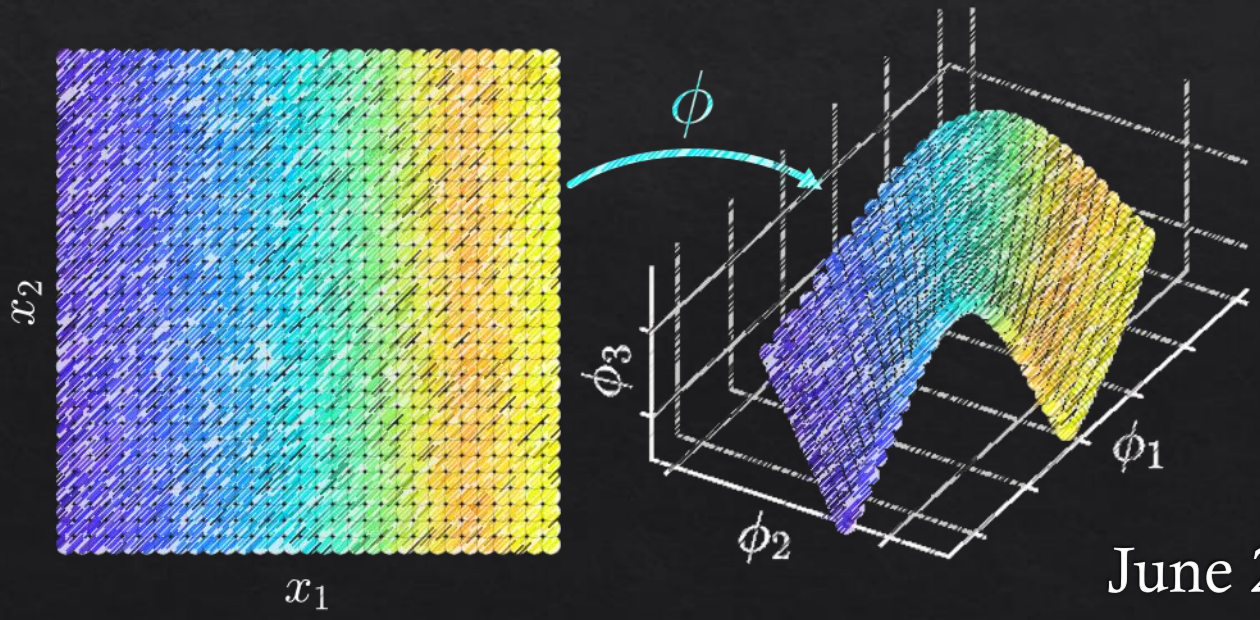
Supervised Learning – Nonlinear Classification

