Introduction to Machine Learning
Lecture 5 - Supervised Learning

1 Introduction

Consider the **training set**:

$$\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$$

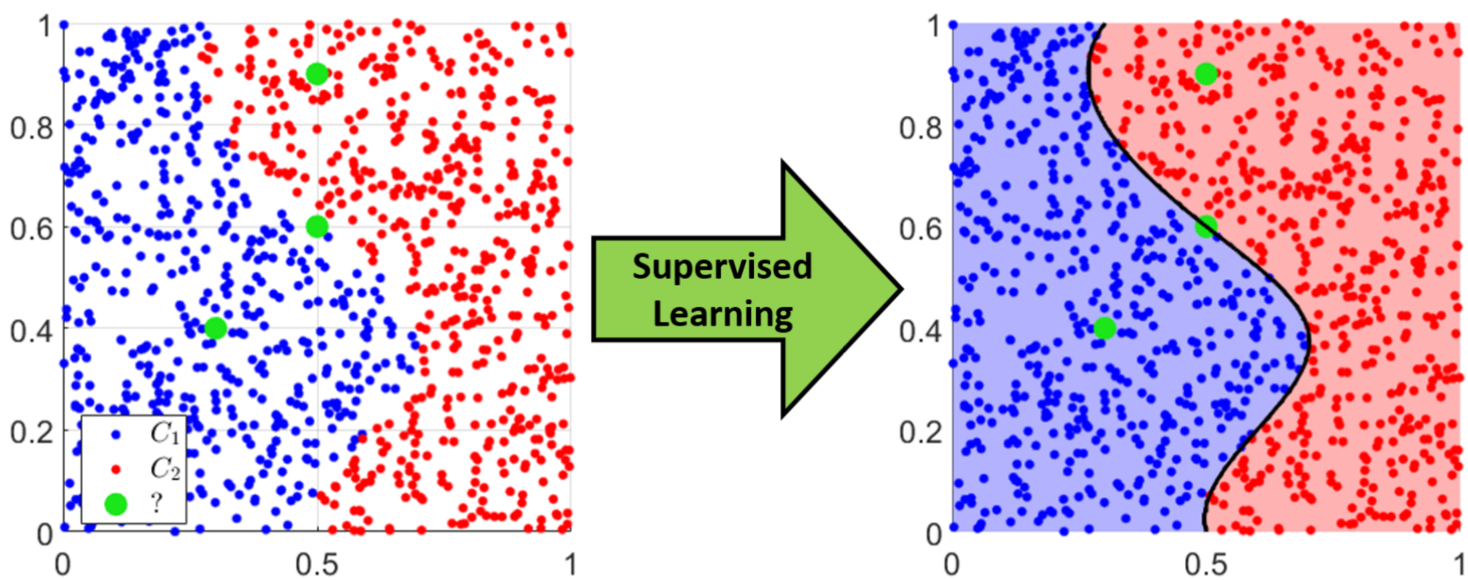
where:

- $\boldsymbol{x}_i \in \mathcal{X} \subseteq \mathbb{R}^D$ is the features vector (also known as a sample \ observation)
- $y_i \in \mathcal{Y} = \{C_1, C_2, \ldots, C_K\}$ or just $y_i \in \{1, 2, \ldots, K\}$ is the class (category) of $\boldsymbol{x}_i$.

Consider a new pair $(\boldsymbol{x}_0, y_0)$ where $\boldsymbol{x}_0$ is known and y_0 is unknown.
Given the set $\mathcal{D}$, our goal is to derive a classifier function $f : \mathcal{X} \longrightarrow \mathcal{Y}$, such that:

$$y_0 = f(\boldsymbol{x}_0)$$

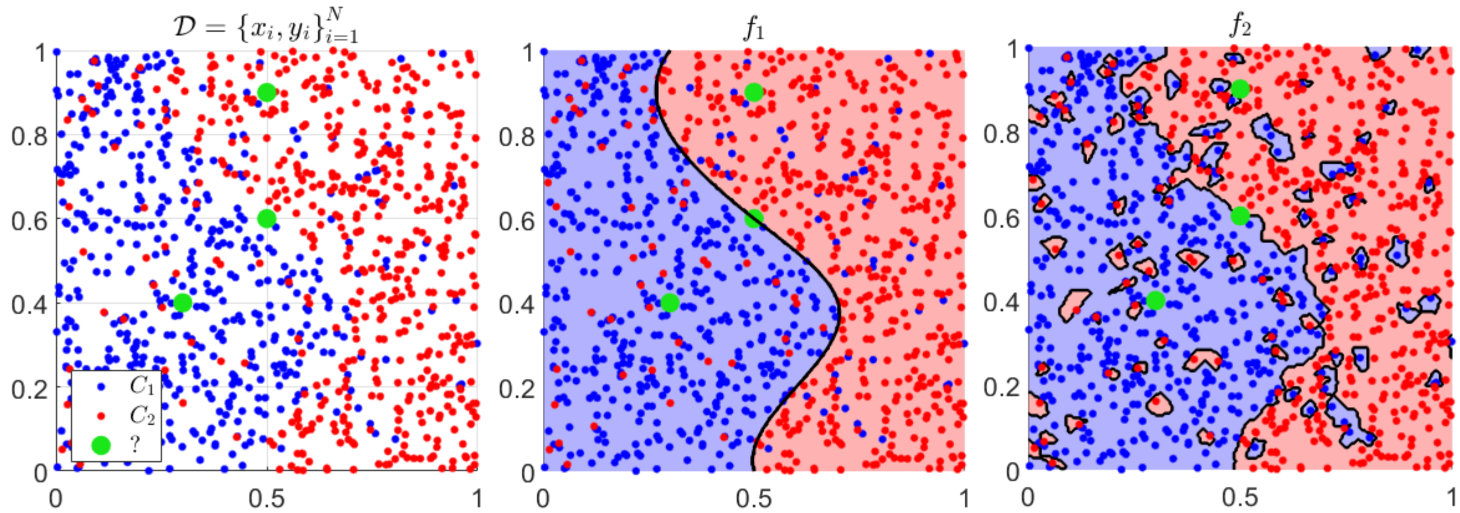
For example, $\mathcal{D}$ is the set of all red and blue points (left figure).
Can we guess the classes of the three green points?



