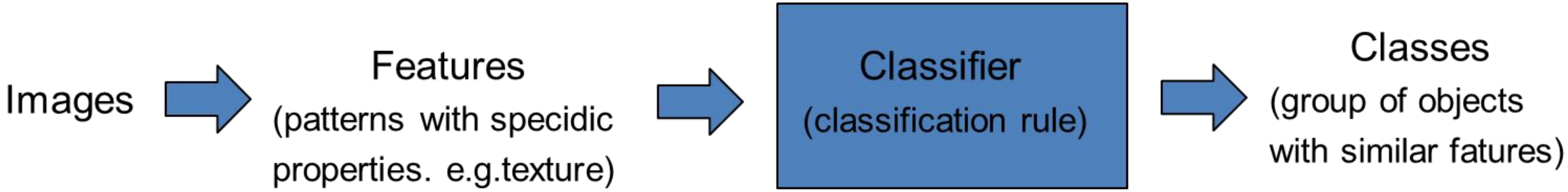
PATTERN RECOGNITION

MININUM DISTANCE CLASSIFIER

PATTERN RECOGNITION AND CLASSIFICATION

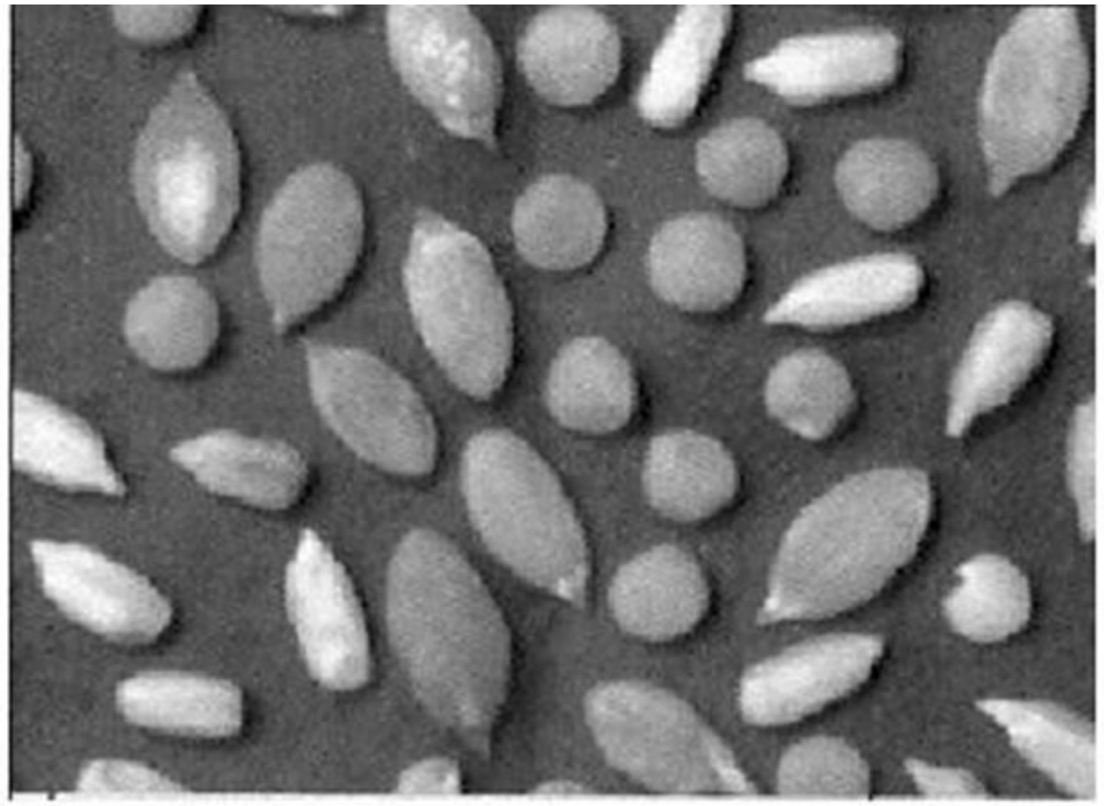
- Automatically assigning measured signals / objects to categories.
- Automatically classifying them with respect to some features.



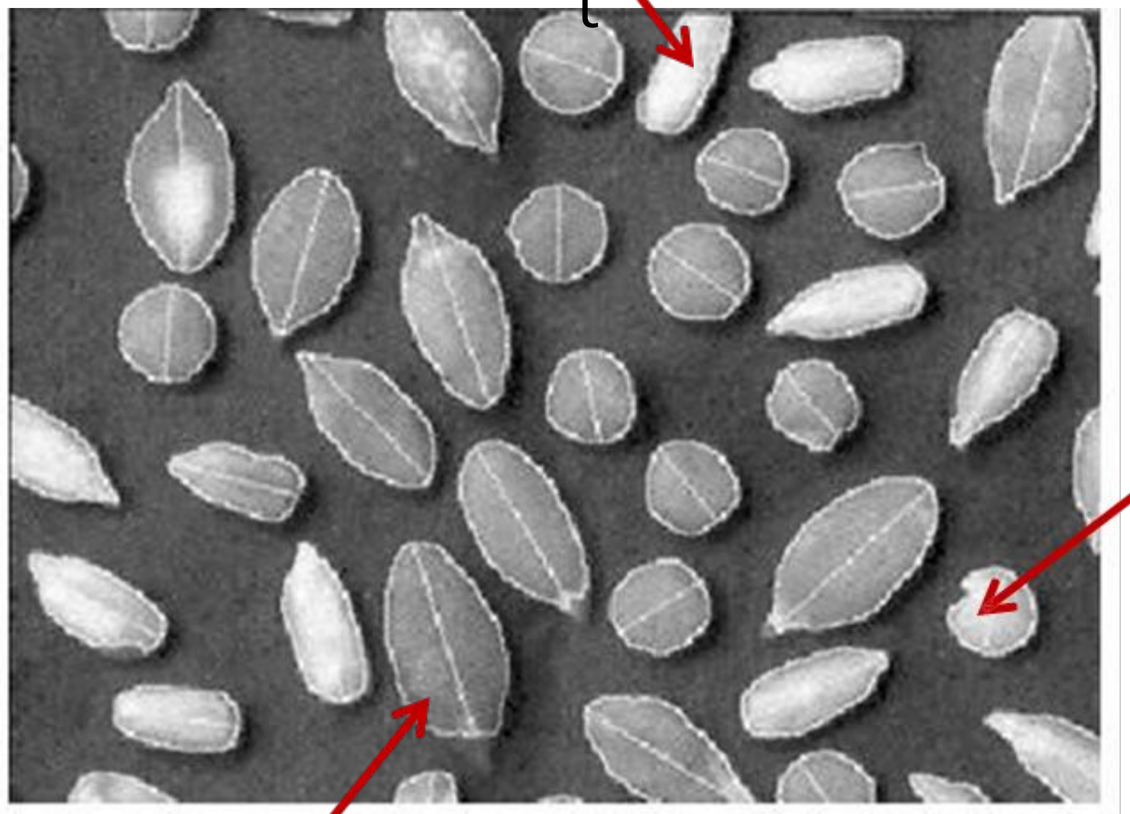
PATTERN RECOGNITION - EXAMPLE

Classifying nuts

Features (circularity, line-fit-error) → Classifier (classification rules) → Pine-nuts (Pinienkerne), Lentils (Linsen), Pumpkin seeds (Kürbiskerne)



pine-nu
t



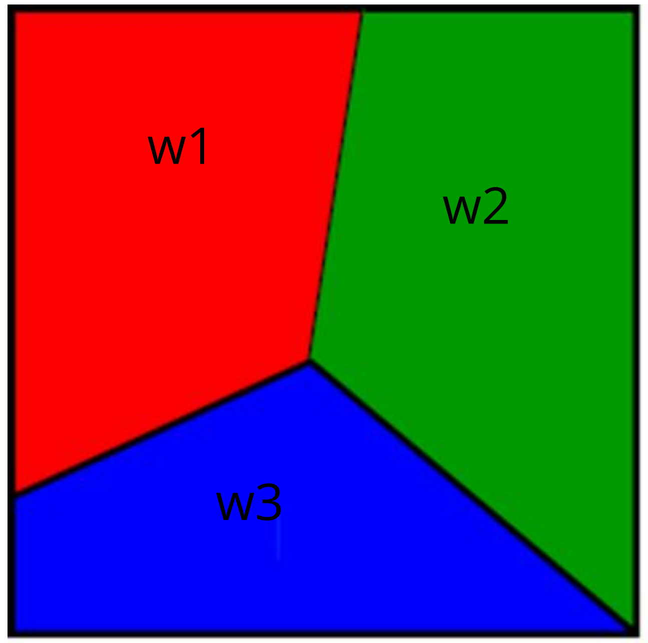
lenti
l

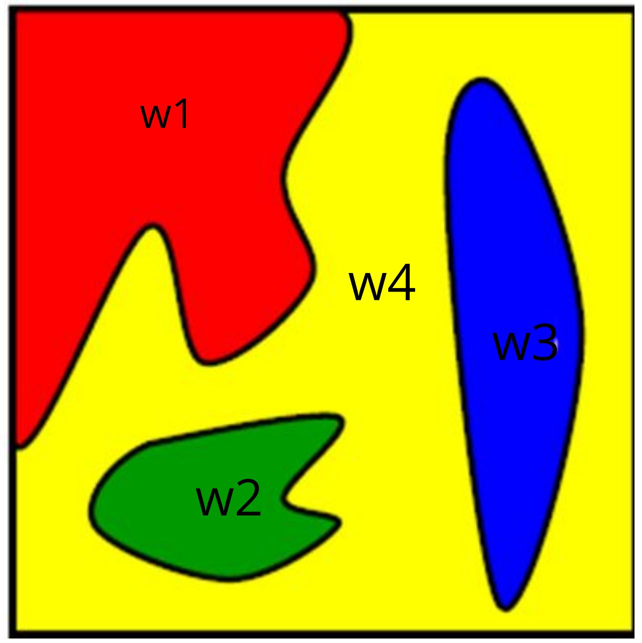
pumpkin seed

CLASSIFIERS

The task of a classifier is to **partition the feature space** into **class-labeled decision regions** 1, 2, 3, ..., *W*, called **classes** w1, w2, ...

- Borders between decision regions are called **decision boundaries**.
- The **classification** of a feature vector x consists of determining which decision region (class) it belongs to, and **assign** x **to this class**.





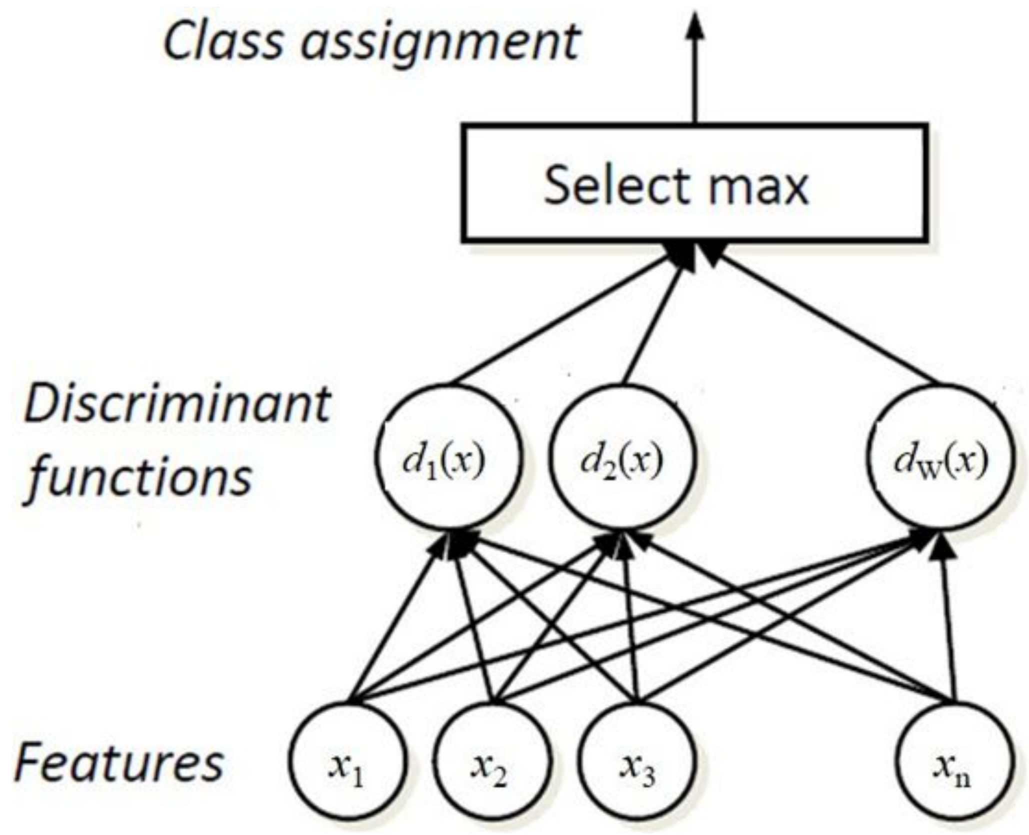
CLASSIFIERS

A **classifier** can be represented as a set of **discriminant (decision) functions** *d*i(x) **for the classes** w*i*.