Machine Learning Methods – Lecture 5



June 2021

fixel your pixels

fixelalgorithms.gitlab.io

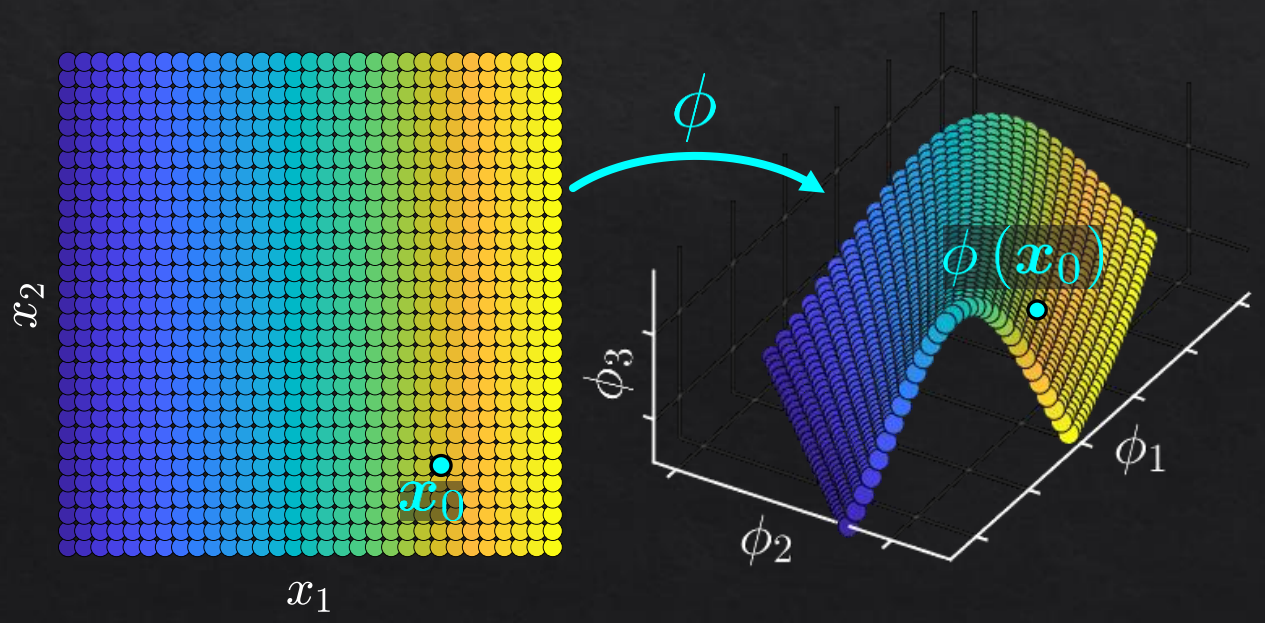
Or Yair

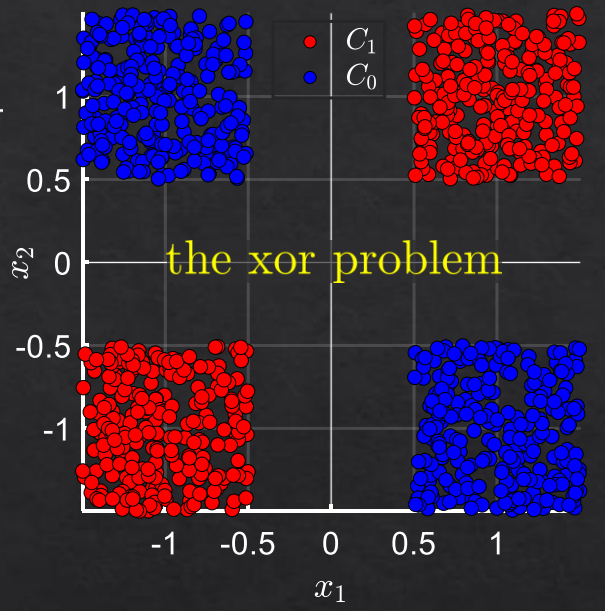
Linearly Non-separable Problems

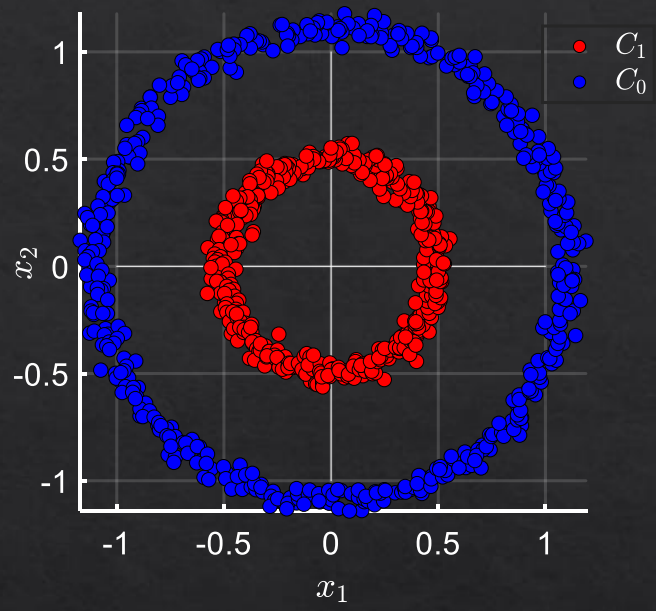
- Consider the following classification problems:
- These problems are linearly non-separable in the space of (x_1, x_2).

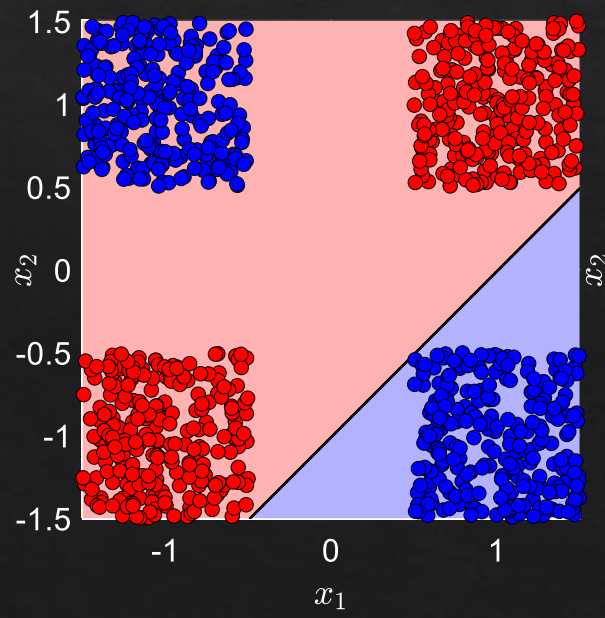
Can we find a mapping $(x_1, x_2) \mapsto \phi(x_1, x_2)$ such that non separable problem will turn into a linearly separable problem in the new space?

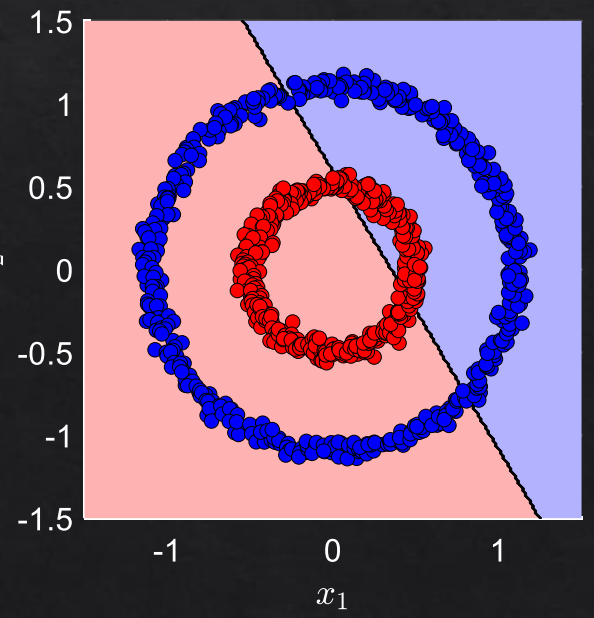
Yes! (and always!)







