CS480/680: Introduction to Machine Learning

Lec 07: Reproducing Kernels

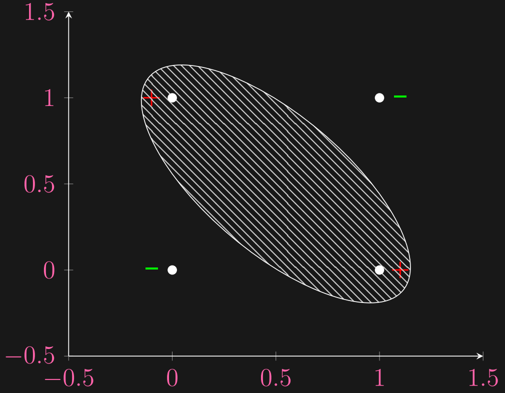
Yaoliang Yu

UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS
DAVID R. CHERITON SCHOOL OF COMPUTER SCIENCE

May 29, 2024

XOR Dataset

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



Quadratic Classifier

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

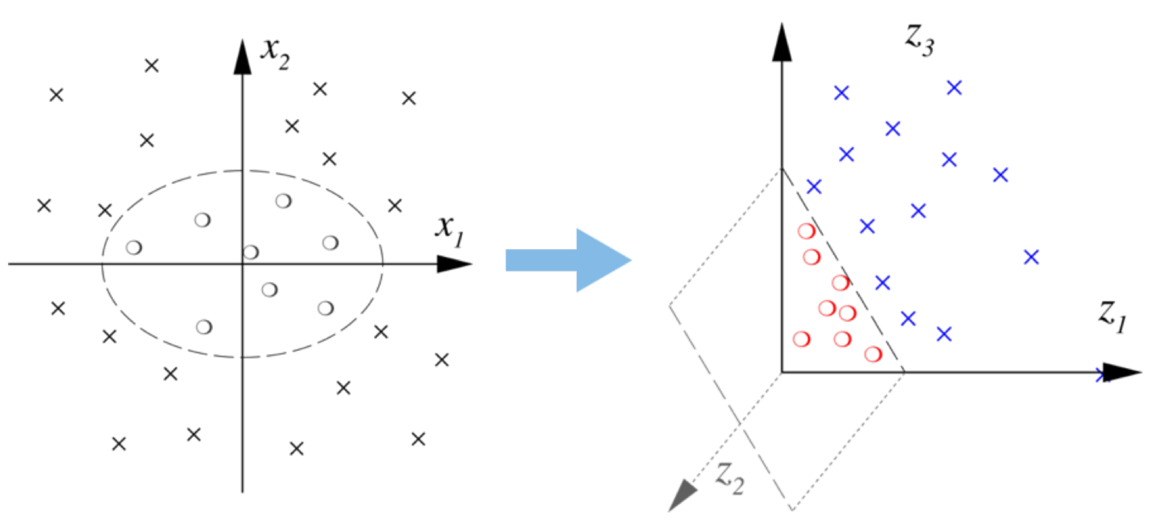
- Predict as before $\hat{\mathrm{y}} = \mathrm{sign}(f(\mathbf{x}))$
- Weights to be learned: $Q \in \mathbb{R}^{d \times d}$, $\mathbf{p} \in \mathbb{R}^{d}$, $b \in \mathbb{R}$
- Setting $Q = \mathbf{0}$ reduces to the linear case

The Power of Lifting 🏋️‍♀️

$$
\begin{aligned}
f(\mathbf{x}) &= \langle \mathbf{x}, Q\mathbf{x} \rangle + \sqrt{2}\,\langle \mathbf{x}, \mathbf{p} \rangle + b \\
&= \langle \mathbf{x}\mathbf{x}^\top, Q \rangle + \left\langle \sqrt{2}\mathbf{x}, \mathbf{p} \right\rangle + b \\
&= \langle \phi(\mathbf{x}), \mathbf{w} \rangle
\end{aligned}
$$

