CS480/680: Introduction to Machine Learning

Lecture 7: Reproducing Kernels

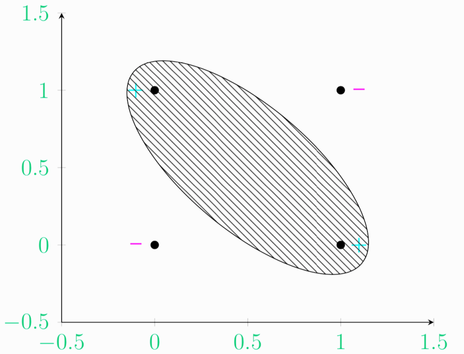
Hongyang Zhang

UNIVERSITY OF WATERLOO

Feb 1, 2024

XOR Dataset

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$
	0	1	0	1
	0	0	1	1
$\mathbf{y}$	$-$	$+$	$+$	$-$



- We have proved that linear classifier cannot separate the data
- Need more complex (non-linear) score function, e.g., a quadratic classifier

Quadratic Classifier

$$f(\mathbf{x}) = \langle \mathbf{x}, Q\mathbf{x} \rangle + \sqrt{2}\, \langle \mathbf{x}, \mathbf{p} \rangle + b$$

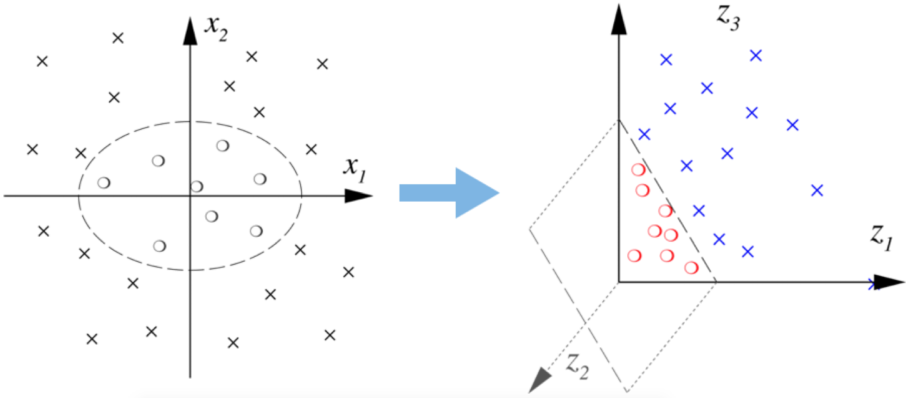
- Predict as before $\hat{\mathrm{y}} = \mathrm{sign}(f(\mathbf{x}))$, $\mathbf{x}$ is a column vector in $\mathbb{R}^d$
- Weights to be learned: $Q \in \mathbb{R}^{d \times d}$, $\mathbf{p} \in \mathbb{R}^d$, $b \in \mathbb{R}$
- Setting $Q = \mathbf{0}$ reduces to the linear case

The Power of Lifting

$$
\begin{aligned}
f(\mathbf{x}) &= \langle \mathbf{x}, Q\mathbf{x} \rangle + \sqrt{2}\,\langle \mathbf{x}, \mathbf{p} \rangle + b \\
&= \left\langle \mathbf{x}\mathbf{x}^\top, Q \right\rangle + \left\langle \sqrt{2}\mathbf{x}, \mathbf{p} \right\rangle + b \\
&= \langle \phi(\mathbf{x}), \mathbf{w} \rangle \quad \text{(no bias term here)}
\end{aligned}
$$

- For any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, let $\overrightarrow{\mathbf{A}}$ be the vectorization operation: $\mathbb{R}^{m \times n} \to \mathbb{R}^{mn}$
- Feature map $\phi(\mathbf{x}) = \begin{bmatrix} \overrightarrow{\mathbf{x}\mathbf{x}^\top} \\ \sqrt{2}\mathbf{x} \\ 1 \end{bmatrix}$, where $\mathbf{x} \in \mathbb{R}^d \mapsto \phi(\mathbf{x}) \in \mathbb{R}^{d \times d + d + 1}$
- Weights to be learned: $\mathbf{w} = \begin{bmatrix} \overrightarrow{Q} \\ \mathbf{p} \\ b \end{bmatrix} \in \mathbb{R}^{d \times d + d + 1}$
- Nonlinear in $\mathbf{x}$ but linear in $\phi(\mathbf{x})$: ϕ must be nonlinear w.r.t. $\mathbf{x}$

From Nonlinear to Linear



- In the high-dimensional space, the data are linearly separable by a hyperplane.