- The **classifier assigns** a **feature vector** x to **class** w*i* if

$$d_i(\mathbf{x}) > d_j(\mathbf{x}) \text{ for } j = 1, 2, \ldots, W; \; j \neq i$$

Based on the **discriminant (decision) functions** *d*i(x) the classifier determines **decision boundaries** *d*ij(x) to **separate the classes**



PATTERN RECOGNITION - EXAMPLE

Classify different kinds of **iris flowers** (Schwertlilie) based on the geometric properties of their blossoms (Blüten).

Iris setosa

Iris versicolor

Iris virginica

Iris flower dataset

- 50 samples from each for the three species of iris flowers.
- 4 features were measured from each sample:
- the length and the width of the sepals (Kelchblatt) and petals (Blütenblatt)

PATTERN RECOGNITION -EXAMPLE

Iris sestosa				Iris versicolor				Iris virginica			
Sepal Leng	Sepal Width	Petal Leng	Petal Width	Sepal Leng	Sepal Width	Petal Leng	Petal Width	Sepal Leng	Sepal Width	Petal Leng	Petal Width
5.1	3.5	1.4	0.2	7.0	3.2	4.7	1.4	6.3	3.3	6.0	2.5
4.9	3.0	1.4	0.2	6.4	3.2	4.5	1.5	5.8	2.7	5.1	1.9
4.7	3.2	1.3	0.2	6.9	3.1	4.9	1.5	7.1	3.0	5.9	2.1
4.6	3.1	1.5	0.2	5.5	2.3	4.0	1.3	6.3	2.9	5.6	1.8
5.0	3.6	1.4	0.2	6.5	2.8	4.6	1.5	6.5	3.0	5.8	2.2
5.4	3.9	1.7	0.4	5.7	2.8	4.5	1.3	7.6	3.0	6.6	2.1
4.6	3.4	1.4	0.3	6.3	3.3	4.7	1.6	4.9	2.5	4.5	1.7
5.0	3.4	1.5	0.2	4.9	2.4	3.3	1.0	7.3	2.9	6.3	1.8
4.4	2.9	1.4	0.2	6.6	2.9	4.6	1.3	6.7	2.5	5.8	1.8
4.9	3.1	1.5	0.1	5.2	2.7	3.9	1.4	7.2	3.6	6.1	2.5
5.4	3.7	1.5	0.2	5.0	2.0	3.5	1.0	6.5	3.2	5.1	2.0
4.8	3.4	1.6	0.2	5.9	3.0	4.2	1.5	6.4	2.7	5.3	1.9
4.8	3.0	1.4	0.1	6.0	2.2	4.0	1.0	6.8	3.0	5.5	2.1
4.3	3.0	1.1	0.1	6.1	2.9	4.7	1.4	5.7	2.5	5.0	2.0
5.8	4.0	1.2	0.2	5.6	2.9	3.6	1.3	5.8	2.8	5.1	2.4
5.7	4.4	1.5	0.4	6.7	3.1	4.4	1.4	6.4	3.2	5.3	2.3
5.4	3.9	1.3	0.4	5.6	3.0	4.5	1.5	6.5	3.0	5.5	1.8
5.1	3.5	1.4	0.3	5.8	2.7	4.1	1.0	7.7	3.8	6.7	2.2
5.7	3.8	1.7	0.3	6.2	2.2	4.5	1.5	7.7	2.6	6.9	2.3
5.1	3.8	1.5	0.3	5.6	2.5	3.9	1.1	6.0	2.2	5.0	1.5
5.4	3.4	1.7	0.2	5.9	3.2	4.8	1.8	6.9	3.2	5.7	2.3
5.1	3.7	1.5	0.4	6.1	2.8	4.0	1.3	5.6	2.8	4.9	2.0
4.6	3.6	1.0	0.2	6.3	2.5	4.9	1.5	7.7	2.8	6.7	2.0
5.1	3.3	1.7	0.5	6.1	2.8	4.7	1.2	6.3	2.7	4.9	1.8
4.8	3.4	1.9	0.2	6.4	2.9	4.3	1.3	6.7	3.3	5.7	2.1
5.0	3.0	1.6	0.2	6.6	3.0	4.4	1.4	7.2	3.2	6.0	1.8
5.0	3.4	1.6	0.4	6.8	2.8	4.8	1.4	6.2	2.8	4.8	1.8
5.2	3.5	1.5	0.2	6.7	3.0	5.0	1.7	6.1	3.0	4.9	1.8
5.2	3.4	1.4	0.2	6.0	2.9	4.5	1.5	6.4	2.8	5.6	2.1
4.7	3.2	1.6	0.2	5.7	2.6	3.5	1.0	7.2	3.0	5.8	1.6
4.8	3.1	1.6	0.2	5.5	2.4	3.8	1.1	7.4	2.8	6.1	1.9
5.4	3.4	1.5	0.4	5.5	2.4	3.7	1.0	7.9	3.8	6.4	2.0
5.2	4.1	1.5	0.1	5.8	2.7	3.9	1.2	6.4	2.8	5.6	2.2
5.5	4.2	1.4	0.2	6.0	2.7	5.1	1.6	6.3	2.8	5.1	1.5
4.9	3.1	1.5	0.2	5.4	3.0	4.5	1.5	6.1	2.0	5.6	1.4
5.0	3.2	1.2	0.2	6.0	3.4	4.5	1.6	7.7	3.0	6.1	2.3
5.5	3.5	1.3	0.2	6.7	3.1	4.7	1.5	6.3	3.4	5.6	2.4
4.9	3.6	1.4	0.1	6.3	2.3	4.4	1.3	6.4	3.1	5.5	1.8
4.4	3.0	1.3	0.2	5.6	3.0	4.1	1.3	6.0	3.0	4.8	1.8
5.1	3.4	1.5	0.2	5.5	2.5	4.0	1.3	6.9	3.1	5.4	2.1
5.0	3.5	1.3	0.3	5.5	2.6	4.4	1.2	6.7	3.1	5.6	2.4
4.5	2.3	1.3	0.3	6.1	3.0	4.6	1.4	6.9	3.1	5.1	2.3
4.4	3.2	1.3	0.2	5.8	2.6	4.0	1.2	5.8	2.7	5.1	1.9
5.0	3.5	1.6	0.6	5.0	2.3	3.3	1.0	6.8	3.2	5.9	2.3
5.1	3.8	1.9	0.4	5.6	2.7	4.2	1.3	6.7	3.3	5.7	2.5
4.8	3.0	1.4	0.3	5.7	3.0	4.2	1.2	6.7	3.0	5.2	2.3
5.1	3.8	1.6	0.2	5.7	2.9	4.2	1.3	6.3	2.5	5.0	1.9
4.6	3.2	1.4	0.2	6.2	2.9	4.3	1.3	6.5	3.0	5.2	2.0
5.3	3.7	1.5	0.2	5.1	2.5	3.0	1.1	6.2	3.4	5.4	2.3
5.0	3.3	1.4	0.2	5.7	2.8	4.1	1.3	5.9	3.0	5.1	1.8

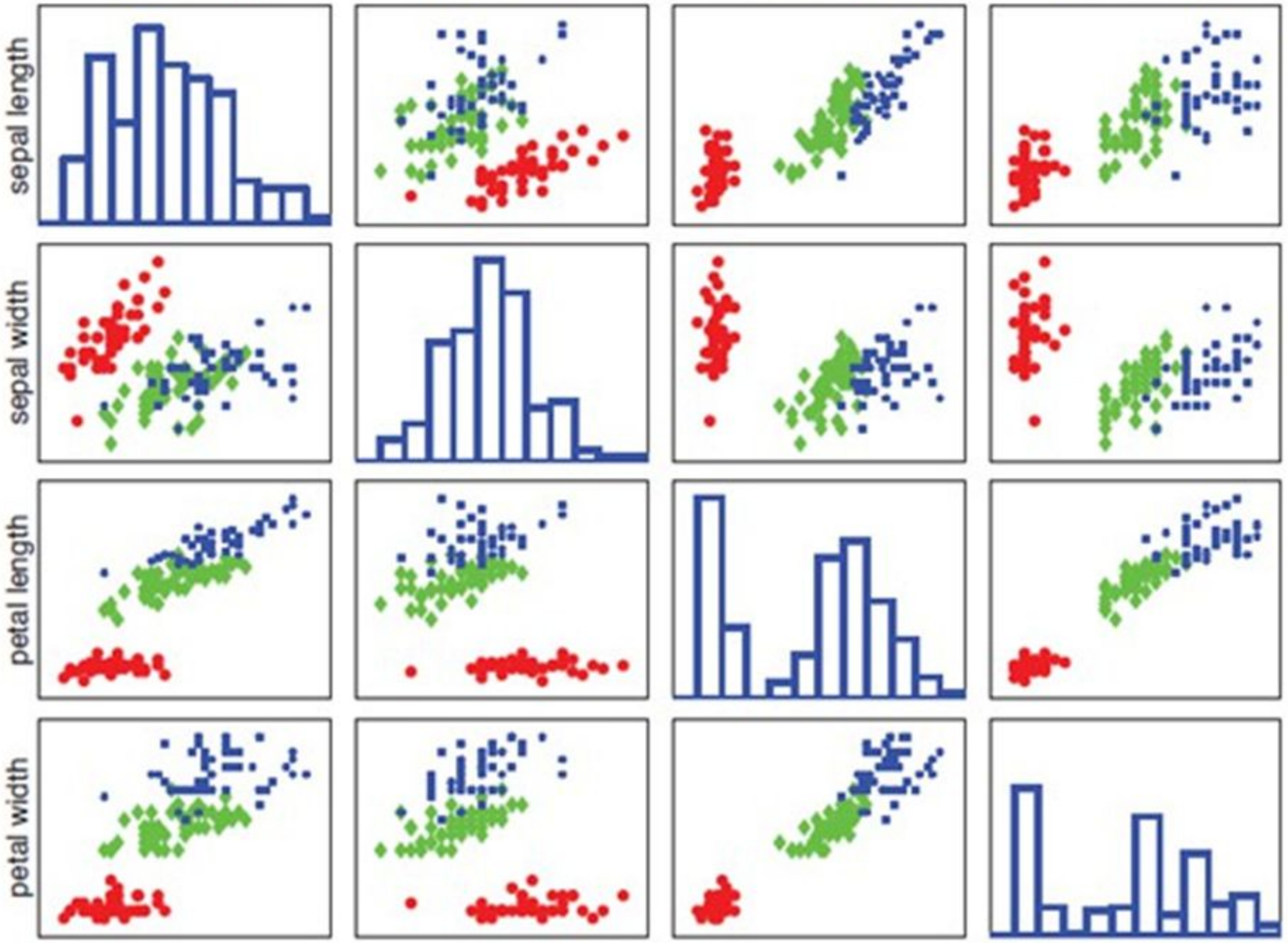
sepal length (Kelchblattlänge)
sepal width (Kelchblattbreite)
petal length (Blütenblattlänge)
petal width (Blütenblattbreite)

