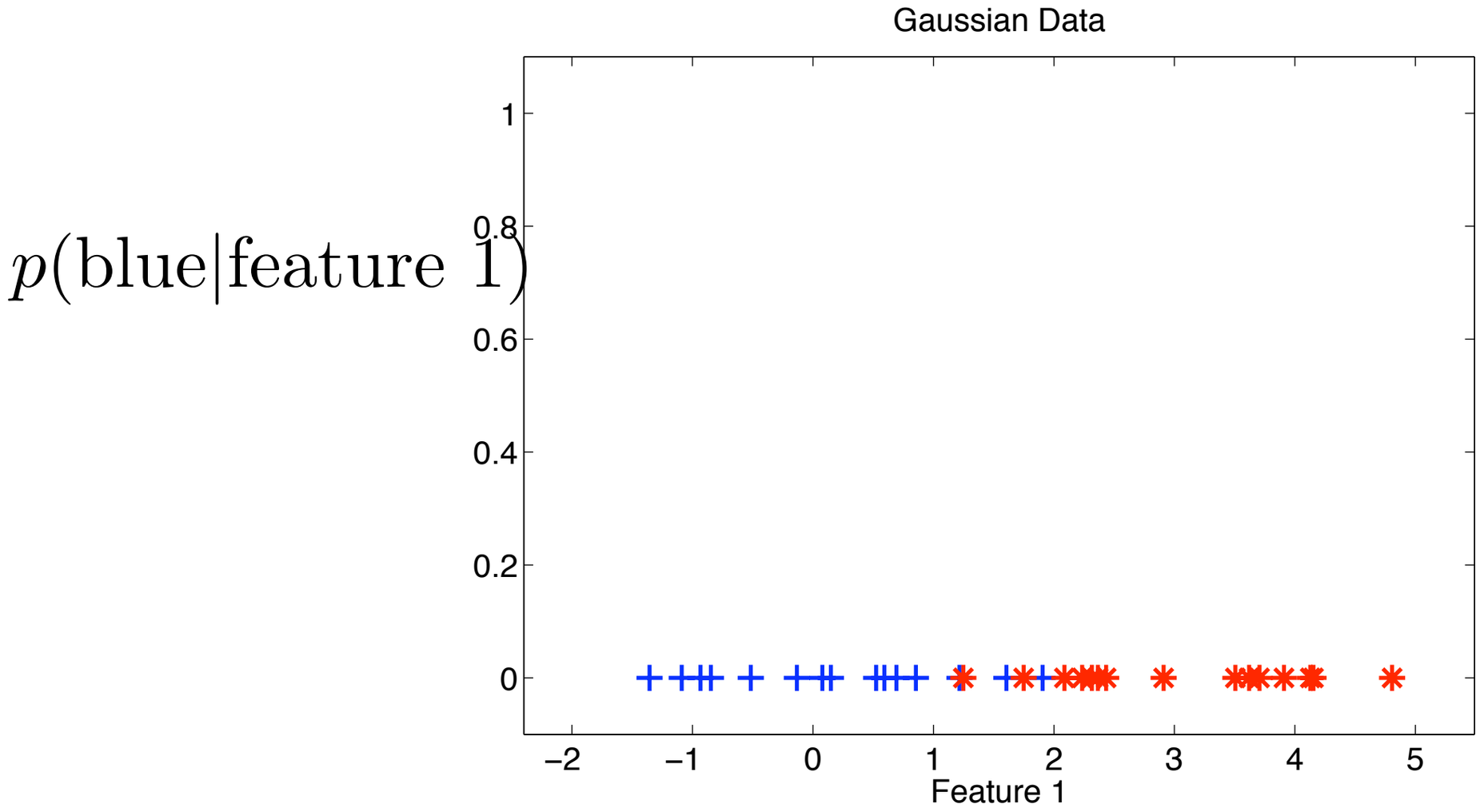
- Given a feature, and a training set, where is the blue class?



TU Delft

Class posterior probability

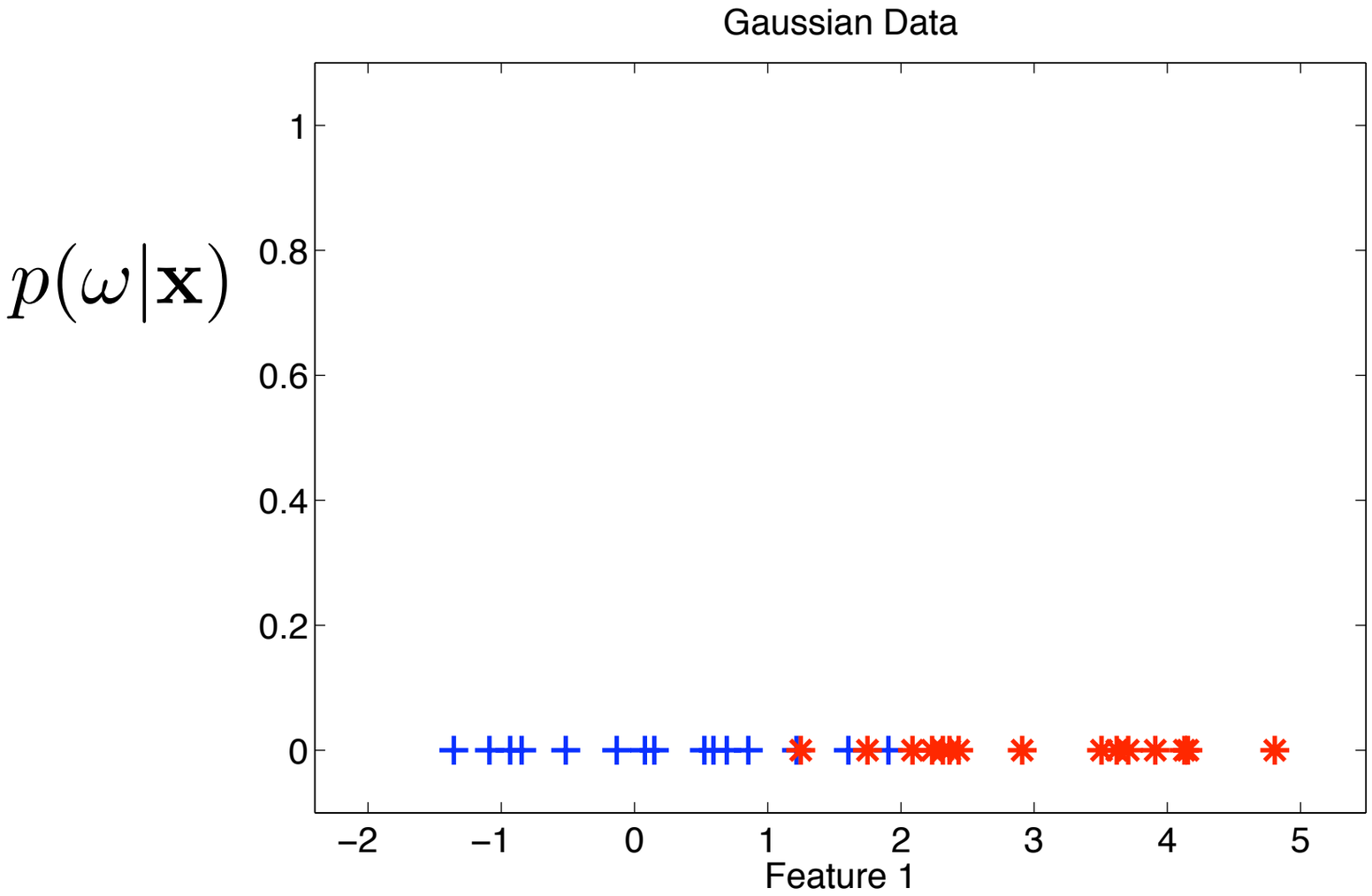
- For each object we want to estimate $p(\text{blue}|\text{feature } 1)$