The Kernel Trick

- Feature map $\phi : \mathbb{R}^d \to \mathbb{R}^{\boxed{d \times d + d + 1}}$ blows up the dimension
- Do we have to operate in the high-dimensional feature space, explicitly?
- In the dual form of SVM, all we need is the inner product!

$$\begin{aligned}\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle &= \left\langle \begin{bmatrix} \overrightarrow{\mathbf{x}\mathbf{x}^\top} \\ \sqrt{2}\mathbf{x} \\ 1 \end{bmatrix}, \begin{bmatrix} \overrightarrow{\mathbf{z}\mathbf{z}^\top} \\ \sqrt{2}\mathbf{z} \\ 1 \end{bmatrix} \right\rangle = (\langle \mathbf{x}, \mathbf{z} \rangle)^2 + 2\langle \mathbf{x}, \mathbf{z} \rangle + 1 \\ &= (\langle \mathbf{x}, \mathbf{z} \rangle + 1)^2\end{aligned}$$

- Inner product in the high-dim space can be computed by the original vectors
- Which can still be computed in $O(d)$ time!

Reverse Engineering

- Given feature map $\phi : \mathcal{X} \to \mathcal{H}$, the resulting inner product

$$\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle =: k(\mathbf{x}, \mathbf{z})$$

  can be computed (e.g., $k(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + 1)^2$)
- Conversely, given $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, does there exist $\phi : \mathcal{X} \to \mathcal{H}$ such that

$$\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle = k(\mathbf{x}, \mathbf{z})?$$

(Reproducing) Kernels

Definition: (Reproducing) Kernels

We call $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a (reproducing) kernel iff there exists some $\phi : \mathcal{X} \to \mathcal{H}$ so that $\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle = k(\mathbf{x}, \mathbf{z})$.

- Choosing a feature transform ϕ determines the corresponding kernel k
- Choosing a kernel k determines some feature transform ϕ too
  - may not be unique
  - $\phi(\mathbf{x}) := [x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1] \in \mathbb{R}^6$
  - $\psi(\mathbf{x}) := [x_1^2, x_1x_2, x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1] \in \mathbb{R}^7$
  - $\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle = \langle \psi(\mathbf{x}), \psi(\mathbf{z}) \rangle$ for any $\mathbf{x} \in \mathbb{R}^2$ and $\mathbf{z} \in \mathbb{R}^2$

N. Aronszajn (1950). “Theory of Reproducing Kernels”. Transactions of the American Mathematical Society, vol. 68, no. 3, pp. 337–404.

Verifying a Kernel

Theorem: Mercer's theorem

$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel iff for any $n \in \mathbb{N}$, for any $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$, the kernel matrix K such that $K_{ij} := k(\mathbf{x}_i, \mathbf{x}_j)$ is symmetric and PSD.

- Symmetric: $K_{ij} = K_{ji}$
- Positive Semi-Definite (PSD): for any $\boldsymbol{\alpha} \in \mathbb{R}^n$,

$$\langle \boldsymbol{\alpha}, K\boldsymbol{\alpha} \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K_{ij} \geq 0.$$

Examples

- Polynomial kernel: $k(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + 1)^p$
  - p is a hyper-parameter
  - larger $p \rightarrow$ higher-degree polynomial mapping ϕ
- Gaussian kernel: $k(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|_2^2 / \sigma)$
- Laplace kernel: $k(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|_2 / \sigma)$
  - σ is a hyper-parameter
  - larger $\sigma \rightarrow$ smooth ϕ: $\phi(\mathbf{x}_1)$ and $\phi(\mathbf{x}_2)$ will not differ too much for close $\mathbf{x}_1$ and $\mathbf{x}_2$