PATTERN RECOGNITION - EXAMPLE

Iris setosa

Iris versicolor

Iris virginica

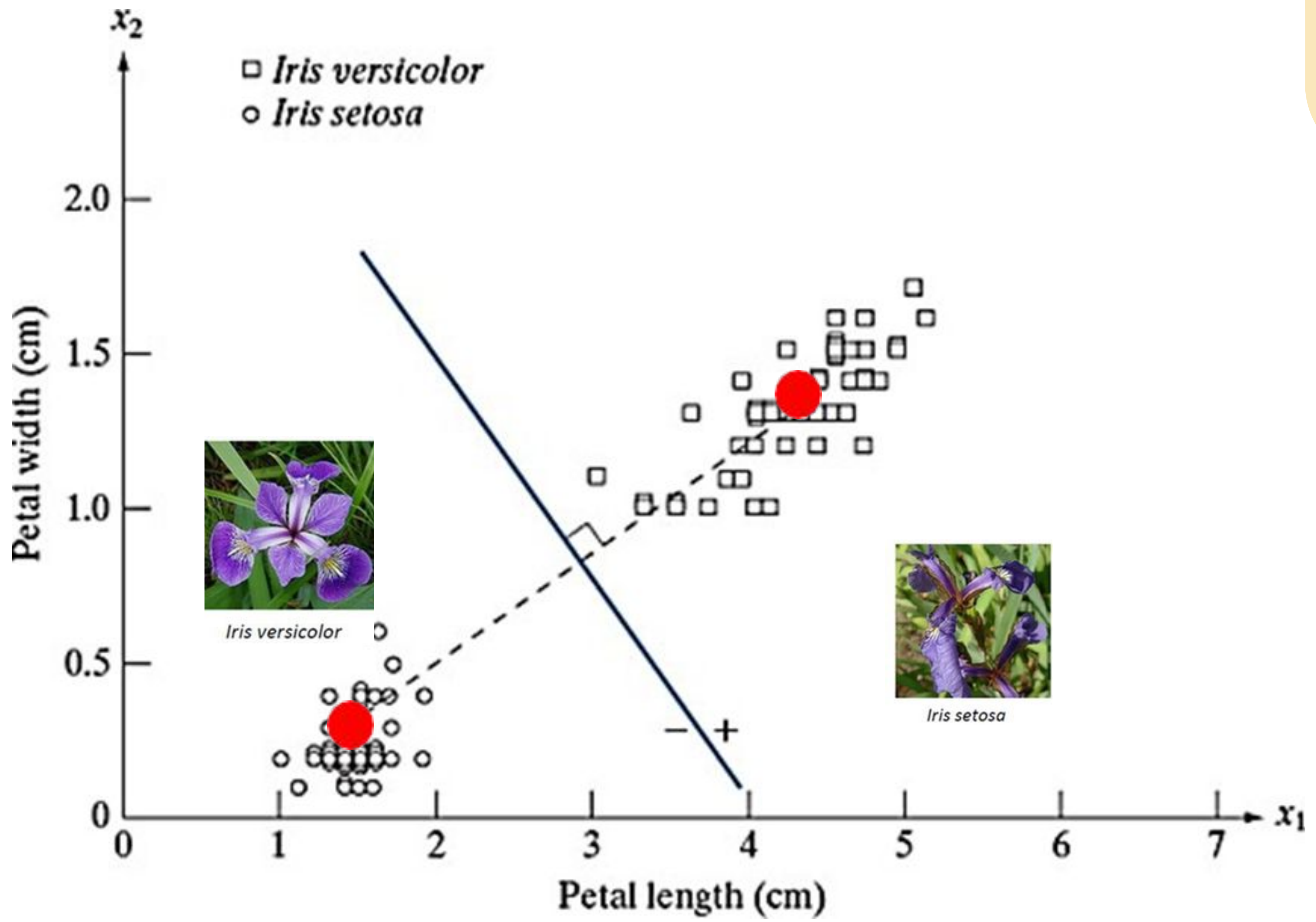


Visualization of the Iris data as a pairwise scatter plot.
The diagonal plots the histograms of the 4 features.
The off diagonals contain scatterplots of all possible pairs of features.

SCATTERPLOTS

A 2D scatter plot is a plot of feature values for two different features.

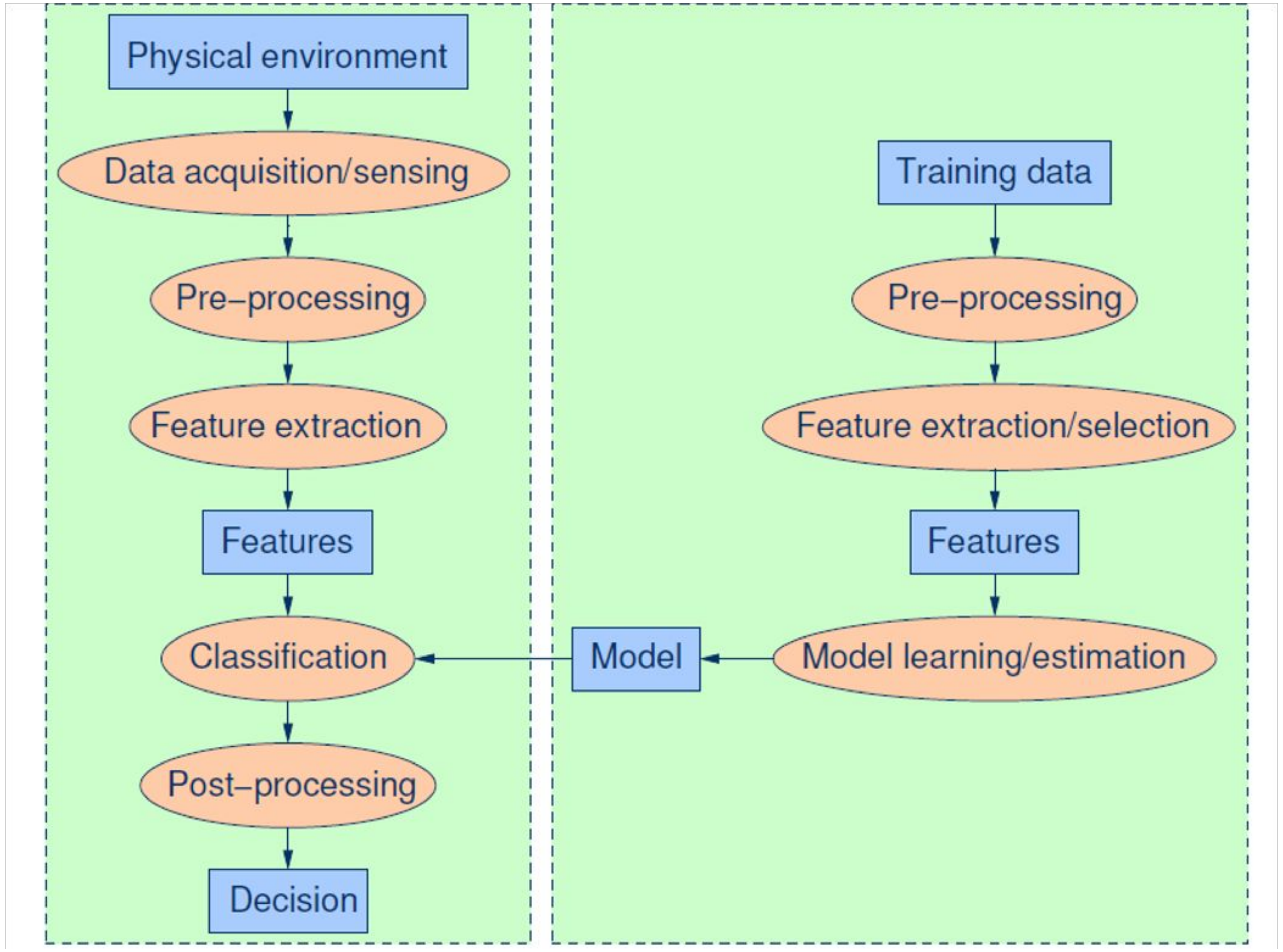
Each object's feature values are plotted in the position given by the feautes values,
and with a class label telling its object class.



Features with good class seperation show clusters for each class, but different clusters shoould ideally be seperated

Classification is done based on more than two features, but this is difficult to visualize

Pattern Recognition Systems



Object/process diagram of a pattern recognition system

Pattern Recognition Systems

- **Data acquisition and sensing** :
  - Measurements of physical variables.
  - Important issues: bandwidth, resolution, distortion, sensitivity, SNR, latency, etc.
- **Pre-processing** :
  - Removal of noice in data.
  - Isolation of patterns of interest from the background.
- **Feature extraction** :
  - Finding a new representation in terms of features.

Supervised and Unsupervised Learning

- **Supervised learning (classification)**

  trained by examples (by humans)

  A teacher provides a category label or cost for each pattern in the training set (i.e., ground truth based on expert's knowledge)

- **Unsupervised learning (clustering)**

  only by feature data

  The system forms clusters or "natural groupings" of the input patterns using the mathematical properties (statistics) of the data set.

- **Semi-supervised learning**

  Use both labeled and un-labeled patterns to reduce the labeling

Classification Methods - Overview

- **Goal of pattern recognition:** Assign patterns to their classes with as little human interaction as possible.
- **Matching (prototype matching):** An unknown pattern is assigned to the class to which it is closest with respect to a metric.
- **Minimum-Distance Classifier:** The Euclidean distance is used as metric between the unknown pattern and prototype vectors representing the classes
- **Statistical classifiers - Bayes Classifiers**
- **Neural networks**

Decision Functions

- **Decision-theoretic methods** are based on **decision (discriminant)** functions. Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ represent a **feature vector** describing a pattern.

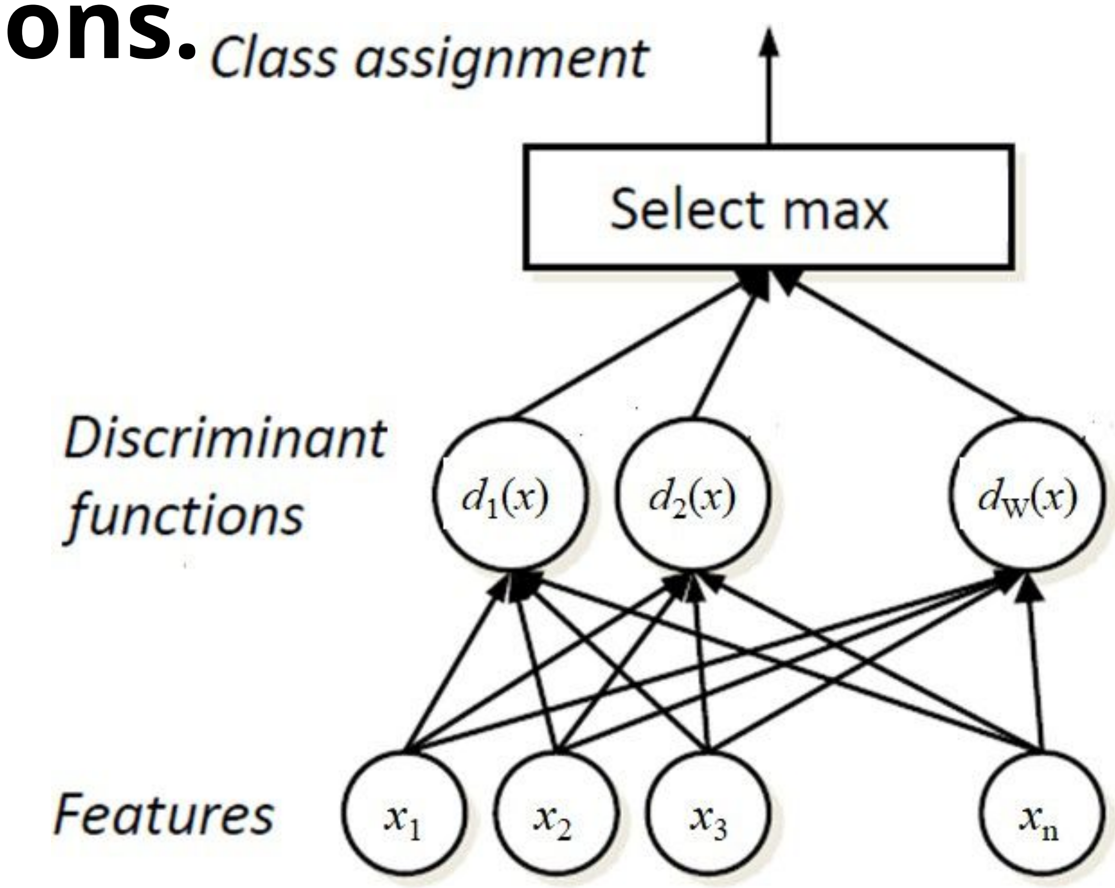
For W pattern classes ω1, ω2, ..., ωW the basic problem is to find **W decision (discriminant) functions.**

