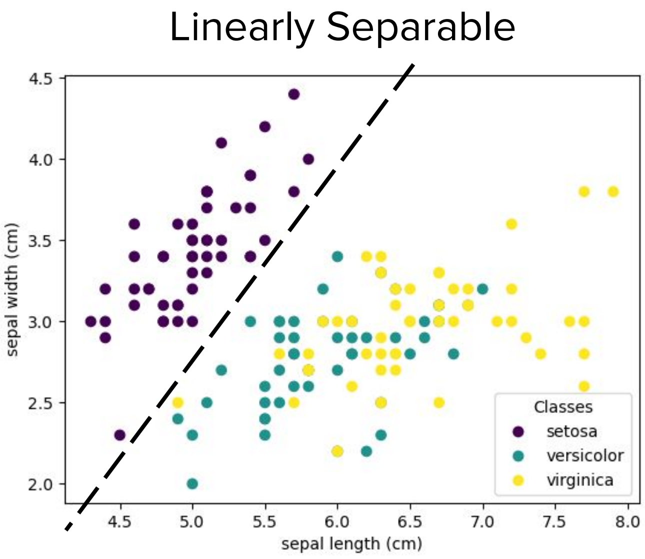
CS 480/680
Introduction to Machine Learning

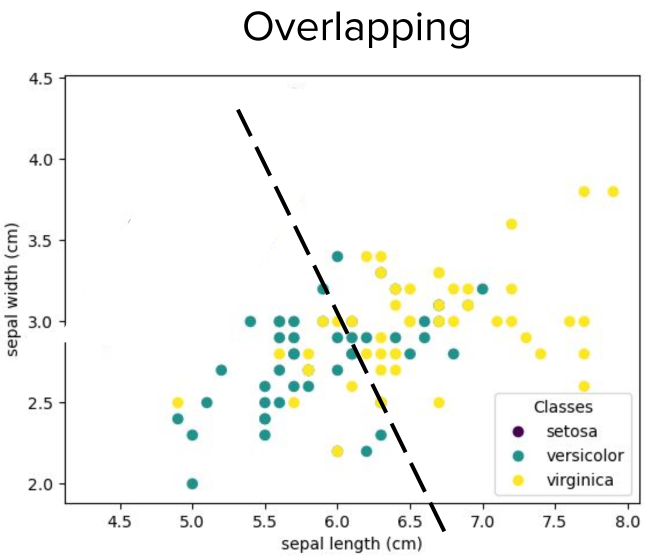
Lecture 8
Nonlinear Feature Maps and Kernel Methods

Kathryn Simone
3 October 2024

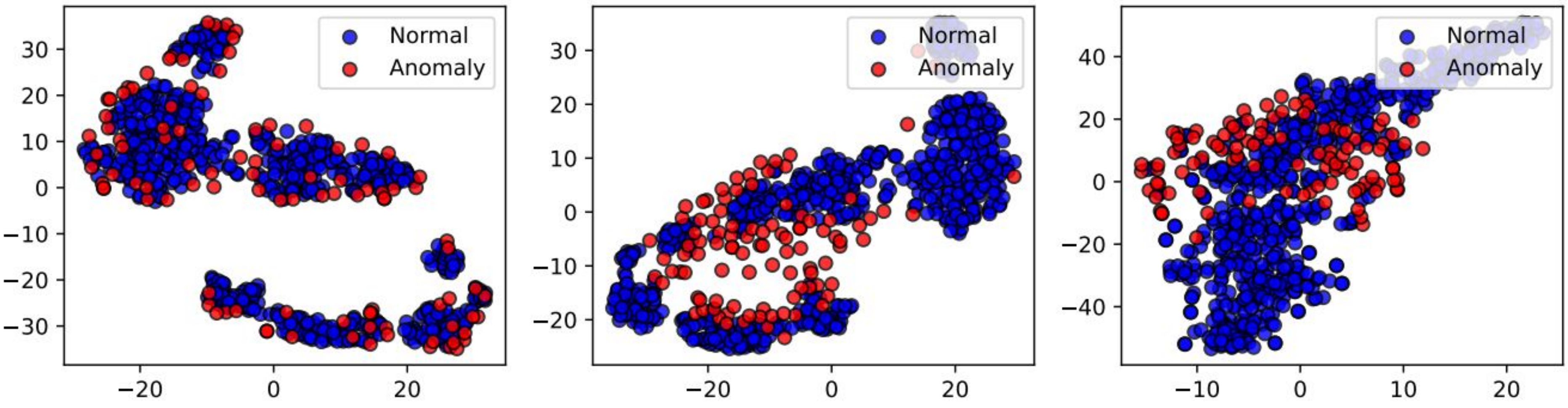
UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

