Rensselaer

why not change the world?®

Support Vector Machines (SVM) for classification

Ahmed Eleish

Data Analytics ITWS-4600/6600 CSCI-4960 MGMT 4600/6600

October 27th 2025

Tetherless World Constellation
Rensselaer Polytechnic Institute

Support Vector Machines

- Rationale
- Hyperplanes, Margins and Support vectors
- Classification using SVM
- Linear Separability of classes (or not)
- Soft Margin SVM
- Kernels

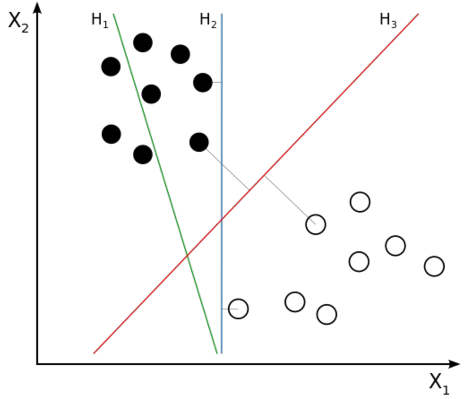
Rationale

- If data points in *p*-dimensional space, belonging to 2 different classes can be separated by a (*p*-1)-dimensional hyperplane, this hyperplane can be used as a linear classifier.

- Example: in 2d space, a line could be linear classifier..

- The hyperplane representing the largest separation or “margin” between the classes maximizes the distance to the nearest data point from each class.

- SVM utilizes the maximum margin hyperplane to solve classification, regression and outlier detection problems.