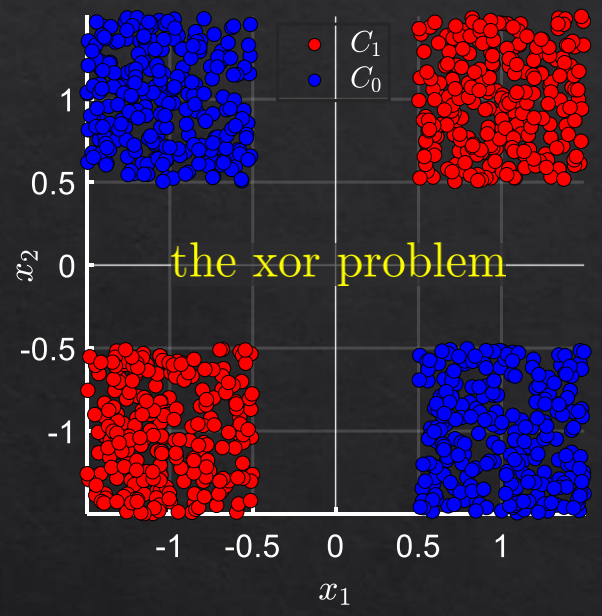


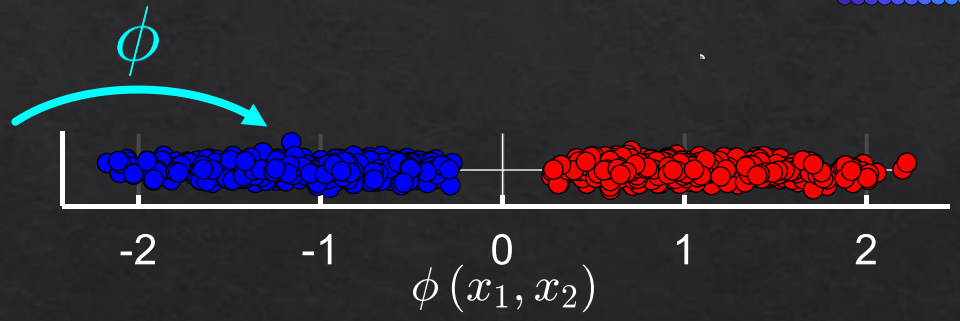


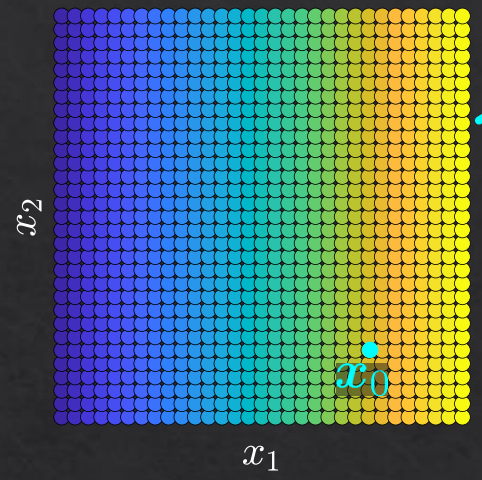
Feature Transform

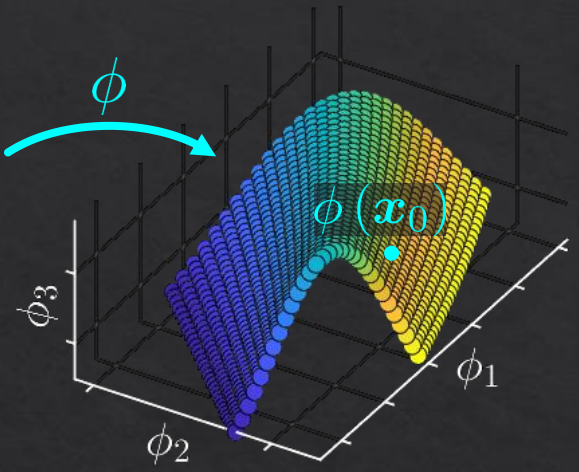
- Consider the following transformation:

$$\phi(x_1, x_2) = x_1 \cdot x_2$$







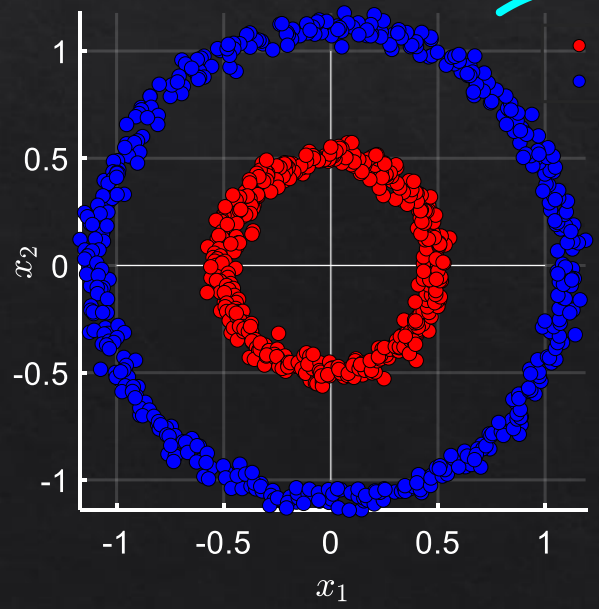


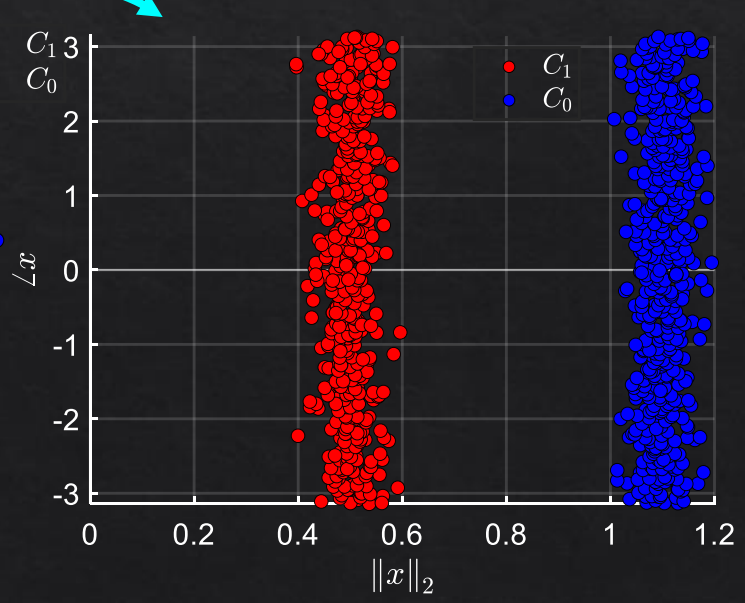
- Let:

$$\psi(x_1, x_2) = \left(\underbrace{\sqrt{x_1^2 + x_2^2}}_{\text{radius}}, \underbrace{\arctan\left(\frac{x_2}{x_1}\right)}_{\text{angle}} \right)$$

radius $\|\boldsymbol{x}\|_2$

angle $\angle \boldsymbol{x}$

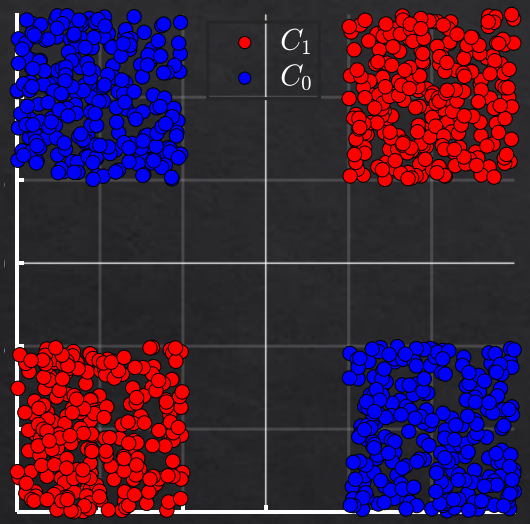




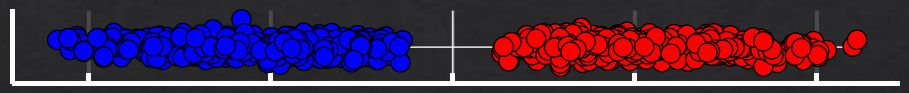
Feature Transform

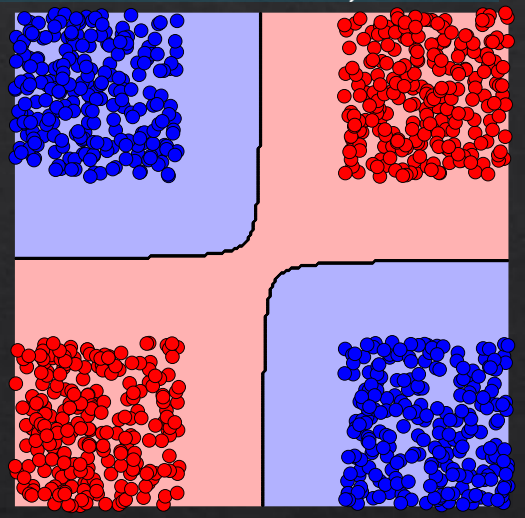
- Instead of training a linear classifier in the original domain: $y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i - b \right) > 0$