Notes:

- For $\mathcal{Y} = \{0, 1\}$, the problem is known as **binary classification**.
- If $\mathcal{Y}$ is a continuous set we called this problem **regression**.

In actual problems the data might not be perfectly separable:



and learning a prefect f is impossible.

2 Quality Index

2.1 Loss Function

A loss function $\ell:\mathcal{Y}\times\mathcal{Y}\longrightarrow\mathbb{R}$ satisfies:

1. $$\ell\left(y,y\right)=0,\qquad\forall y\in\mathcal{Y}$$

2. $$\ell\left(\hat{y},y\right)\geq0$$

Examples

1. Hamming loss:

$$\ell\left(\hat{y},y\right)=\boldsymbol{I}\left\{\hat{y}\neq y\right\}\triangleq\begin{cases}1 & \hat{y}\neq y\\0 & \text{else}\end{cases}$$

2. Squared loss (for continuous $\mathcal{Y}$):

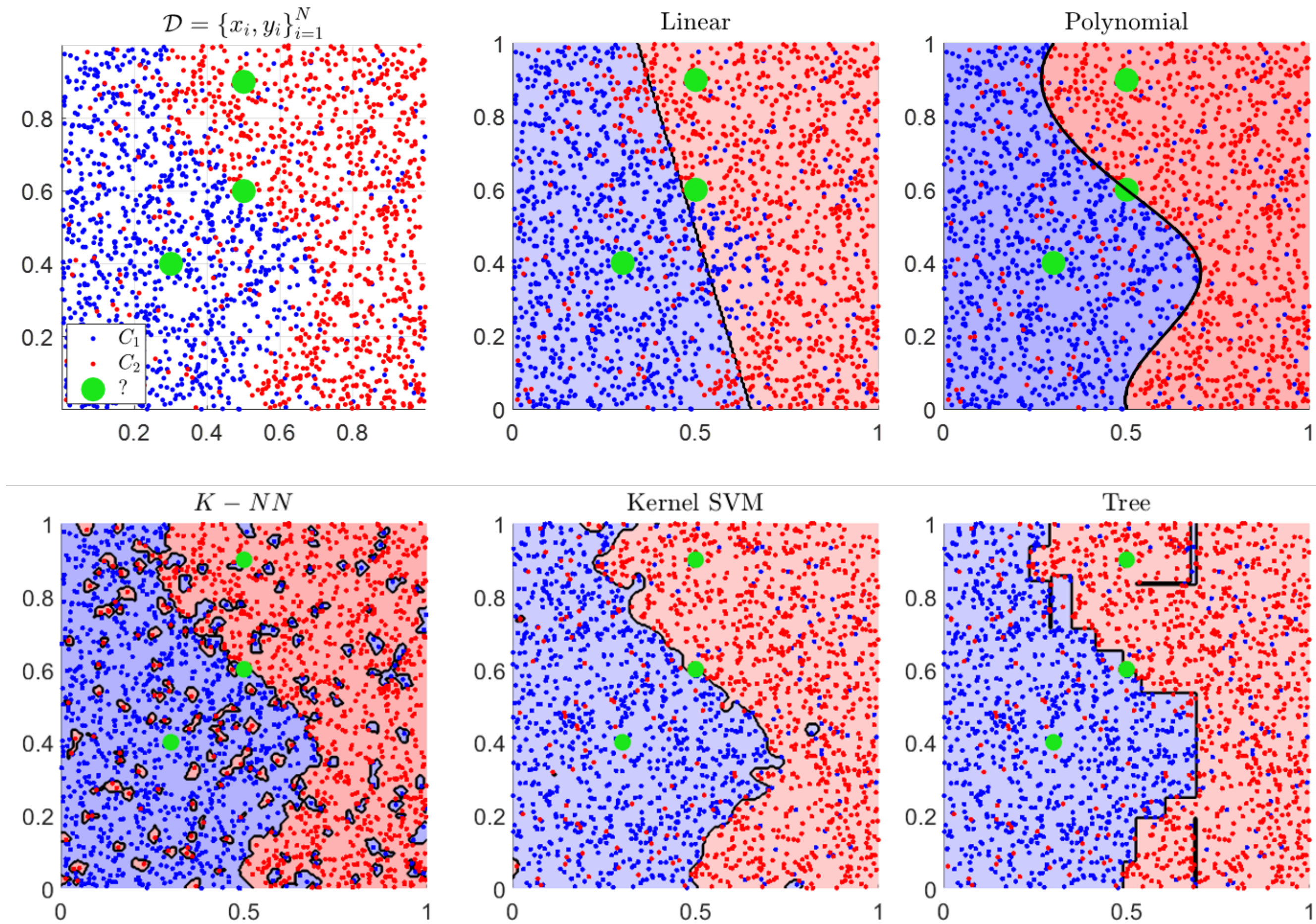
$$\ell\left(\hat{y},y\right)=\left(\hat{y}-y\right)^2$$