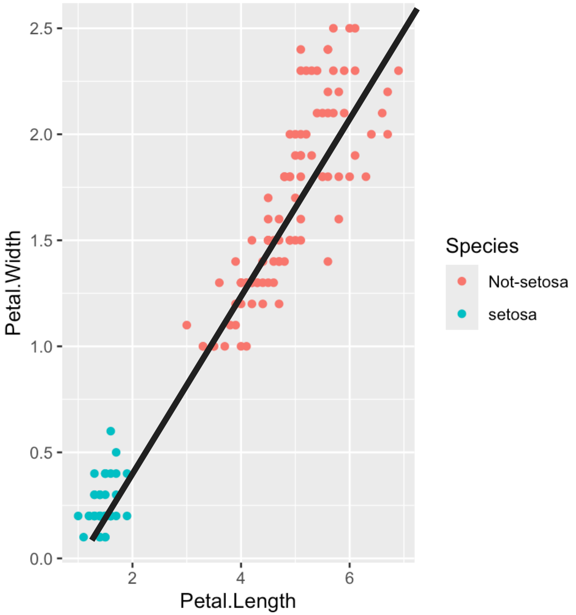


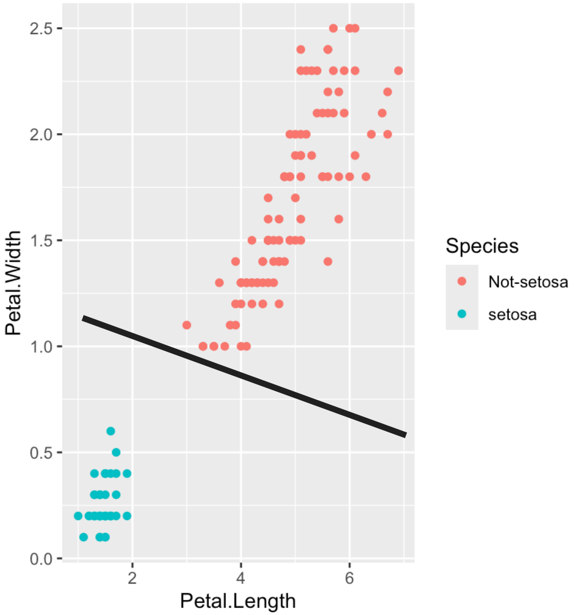
Hyperplane

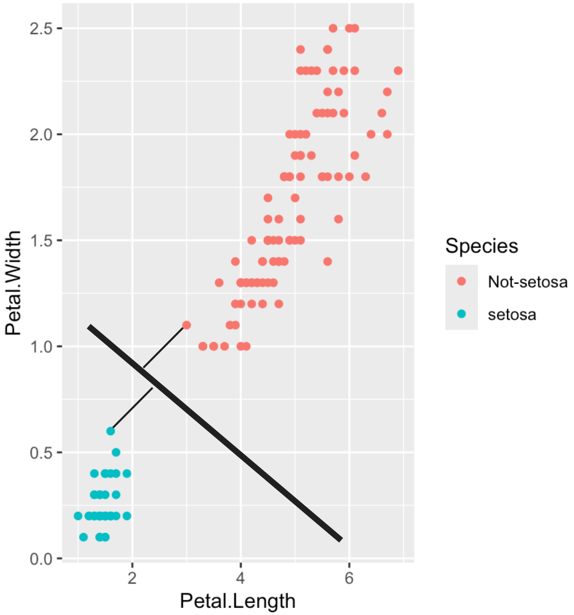
- A hyperplane is a plane of dimension *p-1* in a *p* dimensional space
- “a flat hypersurface, a subspace whose dimension is one less than that of the ambient space”
- “any codimension - 1 vector subspace of a vector space”

https://en.wikipedia.org/wiki/Hyperplane
https://mathworld.wolfram.com/Hyperplane.html

Hyperplanes

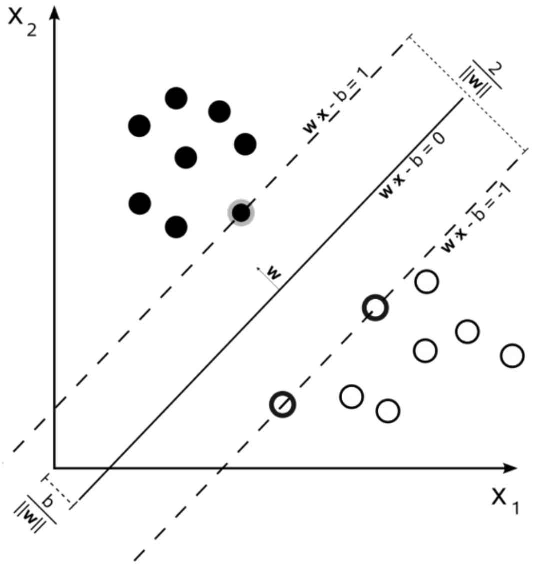






Margin

- The distance between the hyperplane (decision boundary) and the nearest points from each class.
- larger margin = greater confidence in the classifier
- SVMs find the hyperplane that maximizes the margin
  - “maximum-margin classifiers”



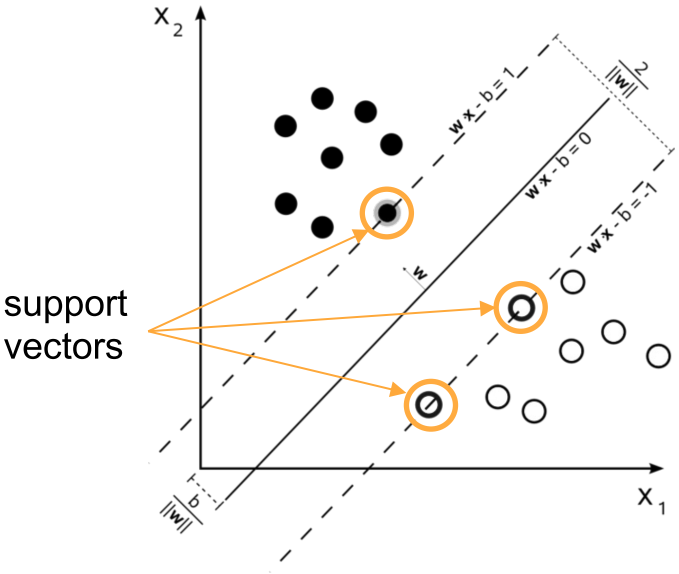
Support Vectors

- The points closest to the decision boundary.
- They determine the position and orientation of the hyperplane, i.e. define the decision boundary.
- They are used to calculate the margin.