TU Delft

Class posterior probability

- For each object we want to estimate $p(\omega|\mathbf{x})$

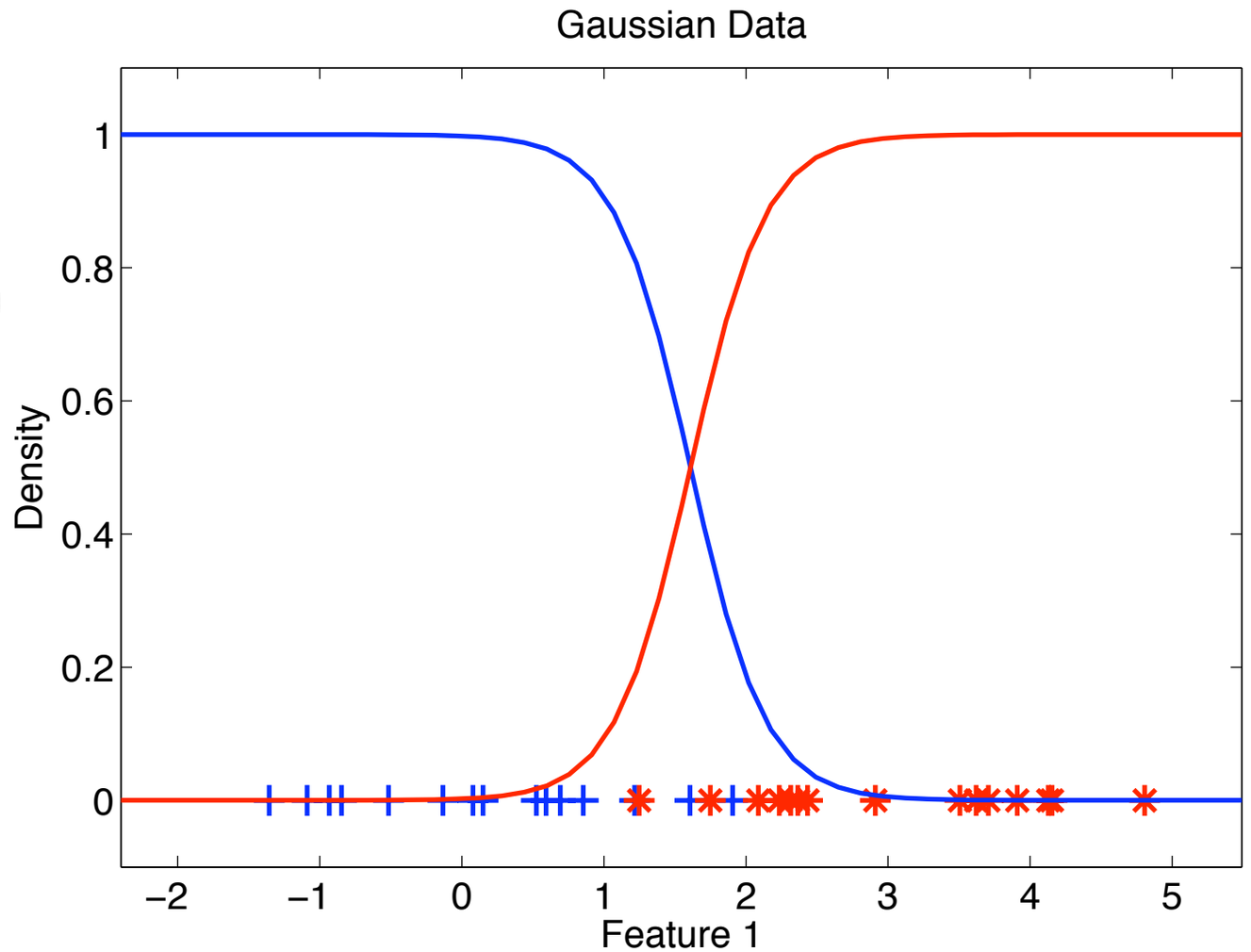


TU Delft

Class posterior probability

- For each object we have to estimate $p(\omega|\mathbf{x})$

$p(\omega|\mathbf{x})$



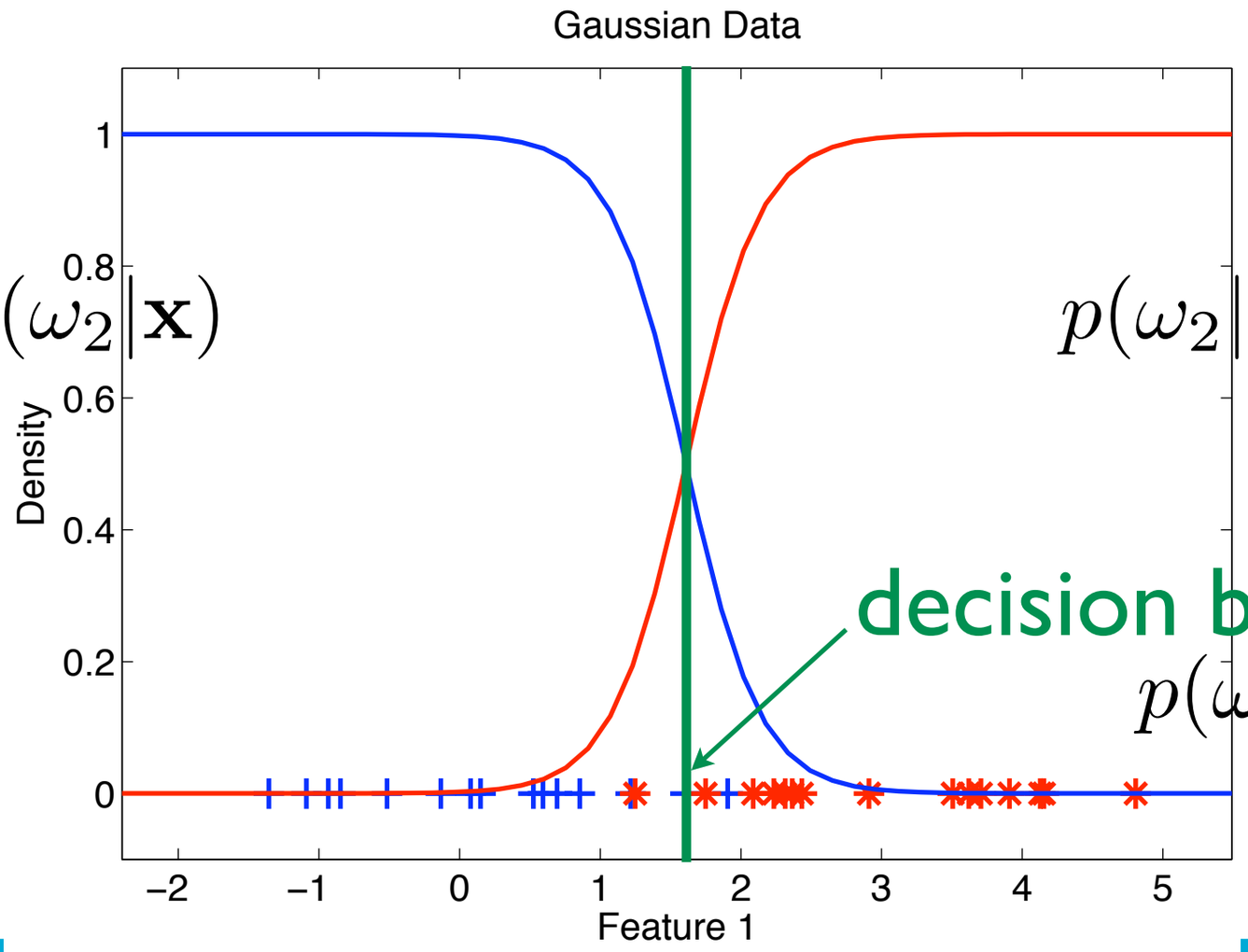
$$\sum_i p(\omega_i|\mathbf{x}) = 1$$

TU Delft

Classify new objects

- Assign the label of the class with the largest posterior probability

$$p(\omega_1|\mathbf{x}) > p(\omega_2|\mathbf{x})$$



$$p(\omega_2|\mathbf{x}) > p(\omega_1|\mathbf{x})$$

$$p(\omega_2|\mathbf{x}) = p(\omega_1|\mathbf{x})$$

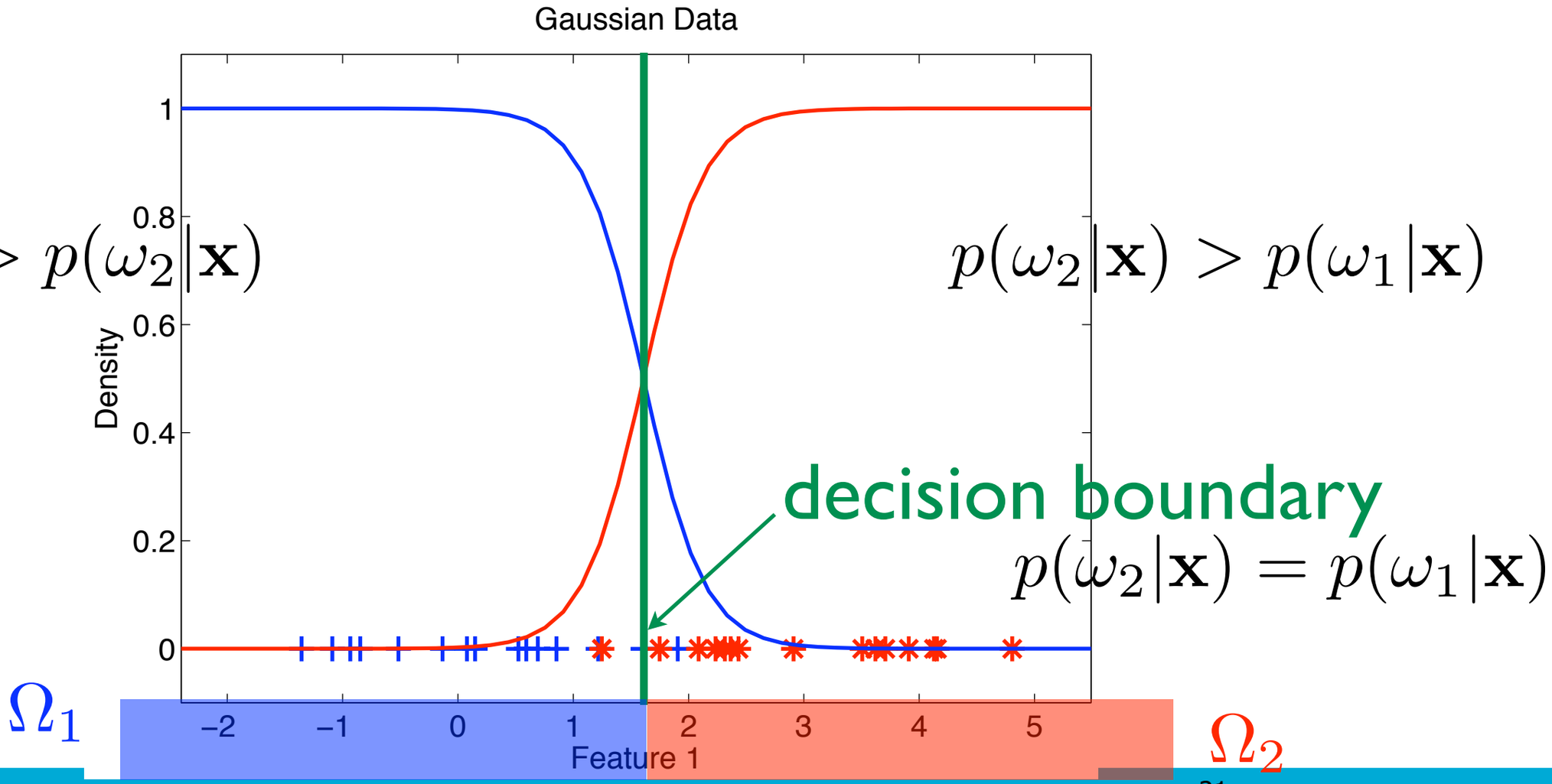
TU Delft

Classify new objects

- Assign the label of the class with the largest posterior probability

$$p(\omega_1|\mathbf{x}) > p(\omega_2|\mathbf{x})$$

$$p(\omega_2|\mathbf{x}) > p(\omega_1|\mathbf{x})$$



TU Delft

Description of a classifier

There are several ways to describe a classifier:

- if $p(\omega_1|\mathbf{x}) > p(\omega_2|\mathbf{x})$ then assign to ω_1
  otherwise ω_2