  We can train a linear classifier in the new domain: $y_i \left(\tilde{\boldsymbol{w}}^T \phi \left(\boldsymbol{x}_i \right) - \tilde{b} \right) > 0$

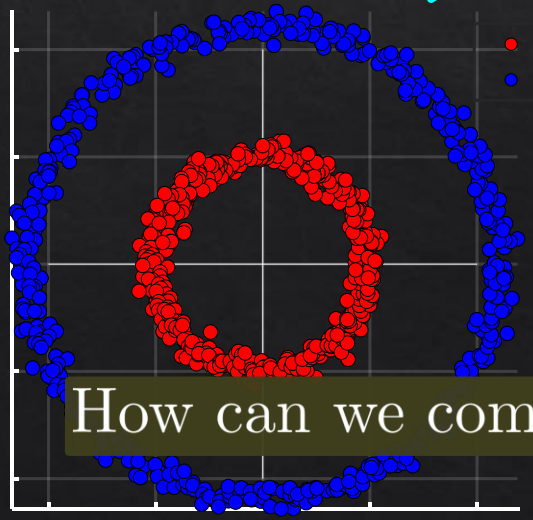


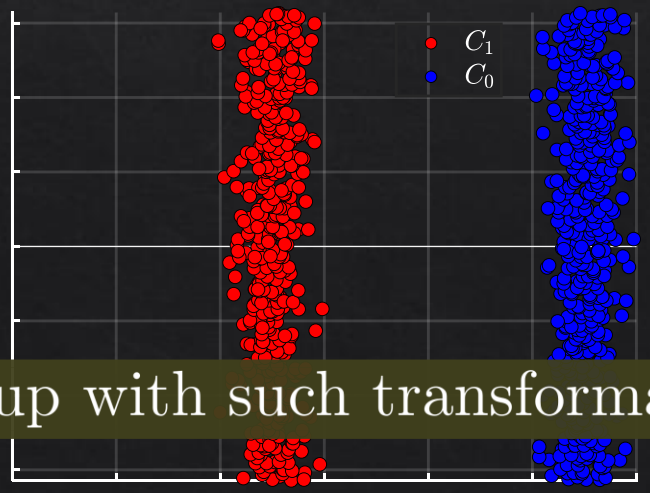
$$\phi \left(x_1, x_2 \right) = x_1 \cdot x_2$$

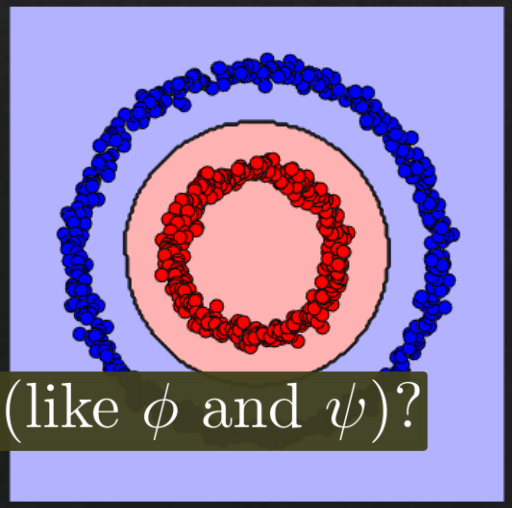




$$\psi \left(\boldsymbol{x} \right) = \begin{bmatrix} \| x \|_2 \\ \angle x \end{bmatrix}$$

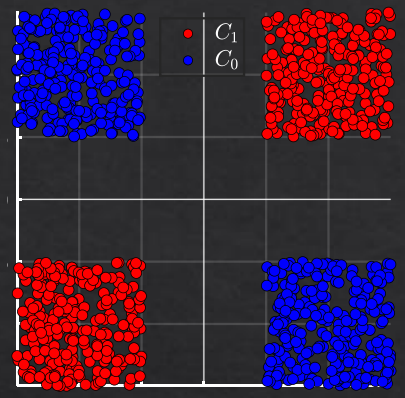




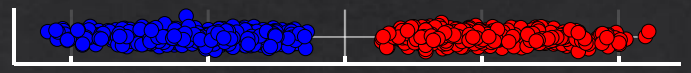


How can we come up with such transformations (like ϕ and ψ)?

The Kernel Trick – Outline



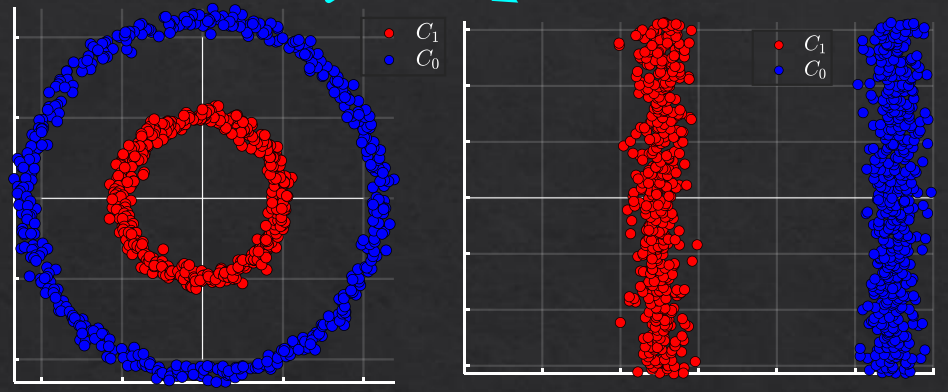
$$\phi(x_1, x_2) = x_1 \cdot x_2$$



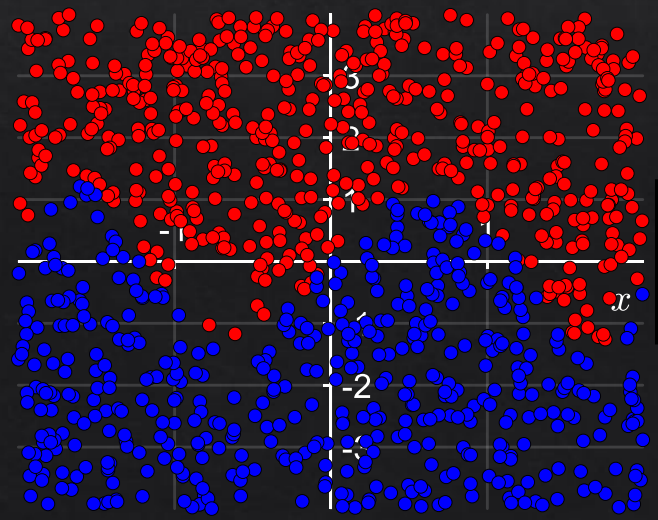
Feature transform:

$$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \longrightarrow \{(\boldsymbol{\phi}(\boldsymbol{x}_i), y_i)\}_{i=1}^N$$

$$\boldsymbol{\psi}(\boldsymbol{x}) = \begin{bmatrix} \|x\|_2 \\ \angle x \end{bmatrix}$$