Hyperplane:

$$W^T X - b = 0$$

W: weight vector
X: input vector
b: bias term



Support Vector Machines

- Given training dataset of points (x_i,y_i) where $\boldsymbol{y_i}$ is equal to **1** or **-1**

* Find the maximum-margin-hyperplane that divides the points x_i for which y_i = 1 from the points for which y_i = -1

Hyperplane:

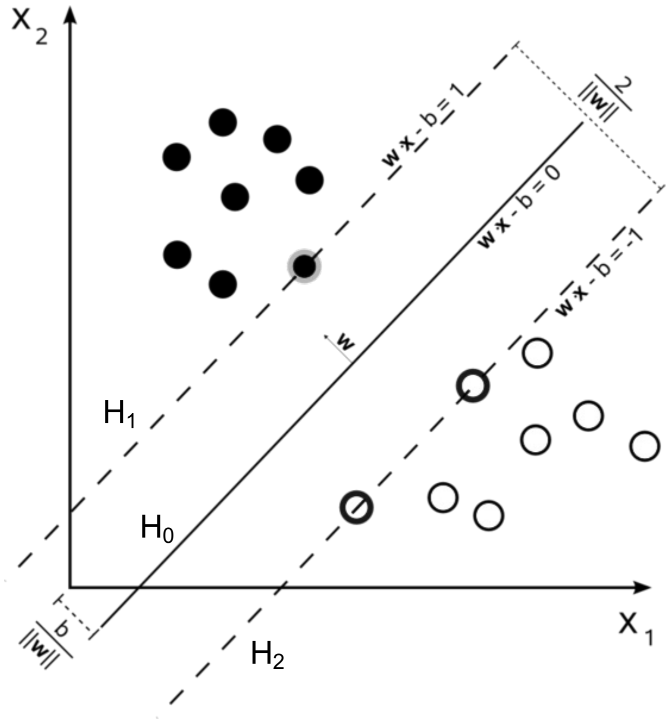
$$W^T X - b = 0$$

Distance from point to line:

$$d = \frac{|w \cdot x + b|}{\|w\|}$$

Distance from hyperplane H_1 to hyperplane H_2 :

$$\frac{2}{\|W\|}$$



Support Vector Machines

Distance from hyperplane H_1 to hyperplane H_2 :

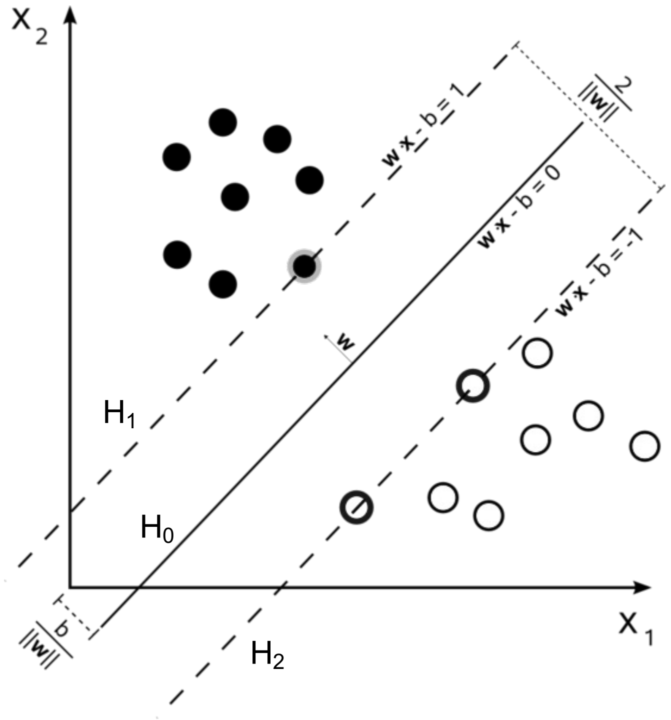
$$\frac{2}{\|W\|}$$

To find *W* and *b*:

$$\underset{\mathbf{w},\, b}{\text{minimize}} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{subject to} \quad y_i(\mathbf{w}^\top \mathbf{x}_i - b) \geq 1 \quad \forall i \in \{1, \dots, n\}$$

* The optimization problem is solved using gradient descent, quadratic programming, etc.