- if $p(\omega_1|\mathbf{x}) - p(\omega_2|\mathbf{x}) > 0$ then assign to ω_1

- or $\dfrac{p(\omega_1|\mathbf{x})}{p(\omega_2|\mathbf{x})} > 1$

- or $\ln(p(\omega_1|\mathbf{x})) - \ln(p(\omega_2|\mathbf{x})) > 0$

TU Delft

Bayes' theorem

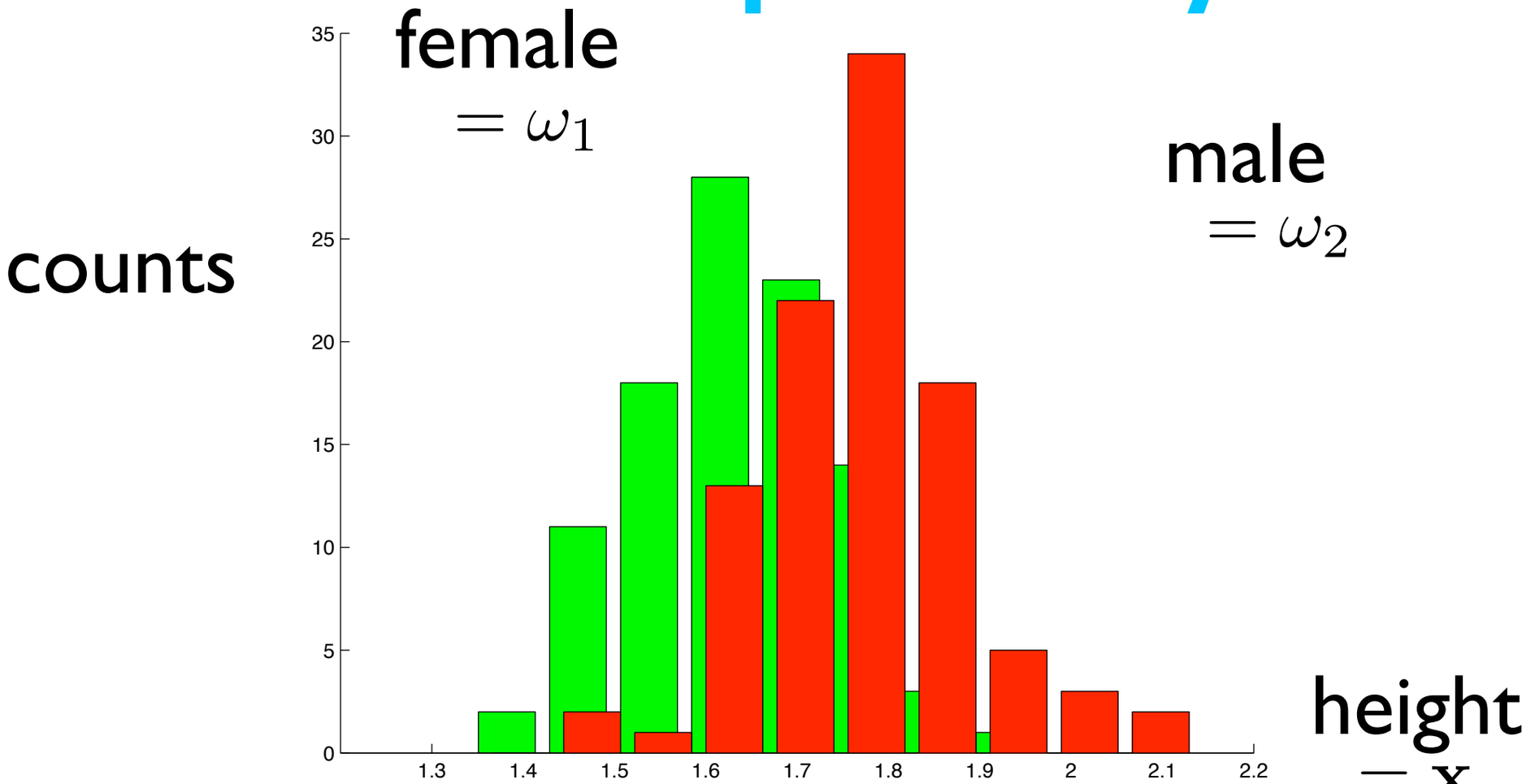
- In many cases the posterior is hard to estimate
- Often a functional form of the class distributions can be assumed
- Use Bayes' theorem to rewrite one into the other:

$$p(\omega|\mathbf{x}) = \frac{p(\mathbf{x}|\omega)p(\omega)}{p(\mathbf{x})}$$

class (conditional) distribution $p(\mathbf{x}|\omega)$
class prior $p(\omega)$
(unconditional) data distribution $p(\mathbf{x})$
posterior probability $p(\omega|\mathbf{x})$

TU Delft

Class conditional probability

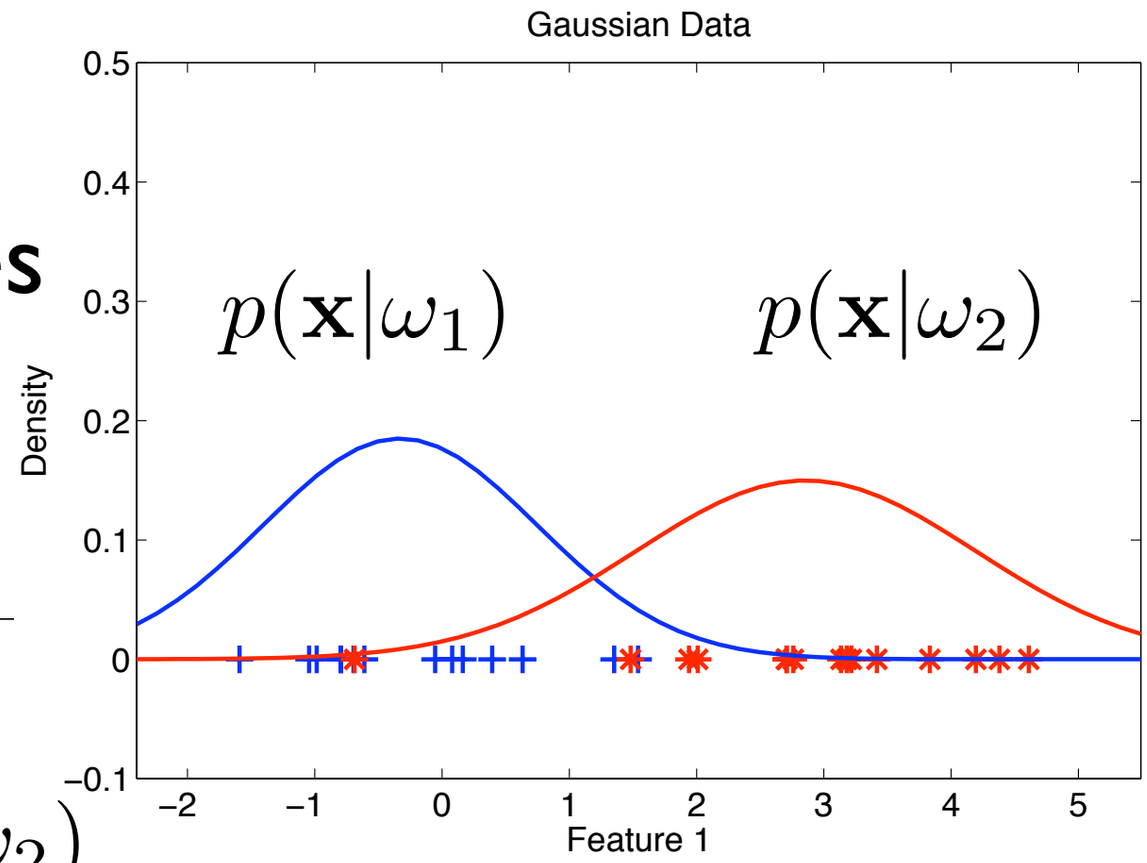


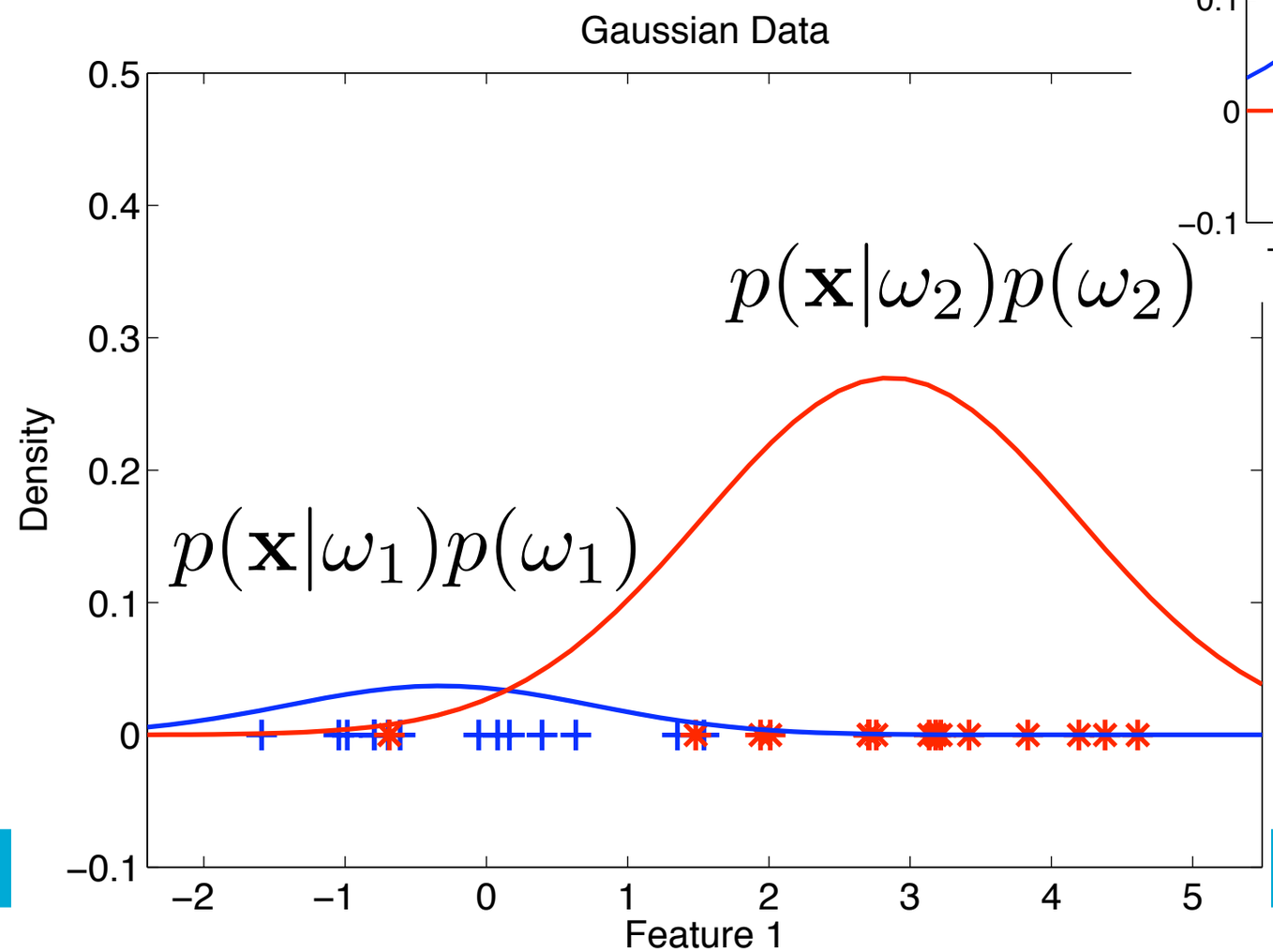
- Class conditional probabilities:
  - The distribution of the females $p(\mathbf{x}|\omega_1)$
  - The distribution of the males $p(\mathbf{x}|\omega_2)$