Hard- and Soft-Margin SVMs find a linear decision boundary





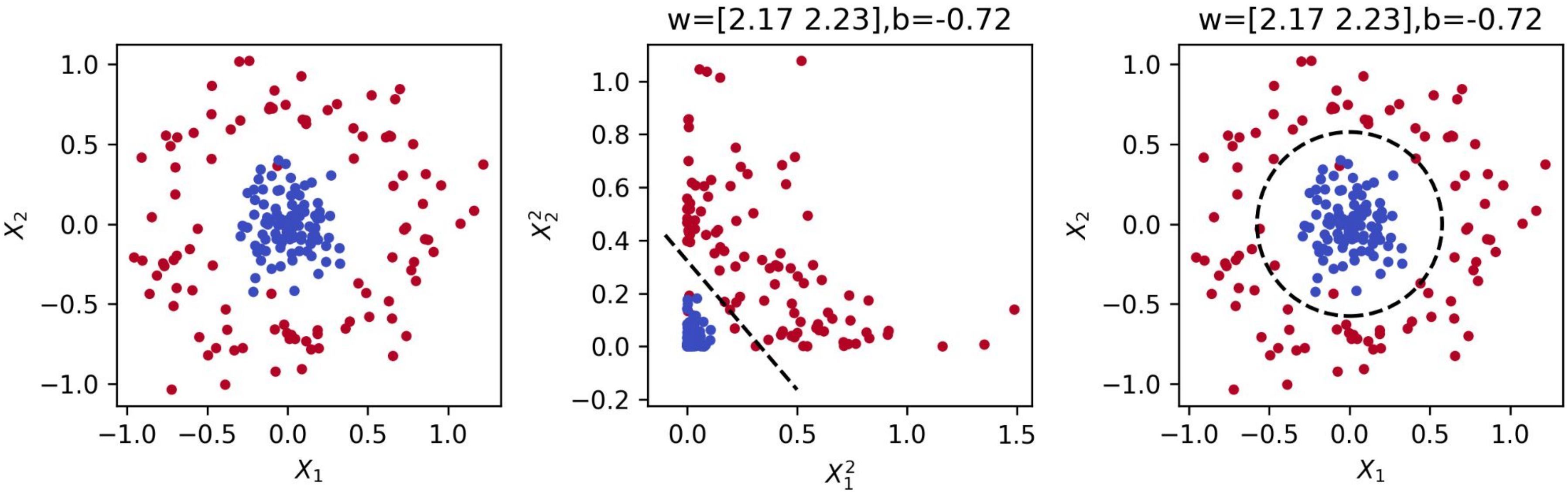
We often require a nonlinear decision boundary



Han et al, 2022

Map the data to a new *feature space*, learn a hyperplane there

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} x_1^2 \\ x_2^2 \end{bmatrix}$$



Key Questions

I. **What kinds of feature maps are possible?**

II. **How can we use these mappings most efficiently?**

Key Questions

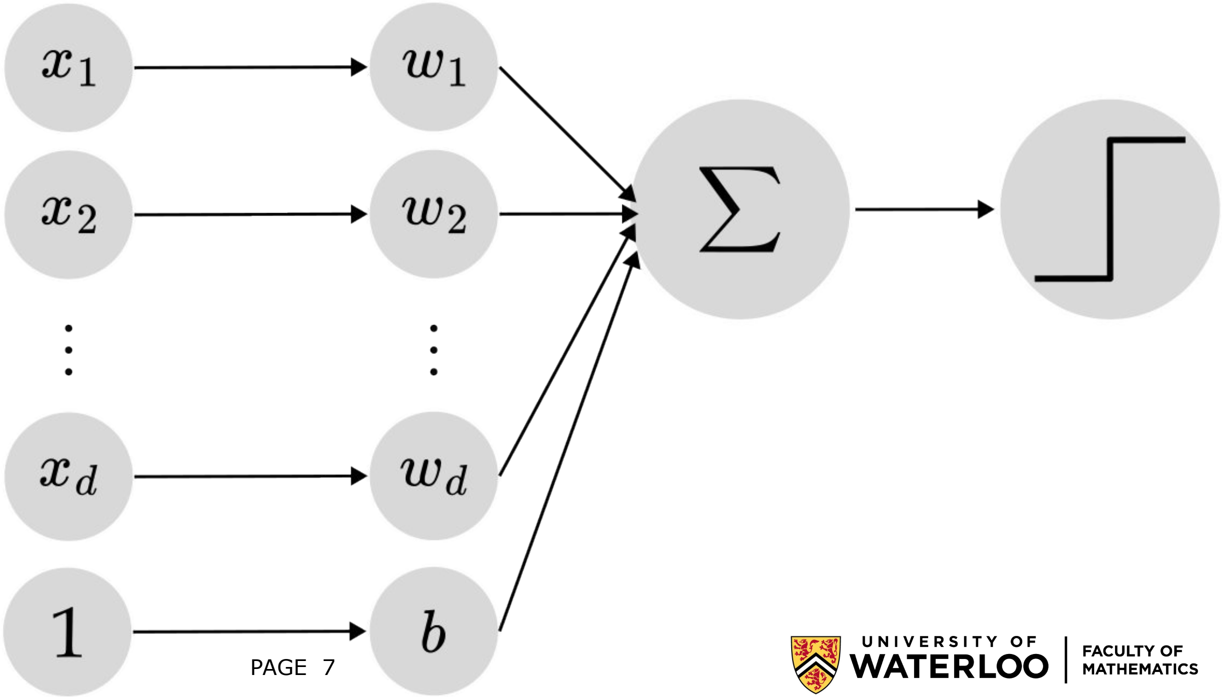
I. **What kinds of feature maps are possible?**

II. **How can we use these mappings most efficiently?**

Learning a classifier in a new feature space

Feature map $\phi(x), x \in \mathbb{R}^{\mathrm{d}} \mapsto \phi(x) \in \mathbb{R}^{\mathrm{m}}$