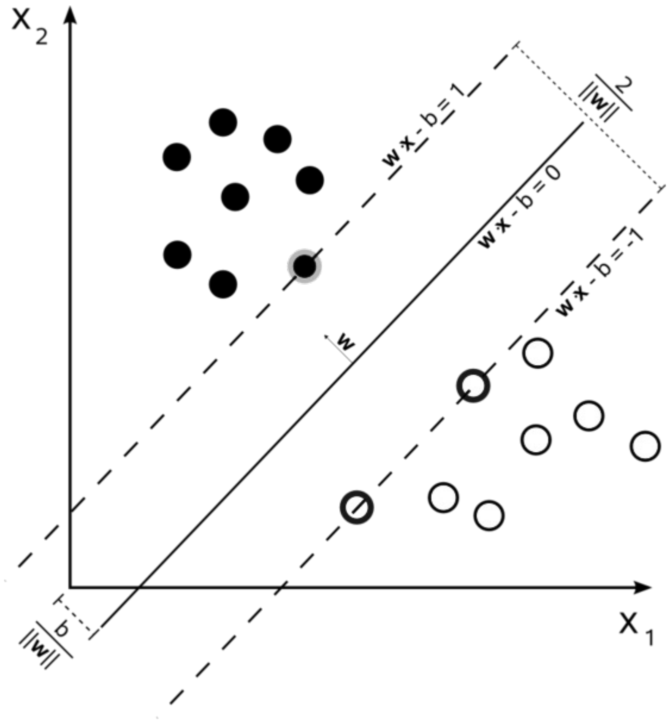


Classification with SVM

- Once the weights *W* and bias term *b* are found, classification is obtained by:

$$\mathrm{sign}(W^T X \ - b)$$

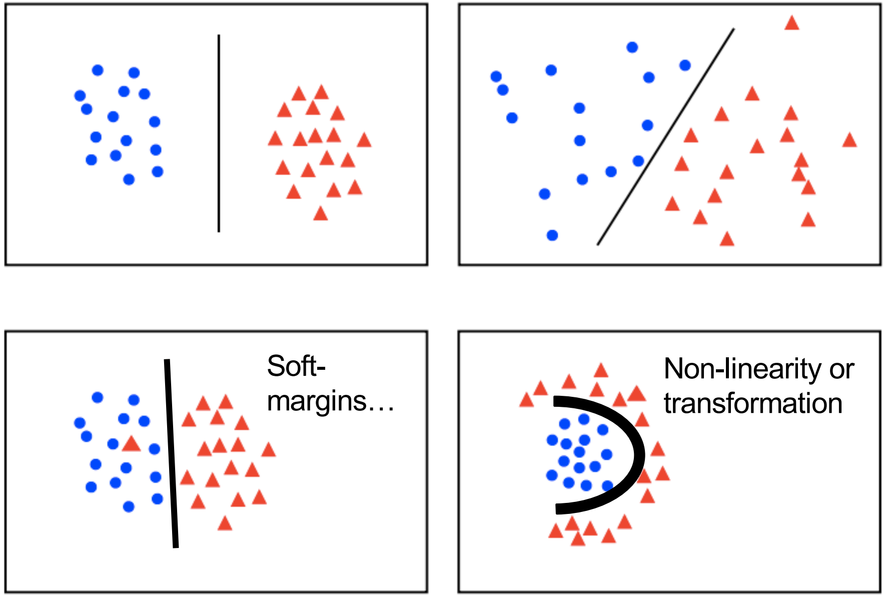
Where sign() is function that returns +1 or -1