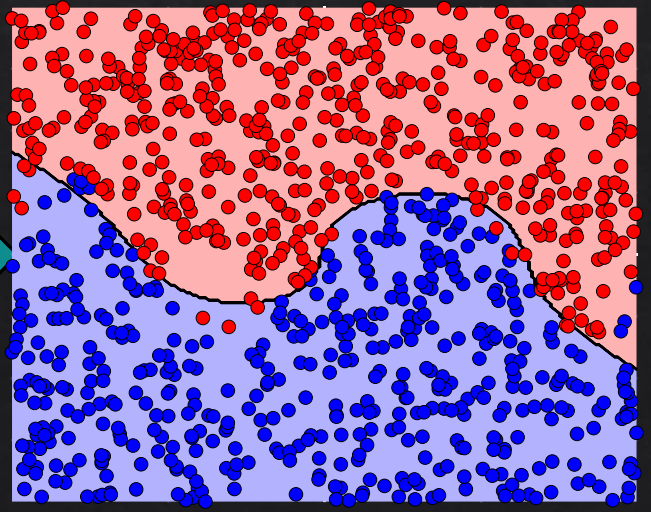


- To obtained a useful transformation, we now review:
  - Optimization dual problem
  - SVM dual problem
  - Kernel functions
  - The kernel trick



linear classification

$$\boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(1 + \boldsymbol{x}_i^T \boldsymbol{x}_j\right)^3$$



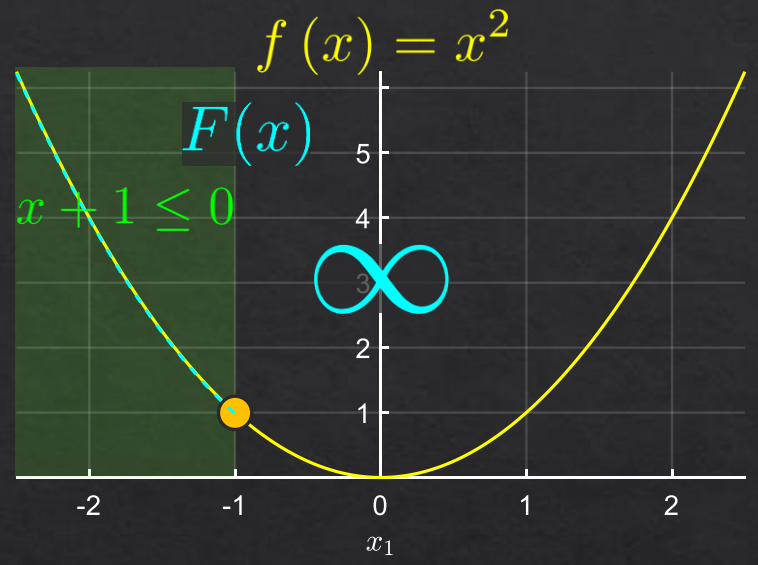
The Dual Problem – Intuition

- Consider the following constraint optimization problem:

$$\begin{cases} \min\limits_{x\in\mathbb{R}} x^2 \\ \text{subject to} \\ x+1\leq 0 \end{cases}$$

same global minimum

$$F(x) := \max_{\lambda\geq 0} x^2 + \lambda(x+1) = \begin{cases} x^2 & (x+1)\leq 0 \\ \infty & (x+1)\geq 0 \end{cases}$$



- We can formulate this (primal) problem as follows:

$$\min_{x\in\mathbb{R}} F(x) = \min_{x\in\mathbb{R}} \max_{\lambda\geq 0} \underbrace{x^2 + \lambda(x+1)}_{=:\, L(x,\lambda)}$$

- The dual problem is defined by:

$$\max_{\lambda\geq 0} \min_{x\in\mathbb{R}} L(x,\lambda) = \max_{\lambda\geq 0} L(x^\star,\lambda) = \underbrace{\max_{\lambda\geq 0} -\frac{\lambda^2}{4} + \lambda}_{\lambda^\star = 2}$$

$$\frac{\partial L(x^\star)}{\partial x} = 0$$

$$2x^\star + \lambda = 0$$

$$x^\star = -\frac{\lambda}{2}$$

$$\implies x^\star = -1$$

$$\underbrace{\min_{x\in\mathbb{R}} \max_{\lambda\geq 0} L(x,\lambda)}_{\text{primal}} = \underbrace{\max_{\lambda\geq 0} \min_{x\in\mathbb{R}} L(x,\lambda)}_{\text{dual}}$$

strong duality (under some conditions)
we can solve the dual instead of the primal

SVM – The Dual Problem

$$\min_{x\in\mathbb{R}} \max_{\lambda\geq 0} L(x,\lambda) = \max_{\lambda\geq 0} \min_{x\in\mathbb{R}} L(x,\lambda)$$

The primal SVM problem:

$$\begin{cases} \min\limits_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_i \xi_i \\ \xi_i = \max\{0, 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i - b)\} \end{cases}$$

The dual SVM problem:

$$\begin{cases} \max\limits_{\{\alpha_i\}} \sum_i \alpha_i - \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle \\ \text{subject to} \\ 0 \leq \alpha_i \leq C \\ \sum_i \alpha_i y_i = 0 \end{cases}$$

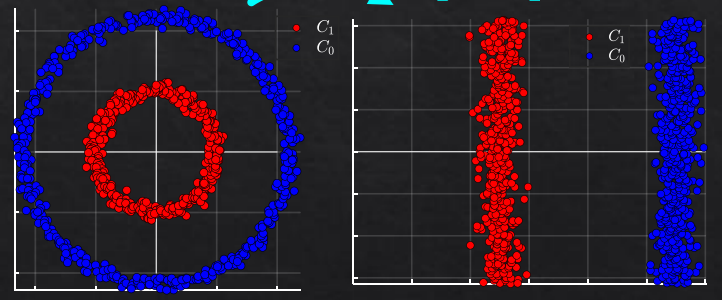
- Notice that the dual problem requires only the inner products $\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle$.