TU Delft

Bayes' rule

1. Estimate the class conditional probabilities





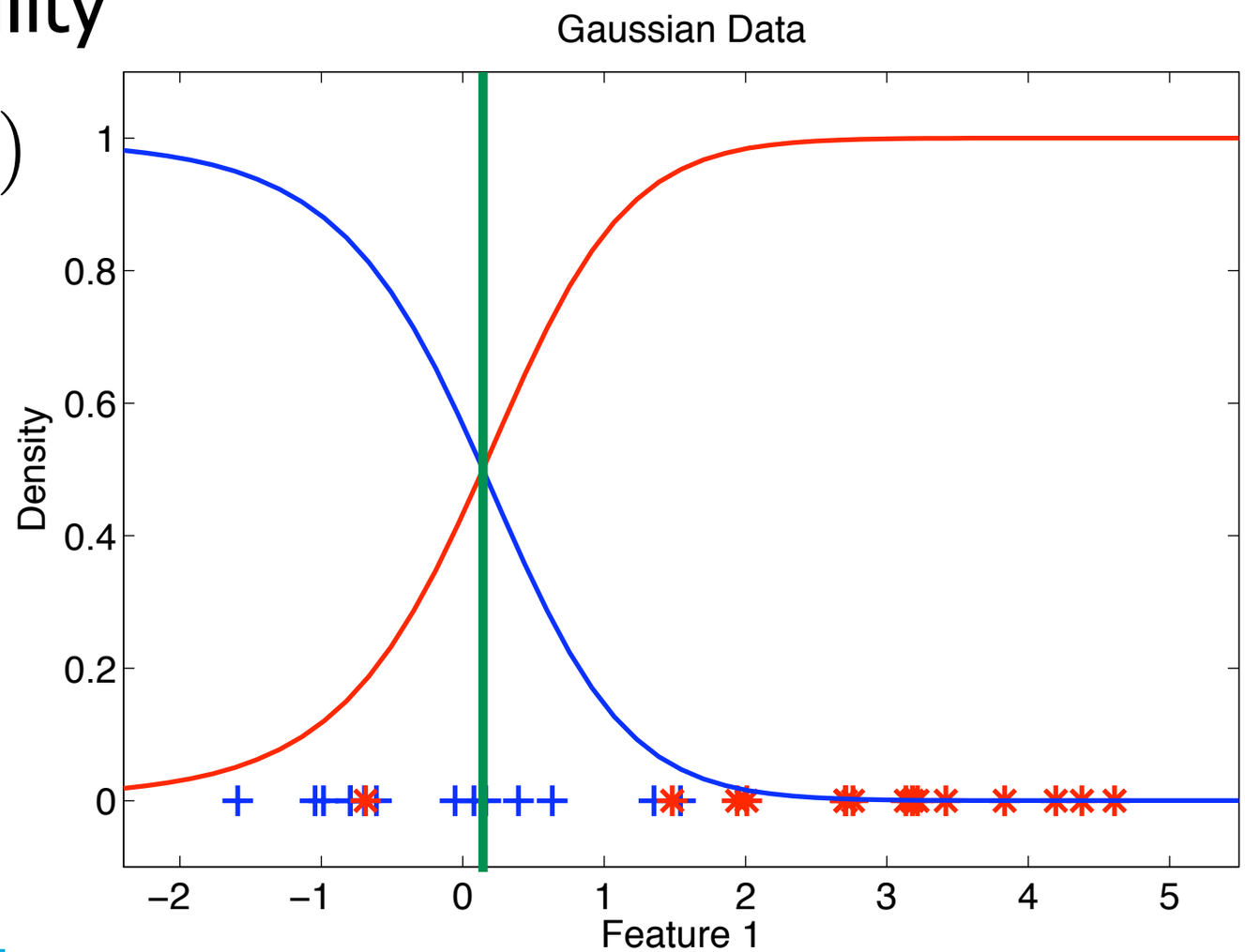
2. Multiply with the class priors

TU Delft

Bayes' rule

3. Compute the class posterior probabilities
4. Assign objects to the class with the highest posterior probability

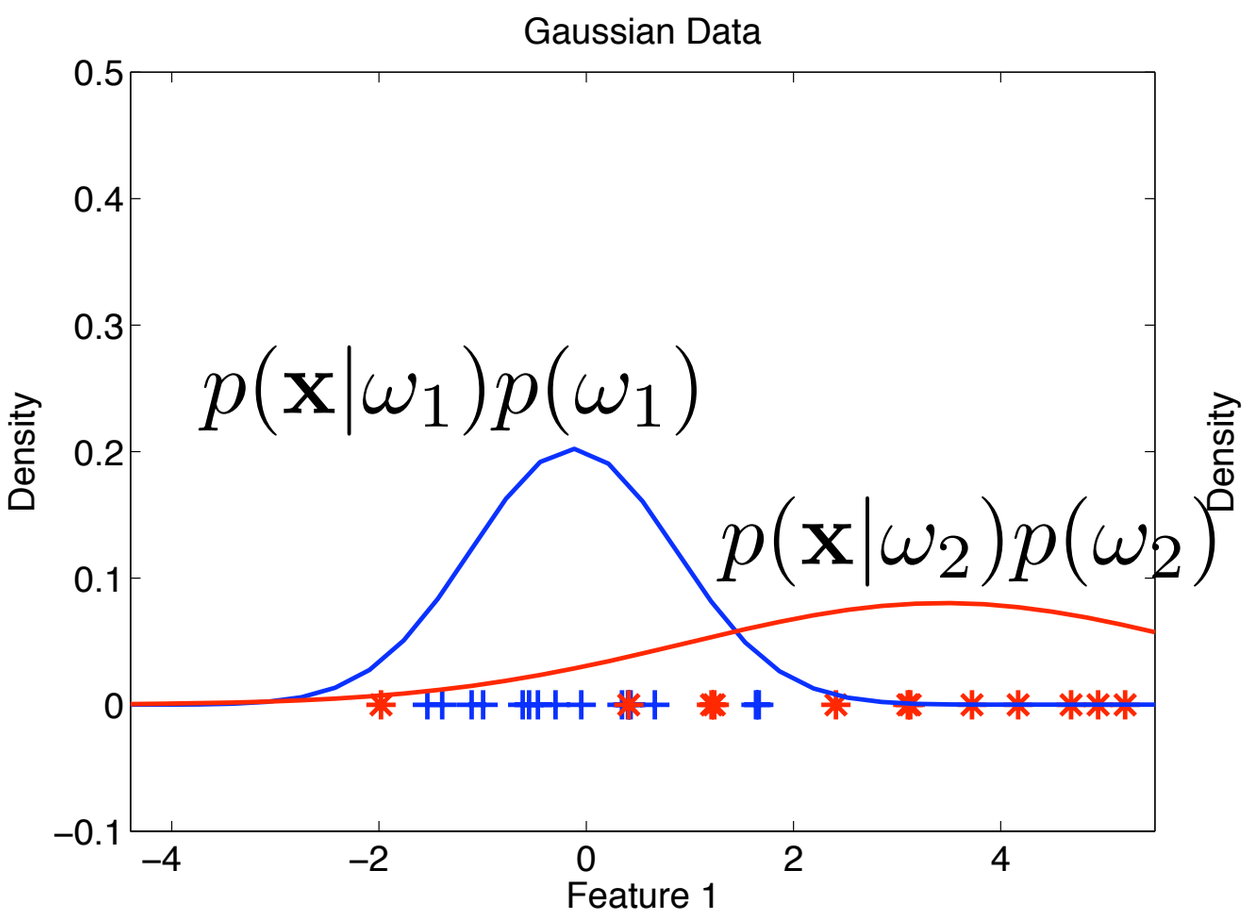