Linear Separability

linearly
separable

not
linearly
separable



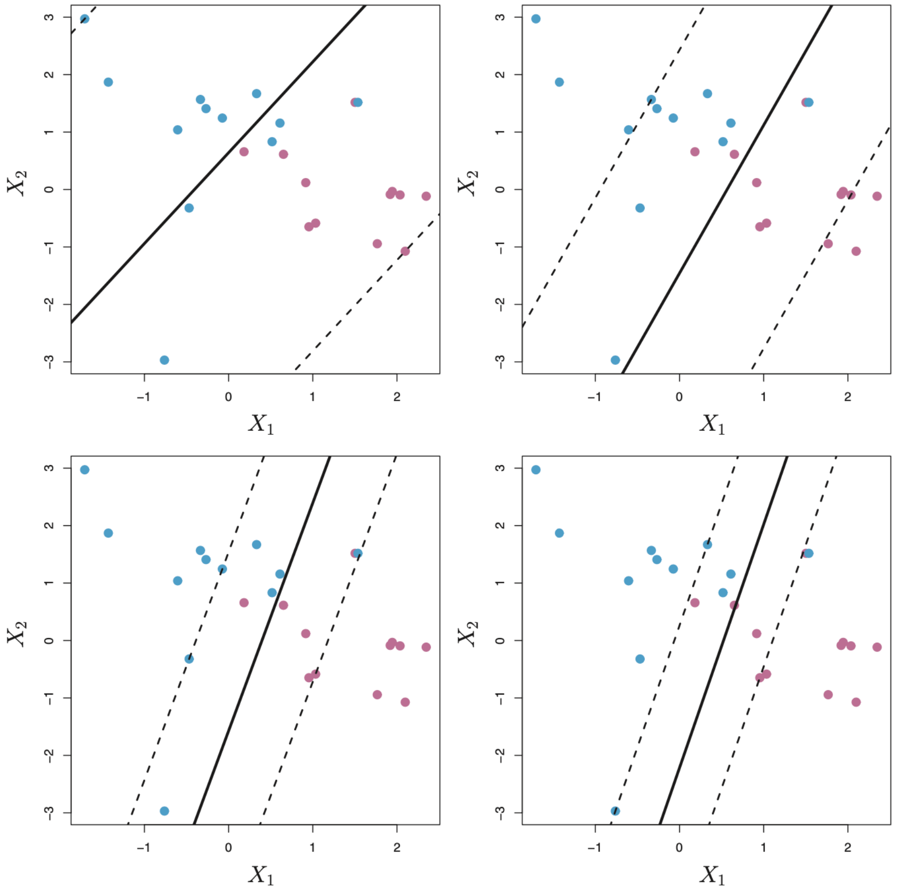
Soft-margin SVM

Allow for some margin violations controlled by the parameter *C*, the *regularization parameter*