Kernel SVM

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}(1 - \mathsf{y}_i\hat{y}_i)^+$$
$$\text{s.t.} \quad \hat{y}_i = \langle \mathbf{x}_i, \mathbf{w} \rangle, \forall i$$

$$\min_{C \geq \alpha \geq \mathbf{0}} -\sum_i \alpha_i + \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \boxed{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}$$
$$\text{s.t.} \quad \sum_i \alpha_i \mathsf{y}_i = 0$$

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}(1 - \mathsf{y}_i\hat{y}_i)^+$$
$$\text{s.t.} \quad \hat{y}_i = \langle \phi(\mathbf{x}_i), \mathbf{w} \rangle, \forall i$$

$$\min_{C \geq \alpha \geq \mathbf{0}} -\sum_i \alpha_i + \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$
$$\text{s.t.} \quad \sum_i \alpha_i \mathsf{y}_i = 0$$

Prediction

- Solve $\boldsymbol{\alpha}^* \in \mathbb{R}^n$, and recover

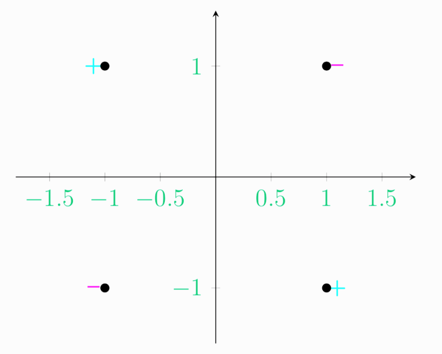
$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* \mathsf{y}_i \phi(\mathbf{x}_i)$$

- But we do not know ϕ, so we cannot compute $\mathbf{w}^*$ explicitly
- For testing, only need to compute the sign of:

$$f(\mathbf{x}) := \langle \phi(\mathbf{x}), \mathbf{w}^* \rangle = \left\langle \phi(\mathbf{x}), \sum_{i=1}^{n} \alpha_i^* \mathsf{y}_i \phi(\mathbf{x}_i) \right\rangle = \sum_{i=1}^{n} \alpha_i^* \mathsf{y}_i k(\mathbf{x}, \mathbf{x}_i)$$

- Knowing the dual vector $\boldsymbol{\alpha}^*$, training set $\{\mathbf{x}_i, \mathsf{y}_i\}$ and the kernel k suffices for getting the score function of the test data $\mathbf{x}$!

An Example on XOR Dataset



We have proved the dataset is non-linearly separable. Consider non-linear mapping:

$$k(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + 1)^2$$

An Example on XOR Dataset

$$\min_{\alpha \geq \mathbf{0}} \ -\sum_i \alpha_i + \tfrac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{s.t.} \quad \sum_i \alpha_i \mathsf{y}_i = 0$$

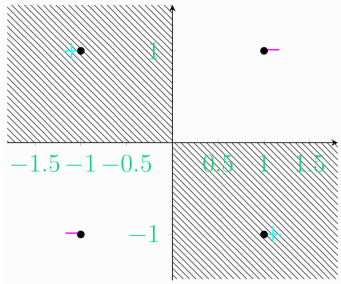
Let the derivative of objective $= 0$, we get

$$\begin{bmatrix} -1 & & & \\ & 1 & & \\ & & -1 & \\ & & & 1 \end{bmatrix} \underbrace{\begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix}}_{K} \begin{bmatrix} -1 & & & \\ & 1 & & \\ & & -1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

An Example on XOR Dataset

$$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \tfrac{1}{8}, \text{ which happens to satisfy } \sum_i \alpha_i \mathsf{y}_i = 0.$$

$$f(\mathbf{x}) = \langle \phi(\mathbf{x}), \mathbf{w} \rangle = \sum_i \alpha_i \mathsf{y}_i k(\mathbf{x}, \mathbf{x}_i) = -x_1 x_2$$




Questions
?
?
Answers
?