Before: $\hat{y} = w^T x + b$

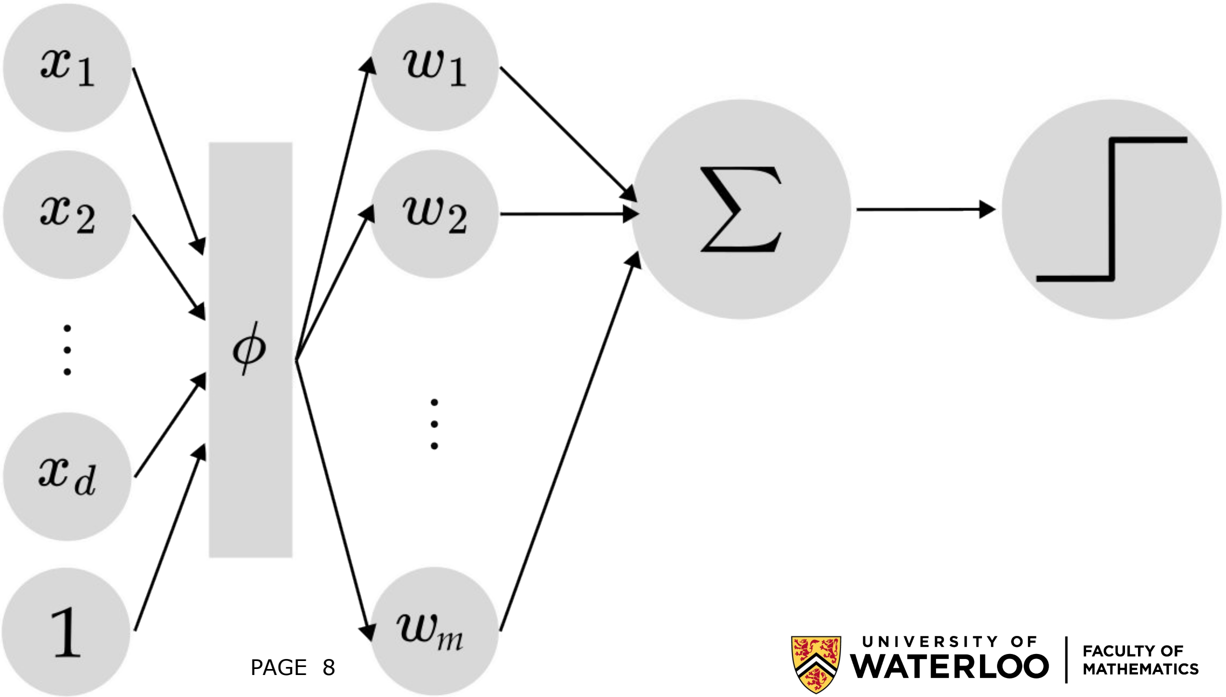


Learning a classifier in a new feature spaces

Feature map $\phi(x), x \in \mathbb{R}^{\mathrm{d}} \mapsto \phi(x) \in \mathbb{R}^{\mathrm{m}}$

Before: $\hat{y} = w^T x + b$

Now: $\hat{y} = w^T \phi(x) + b$



Quadratic feature map

Consider a classifier of the form

$$\hat{y} = x^T Q x + \sqrt{2} x^T p + b$$

Where $x \in \mathbb{R}^{\mathrm{d}}$, Q is a symmetric matrix, $Q \in \mathbb{R}^{\mathrm{d \times d}}$, $p \in \mathbb{R}^{\mathrm{d}}$, $\mathrm{b} \in \mathbb{R}$. Suppose $d = 2$:

$$\begin{aligned}
\hat{y} &= x^T Q x + \sqrt{2} x^T p + b \\
&= \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \sqrt{2} \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + b \\
&= \begin{bmatrix} x_1 q_{11} + x_2 q_{21} & x_1 q_{12} + x_2 q_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \sqrt{2}(x_1 p_1 + x_2 p_2) + b \\
&= (x_1 q_{11} + x_2 q_{21}) x_1 + (x_1 q_{12} + x_2 q_{22}) x_2 + \sqrt{2}(x_1 p_1 + x_2 p_2) + b \\
&= q_{11} x_1^2 + q_{21} x_1 x_2 + q_{22} x_2^2 + q_{12} x_1 x_2 + \sqrt{2} p_1 x_1 + \sqrt{2} p_2 x_2 + b \\
&= q_{11} x_1^2 + q_{22} x_2^2 + 2 q_{21} x_1 x_2 + \sqrt{2} p_1 x_1 + \sqrt{2} p_2 x_2 + b
\end{aligned}$$

Quadratic feature map (continued)

$$\hat{y} = q_{11}x_1^2 + q_{22}x_2^2 + 2q_{21}x_1x_2 + \sqrt{2}p_1x_1 + \sqrt{2}p_2x_2 + b$$

$$w = \begin{bmatrix} q_{11} & q_{22} & 2q_{21} & p_1 & p_2 & b \end{bmatrix}$$

$$\phi(x) = \begin{bmatrix} x_1^2 & x_2^2 & x_1x_2 & x_1 & x_2 & 1 \end{bmatrix}, \text{then}$$

$$\hat{y} = \langle w, \phi(x) \rangle, \text{where } \phi : \mathbb{R}^{\mathrm{d}} \to \mathbb{R}^{\mathrm{d}^2+\mathrm{d}+1}$$