Feature transform:

$$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \longrightarrow \{(\boldsymbol{\phi}(\boldsymbol{x}_i), y_i)\}_{i=1}^N$$

$$\begin{cases} \max\limits_{\{\alpha_i\}} \sum_i \alpha_i - \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)\rangle \\ \text{subject to} \\ 0 \leq \alpha_i \leq C \\ \sum_i \alpha_i y_i = 0 \end{cases}$$

$$\psi(\boldsymbol{x}) = \begin{bmatrix} \|x\|_2 \\ \angle x \end{bmatrix}$$



using kernel functions we can compute the inner product $\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)\rangle$ without knowing $\phi(\boldsymbol{x}_i)$ and $\phi(\boldsymbol{x}_j)$

Kernel Function – Example

- Let $x_i, x_j \in \mathbb{R}$:
- Consider the following kernel:

$$K(x_i, x_j) = (1 + x_i x_j)^2$$

$$= 1 + 2x_i x_j + x_i^2 x_j^2$$

$$= \left\langle \begin{bmatrix} 1 \\ \sqrt{2}x_i \\ x_i^2 \end{bmatrix}, \begin{bmatrix} 1 \\ \sqrt{2}x_j \\ x_j^2 \end{bmatrix} \right\rangle$$

$$\implies K(x_i, x_j) = \langle \boldsymbol{\phi}(x_i), \boldsymbol{\phi}(x_j) \rangle$$

$$\begin{cases} \max\limits_{\{\alpha_i\}} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle \\ \text{subject to} \\ 0 \le \alpha_i \le C \\ \sum_i \alpha_i y_i = 0 \end{cases}$$

using kernel functions we can compute $\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$ without knowing $\phi(\boldsymbol{x}_i)$ and $\phi(\boldsymbol{x}_j)$

$$\boldsymbol{\phi}(x) := \begin{bmatrix} 1 \\ \sqrt{2}x \\ x^2 \end{bmatrix} \in \mathbb{R}^3$$

- $K(x_i, x_j)$ is a kernel because we can write it as an inner product:

  $K(x_i, x_j) = \langle \boldsymbol{\phi}(x_i), \boldsymbol{\phi}(x_j) \rangle$ for some (possibly unknown) $\boldsymbol{\phi}$.

Kernel Function

$$\begin{cases} \max\limits_{\{\alpha_i\}} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \underbrace{\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle}_{K(\boldsymbol{x}_i, \boldsymbol{x}_j)} \\ \text{subject to} \\ 0 \le \alpha_i \le C \\ \sum_i \alpha_i y_i = 0 \end{cases}$$

- A kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ satisfies:
  1. Symmetry: $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = K(\boldsymbol{x}_j, \boldsymbol{x}_i)$.
  2. K is positive (semi) definite.
- A kernel function K can be written as:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{m=1}^{M} \phi_m(\boldsymbol{x}_i) \phi_m(\boldsymbol{x}_j) = \langle \boldsymbol{\phi}(\boldsymbol{x}_i), \boldsymbol{\phi}(\boldsymbol{x}_j) \rangle$$

$$\boldsymbol{\phi}(\boldsymbol{x}_i) = \begin{bmatrix} \phi_1(\boldsymbol{x}_i) \\ \phi_2(\boldsymbol{x}_i) \\ \vdots \\ \phi_M(\boldsymbol{x}_i) \end{bmatrix}$$

M might be infinity

Useful kernels

- Polynomial kernel ($d \ge 1$):

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (1 + \boldsymbol{x}_i^T \boldsymbol{x}_j)^d$$

- Gaussian kernel ($\sigma > 0$):

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right)$$

- Laplacian kernel ($\sigma > 0$):

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\sigma}\right)$$

- Sigmoid kernel:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh\left(\frac{1}{d} \boldsymbol{x}_i^T \boldsymbol{x}_j + c\right)$$

The Kernel Trick – Example

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

- As a simple example, let $x_i, x_j \in \mathbb{R}$.
- Consider the following kernel:

$$\begin{aligned} K(x_i, x_j) &= (1 + x_i x_j)^2 \\ &= 1 + 2x_i x_j + x_i^2 x_j^2 \\ &= \left\langle \begin{bmatrix} 1 \\ \sqrt{2}x_i \\ x_i^2 \end{bmatrix}, \begin{bmatrix} 1 \\ \sqrt{2}x_j \\ x_j^2 \end{bmatrix} \right\rangle \end{aligned}$$

$$\phi(x) := \begin{bmatrix} 1 \\ \sqrt{2}x \\ x^2 \end{bmatrix} \in \mathbb{R}^3$$

$$\implies K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

- The Gaussian kernel:

$$K(x_i, x_j) = \exp\left(-\frac{(x_i - x_j)^2}{2\sigma^2}\right)$$

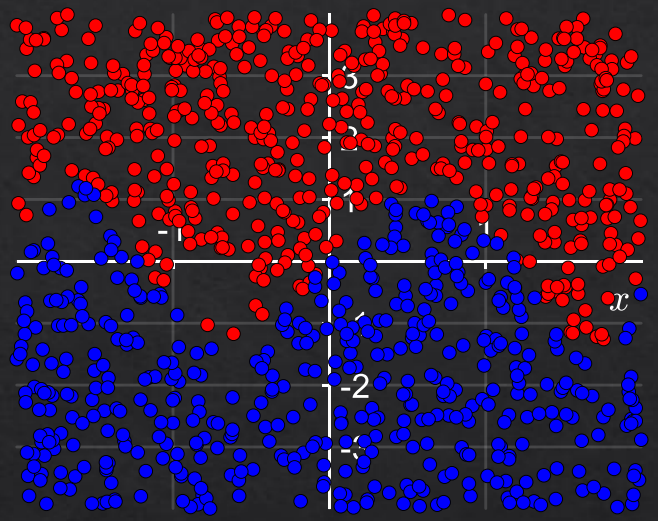
$$\phi(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right) \cdot \begin{bmatrix} 1 \\ \frac{x}{\sigma} \\ \frac{1}{\sqrt{2!}}\frac{x^2}{\sigma^2} \\ \frac{1}{\sqrt{3!}}\frac{x^3}{\sigma^3} \\ \vdots \end{bmatrix} \in \mathbb{R}^\infty$$

The Kernel Trick – Polynomial Kernel

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

SVM dual problem:

$$\begin{cases} \max\limits_{\{\alpha_i\}} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{subject to} \\ 0 \le \alpha_i \le C \\ \sum_i \alpha_i y_i = 0 \end{cases}$$