$$d_1(\mathbf{x}), d_2(\mathbf{x}), \ldots, d_W(\mathbf{x})$$

with the property that if x belongs to ωi

$$d_i(\mathbf{x}) > d_j(\mathbf{x})$$

$$\text{for} \quad j = 1, 2, \ldots, W; \quad j \neq i$$



Decision Boundaries

- The classes can be separated using **decision boundaries** dij**(**x**)** **based on the decision functions** di(x) and dj(x).
- The **decision boundary** dij**(**x**)** separating class ωi from class ωj is given by the values of x for which the decision functions:

$$d_i(\mathbf{x}) = d_j(\mathbf{x})$$

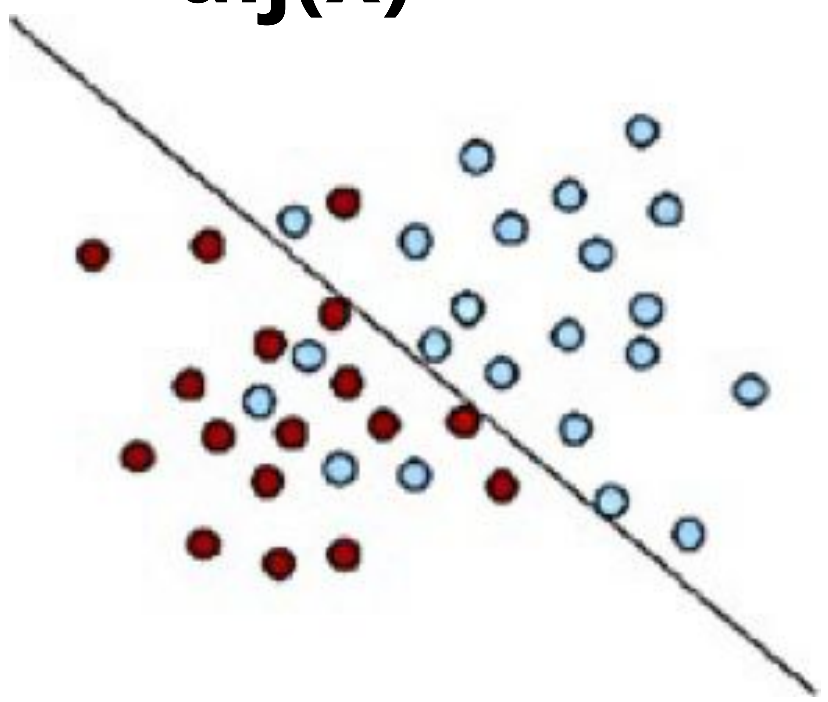
or

$$d_{ij}(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0$$

If x belongs to class ωi we have

$$d_{ij}(\mathbf{x}) > 0 \text{ for } j = 1, 2, \ldots, W; \; j \neq i$$

dij(x)



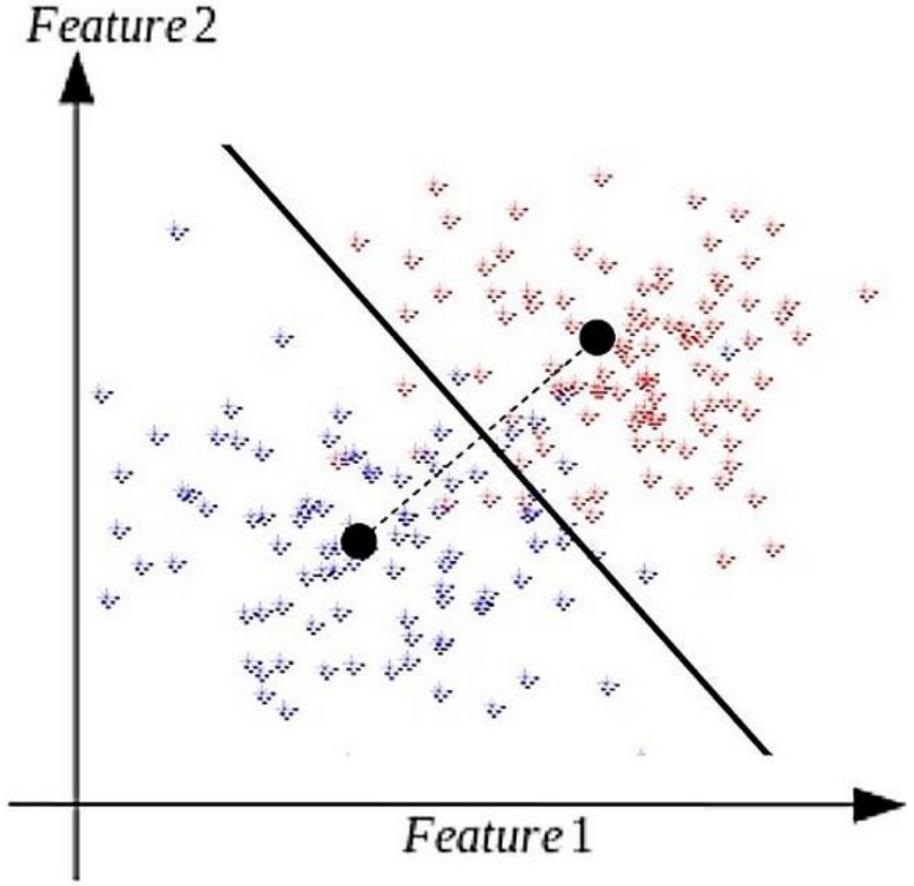
Decision Boundaries

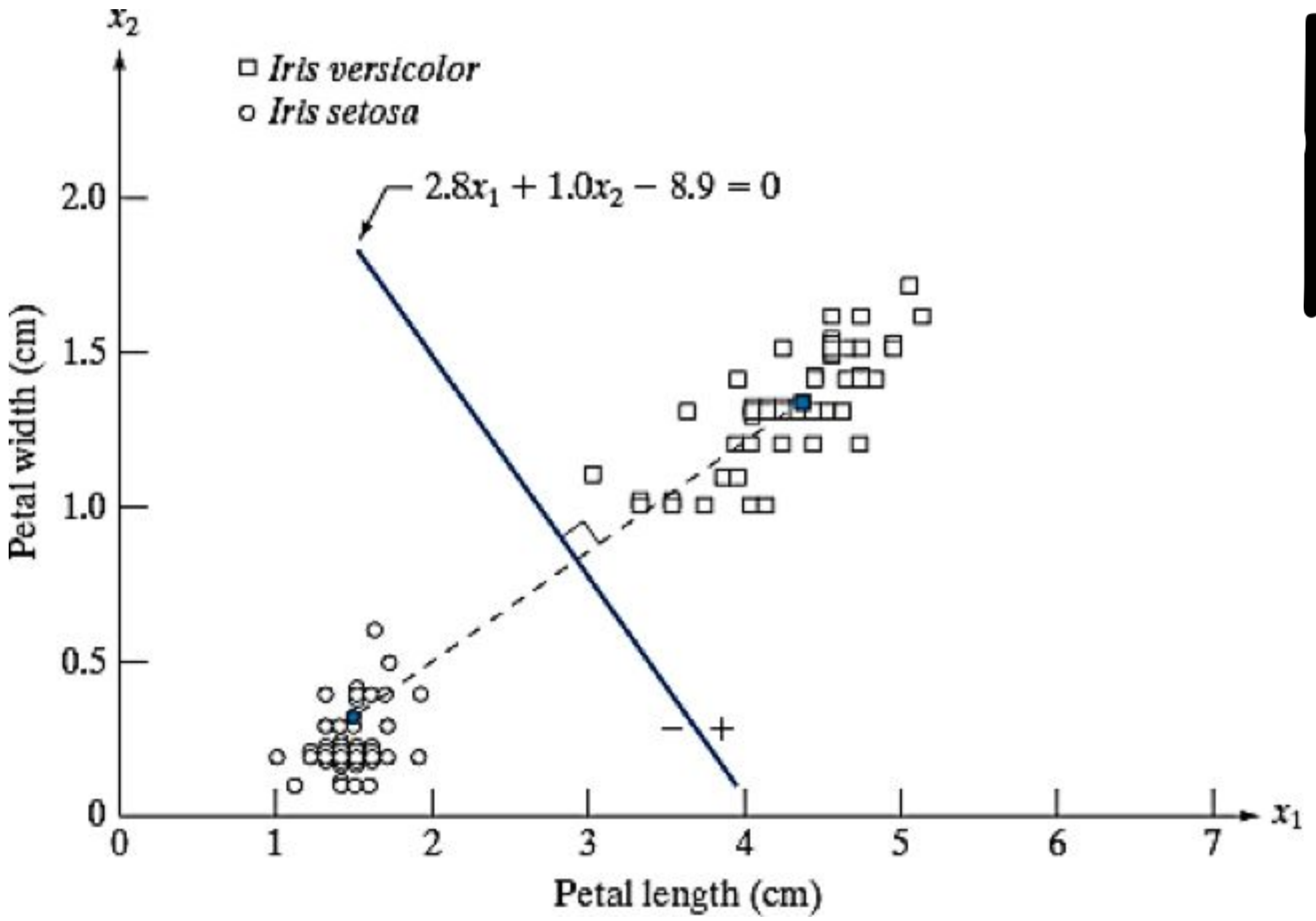
- The decision boundary separating class ωi from class ωj is given

$$d_{ij}(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x}) = 0$$

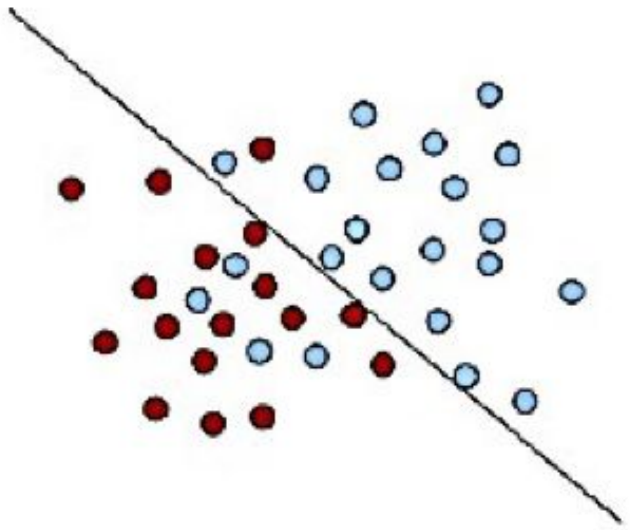
For two classes ωi and ωj we have

$$d_{ij}(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x}) \begin{cases} > 0 & \mathbf{x} \in \omega_i \\ = 0 & ? \\ < 0 & \mathbf{x} \in \omega_j \end{cases}$$