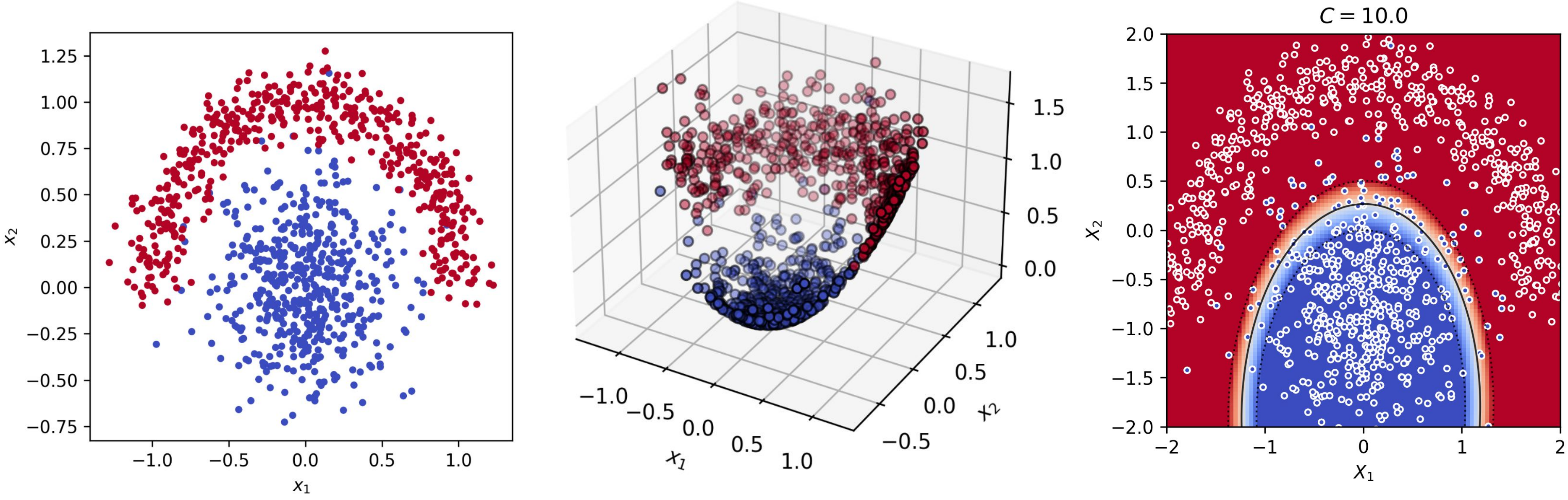
Nonlinear feature maps in SVM

$$L_D = \max_{0\le\lambda_i\le C} \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j x_i^T x_j \ \text{ s.t.} \sum_{i=1}^{n}\lambda_i y_i$$

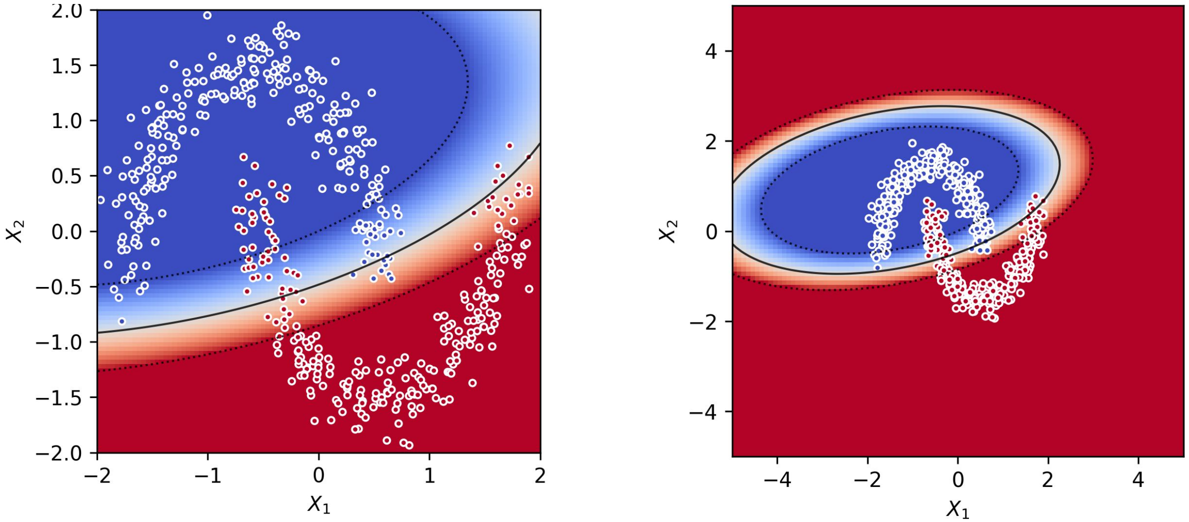
$$= \min_{0\le\lambda_i\le C} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j x_i^T x_j - \sum_{i=1}^{n}\lambda_i \ \text{ s.t.} \sum_{i=1}^{n}\lambda_i y_i$$

$$= \min_{0\le\lambda_i\le C} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_i\lambda_j y_i y_j \phi(x_i)^T\phi(x_j) - \sum_{i=1}^{n}\lambda_i \ \text{ s.t.} \sum_{i=1}^{n}\lambda_i y_i$$