2.2 Empirical Risk (Training and Test Loss)

Let $f:\mathcal{X}\longrightarrow\mathcal{Y}$ be a classifier function.
Based on a loss function ℓ,
we can evaluate the performance of f on a set $\mathcal{D}=\left\{\left(\boldsymbol{x}_i,y_i\right)\right\}$ by:

$$L\left(f\right)\triangleq\frac{1}{N}\sum_{i=1}^{N}\ell\left(f\left(\boldsymbol{x}_i\right),y_i\right)$$

Example – Models and decision regions



3 Linear Binary Classification Using Perceptron

Let $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$ be a training set where:

- $\boldsymbol{x}_i \in \mathcal{X} = \mathbb{R}^D$.
- $y_i \in \mathcal{Y} = \{-1, 1\}$ (binary classification)

We search for a linear classifier:

$$f : \mathcal{X} \longrightarrow \mathcal{Y}$$

such that:

$$f(\boldsymbol{x}) = \text{sign}\left(\boldsymbol{w}^T \boldsymbol{x} - b\right) = \begin{cases} 1 & \boldsymbol{w}^T \boldsymbol{x} - b \geq 0 \\ -1 & \boldsymbol{w}^T \boldsymbol{x} - b < 0 \end{cases}$$

where $\boldsymbol{w} \in \mathbb{R}^D$ and $b \in \mathbb{R}$ are the classifier model parameters.

- Notice that for any value of C:

$$\text{sign}\left(\boldsymbol{w}^T \boldsymbol{x} - b\right) = \text{sign}\left(\frac{\boldsymbol{w}^T \boldsymbol{x}}{C} - \frac{b}{C}\right), \qquad \forall \boldsymbol{x} \in \mathbb{R}^D$$

  Thus, we can assume (without loss of generality) that $\|\boldsymbol{w}\| = 1$.

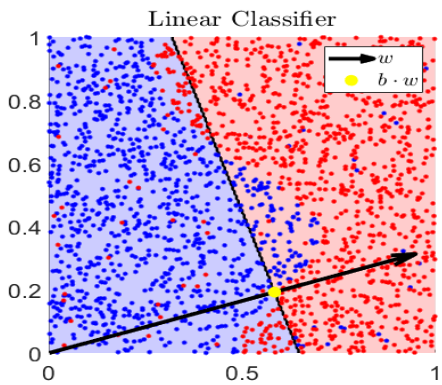
3.1 Analysis

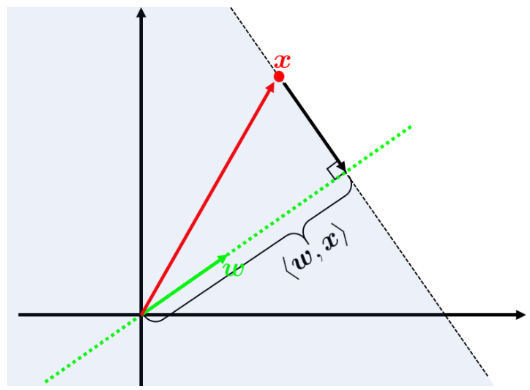
Consider the function:

$$f(\boldsymbol{x}) = \operatorname{sign}\left(\boldsymbol{w}^T\boldsymbol{x} - b\right), \qquad \|\boldsymbol{w}\| = 1$$

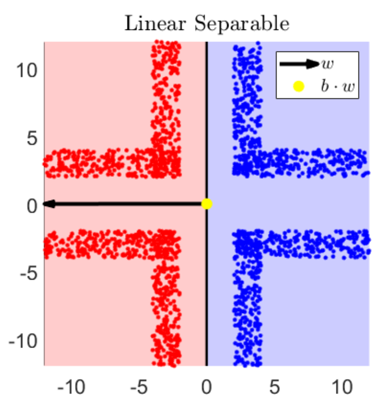
This function split the input space $\mathcal{X}$ into two half-planes.
The decision boundary is given by:

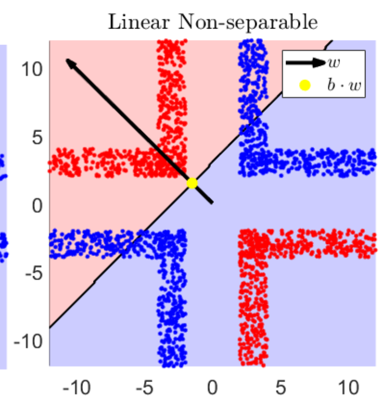
$$\boldsymbol{w}^T\boldsymbol{x} - b = 0$$
$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle = b$$





Linear separable and linear non-separable





3.2 The Perceptron Algorithm

For convenience purposes, we denote:

$$\tilde{\boldsymbol{x}}_i \triangleq \begin{bmatrix} 1 \\ \boldsymbol{x}_i \end{bmatrix}, \qquad \tilde{\boldsymbol{w}} = \begin{bmatrix} -b \\ \boldsymbol{w} \end{bmatrix}$$

So, we have:

$$\Rightarrow \boxed{\boldsymbol{w}^T \boldsymbol{x} - b = \tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}}}$$

Algorithm 1 The Perceptron Algorithm

Input: Training set $\mathcal{D} = \{\tilde{\boldsymbol{x}}_i, y_i\}$

Output: The linear classifiers parameters: $\tilde{\boldsymbol{w}}$ (that is, $\boldsymbol{w}$ and b).