$$\min_{w,b,z} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell} z_i$$

$$s.t \quad z_i \geq 1 - y_i\left(x_i \cdot w + b\right)$$

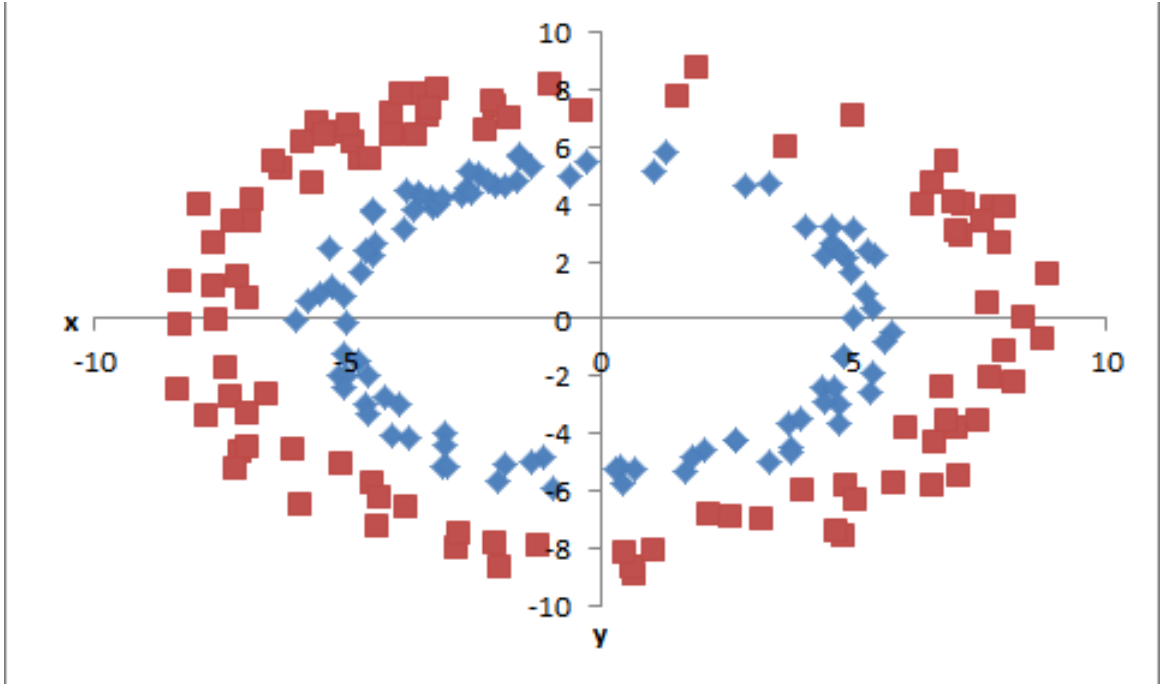
$$z_i \geq 0 \quad i = 1,.., N$$



Top left: Highest C value, decreasing C narrows the margin

Non-linearity

What to do??



Non-linearity (ideal example)

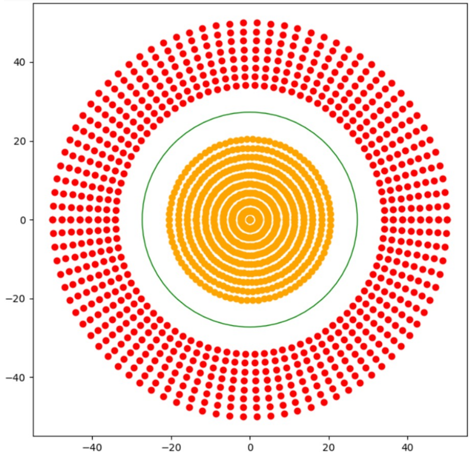
Transform the input:

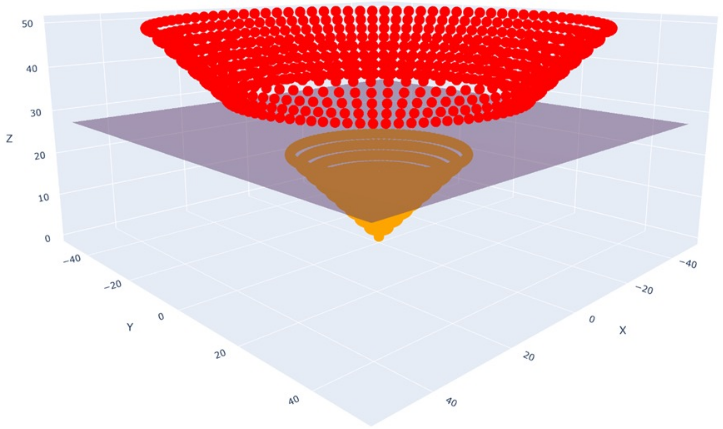
- Add a new dimension where the data are linearly separable