Applying the quadratic feature map in SVM



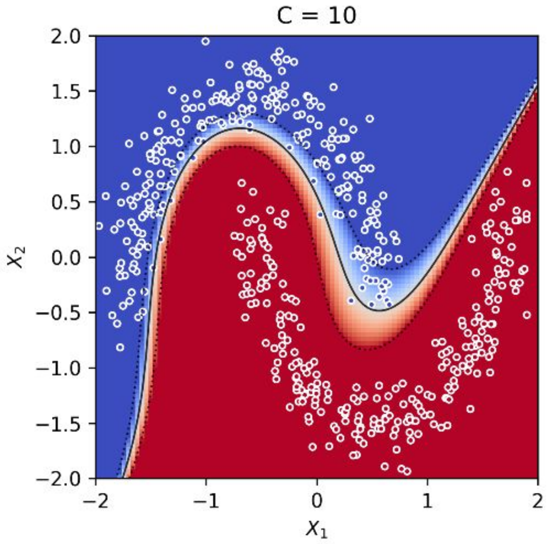
Quadratic feature map fails on another task

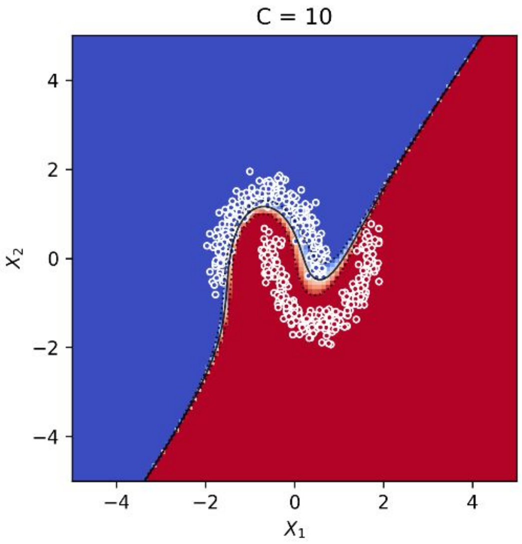


UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

Polynomial feature map (degree 3)

$$\phi(x) = \begin{bmatrix} x_1^3 & x_2^3 & x_1^2 & x_2^2 & x_1^2 x_2 & x_1 x_2^2 & x_1 x_2 & x_1 & x_2 & 1 \end{bmatrix}$$





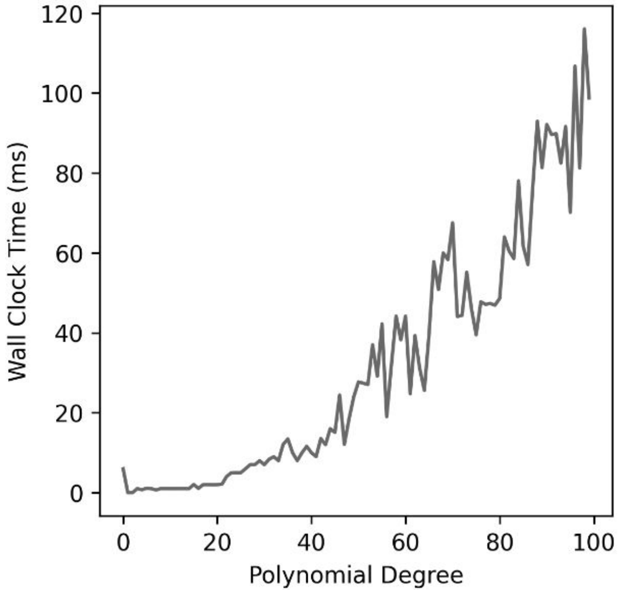
High-dimensional feature mappings in SVM

$$\min_{0 \leq \lambda_i \leq C} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \phi(x_i)^T \phi(x_j) - \sum_{i=1}^{n} \lambda_i \quad \text{s.t.} \sum_{i=1}^{n} \lambda_i y_i$$

Computing dot products between feature vectors, for samples $\vec{x} \in \mathbb{R}^{\mathrm{d}}$:

$$\phi(x) = x : \mathcal{O}(d)$$

$$\phi(x) = [x_1^2, x_2^2, x_1 x_2, x_1, x_2, 1] : \mathcal{O}(d^2)$$



Key Questions

I. What kinds of feature maps are possible?

II. How can we use these mappings most efficiently?

The inner product is all you need in the dual form of SVM

$$= \min_{0 \le \lambda_i \le C} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \phi(x_i)^T \phi(x_j) - \sum_{i=1}^{n} \lambda_i \;\; \text{s.t.} \sum_{i=1}^{n} \lambda_i y_i$$

$$= \min_{0 \le \lambda_i \le C} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \langle \phi(x_i), \phi(x_j) \rangle - \sum_{i=1}^{n} \lambda_i \;\; \text{s.t.} \sum_{i=1}^{n} \lambda_i y_i$$

Is there another way to evaluate the inner product?