1. Set $\tilde{\boldsymbol{w}}_1$ with some initial guess.
2. **for** $k = 1, 2, 3, \ldots$
   (a) Choose some $(\tilde{\boldsymbol{x}}_k, y_k) \in \mathcal{D}$
   (b) Compute:
   $$\hat{y}_k = \text{sign}\left(\tilde{\boldsymbol{w}}_k^T \tilde{\boldsymbol{x}}_k\right)$$
   (c) Update:
   $$\tilde{\boldsymbol{w}}_{k+1} = \tilde{\boldsymbol{w}}_k + \frac{1}{2}\left(y_k - \hat{y}_k\right)\tilde{\boldsymbol{x}}_k$$

Notes:

- If $\hat{y}_k = y_k$, there is no update: $\tilde{\boldsymbol{w}}_{k+1} = \tilde{\boldsymbol{w}}_k$
- If $\hat{y}_k = -1$ and $y_k = 1$, then, the value $\tilde{\boldsymbol{w}}_k^T \tilde{\boldsymbol{x}}_k < 0$ is too small (and we should increase it).
  - Step (c) increases the value of $\tilde{\boldsymbol{w}}_k^T \tilde{\boldsymbol{x}}_k$:

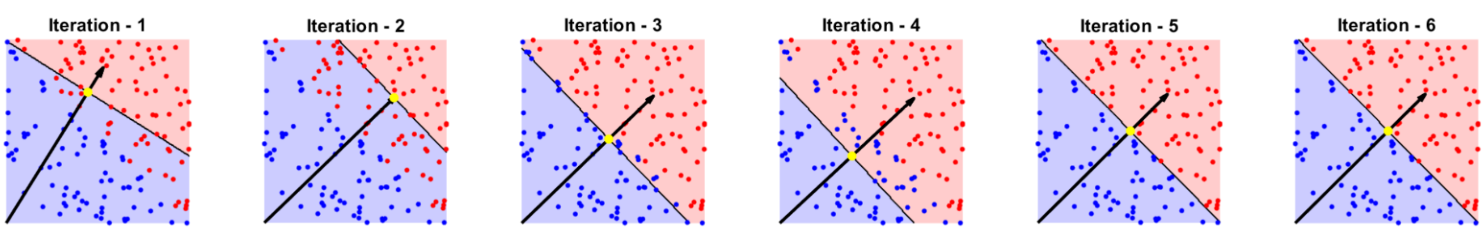
$$\tilde{\boldsymbol{w}}_{k+1} = \tilde{\boldsymbol{w}}_k + \frac{1}{2}\left(y_k - \hat{y}_k\right)\tilde{\boldsymbol{x}}_k = \tilde{\boldsymbol{w}}_k + \tilde{\boldsymbol{x}}_k$$

$$\Rightarrow \boxed{\tilde{\boldsymbol{w}}_{k+1}^T \tilde{\boldsymbol{x}}_k = \left(\tilde{\boldsymbol{w}}_k + \tilde{\boldsymbol{x}}_k\right)^T \tilde{\boldsymbol{x}}_k = \tilde{\boldsymbol{w}}_k^T \tilde{\boldsymbol{x}}_k + \|\tilde{\boldsymbol{x}}_k\|_2^2 > \tilde{\boldsymbol{w}}_k^T \tilde{\boldsymbol{x}}_k}$$

- In the same manner, the algorithm updates the miss-classification $\hat{y}_k = 1$ and $y_k = -1$.
- The algorithm converge in finite number of iterations if the problem is linear separable.
- In the linear non-separable case, there is no guarantee for convergence.

Example

Let $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$ with $N = 150$.
We apply the algorithm for $5N$ iterations, namely, we run over $\mathcal{D}$, 5 times:



4 Quadratic Classifier Using Bayesian Approach

4.1 Introduction

Given the statistic $p_{X,Y}$ we can use the MAP classifier:

$$f_{MAP}\left(\boldsymbol{x}\right) = \arg\max_{C_k \in \mathcal{Y}} p\left(\boldsymbol{x}|C_k\right) P_{\mathcal{Y}}\left(C_k\right)$$

In practice, the statistic is usually unknown.
However, we can use the training set $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$ to estimate it.

Prior Estimation The estimation of $P_{\mathcal{Y}}$ is given by:

$$\boxed{\hat{P}_{\mathcal{Y}}\left(C_k\right) = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{I}\left\{y_i = C_k\right\}}$$

Likelihood Estimation Lets assume that given C_k:

$$\boldsymbol{x}|C_k \sim \mathcal{N}\left(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right), \qquad \forall C_k \in \mathcal{Y}$$

Thus, we can estimate the mean and covariance of each class:

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{N_k}\sum_{\boldsymbol{x}_i \in \mathcal{C}_k} \boldsymbol{x}_i, \qquad N_k \triangleq \sum_{i=1}^{N} \boldsymbol{I}\left\{y_i = C_k\right\}$$

$$\hat{\boldsymbol{\Sigma}}_k = \frac{1}{N_k}\sum_{\boldsymbol{x}_i \in \mathcal{C}_k} \left(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k\right)\left(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_k\right)^T$$

$$\Rightarrow \boxed{p\left(\boldsymbol{x}|C_k\right) = \left|2\pi\boldsymbol{\Sigma}_k\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{x} - \boldsymbol{\mu}_k\right)^T \boldsymbol{\Sigma}_k^{-1}\left(\boldsymbol{x} - \boldsymbol{\mu}_k\right)\right)}$$

4.2 Binary Case Analysis

Consider the binary case: $\mathcal{Y} = \{C_1, C_2\}$.
The MAP classifier is given by:

$$\begin{aligned} p\left(\boldsymbol{x}|C_1\right) P_{\mathcal{Y}}\left(C_1\right) &\underset{C_2}{\overset{C_1}{\gtrless}} p\left(\boldsymbol{x}|C_2\right) P_{\mathcal{Y}}\left(C_2\right) \\ p_1 p\left(\boldsymbol{x}|C_1\right) &\underset{C_2}{\overset{C_1}{\gtrless}} \left(1 - p_1\right) p\left(\boldsymbol{x}|C_2\right), \qquad p_1 \triangleq P_{\mathcal{Y}}\left(C_1\right) \end{aligned}$$