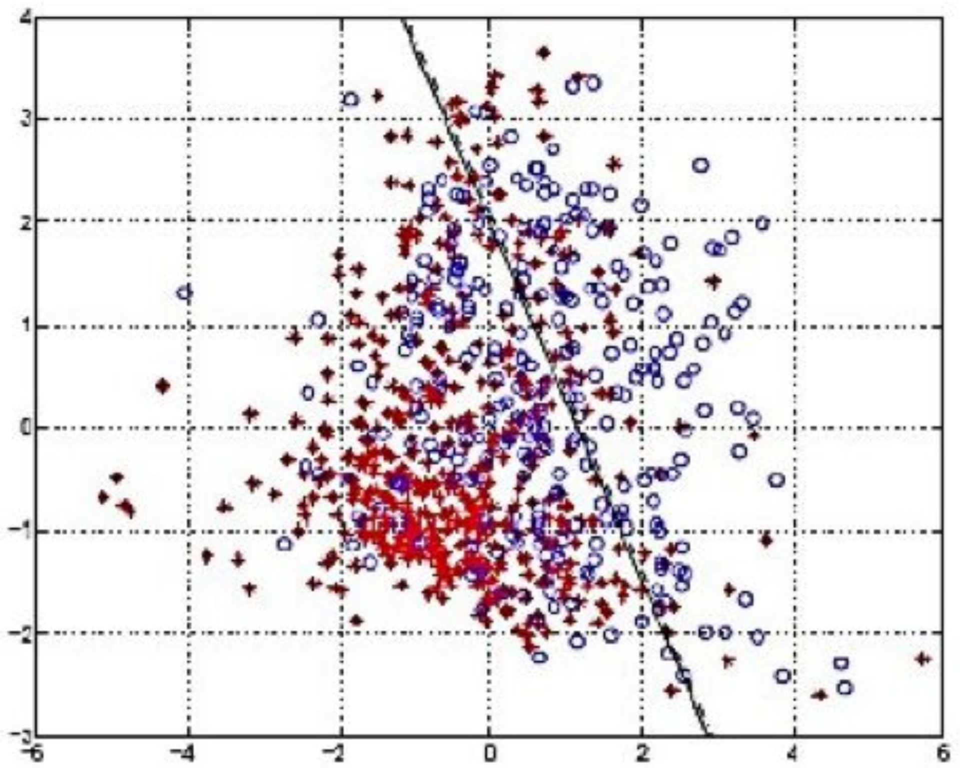




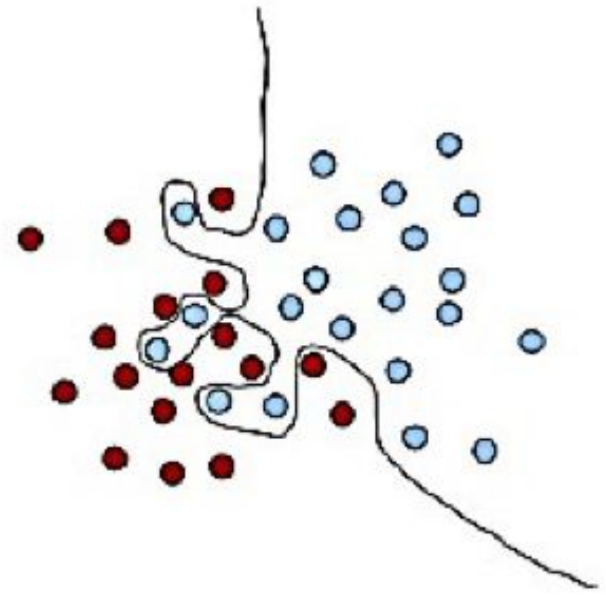
Decision Boundaries



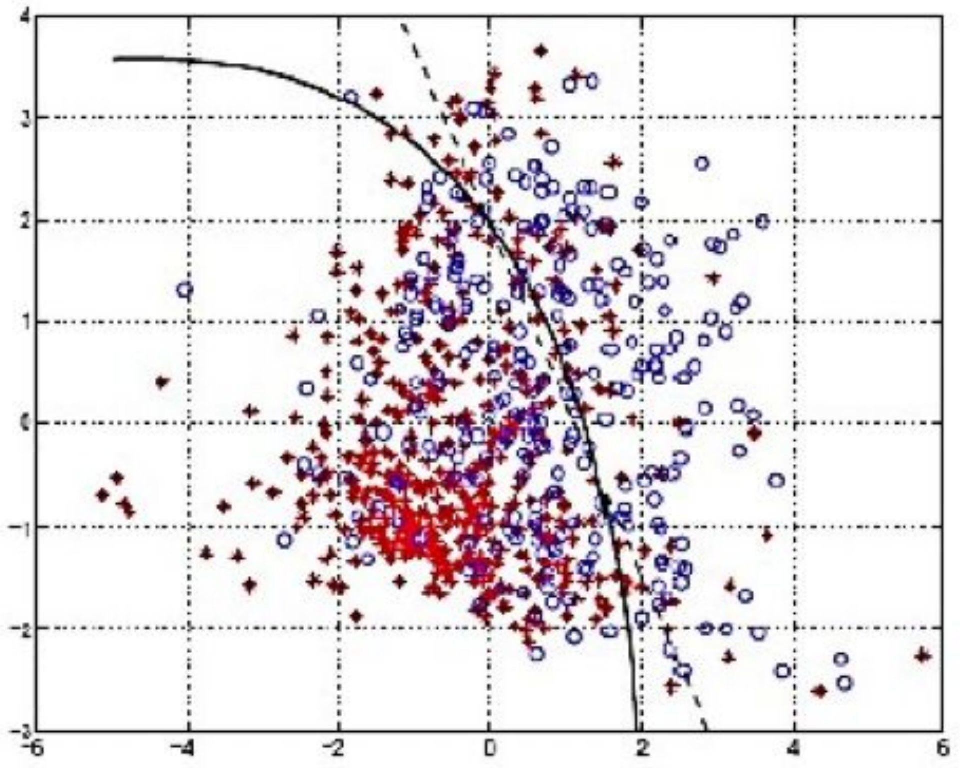
Linear decision boundary?
Non-separable data



Linear
Decision boundary



Non-linear decision boundary
– overfitting?



Quadratic
decision boundary

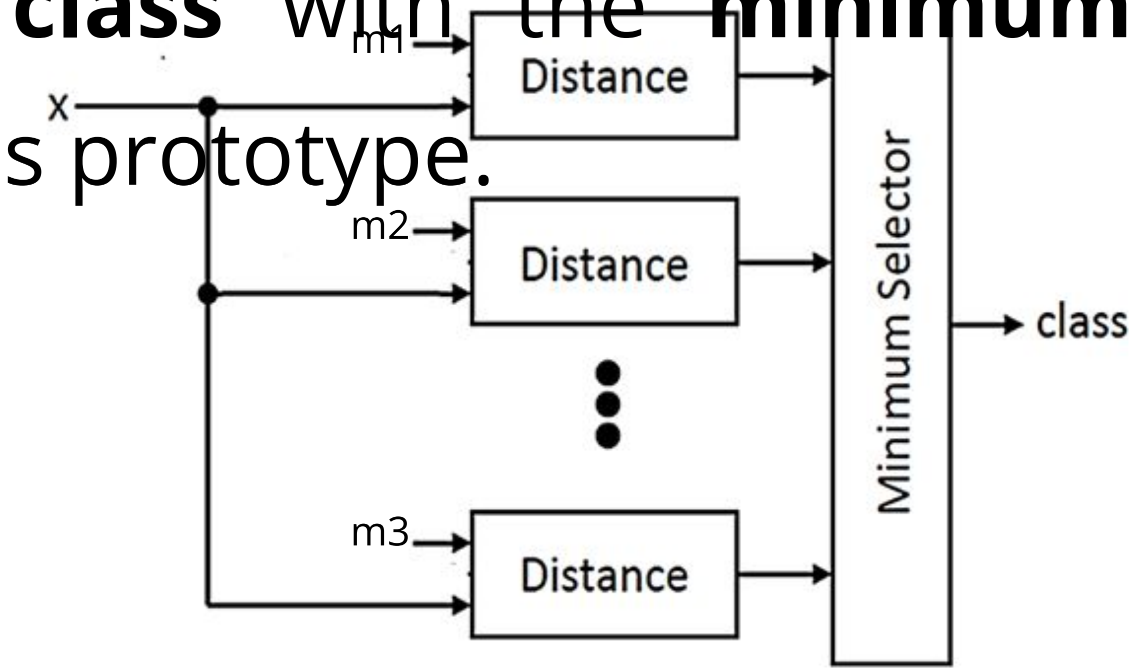
PROTOTYPE MATCHING

- **Classification based on matching** involves
  1. **Comparing** an unknown (actual) **pattern against a set of prototypes** characterizing the different classes,
  2. **Assigning** the **unknown pattern** to the class of the **prototype** that is the **most "similar" to the unknown.**
- Each prototype represents a unique pattern class.
- What distinguishes one matching method from another is the measure used to determine similarity.

MINIMUM-DISTANCE CLASSIFIER

- **Minimum-Distance Classifier**

1. one of the simplest and most widely used prototype matching methods
2. computes a **distance-based measure** between the **feature vector** of an unknown pattern **x** and each of the class **prototype vectors mj.**
3. assigns the unknown **pattern** to the **class** with the **minimum distance** between feature vector and class prototype.



distance for class j

$$D_j(\mathbf{x}) = \| \mathbf{x} - \mathbf{m}_j \| \qquad j = 1, 2, \ldots, N_c$$ = W

feature vector

prototype of class j

how to find the prototyps: discussed later

MINIMUM-DISTANCE CLASSIFIER

- A **distance (decision) measure** determines the similarity. That is, the smallest distance implies the best match.
- Often we use the **Euclidean distance**

$$D_j(\mathbf{x}) = \| \mathbf{x} - \mathbf{m}_j \| \qquad j = 1, 2, \ldots, N_c = W$$

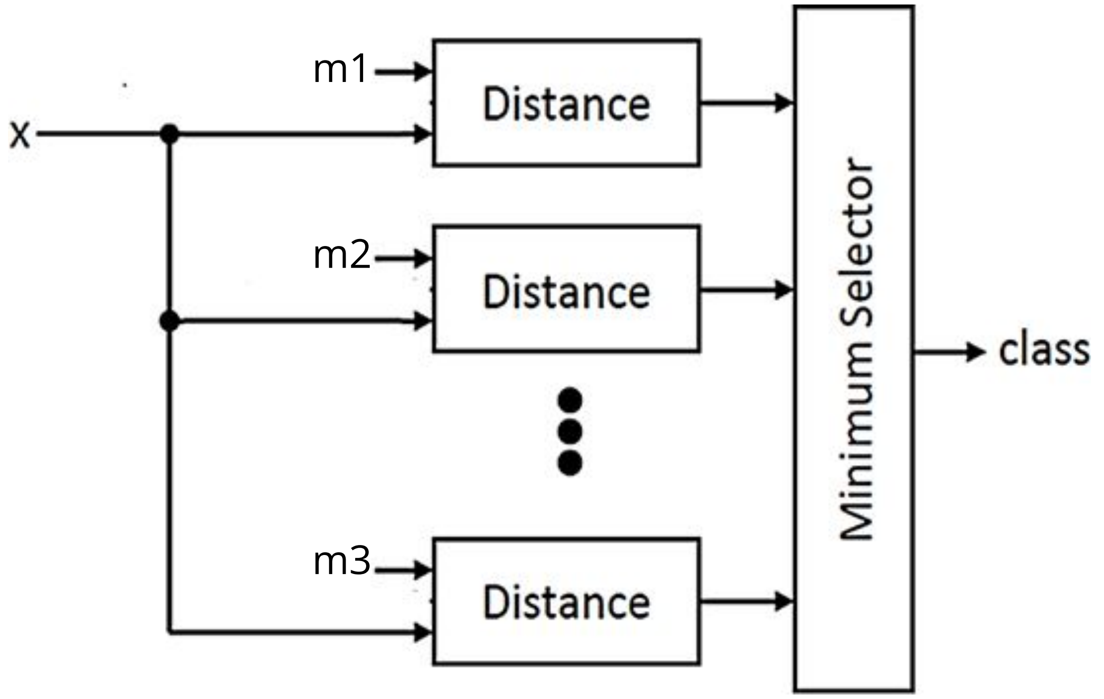
with the Euclidean norm $\| \mathbf{a} \| = (\mathbf{a}^T \mathbf{a})^{1/2}$ resp. $L_2(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{d} |\mathbf{x}_i - \mathbf{y}_i|^2 \right)^{1/2}$

- The classifier then assigns an unknown pattern x to class ci if

$$D_i(\mathbf{x}) < D_j(\mathbf{x})$$

minimize

$$j = 1, 2, \ldots, N_c, \; j \neq i$$



Decision Functions

Selecting the smallest distance is equivalent to evaluating the **decision or discriminant functions**.

$$d_j(x) = m_j^T x - \frac{1}{2} m_j^T m_j \quad j = 1, 2, \ldots W$$

and assigning an unknown pattern **x** to the class whose prototype yielded the **largestvalue of *dj*(x)**.