If are dataset contains variables X1, X2:
we can add $X3 = f(X1,X2)$

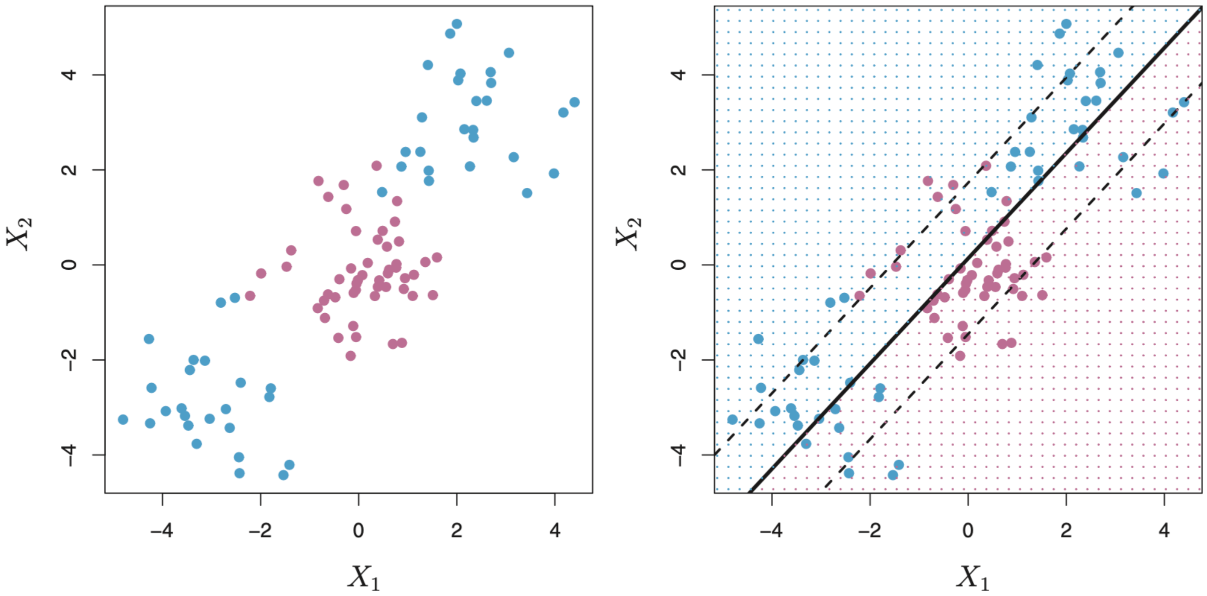
e.g. $X3 = (X1^2 + X2^2)^\wedge(1/2)$

- Computationally expensive!





Less ideally…



The Kernel Trick

- Instead of computing $\mathbf{w}^T\mathbf{x}$,we compute:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i \langle \mathbf{x}, \mathbf{x}_i \rangle + b$$

where $\langle \mathbf{x}, \mathbf{x}_i \rangle$ is the dot product of a new vector $\mathbf{x}$ and all training samples $\mathbf{x}_i$

- We replace the dot product with a kernel function:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b$$

Where $K(\mathbf{x}_i, \mathbf{x})$ is the kernel function and α_i is a weight coefficient

The Kernel Trick

- Kernel Functions:

1. Linear Kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$$

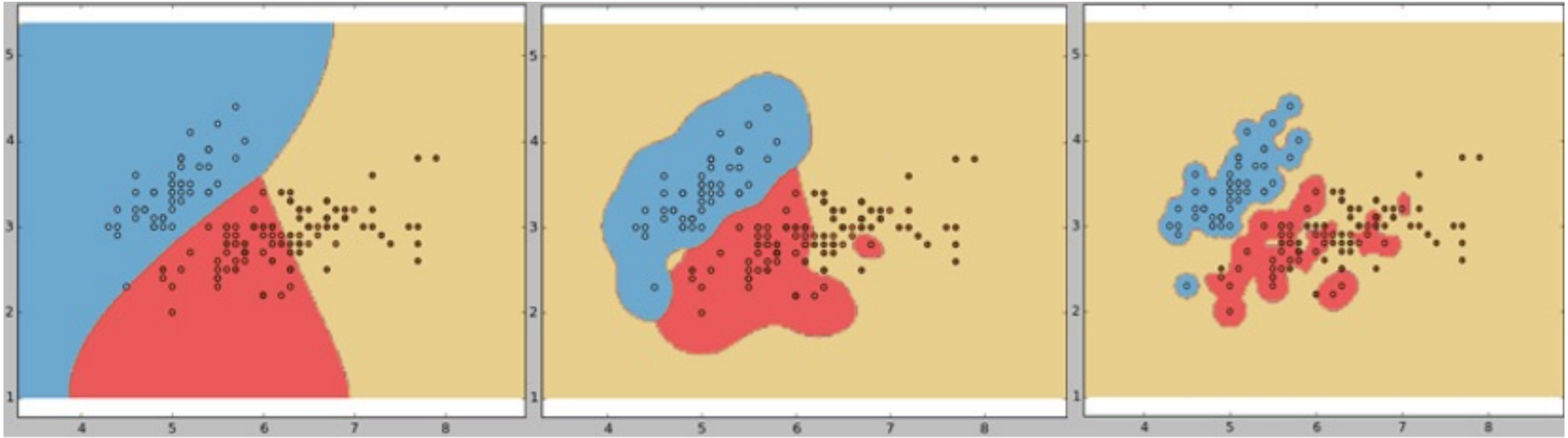
2. Polynomial Kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + c)^d$$