In other words, the boundary decision is given by:

$$\begin{aligned}
p_1 p(\boldsymbol{x}|C_1) &= (1-p_1)\, p(\boldsymbol{x}|C_2) \\
p_1 |2\pi\boldsymbol{\Sigma}_1|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)\right) &= (1-p_1)|2\pi\boldsymbol{\Sigma}_2|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2)\right) \\
p_1 d_1 \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)\right) &= (1-p_1) d_2 \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2)\right), \qquad d_{1,2} \triangleq |\boldsymbol{\Sigma}_{1,2}|^{-\frac{1}{2}} \\
\log(p_1 d_1) - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) &= \log((1-p_1)d_2) - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) \\
\underbrace{2\log\left(\frac{p_1 d_1}{(1-p_1)d_2}\right)}_{\triangleq b} - (\boldsymbol{x}-\boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) &= -(\boldsymbol{x}-\boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2)
\end{aligned}$$

$$\begin{aligned}
b - (\boldsymbol{x}-\boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + (\boldsymbol{x}-\boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) &= 0 \\
\underbrace{b - \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2^T\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2}_{\tilde{b}} - \underbrace{2\left(\boldsymbol{\mu}_2^T\boldsymbol{\Sigma}_2^{-1} - \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}_1^{-1}\right)}_{\triangleq \boldsymbol{w}^T}\boldsymbol{x} + \boldsymbol{x}^T\underbrace{\left(\boldsymbol{\Sigma}_2^{-1} - \boldsymbol{\Sigma}_1^{-1}\right)}_{\triangleq \boldsymbol{M}}\boldsymbol{x} &= 0
\end{aligned}$$

$$\Rightarrow \boxed{\tilde{b} - \boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{M}\boldsymbol{x} = 0}$$

This is a **quadratic** decision boundary.

Examples Consider two states with equal a-priori probability:

$$\mathcal{Y} = \{C_1, C_2\}$$

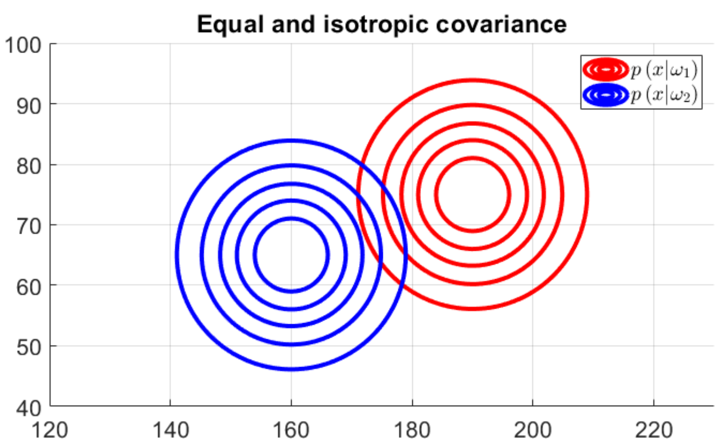
$$P_{\mathcal{Y}}(C_1) = P_{\mathcal{Y}}(C_2) = \frac{1}{2}$$

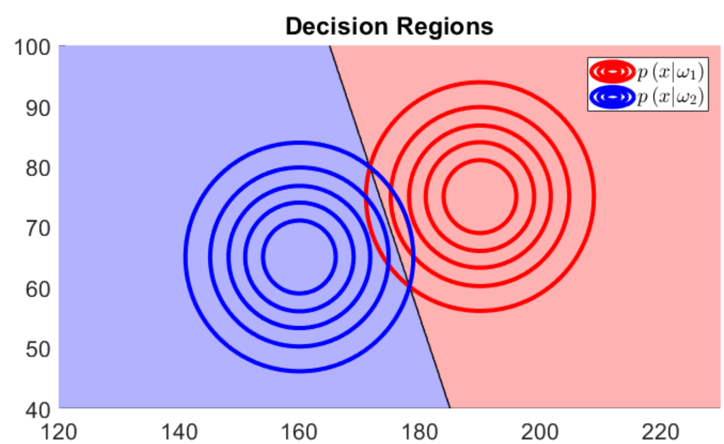
Given each state C_i, the observation X is a random Gaussian vector (with $\boldsymbol{\Sigma}_i$ and $\boldsymbol{\mu}_i$):

$$p_{X|\mathcal{Y}}(\boldsymbol{x}|C_i) = \frac{1}{2\pi}|\boldsymbol{\Sigma}_i|^{-1}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right)$$

Equal isotropic covariance

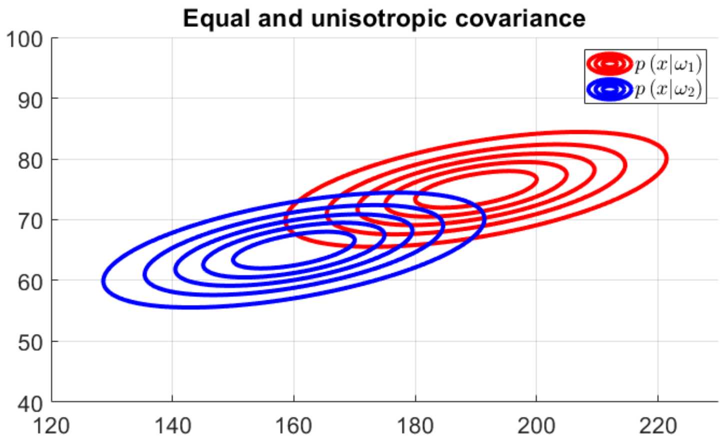
$$\begin{cases} \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{I} \\ \boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2 \end{cases}$$