$p(\omega_1|\mathbf{x})$

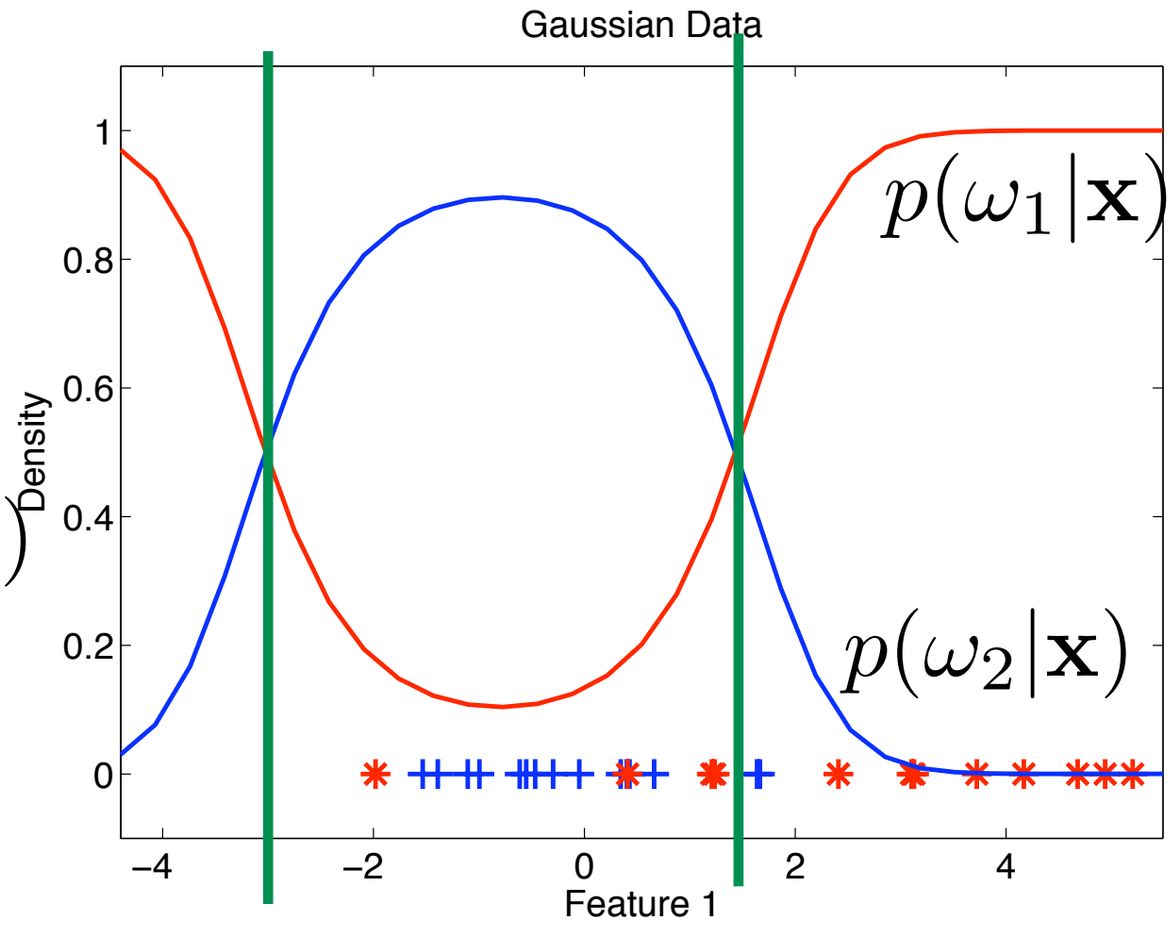


$p(\omega_2|\mathbf{x})$

TU Delft

Complicated decision boundary

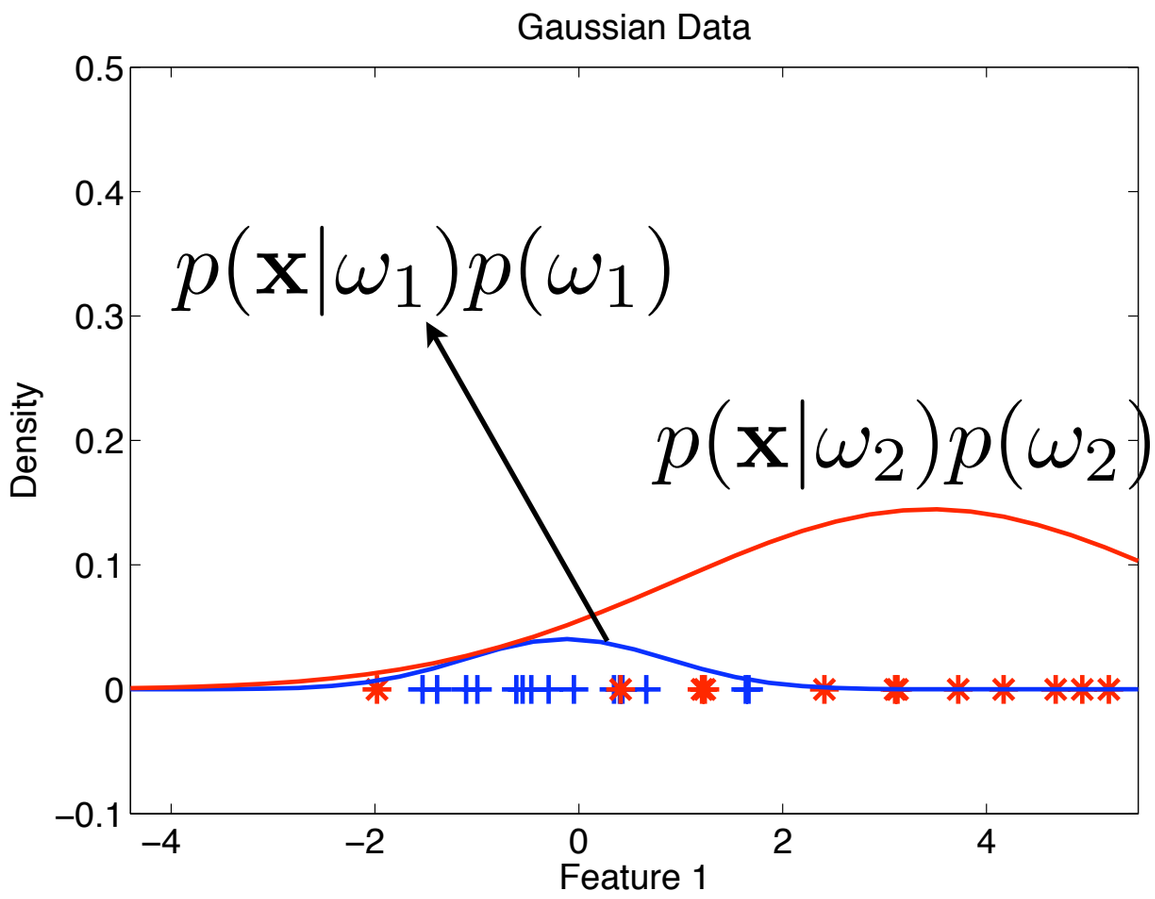


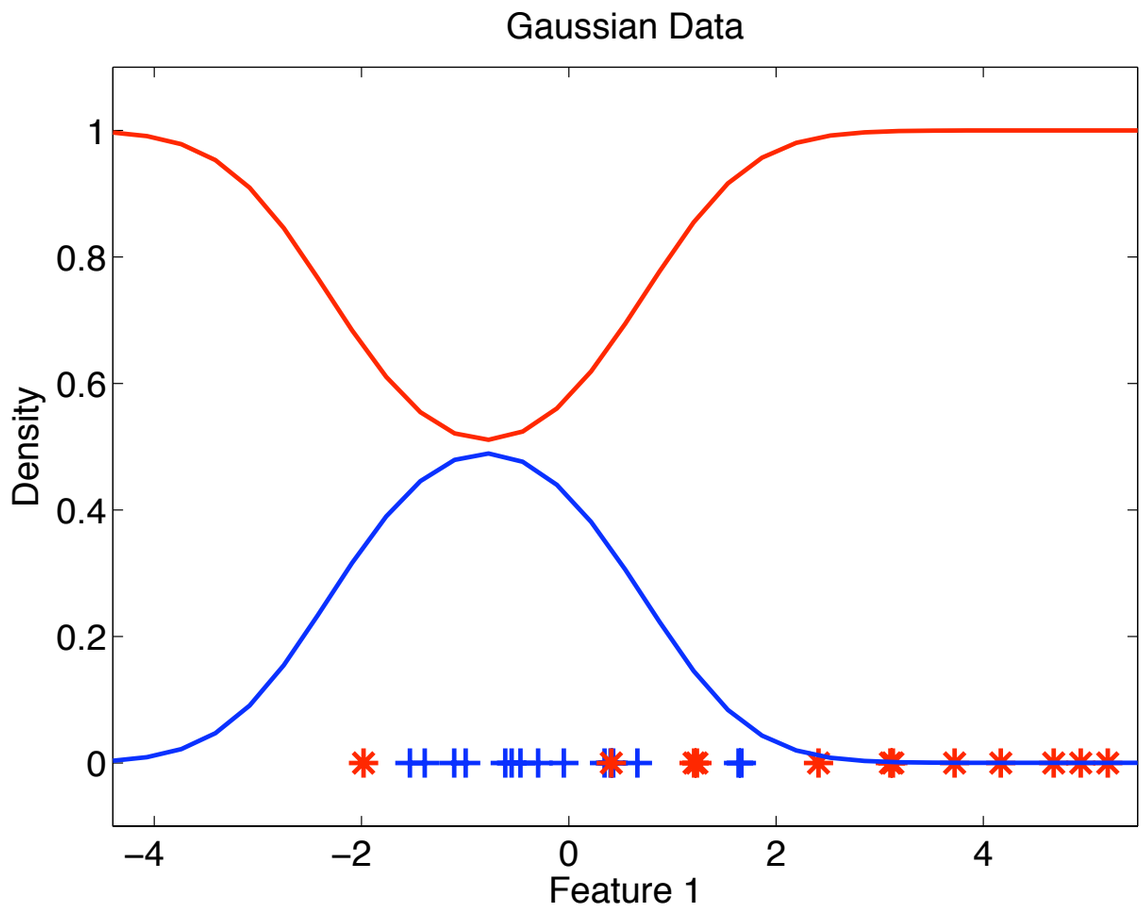


- Depending on the class-conditional probability densities, complicated decision boundaries can appear