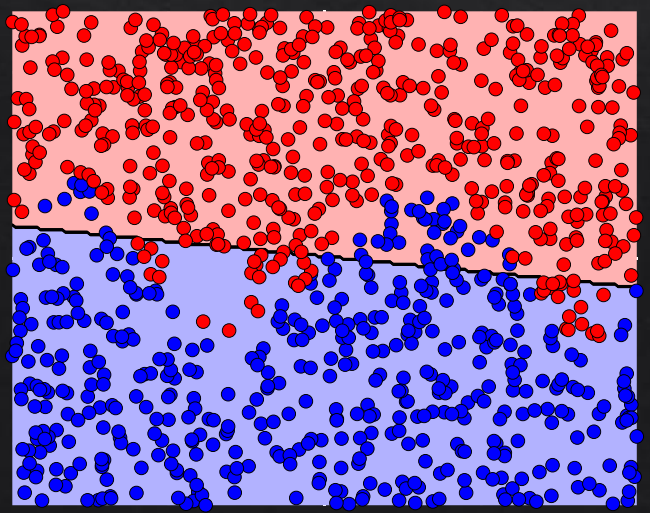
- Polynomial kernel ($d \ge 1$):

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(1 + \boldsymbol{x}_i^T \boldsymbol{x}_j\right)^d$$

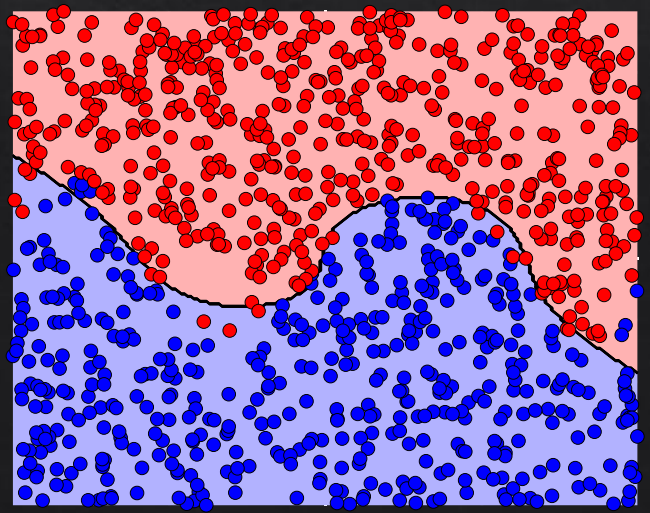
- Gaussian kernel ($\sigma > 0$):

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right)$$

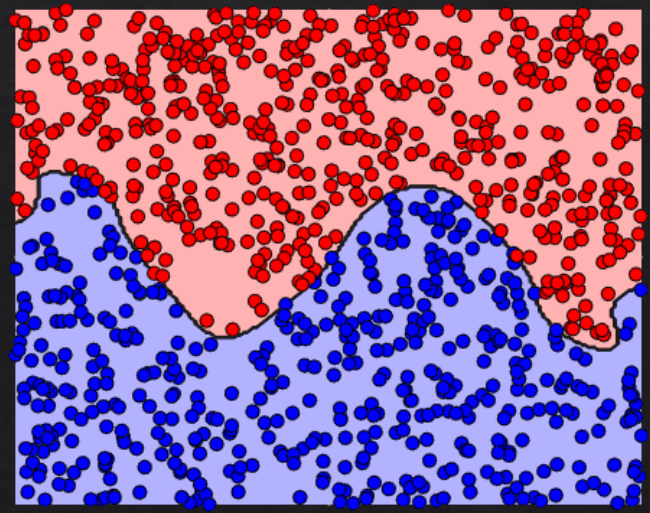
$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(1 + \boldsymbol{x}_i^T \boldsymbol{x}_j\right)^1$



$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(1 + \boldsymbol{x}_i^T \boldsymbol{x}_j\right)^3$



$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(1 + \boldsymbol{x}_i^T \boldsymbol{x}_j\right)^5$



Kernel SVM

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

- Polynomial kernel ($d \geq 1$):

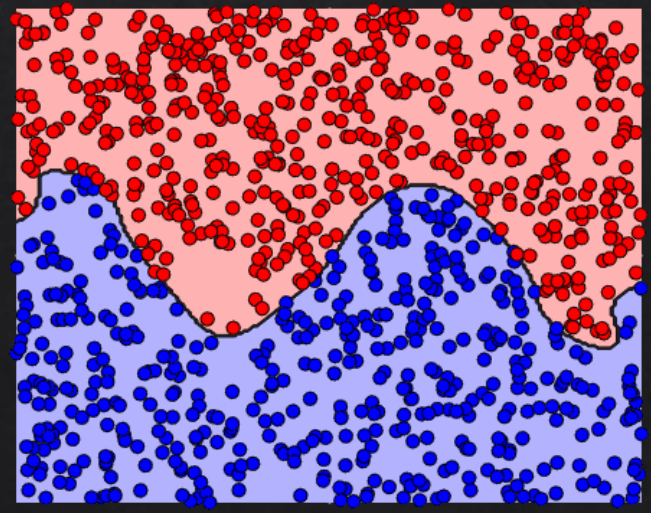
$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(1 + \boldsymbol{x}_i^T \boldsymbol{x}_j\right)^d$$

- Gaussian kernel ($\sigma > 0$):

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right)$$

SVM with a Gaussian Kernel

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(1 + \boldsymbol{x}_i^T \boldsymbol{x}_j\right)^5$$