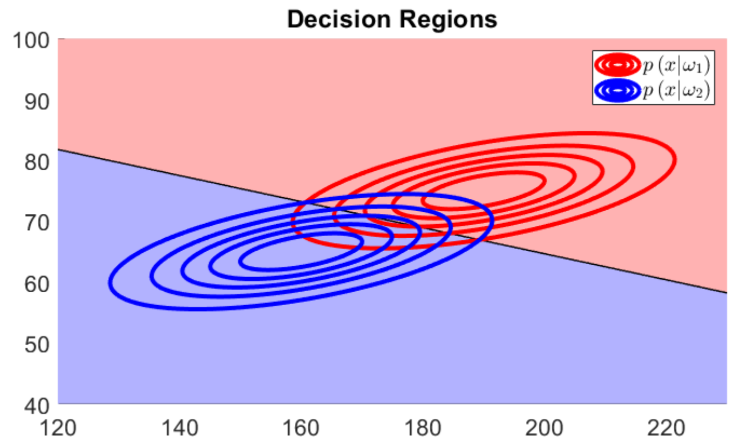




Equal covariances

$$\begin{cases} \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 \\ \boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2 \end{cases}$$

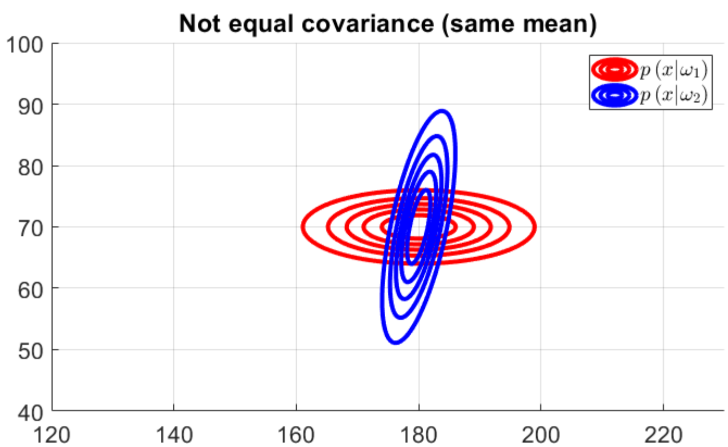


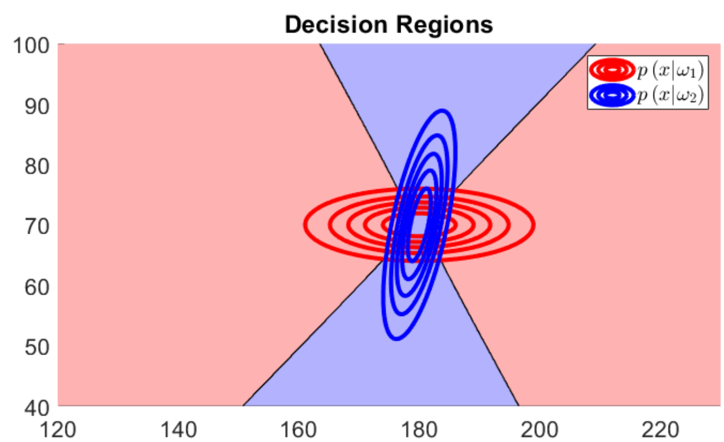


Notice that when $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2$ the decision boundary is linear (a single straight line).

Equal mean

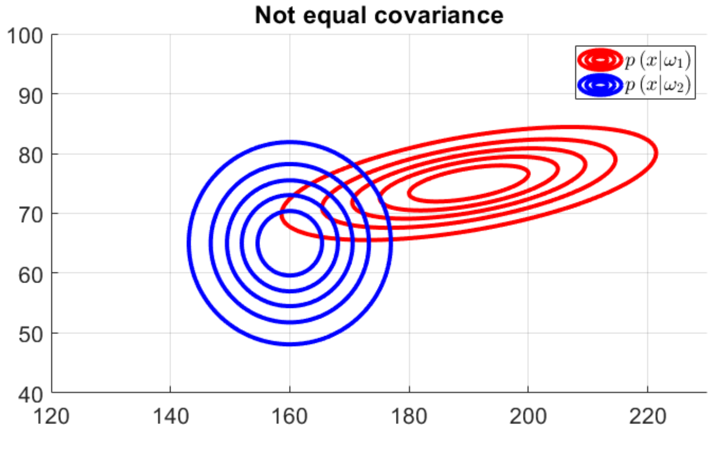
$$\begin{cases} \boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2 \\ \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 \end{cases}$$

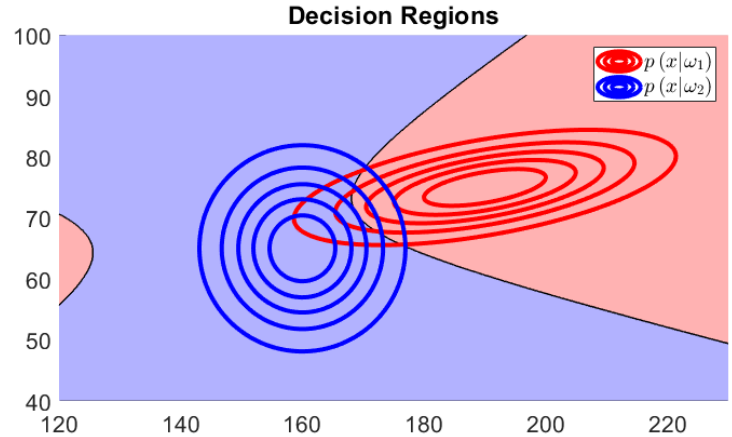




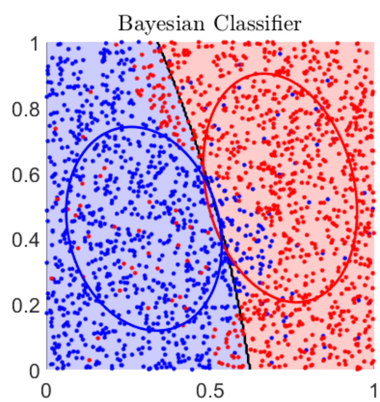
Non-equal covariance

$$\begin{cases} \boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2 \\ \boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2 \end{cases}$$