TU Delft

Missing decision boundary

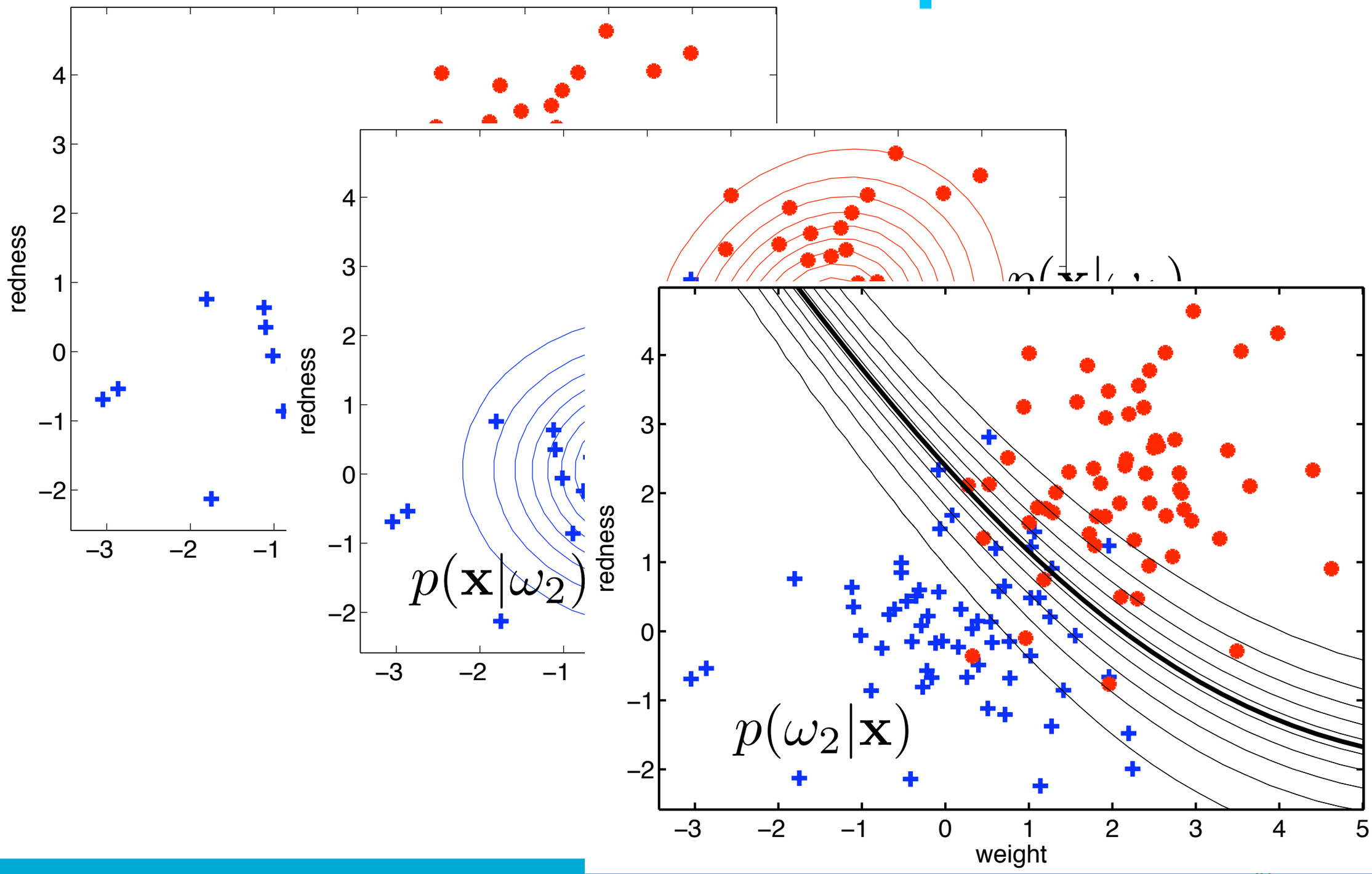




- A class can be too small (class prior is low) or too dispersed, that no objects are assigned to that class

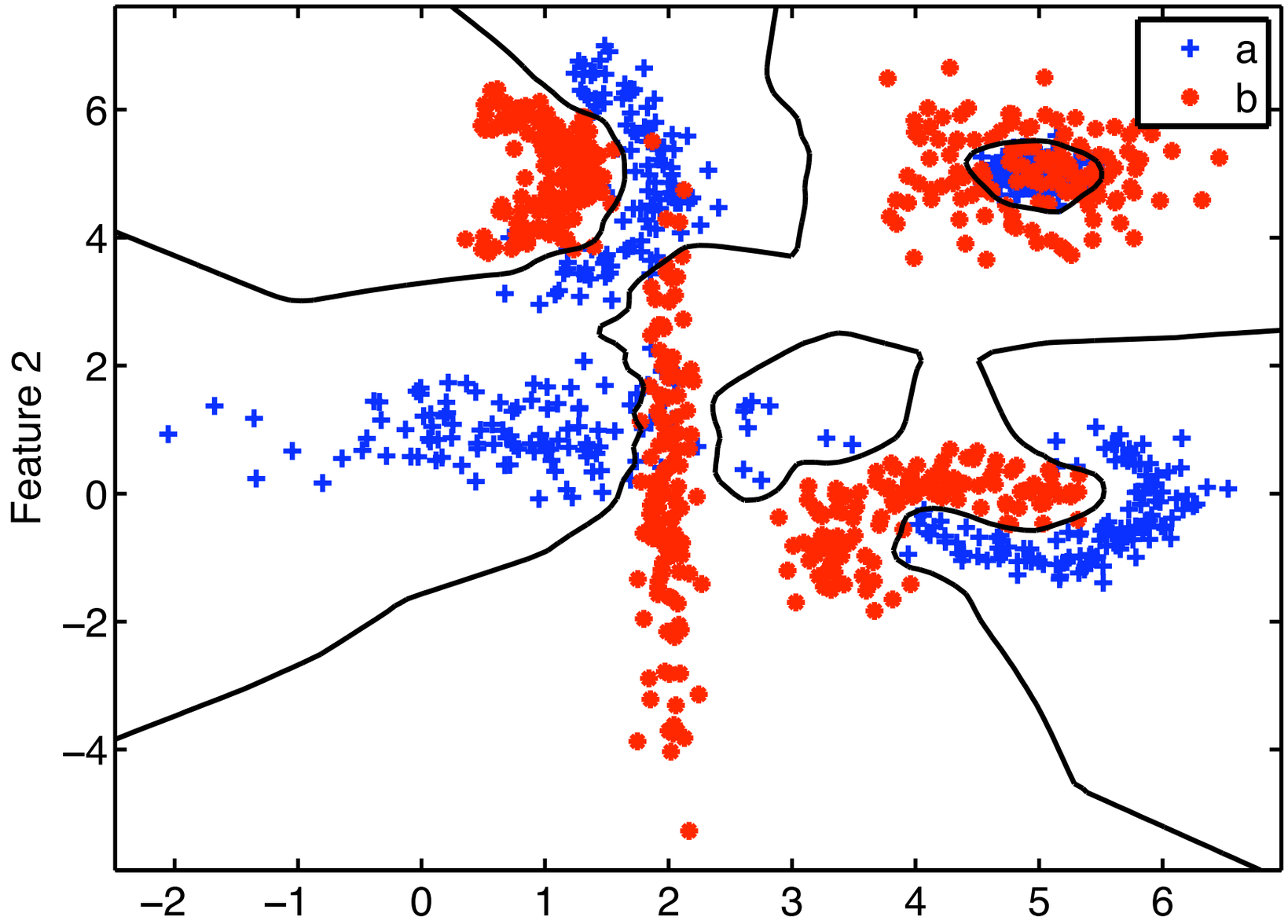
TU Delft

2-dimensional feature space



TU Delft

Multi-modal distributions



- Depending on the class distributions, the decision boundary can have arbitrary shapes

TU Delft

The class conditional probabilities

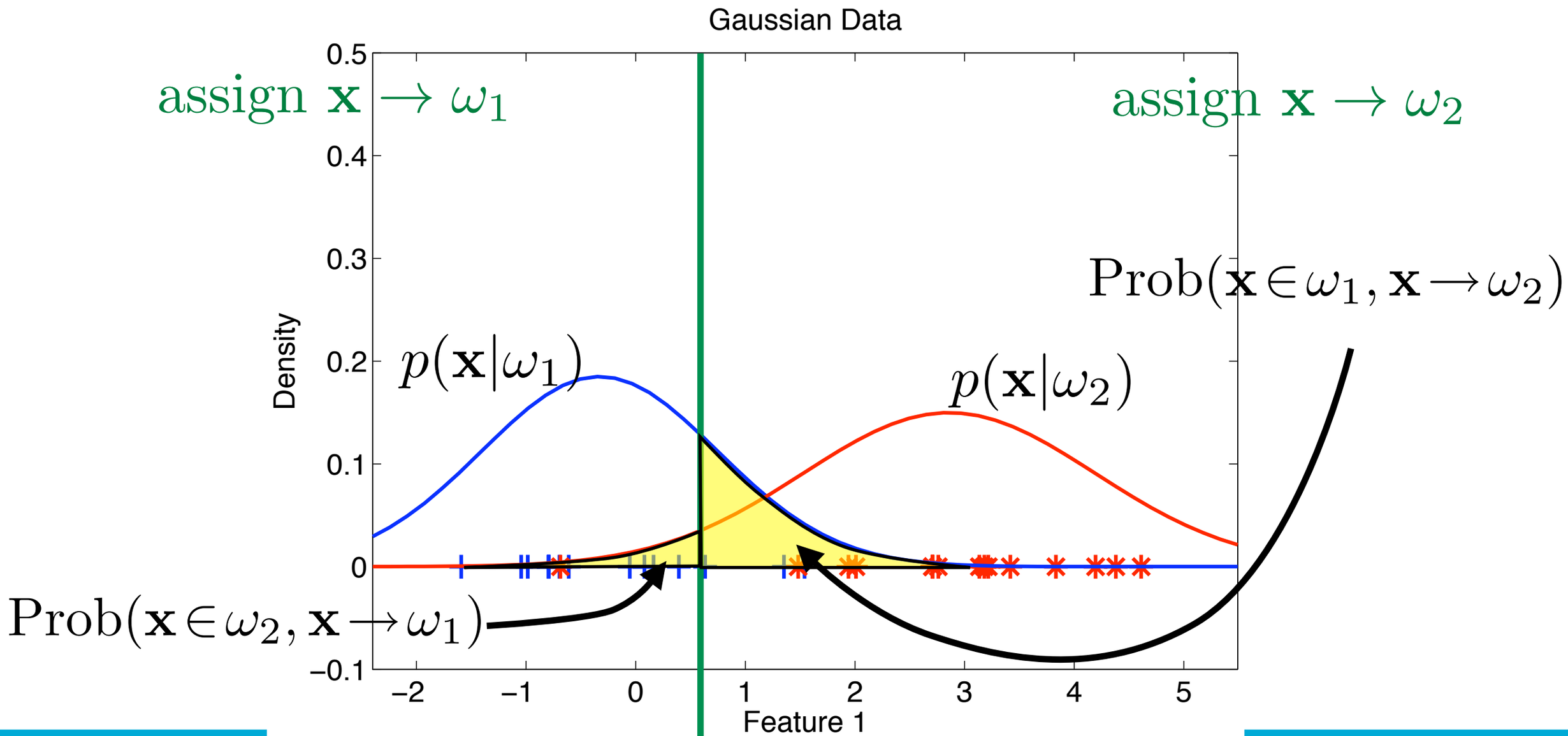
- But, how do we obtain the class conditional probabilities $p(\mathbf{x}|\omega_i)$?
- Typically, you need to assume a model
- Estimate the model parameters such that the example objects fit well:
  maximum likelihood estimators
- This will be the topic for the coming weeks

- Note: other approaches than Bayes' rule are possible. We discuss it later…

TU Delft

How good is it?