Consider $x \in \mathbb{R}^2, \phi : \mathbb{R}^2 \mapsto \mathbb{R}^3$:

$$\phi(x) = \begin{bmatrix} x_1^2 & \sqrt{2}x_1x_2 & x_2^2 \end{bmatrix}$$

$$\begin{aligned} \phi(y) \cdot \phi(z) &= \begin{bmatrix} y_1^2 & \sqrt{2}y_1y_2 & y_2^2 \end{bmatrix} \cdot \begin{bmatrix} z_1^2 \\ \sqrt{2}z_1z_2 \\ z_2^2 \end{bmatrix} \\ &= y_1^2z_1^2 + 2y_1y_2z_1z_2 + y_2^2z_2^2 \\ &= (y_1z_1 + y_2z_2)^2 \\ &= (y \cdot z)^2 \end{aligned}$$

$\mathcal{O}(d)$

Probabilistic Machine Learning, Section 17.1.1

UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

(Mercer) kernels

$$k(x, x') = (x \cdot x')^2$$
$$= \langle \phi(x), \phi(x') \rangle \quad \text{for } \phi : \begin{bmatrix} x_1^2 & \sqrt{2} x_1 x_2 & x_2^2 \end{bmatrix}$$

Any symmetric function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a *kernel* if and only if there exists some $\phi : \mathcal{X} \mapsto \mathcal{H}$ such that

$$k(x, x') = \langle \phi(x), \phi(x') \rangle$$

Mercer's theorem

A function $k : \mathbb{R}^{\mathrm{d}} \times \mathbb{R}^{\mathrm{d}} \to \mathbb{R}$ is a *Mercer kernel* if and only if, for any $n \in \mathcal{N}$, for any $x_1, \ldots x_n \in \mathcal{X}$ the kernel matrix K for which $K_{ij} = k(x_i, x_j)$ is symmetric and positive semidefinite.

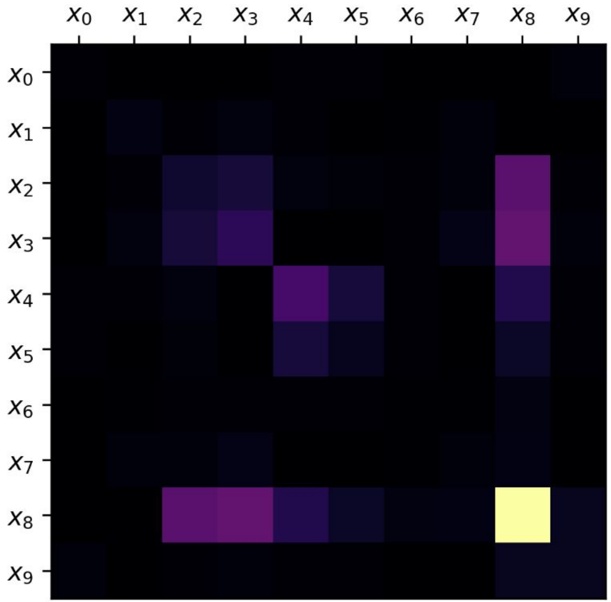
- Symmetric: $K_{ij} = K_{ji}$
- Positive Semidefinite:

$$\langle \boldsymbol{c}, K\boldsymbol{c} \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} k(x_i, x_j) c_i c_j \qquad \geq 0$$

$$\forall x_i \in \mathcal{X}, \quad \forall c_i \in \mathbb{R}.$$

$$k(x, x') = (x \cdot x' + c)^p$$

$$\text{with} \quad c = 1, p = 2$$



Examples of Mercer Kernels

Gaussian:

$$k(x, x') = e^{-\gamma \|x - x'\|_2^2}, \text{where } \gamma = \frac{1}{2\sigma^2}$$

$$= e^{-\gamma x - \gamma x'} \left[1 \cdot 1 + \sqrt{\frac{2\gamma}{1!}} x \cdot \sqrt{\frac{2\gamma}{1!}} x' + \sqrt{\frac{(2\gamma)^2}{2!}} x^2 \cdot \sqrt{\frac{(2\gamma)^2}{2!}} x'^2 + \dots \right]$$

Laplace:

$$k(x, x') = e^{-\gamma \|x - x'\|}$$

The Kernel Trick

$$\min_{0 \leq \lambda_i \leq C} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j \langle \phi(x_i), \phi(x_j) \rangle - \sum_{i=1}^{n} \lambda_i \ \text{ s.t.} \sum_{i=1}^{n} \lambda_i y_i$$

$$\min_{0 \leq \lambda_i \leq C} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j k(x_i, x_j) - \sum_{i=1}^{n} \lambda_i \ \text{ s.t.} \sum_{i=1}^{n} \lambda_i y_i$$

Solving and making predictions with Kernel SVM

$$\min_{0 \le \lambda_i \le C} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j K_{ij} - \sum_{i=1}^{n} \lambda_i \quad \text{s.t.} \sum_{i=1}^{n} \lambda_i y_i$$

$$w^* = \sum_{i=1}^{N_{sv}} \lambda_i y_i \phi(x_i)$$

but it is inconvenient or impossible to compute $\phi(x)$

$$\hat{y} = \langle \phi(x), w^* \rangle + b^*$$

$$\implies \hat{y} = \langle \phi(x), \sum_{i=1}^{N_{sv}} \lambda_i y_i \phi(x_i) \rangle + b^*$$