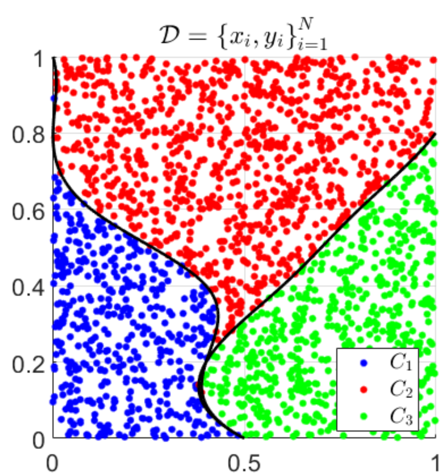


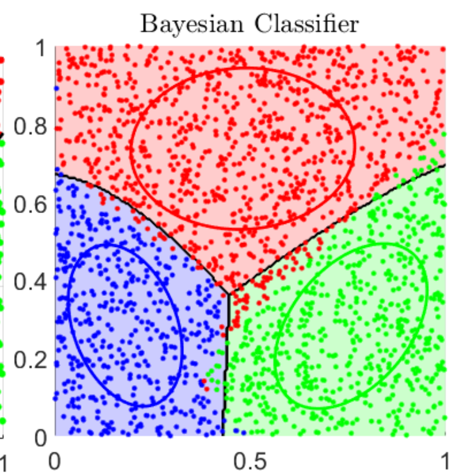


Binary case – with estimation

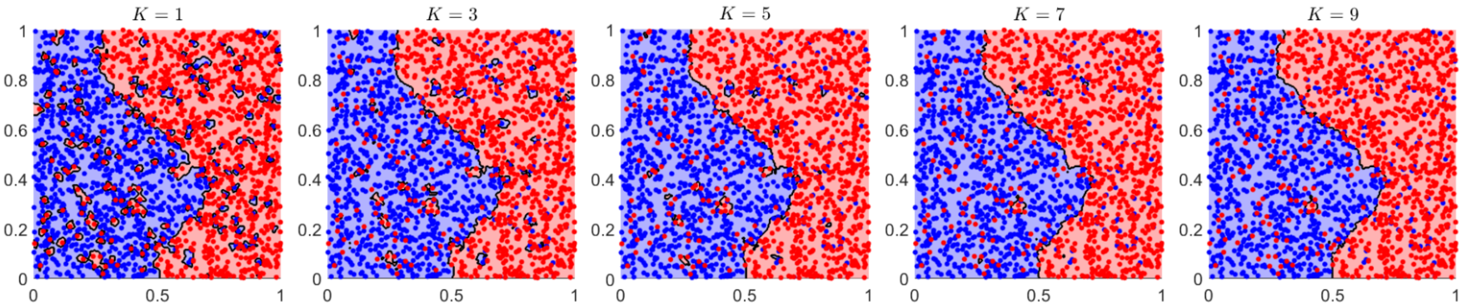


Trinary case – with estimation





5 K Nearest Neighbors (K-NN)



Let $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$ be a training set.

5.1 1-NN $(K = 1)$

The 1-NN classifier is given by:

$$\hat{y} = f_{NN}(\boldsymbol{x}) = y_{k(\boldsymbol{x})}$$

such that:

$$k(\boldsymbol{x}) = \arg \min_{i \in \{1,2,\ldots,N\}} d(\boldsymbol{x}, \boldsymbol{x}_i)$$

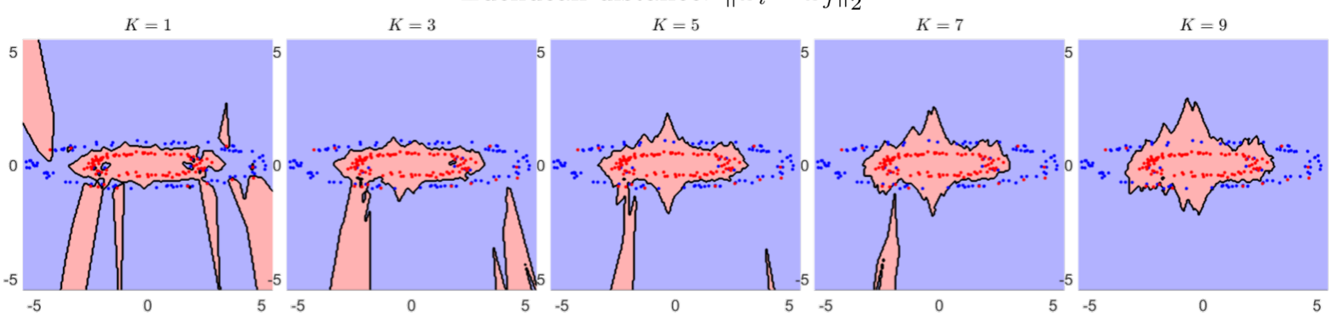
where $d : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$ is a metric function (for example: $d(\boldsymbol{x}, \boldsymbol{x}_i) = \|\boldsymbol{x} - \boldsymbol{x}_i\|_2$).
In words, a new sample $\boldsymbol{x}$ will labeled by the class of the nearest neighbor $\boldsymbol{x}_i \in \mathcal{D}$.