- The error of the green decision boundary:



TU Delft

Error of type I and II

- For a two-class classification problem, the following two errors are defined:

- Type I error: $$\varepsilon_1 = \int_{\Omega_2} p(\mathbf{x}|\omega_1)d\mathbf{x}$$

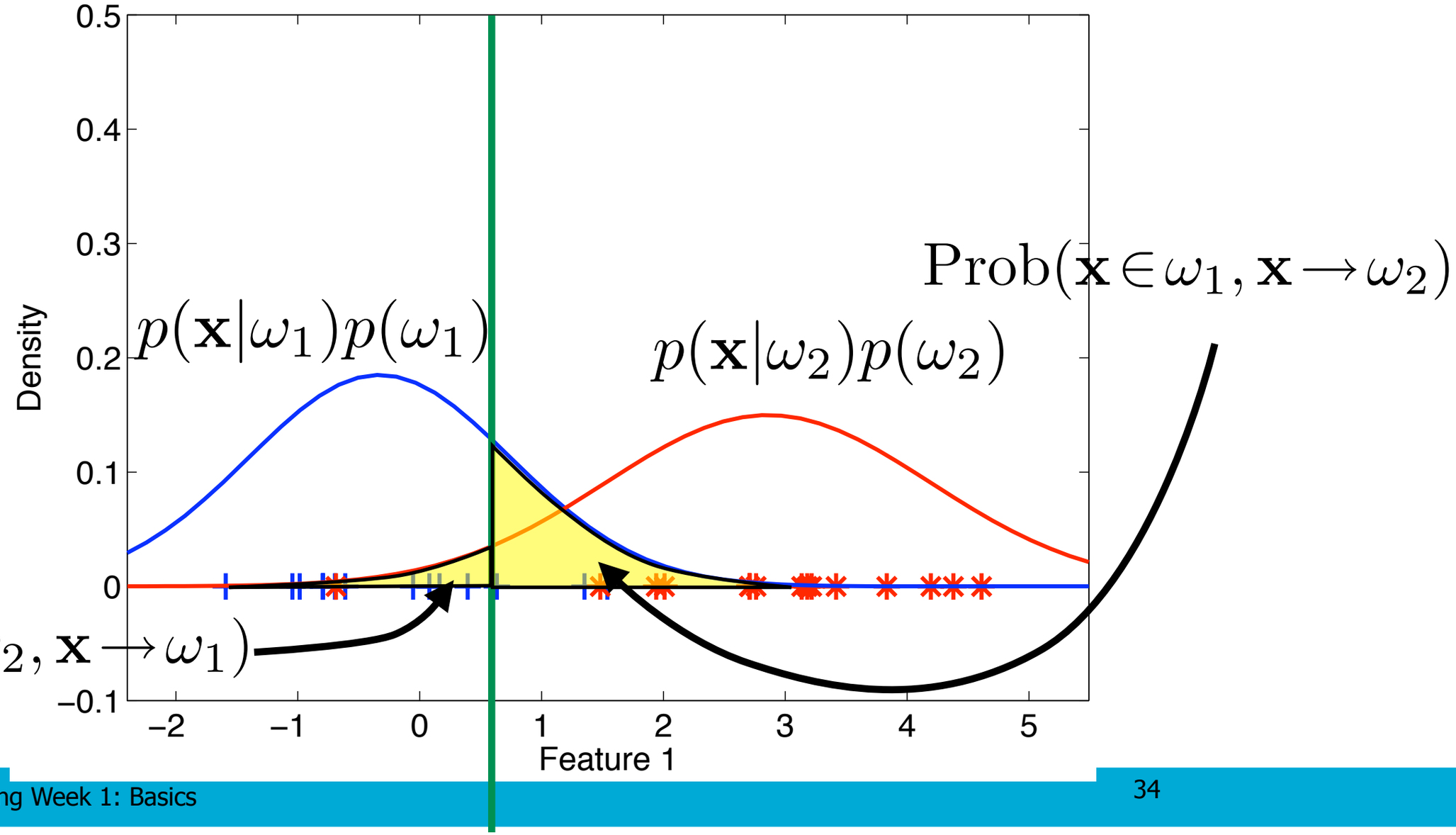
- Type II error: $$\varepsilon_2 = \int_{\Omega_1} p(\mathbf{x}|\omega_2)d\mathbf{x}$$

- When we call ω_1 the positive class, and ω_2 the negative class, then ε_1 is the false negative fraction, and ε_2 is the false positive fraction.

TU Delft

Classification error

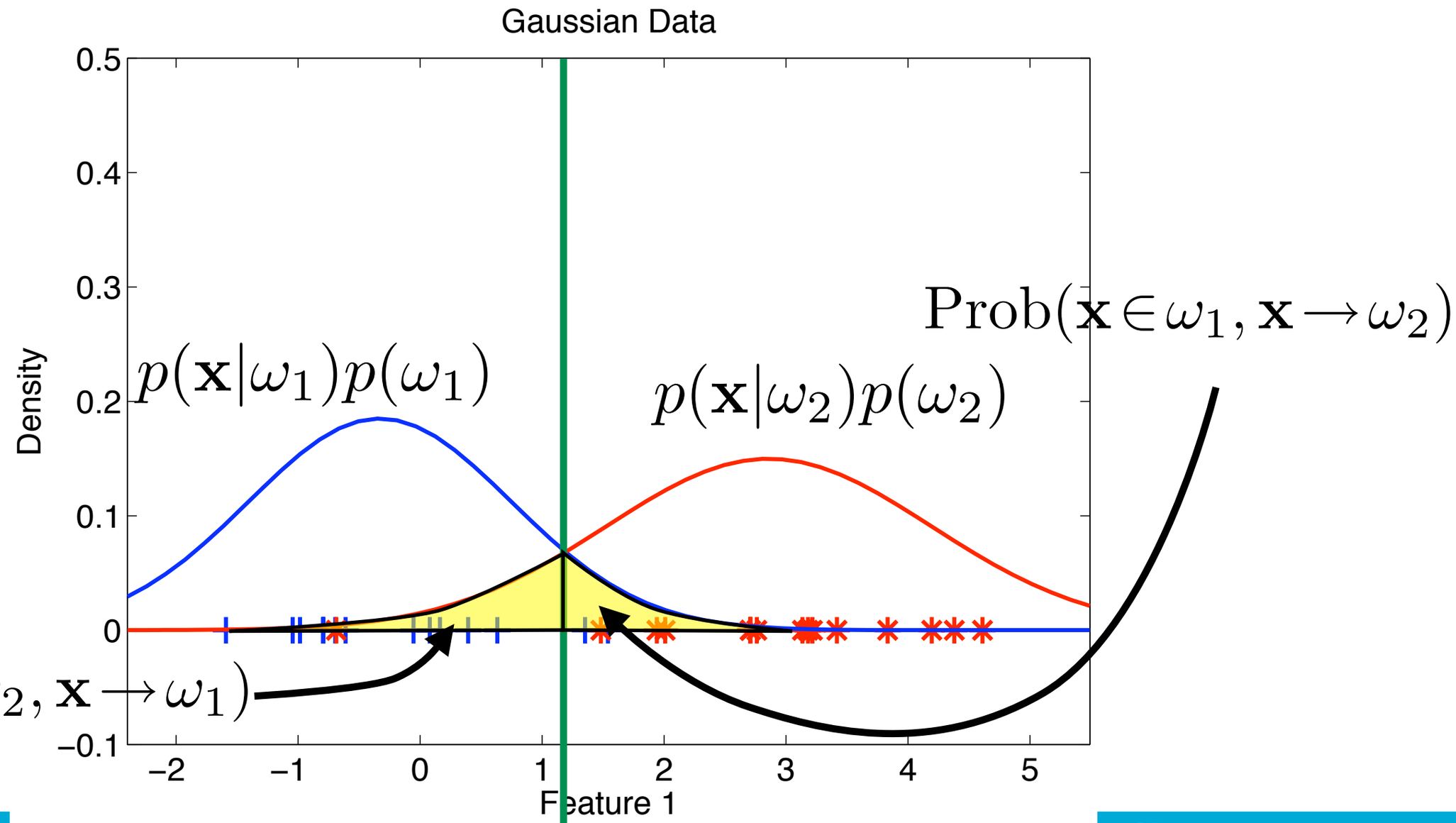
- The error: $p(\text{error}) = \sum_{i=1}^{C} p(\text{error}|\omega_i)p(\omega_i)$



TU Delft

Bayes error ε^*

Bayes error is the **minimum** error: typically >0 !!



TU Delft

Bayes' Error

- Bayes' error is the **minimum** attainable error ε^*
- In practice, we do not have the true distributions, and we can not obtain
- The Bayes' error does not depend on the classification rule that you apply, but on the distribution of the data
- In general you can not compute the Bayes' error:
  - you don't know the true class conditional probabilities
  - the (high) dimensional integrals are very complicated

TU Delft

Misclassification Costs

- Sometimes: misclassification of class A to class B is much more dangerous than misclassification of class B to class A

misclassification:
classify 'healthy' to 'ill'

misclassification:
classify 'ill' to 'healthy'

TU Delft

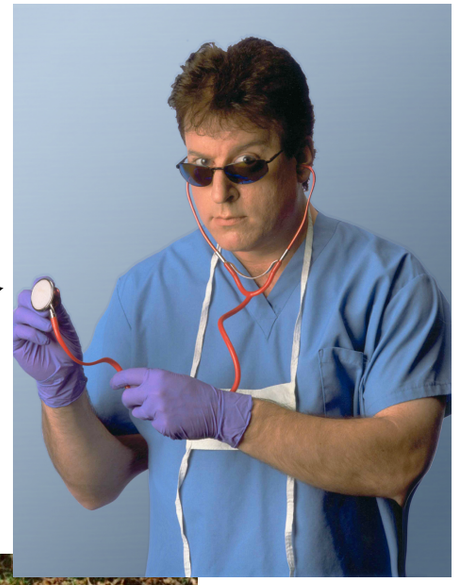
Misclassification cost

- Introduce a loss that measures the cost of assigning an object that came from class ω_j to class ω_i :

$$\lambda_{ji}$$

$$\lambda_{\text{healthy,ill}} = 10$$

$$\lambda_{\text{ill,healthy}} = 100$$



TU Delft

Misclassification cost of some dataset

- Assume I have a labeled dataset $\mathcal{D} = \{(\mathbf{x}_i, \omega_i)\}_{i=1}^{N}$
- Further assume that these objects are classified by a classifier, and the estimated class labels are $\hat{\omega}_i$