That is, **x** is assigned to class c_i with

$$d_i(x) > d_j(x) \qquad \text{for } j = 1, 2, 3, \ldots W \text{ and } j \neq i$$

i.e the largest $d_j(x)$.

DECISION BOUNDARIES

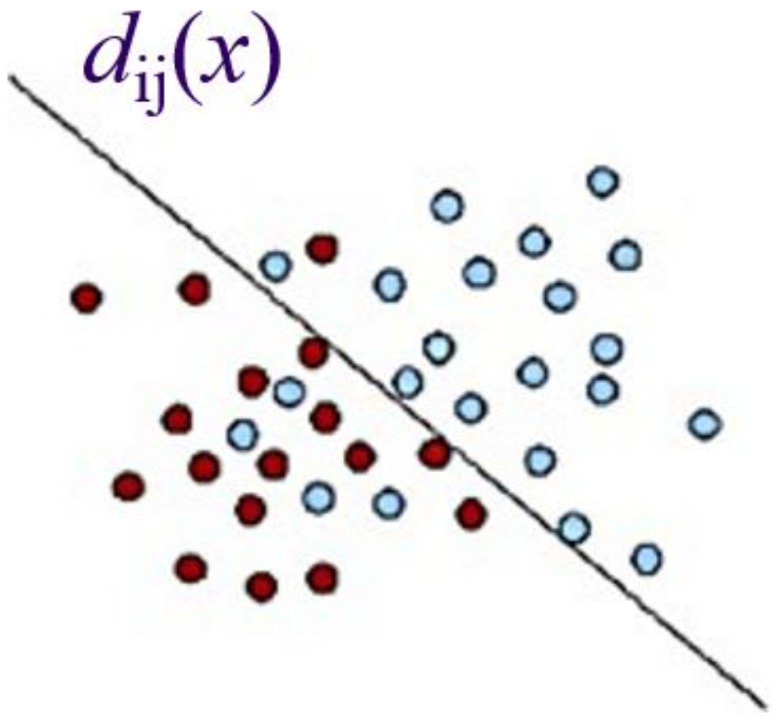
The **decision boundary dij(x)** separating class ci from cj is given by the values of x for which the decision functions

di(x) = dj(x)

have the same value or, equivalently, by values of x for which

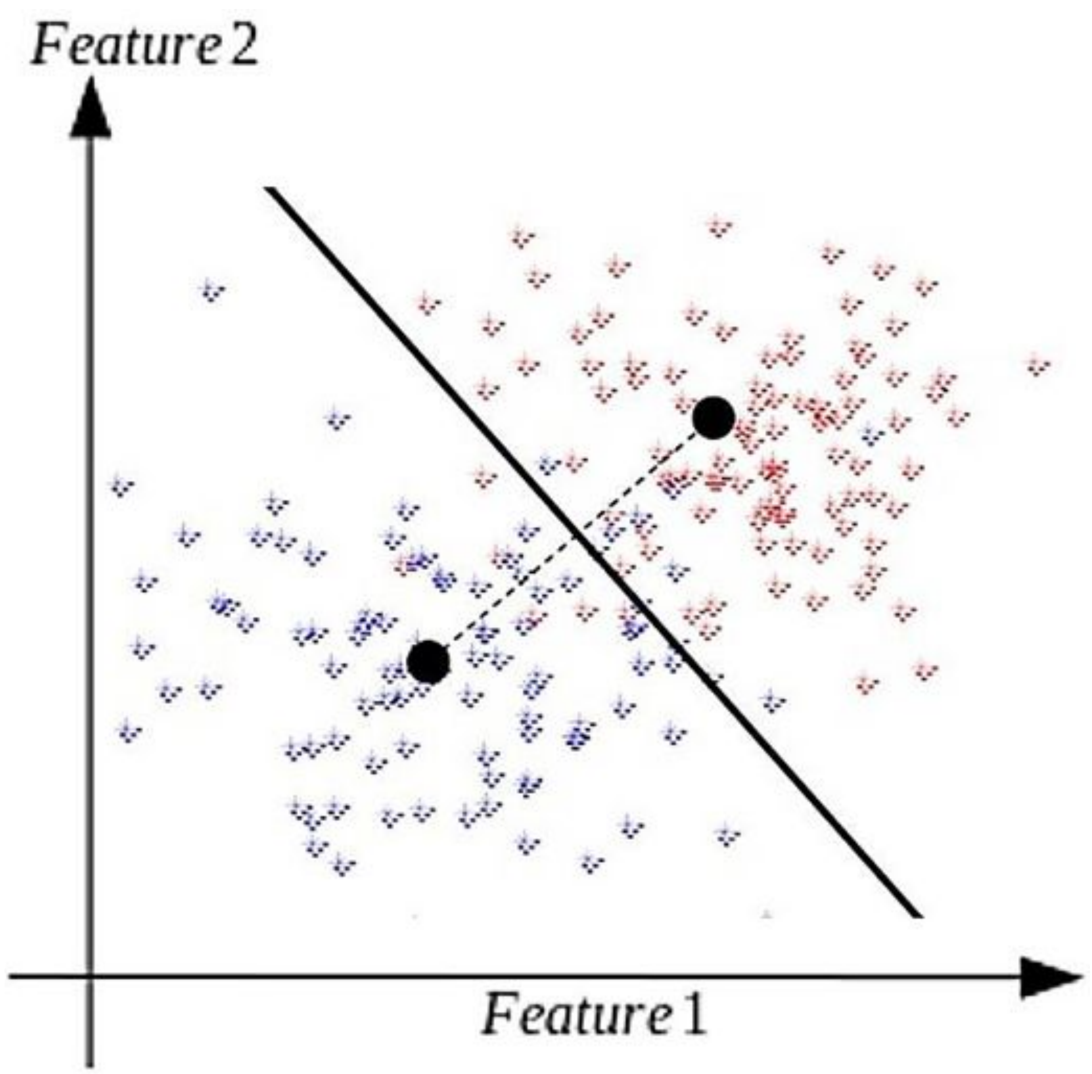
di(x) - dj(x) = 0

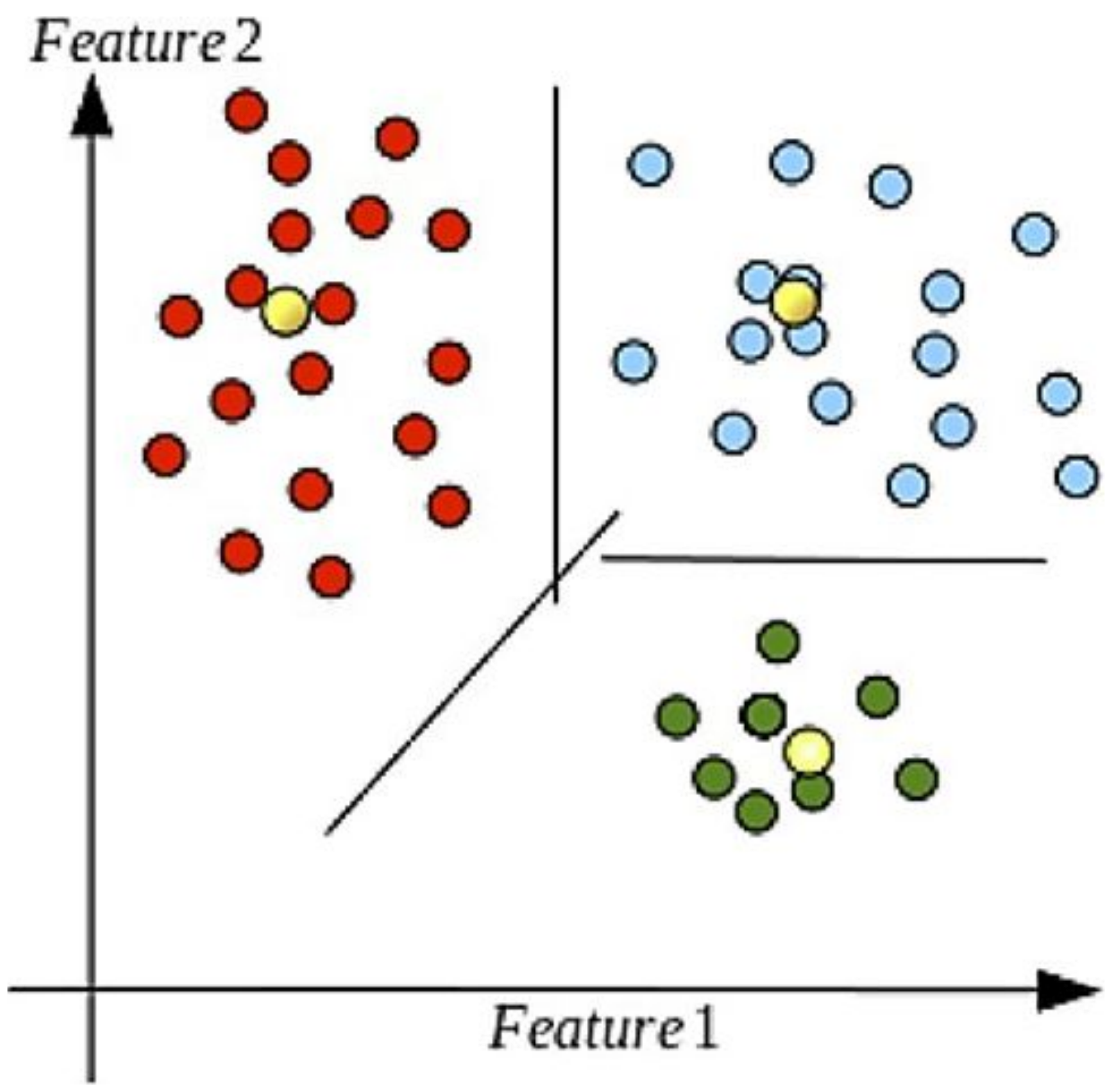
The decision boundaries for a **minimum-distance classifier** is given by



$$
\begin{aligned}
d_{ij}(\mathbf{x}) &= d_i(\mathbf{x}) - d_j(\mathbf{x}) \\
&= (\mathbf{m}_i - \mathbf{m}_j)^T \mathbf{x} - \frac{1}{2}(\mathbf{m}_i - \mathbf{m}_j)^T(\mathbf{m}_i + \mathbf{m}_j) = 0
\end{aligned}
$$

DECISION BOUNDARIES





This boundary is the perpendicular bisector of the line joining mi and mj.

In the 2-dimensional case (i.e. W=2), the perpendicular bisector is a line, for W=3 it is a plane, and for W>3 it is called a hyperplane.

PROTOTYPES AND CLASSES

- How to **define the prototypes m***j* for the different classes *cj*

  \- They may be well defined and **known in advance**;

  e.g. optical inspection: correct, defect-free reference part

  \- They may be **not exactly known in advance**, but **estimated** based on a training (sample) data set.