5.2 K-NN

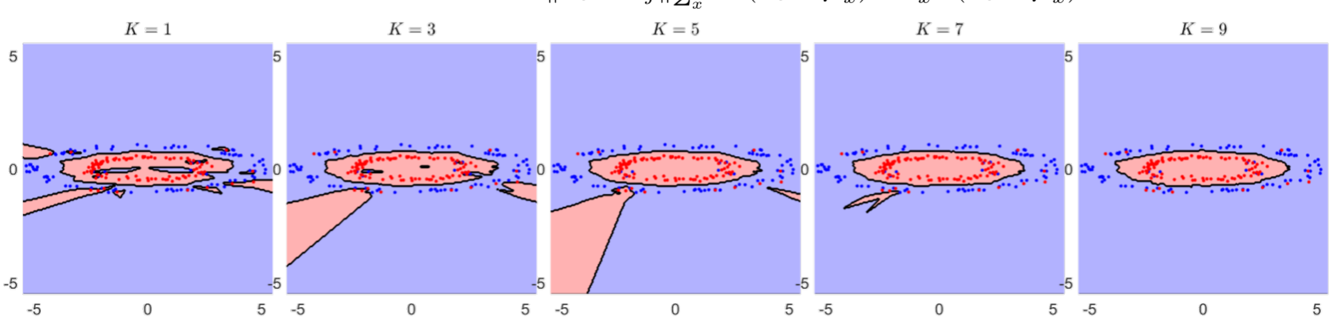
For $K > 1$, a new sample $\boldsymbol{x}$ will be labeled by the class most common among its K nearest neighbors.

K-NN with different metric (example)

Euclidean distance: $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$



Mahalanobis distance: $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_{\boldsymbol{\Sigma}_x} \triangleq (\boldsymbol{x}_i - \boldsymbol{\mu}_x)^T \boldsymbol{\Sigma}_x^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_x)$



5.3 Lower and Upper Performance Bounds (Extra)

Consider the case of 1-Nearest Neighbors.
The optimal MAP estimator is given by:

$$C_m \triangleq \arg \max_{C_i \in \{1,2,\ldots,K\}} P\left(C_i|\boldsymbol{x}\right), \qquad K\text{-number of classes}$$

The error of the MAP classifier is given by:

$$P^*\left(\text{error}|\boldsymbol{x}\right) = 1 - P\left(C_m|\boldsymbol{x}\right)$$

We denote the error of the 1-NN classifier by:

$$P\left(\text{error}|\boldsymbol{x}\right) = 1 - P\left(f_{NN}\left(\boldsymbol{x}\right)|\boldsymbol{x}\right)$$

Lemma 1. *In the limit $N \longrightarrow \infty$:*
(We also assume that p_X is continuous and non-zero everywhere)

$$P^*\left(error|\boldsymbol{x}\right) \leq P\left(error|\boldsymbol{x}\right) \leq 2P^*\left(error|\boldsymbol{x}\right)$$

Proof. Given a new point $\boldsymbol{x}$ with the (unknown) label $y \in \mathcal{Y} = \{C_1, C_2, \ldots, C_K\}$.
By the optimality of the MAP classifier:

$$\boxed{P^*\left(\text{error}|\boldsymbol{x}\right) \leq P\left(\text{error}|\boldsymbol{x}\right)}$$

Since $N \longrightarrow \infty$ and p_X is non-zeros everywhere there is some pair $(\boldsymbol{x}_0, y_0) \in \mathcal{D}$ such that $\boldsymbol{x}_0 = \boldsymbol{x}$.
Thus:

$$f_{NN}\left(\boldsymbol{x}\right) = y_0$$

Thus, the error is given by:

$$\begin{aligned}
P\left(\text{error}|\boldsymbol{x}\right) &= P\left(y \neq y_0|\boldsymbol{x}, \boldsymbol{x}_0\right) \\
&= P\left(y \neq y_0|\boldsymbol{x}\right) \\
&= 1 - \sum_{i=1}^{C} P\left(y = C_i, y_0 = C_i|\boldsymbol{x}\right) \\
&= 1 - \sum_{i=1}^{C} P_{\mathcal{Y}}^2\left(C_i|\boldsymbol{x}\right) \\
&\underset{(*)}{\leq} 2P^*\left(\text{error}|\boldsymbol{x}\right)
\end{aligned}$$

where $(*)$:

$$\begin{aligned}
\left(1 - P\left(C_i|\boldsymbol{x}\right)\right)^2 \geq 0 &\geq -\sum_{j \neq i} P^2\left(C_j|\boldsymbol{x}\right) \\
\Rightarrow \left(1 - P\left(C_i|\boldsymbol{x}\right)\right)^2 &\geq -\sum_{j \neq i} P^2\left(C_j|\boldsymbol{x}\right) \\
1 - 2P\left(C_i|\boldsymbol{x}\right) + P^2\left(C_i|\boldsymbol{x}\right) &\geq -\sum_{j \neq i} P^2\left(C_j|\boldsymbol{x}\right) \\
1 - 2P\left(C_i|\boldsymbol{x}\right) &\geq -\sum_{j} P^2\left(C_j|\boldsymbol{x}\right) \\
2 - 2P\left(C_i|\boldsymbol{x}\right) &\geq 1 - \sum_{j} P^2\left(C_j|\boldsymbol{x}\right) \\
2P^*\left(\text{error}|\boldsymbol{x}\right) &\geq 1 - \sum_{j} P^2\left(C_j|\boldsymbol{x}\right)
\end{aligned}$$

Overall:

$$\Rightarrow \boxed{P\left(\text{error}|\boldsymbol{x}\right) \leq 2P^*\left(\text{error}|\boldsymbol{x}\right)}$$

□