The Kernel Trick – Gaussian Kernel

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

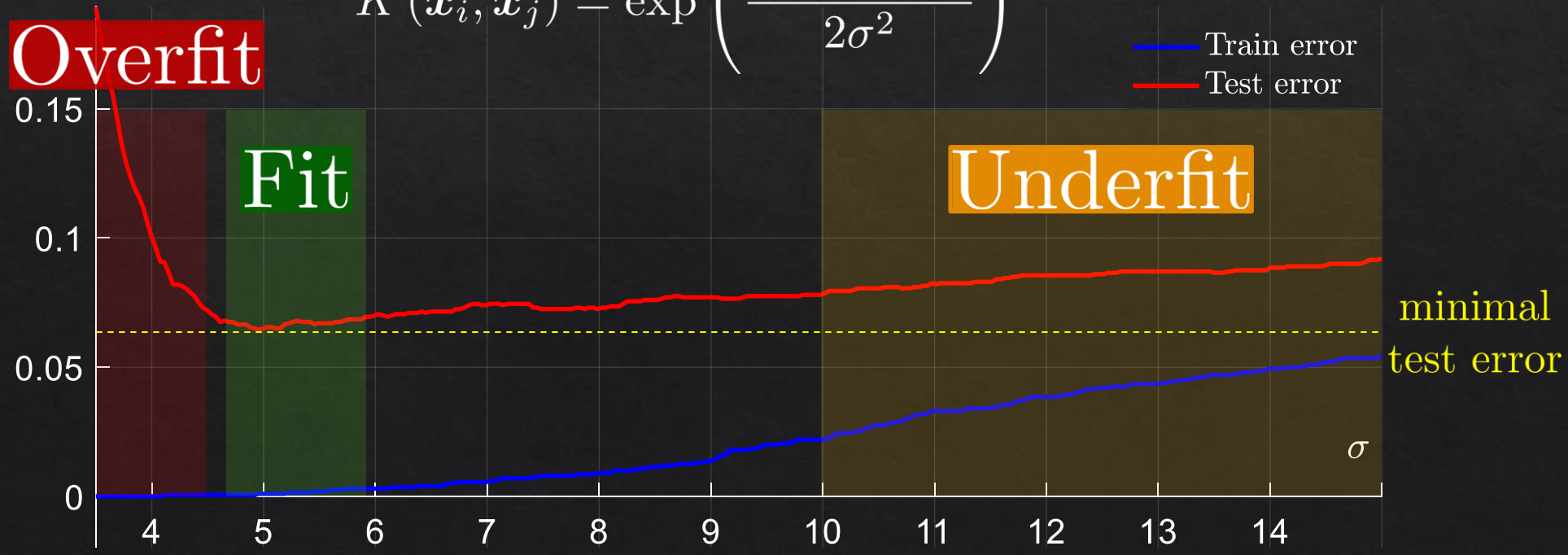
- Polynomial kernel ($d \geq 1$):

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(1 + \boldsymbol{x}_i^T \boldsymbol{x}_j\right)^d$$

- Gaussian kernel ($\sigma > 0$):

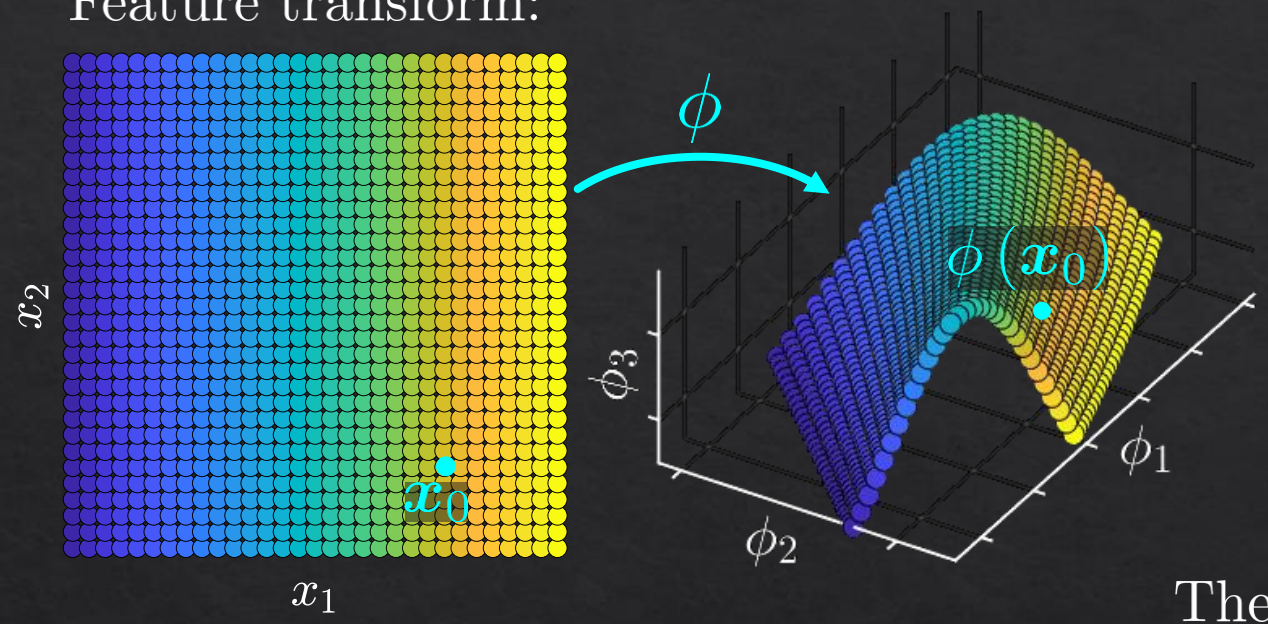
$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right)$$

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right)$$

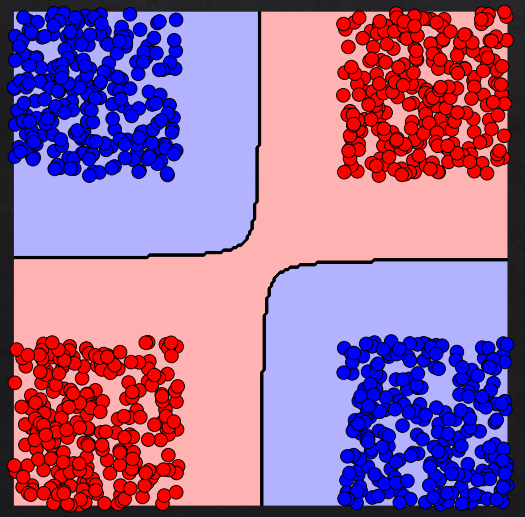


Questions

Feature transform:

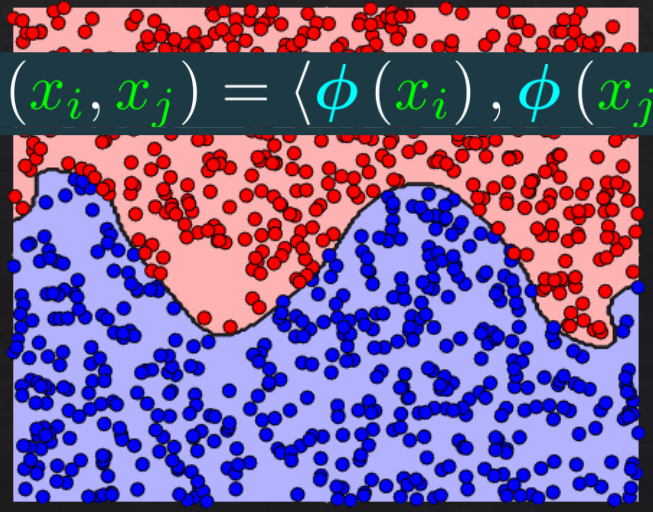


Non-linear classification:



The kernel trick:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (1 + \boldsymbol{x}_i^T \boldsymbol{x}_j)^5$$



$$K(x_i, x_j) = \langle \boldsymbol{\phi}(x_i), \boldsymbol{\phi}(x_j) \rangle$$

fixelalgorithms.gitlab.io