- Often the **means of feature vectors x** of samples of the various

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{\mathbf{x} \in \omega_j} \mathbf{x}_j, \quad j = 1, 2, \ldots, W$$

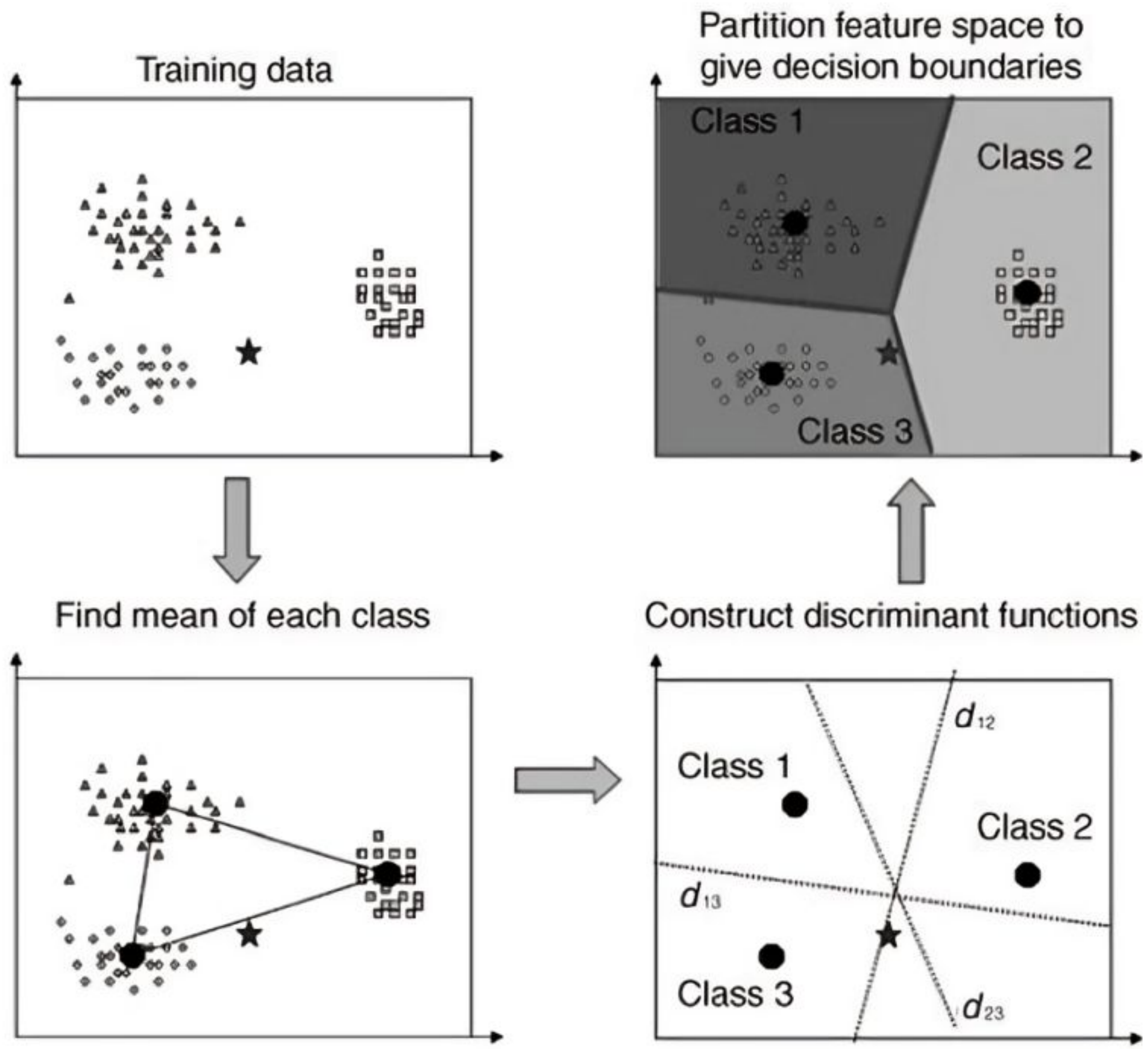
Nj is the number of pattern vectors used to compute the j-th mean vector

cj is the j-th pattern class, and W is the number of classes.

PROTOTYPES AND CLASSES

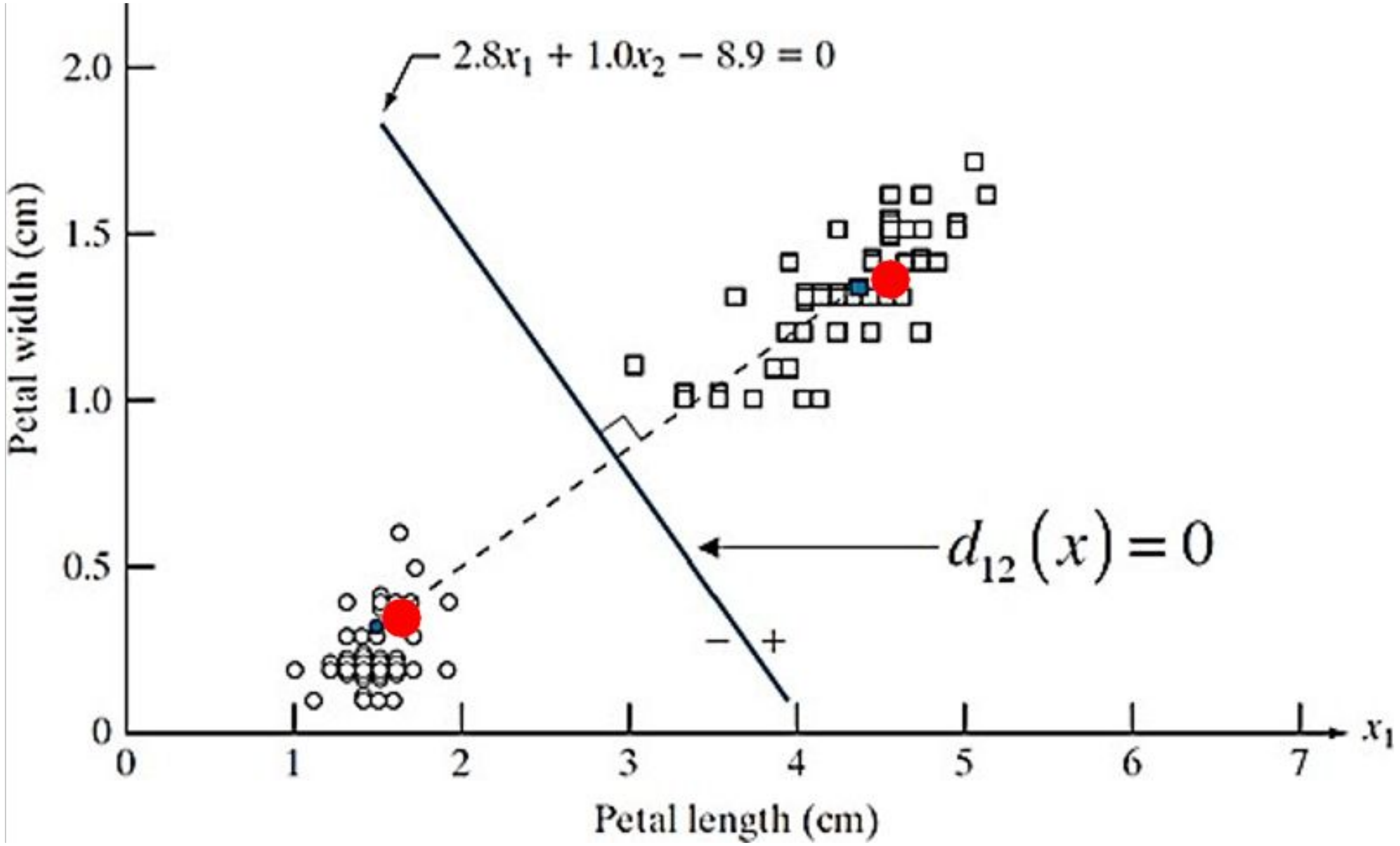
- If the **prototypes** of the classes are not well known or if they are not predefined, they **have to be learned (estimated)** by a set of training data.
- If the prototypes of each pattern class is the mean

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{\mathbf{x} \in \omega_j} \mathbf{x}_j, \quad j = 1, 2, \ldots, W$$



MINIMUM DISTANCE CLASSIFIER - EXAMPLE

- We denote the iris versicolor and iris setosa data as classes c1 and c2, respectively.
- The means of the two classes are $\mathbf{m}_1 = \begin{bmatrix} 4.3 \\ 1.3 \end{bmatrix}$ $\mathbf{m}_2 = \begin{bmatrix} 1.5 \\ 0.3 \end{bmatrix}$



MINIMUM DISTANCE CLASSIFIER - EXAMPLE

$$\mathbf{m}_1 = \begin{bmatrix} 4.3 \\ 1.3 \end{bmatrix} \qquad \mathbf{m}_2 = \begin{bmatrix} 1.5 \\ 0.3 \end{bmatrix}$$

$$d_1(\mathbf{x}) = \mathbf{m}_1^T\mathbf{x} - \tfrac{1}{2}\mathbf{m}_1^T\mathbf{m}_1 = 4.3x_1 + 1.3x_2 - 10.1$$

$$d_2(\mathbf{x}) = \mathbf{m}_2^T\mathbf{x} - \tfrac{1}{2}\mathbf{m}_2^T\mathbf{m}_2 = 1.5x_1 + 0.3x_2 - 1.17$$

$$d_{12}(\mathbf{x}) = d_1(\mathbf{x}) - d_2(\mathbf{x}) = 2.8x_1 + 1.0x_2 - 8.9 = 0$$

Example: $\mathbf{x} = \begin{bmatrix} 3.0 \\ 1.0 \end{bmatrix}$

$$\begin{aligned} d_{12}(\mathbf{x}) &= 2.8(3.0) + 1.0(1.0) - 8.9 \\ &= 8.4 + 1.0 - 8.9 \\ &= 0.5 > 0 \end{aligned}$$

$\therefore$ $\mathbf{x}$ is classified into Class 1.

DISTANCE FUNCTIONS - METRICS

- The classifier relies on a metric or a distance function between points.
- For all points x, y and z, a metric $D(\cdot,\cdot)$ must satisfy the following properties:
    - Nonnegativity: $D(x,y) \geq 0$
    - Reflexivity: $D(x,y)=0$ if and only if $x=y$
    - Symmetry: $D(x,y)=D(y,x)$
    - Triangle inequality: $D(x,y)+D(y,z) \geq D(x,z)$

DISTANCE FUNCTIONS - METRICS

- The Euclidean distance is the L2 norm

$$L_2(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^2 \right)^{1/2}$$

- The Manhattan or city block distance is the L1 norm

$$L_1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} |x_i - y_i|$$

- A general class of metrics for d -dimensional patterns is the Minkowski metric, also referred to as the Lp norm

$$L_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p}$$

- The L∞ norm is the maximum of the distances along individual coordinate axes

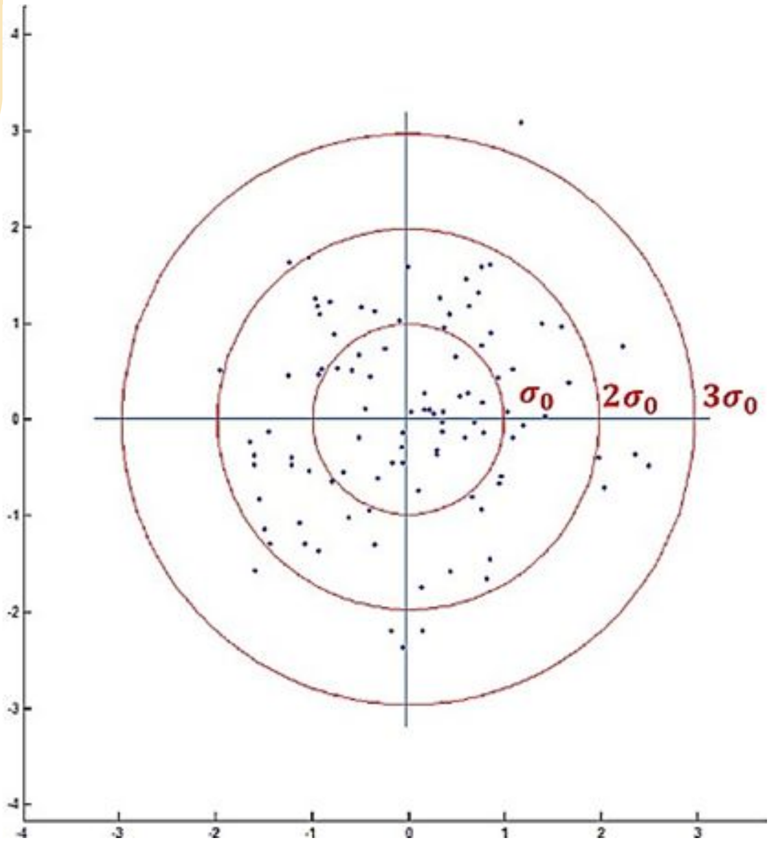
$$L_\infty(\mathbf{x}, \mathbf{y}) = \max_{1 \le i \le d} |x_i - y_i|$$