- Then the total empirical risk on this dataset is

$$R = \frac{1}{N} \sum_{i=1}^{N} \lambda_{\omega_i, \hat{\omega}_i}$$

TU Delft

Conditional risk, total risk

- The conditional risk of assigning object **x** to class ω_i :

$$l^i(\mathbf{x}) = \sum_{j=1}^{C} \lambda_{ji} p(\omega_j|\mathbf{x})$$

- The average risk over a region:

$$r^i = \int_{\Omega_i} l^i(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

- Overall risk:

$$= \int_{\Omega_i} \sum_{j=1}^{C} \lambda_{ji} p(\omega_j|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

$$r = \sum_{i=1}^{C} r^i = \sum_{i=1}^{C} \int_{\Omega_i} \sum_{j=1}^{C} \lambda_{ji} p(\omega_j|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

TU Delft

Minimum total risk

- We minimize the risk when we define the regions Ω_i are chosen such that each of the integrals are as small as possible:

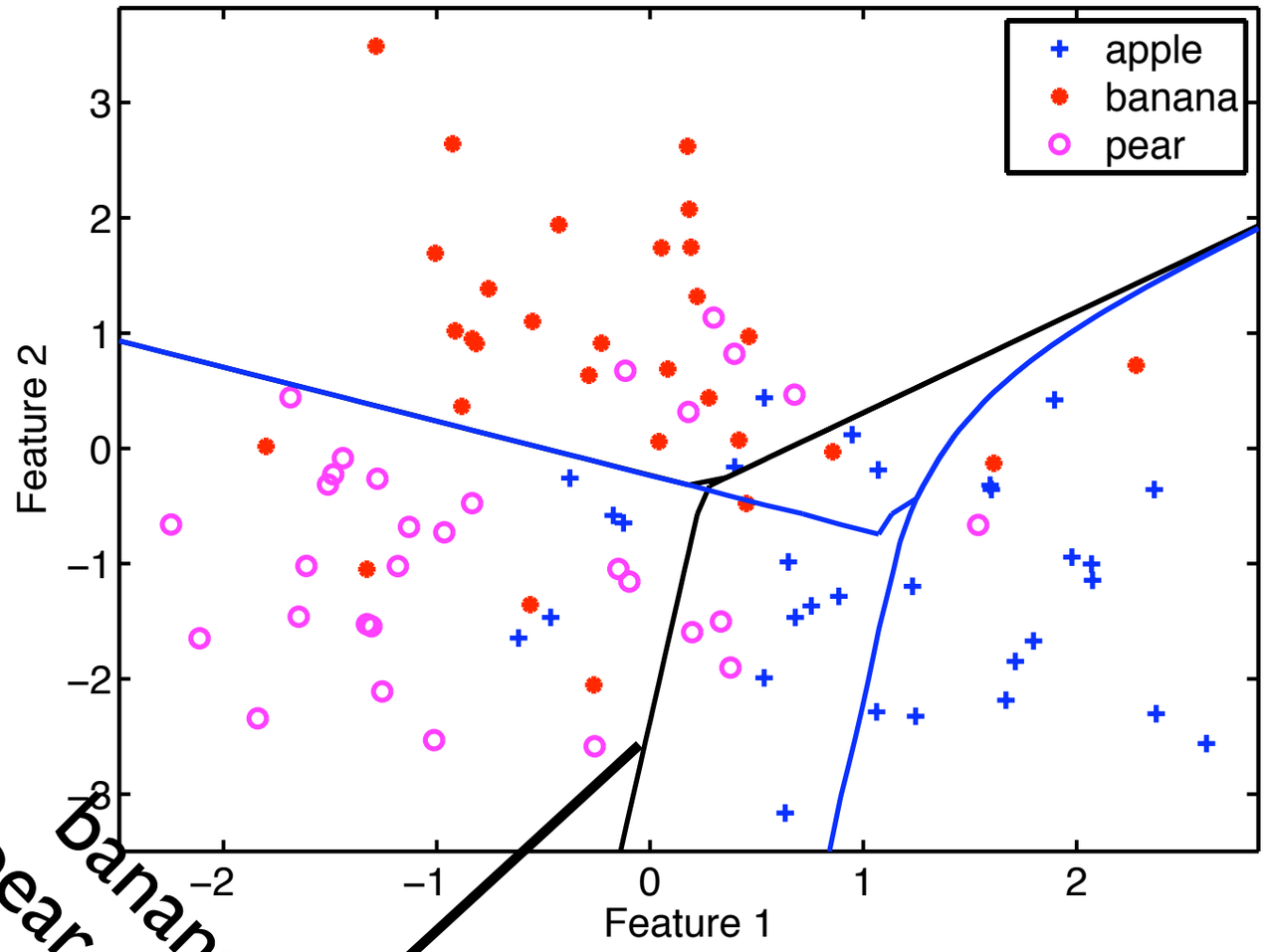
$$r = \sum_{i=1}^{C} r^i = \sum_{i=1}^{C} \int_{\Omega_i} \sum_{j=1}^{C} \lambda_{ji} p(\omega_j|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

- So make **x** part of Ω_i if:

$$\sum_{j=1}^{C} \lambda_{ji} p(\omega_j|\mathbf{x}) \leq \sum_{j=1}^{C} \lambda_{jk} p(\omega_j|\mathbf{x}) \quad k = 1, .., C$$

TU Delft

Example cost

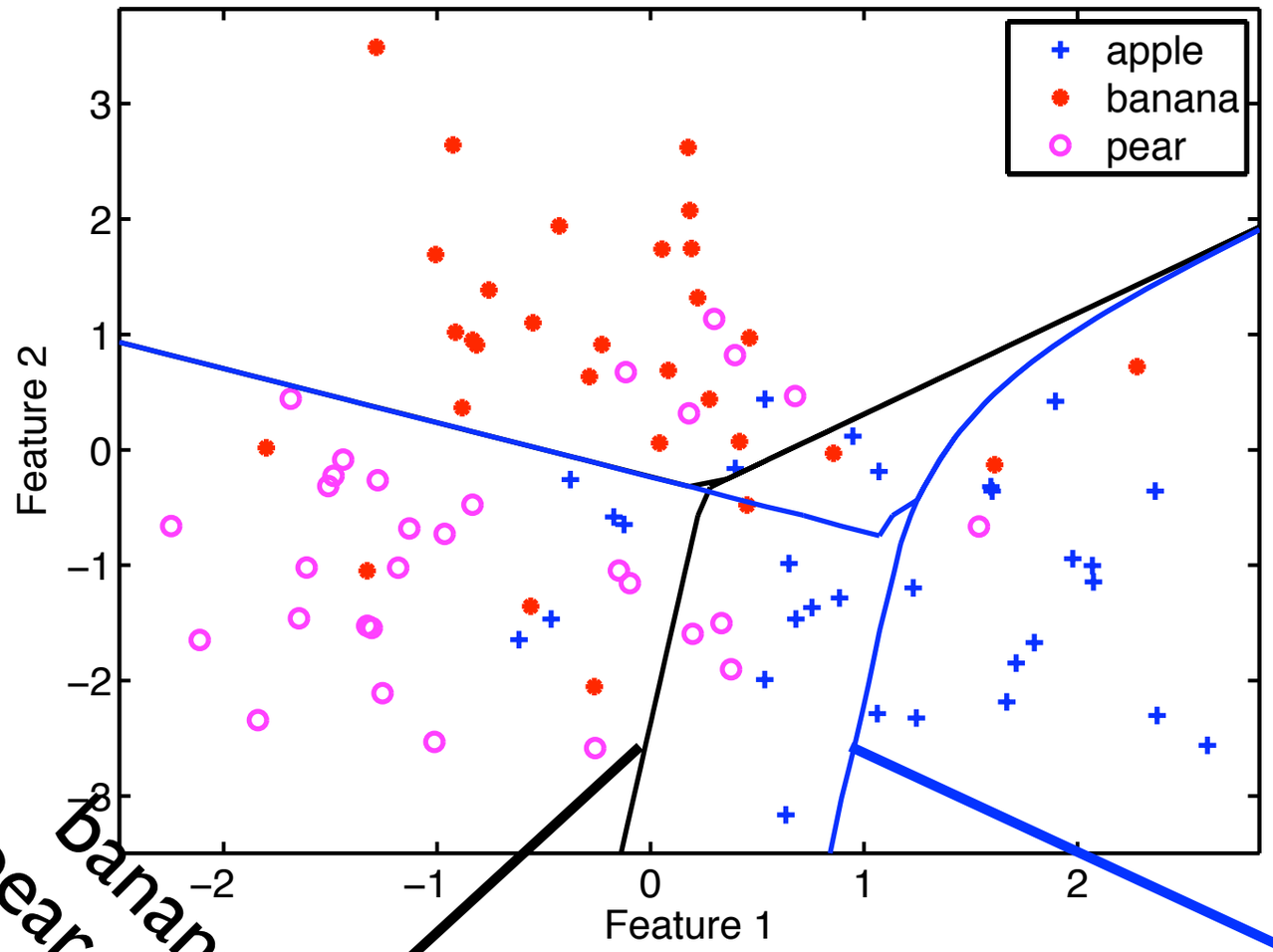


classified

True	apple	pear	banana
apple	0	1	1
pear	1	0	1
banana	1	1	0

TU Delft

Example cost



$$\lambda_{2,1} = 9$$

Predicted: apple, pear, banana

True:

	apple	pear	banana
apple	0	1	1
pear	1	0	1
banana	1	1	0

Predicted: apple, pear, banana

	apple	pear	banana
apple	0	1	1
pear	9	0	1
banana	1	1	0

TU Delft

What did we do?

- Objects, features, measurements,
- ... datasets and feature space
- Traditional pattern recognition: classification
- Class posterior probabilities and Bayes' rule
- Bayes' classifier and Bayes' error
- Misclassification costs

- Next week: how do we get these class-conditional probabilities?

TU Delft