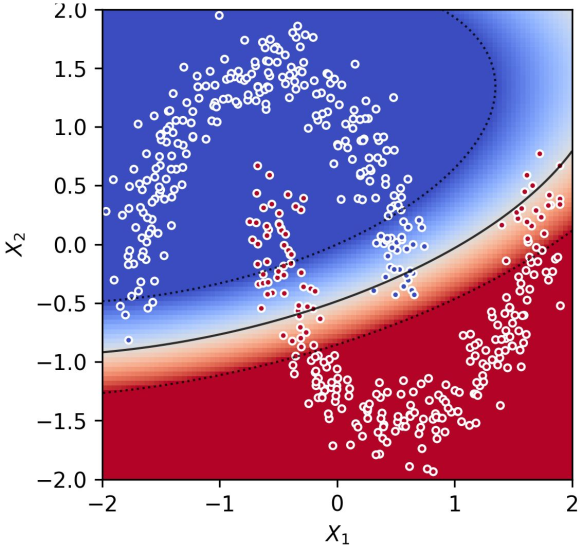
$$= \sum_{i=1}^{N_{sv}} \lambda_i y_i \langle \phi(x), \phi(x_i) \rangle + b^*$$

$$= \sum_{i=1}^{N_{sv}} \lambda_i y_i k(x, x_i) + b^*$$

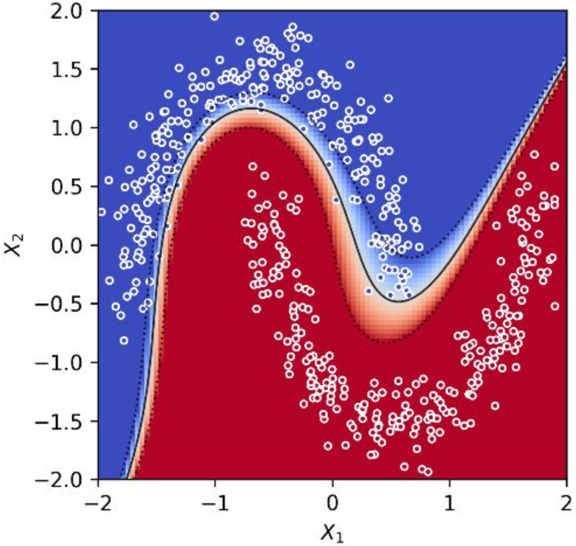
$$\begin{aligned}
b^* &= \frac{1}{N_{sv}} \sum_{i=1}^{N_{sv}} \left(y_i - w^{*T} \phi(x_i) \right) \\
&= \frac{1}{N_{sv}} \sum_{i=1}^{N_{sv}} \left(y_i - \left(\sum_{j=1}^{N_{sv}} \lambda_j y_j \phi(x_j) \right)^T \phi(x_i) \right) \\
&= \frac{1}{N_{sv}} \sum_{i=1}^{N_{sv}} \left(y_i - \sum_{j=1}^{N_{sv}} \lambda_j y_j \langle \phi(x_j), \phi(x_i) \rangle \right) \\
&= \frac{1}{N_{sv}} \sum_{i=1}^{N_{sv}} \left(y_i - \sum_{j=1}^{N_{sv}} \lambda_j y_j k(x_j, x_i) \right)
\end{aligned}$$

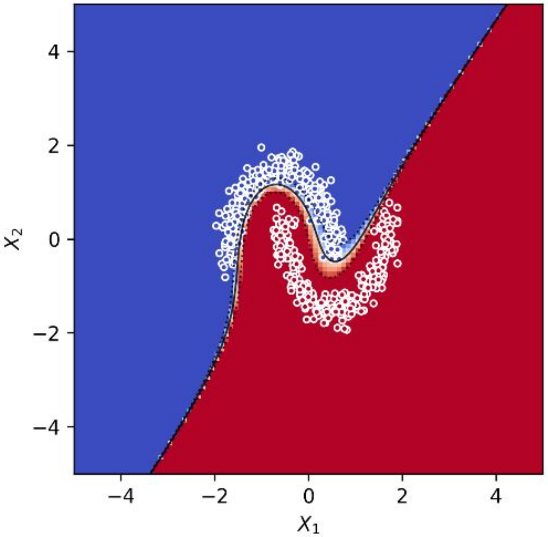
Revisiting the “moons” task with kernel SVMs

Quadratic:



Polynomial:

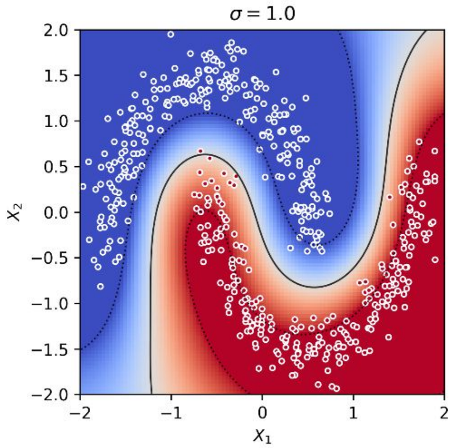


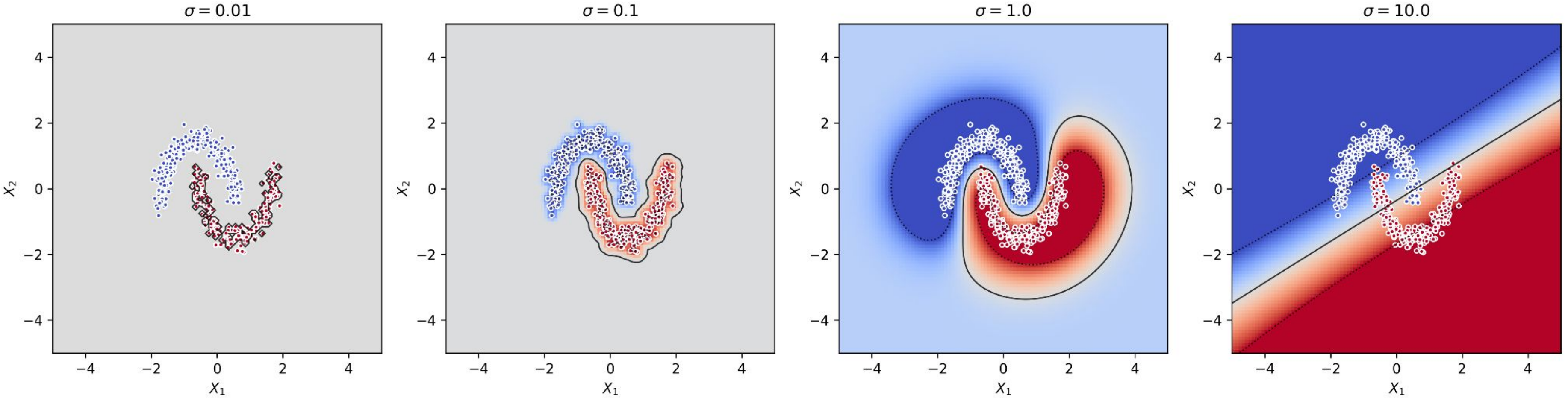


Radial Basis Function Kernel (RBF)

$$k(x, x') = e^{-\gamma \|x - x'\|_2^2}, \text{where } \gamma = \frac{1}{2\sigma^2}$$

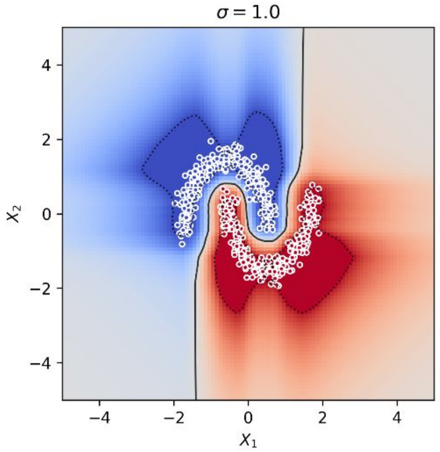
$$= e^{-\gamma x - \gamma x'} \left[1 \cdot 1 + \sqrt{\frac{2\gamma}{1!}} x \cdot \sqrt{\frac{2\gamma}{1!}} x' + \sqrt{\frac{(2\gamma)^2}{2!}} x^2 \cdot \sqrt{\frac{(2\gamma)^2}{2!}} x'^2 + \dots \right]$$

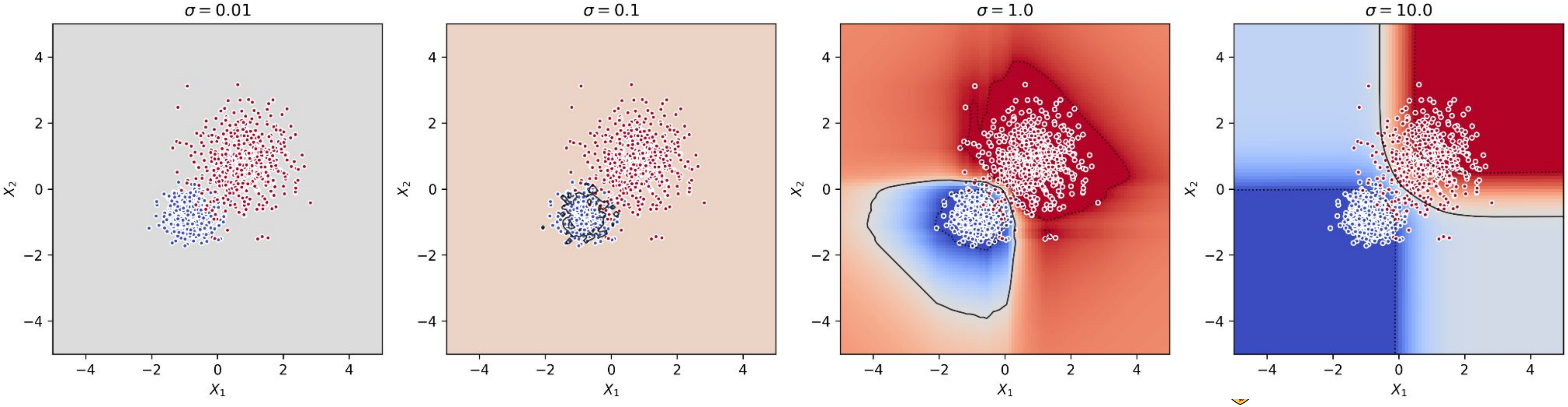




Laplace Kernel

$$k(x, x') = e^{-\gamma \|x - x'\|}$$





Now that we're at the end of the lecture, you should be able to...

- ★ Discriminate between feature maps with local and global effects.
- ★ Construct kernel functions for specialized classification tasks.
- ★ Recall **widely-used kernels** and describe their properties and parameters.
- ★ Verify whether a **kernel function** is a **Mercer kernel** using formal proofs or inspection of its associated **Gram matrix**.
- ★ Recognize and apply the **kernel trick** in SVM classification.
- ★ Defend the **kernel trick** with reference to **expressivity, implicit computation, computational complexity**.