- Mehalanobis distance

$$L(\mathbf{x}, \mathbf{y}) = [(\mathbf{x}\ \mathbf{y})^\mathsf{T} \boldsymbol{\Sigma}^{-1} (\mathbf{x}\ \mathbf{y})]^{1/2}$$

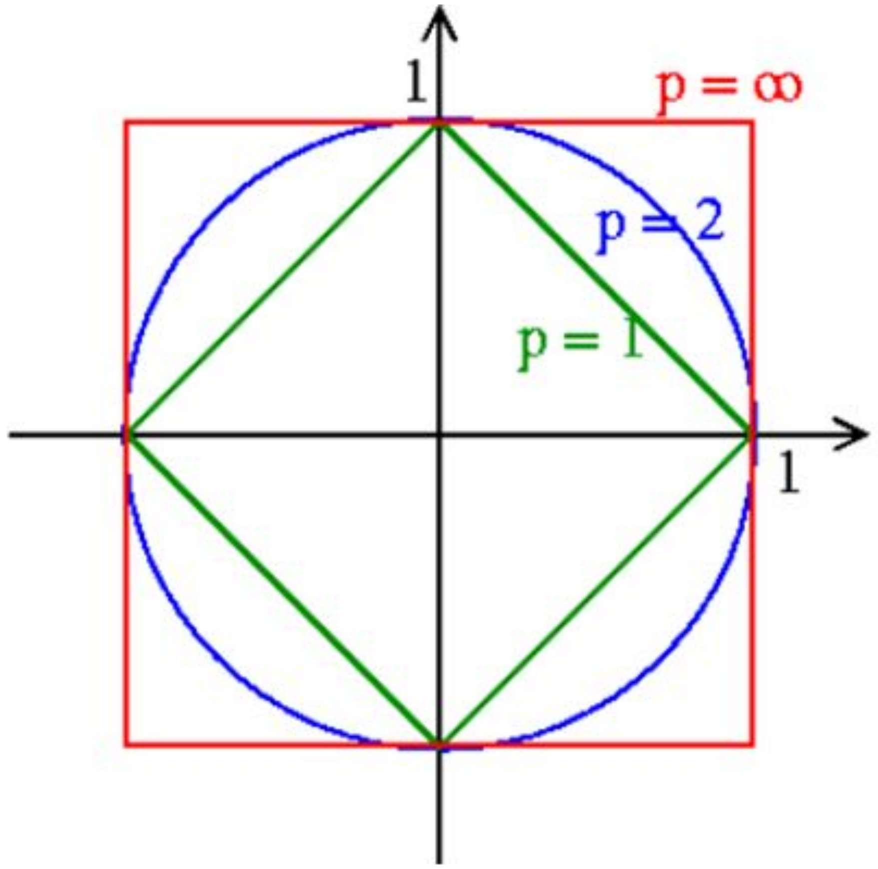
$$\boldsymbol{\Sigma} = \langle (\mathbf{x} - \mu_\mathbf{x})(\mathbf{y} - \mu_\mathbf{y})^\mathsf{T} \rangle$$

DISTANCE FUNCTIONS - METRICS

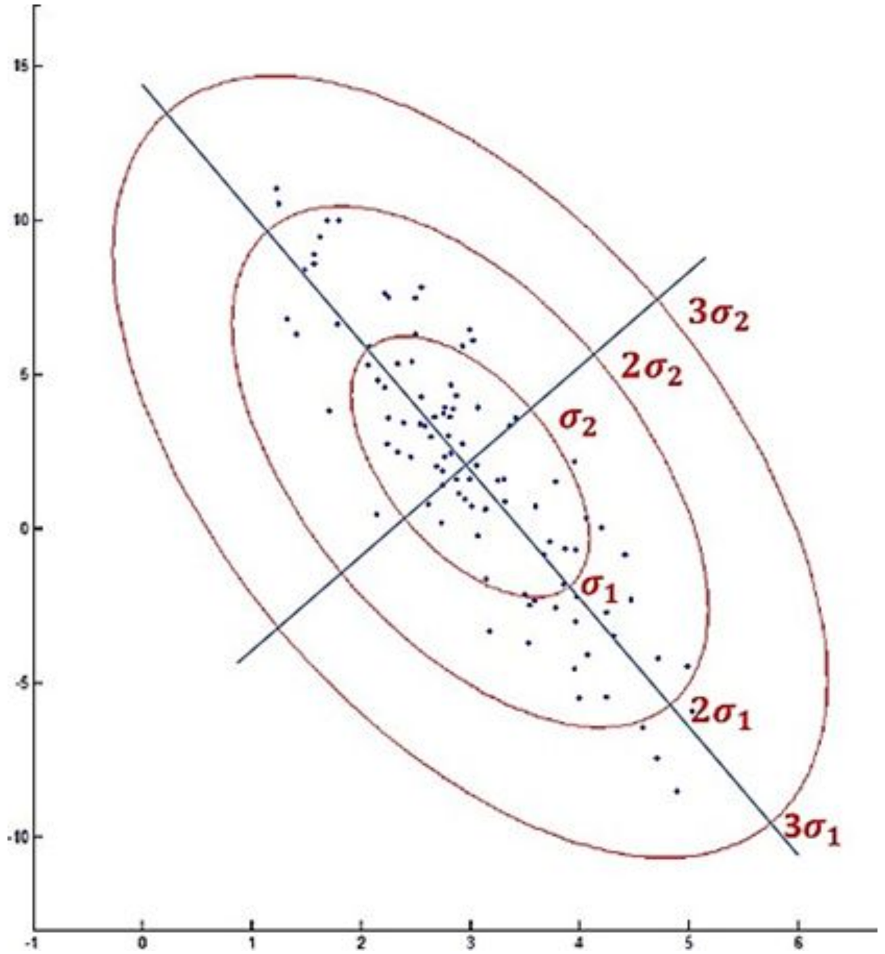


- Euclidean Distance

Points with equal distance to μ lie on a circle



- Mehalanobis Distance



Points with equal distance to μ lie on an ellipse.

Each colored shape consists of points at a distance 1.0 from the origin, measured using different values of p in the Minkowski Lp metric

FEATURE NORMALIZATION

- So far we assumed we use Euclidian Distance to find the nearest neighbour:

$$D(a,b) = \sqrt{\sum_k (a_k - b_k)^2}$$

- Euclidean distance treats each feature as equallly important

- However some features (dimensions) may be much more discriminative than other features

FEATURE NORMALIZATION

- We should be careful about scaling of the coordinate axes when we compute these metrics.
- When there is great difference in the range of the data along different axes in a multidimensional space, these metrics implicitly assign more weighting to features with large ranges than those with small ranges.
- Feature normalization can be used to approximately equalize ranges of the features and make them have approximately the same effect in the distance computation.
- The following methods can be used to independently normalize each feature.

FEATURE NORMALIZATION

- Linear scaling to unit range:

  Given a lower bound l and an upper bound u for a feature x

$$\tilde{x} = \frac{x - l}{u - l}$$

results in the [0,1] range

- Linear scaling to unit variance:

  A feature x can be transformed to a random variable with zero mean and unit variance as

$$\tilde{x} = \frac{x - \mu}{\sigma}$$

where μ and σ are the sample mean and the sample standard deviation of that feature, respectively.

FEATURE NORMALIZATION

- Feature normalization does not help in high dimensional spaces if most features are irrelevant

$$D(a,b)=\sqrt{\sum_k (a_k-b_k)^2}=\sqrt{\underbrace{\sum_i (a_i-b_i)^2}_{\text{discriminative features}}+\underbrace{\sum_j (a_j-b_j)^2}_{\text{noisy features}}}$$

- If the number of useful features is smaller than the number of noisy features, Euclidean distance is dominated by noise

FEATURE NORMALIZATION

- Scale each feature by its importance for classification

$$D(a, b) = \sqrt{\sum_k w_k \left(a_k - b_k\right)^2}$$

- Try to use our prior knowledge about which features are more important
- Try to learn the weights wk

ADVANTAGES

1. **Simple & Fast** – The classification rule is based only on the distance to class means $\hat{y} = \arg\min_i \|x - \mu_i\|$, making it computationally light and efficient even for large datasets.
2. **Low Memory Requirement** – Only class means (μ_i) need to be stored, unlike k-NN which requires the entire dataset, making MDC memory-efficient.
3. **Interpretability** – The decision boundaries are straight lines (in 2D) or hyperplanes (in higher dimensions), which are easy to visualize and explain.
4. **Connection to Bayes Classifier** – MDC is theoretically equivalent to a Bayes classifier when classes follow Gaussian distributions with equal covariance matrices.

APPLICATIONS

- **Pattern Recognition** – Handwritten digit and image classification (e.g., OCR systems).
- **Speech & Audio** – Speaker identification, speech signal classification.
- **Remote Sensing** – Land cover classification in satellite imagery.
- **Medical Diagnosis** – Quick classification of patient data (healthy vs diseased).
- **Document Classification** – Classifying texts/news by topic using feature vectors.

DISDVANTAGES

- **Sensitive to Feature Scaling** – Features with larger numerical ranges dominate the distance measure, requiring normalization or standardization.
- **Outlier Influence** – A single extreme point can shift the class mean significantly, leading to misclassification.
- **Poor with Overlapping Classes** – When class distributions overlap heavily, MDC struggles since it only relies on mean separation.
- **Linear Decision Boundaries** – MDC cannot handle classes separated by non-linear boundaries, limiting performance on complex datasets.

LIMITATIONS

- **Curse of Dimensionality** – In high dimensions, distance measures become less meaningful and the estimation of class means becomes unreliable.
- **Class Imbalance Issue** – For imbalanced datasets, the minority class mean is poorly estimated, making classification biased toward majority classes.
- **Loss of Variance Information** – Since MDC uses only the mean, it ignores the spread or variance within each class, which can be important for accurate classification.
- **Not Suitable for Complex Distributions** – If a class has multiple clusters (multimodal), a single mean cannot represent it, leading to poor accuracy.

CONTRIBUTIONS

2201CS91: 4.16%
2201CS70: 4.16%
2201CS20: 4.16%
2201CS88: 4.16%
2201CS68: 4.16%
2201CS10: 4.16%
2201CS06: 4.16%
2201CS92: 4.16%
2201CS65: 4.16%
2201CS39: 4.16%
2201AI38: 4.16%
2201AI55: 4.16%

2201AI35: 4.16%
2201CS04: 4.16%
2201CS40: 4.16%
2201CS41: 4.16%
2201CS42: 4.16%
2201CS57: 4.16%
2201CS63: 4.16%
2201CS64: 4.16%
2201CS67: 4.16%
2201CS95: 4.16%
2201AI07: 4.16%
2201AI14: 4.16%