3. Radial Basis Function (RBF) Kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \mid\mid \mathbf{x}_i - \mathbf{x}_j \mid\mid^2)$$

Parameter Gamma (γ) in RBF

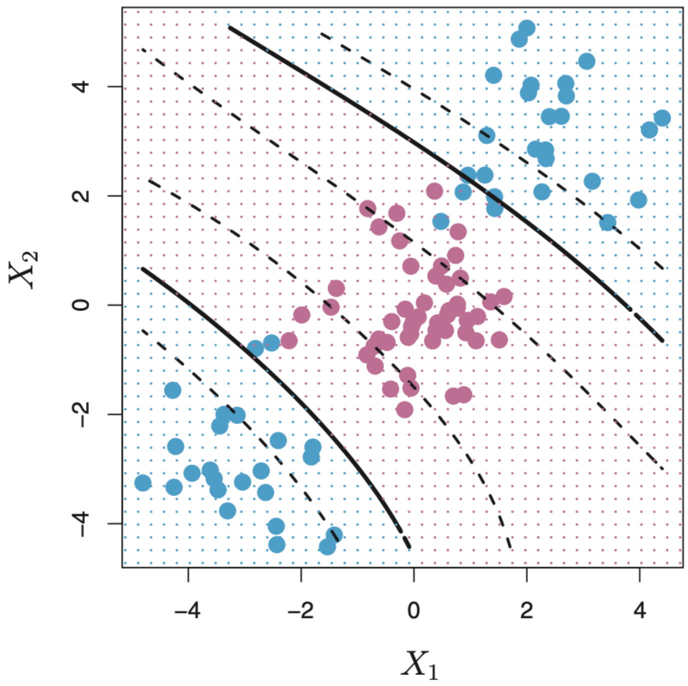


γ = 0.1 γ = 10 γ = 100

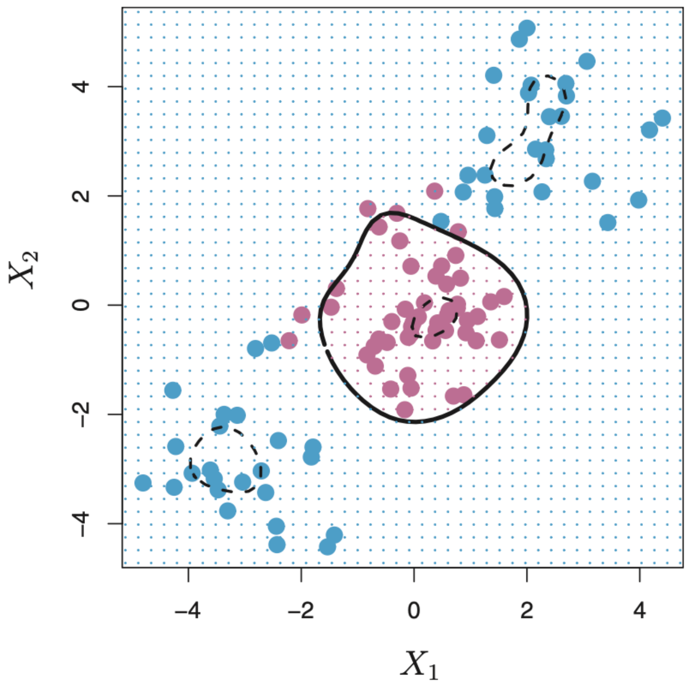
Many Kernels

- Polynomial Kernel
- Gaussian Kernel
- Gaussian RBF Kernel
- Laplace RBF Kernel
- Hyperbolic Tangent Kernel
- Sigmoid Kernel
- Bessel function of first kind Kernel
- ANOVA radial basis Kernel
- Linear Splines Kernel

Applying Kernels



Polynomial Kernel



Radial Kernel

In-class exercise

https://rpi.box.com/s/mgyeuj7ncv3rmxfy74n0yzc57ctiy9or

Thanks!