- 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

From Nonlinear to Linear



The Kernel Trick

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

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

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

- 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, albeit inefficiently due to large dimension of $\mathcal{H}$

- Conversely, given $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, does there exist $\phi : \mathcal{X} \to \mathcal{H}$ such that

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

- For computational purposes, all we need is the existence of such ϕ

- Later, neural nets learn ϕ simultaneously with $\mathbf{w}$

(Reproducing) Kernels

We call $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a (reproducing) kernel if there exists some feature transform $\phi : \mathcal{X} \to \mathcal{H}$ 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; cannonical choice $\varphi(\mathbf{x}) := k(\cdot, \mathbf{x})$
  - $\phi(x_1, x_2) := [x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1]$
  - $\psi(x_1, x_2) := [x_1^2, x_1x_2, x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1]$
- Unique RKHS: $\mathcal{H}_k := \{\mathbf{x} \mapsto \langle \phi(\mathbf{x}), \mathbf{w} \rangle : \mathbf{w} \in \mathcal{H}\} \subseteq \mathbb{R}^{\mathcal{X}}$
- Reproducing: $\langle f, k(\cdot, \mathbf{x}) \rangle = f(\mathbf{x})$ and $\langle k(\cdot, \mathbf{x}), k(\cdot, \mathbf{z}) \rangle = k(\mathbf{x}, \mathbf{z})$

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

Verifying a Kernel

Theorem: Positive Semi-definite (PSD)

$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ 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 $K_{ij} := k(\mathbf{x}_i, \mathbf{x}_j)$ is symmetric and PSD. In notation: $K \in \mathbb{S}^n_+$.

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

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

  - if equality is attained only at $\boldsymbol{\alpha} = \mathbf{0}$, then it is called positive definite or strictly PSD
- Can think of a kernel as some form of similarity measure

Examples

- Polynomial kernel: $k(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + 1)^p$
  - underlying RKHS?
- Gaussian kernel: $k(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|_2^2 / \sigma)$
  - infinite-dimensional RKHS!
- Laplace kernel: $k(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|_2 / \sigma)$
- Brownian motion: $k(s, t) := s \wedge t$ for $s, t \geq 0$

A Word About Universality

- 1-1 correspondence between a kernel k and its RKHS $\mathcal{H}_k$
- RKHS is a linear space of functions from $\mathcal{X}$ to $\mathbb{R}$
- A kernel is called universal if its RKHS is large enough to approximate any continuous function (over a compact domain $\mathcal{X}$)
- Kernel mean embedding: $P \mapsto \underset{\mathsf{X}\sim P}{\mathbb{E}} \varphi(\mathsf{X}) \in \mathcal{H}_k$, 1-1 iff k is characteristic

C. A. Micchelli et al. "Universal Kernels". *Journal of Machine Learning Research*, vol. 7, no. 95 (2006), pp. 2651–2667, B. K. Sriperumbudur et al. "Universality, Characteristic Kernels and RKHS Embedding of Measures". *Journal of Machine Learning Research*, vol. 12, no. 70 (2011), pp. 2389–2410.

Kernel Calculus

- If k is a kernel, so is λk for any $\lambda \geq 0$
  - if k has feature map ϕ, what could be the feature map of λk?
- If k_1 and k_2 are kernels, so is $k_1 + k_2$
  - if k_i has feature map ϕ_i, what could be the feature map of $k_1 + k_2$?
- If k_1 and k_2 are kernels, so is $k_1 k_2$
  - if k_i has feature map ϕ_i, what could be the feature map of $k_1 k_2$?
- If k_t are kernels then the limit $\lim_t k_t$ (when exists) is also a kernel

Kernel SVM

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

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

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

Testing

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

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

- We do not know ϕ so cannot compute $\mathbf{w}$ explicitly
- For testing, only need to compute

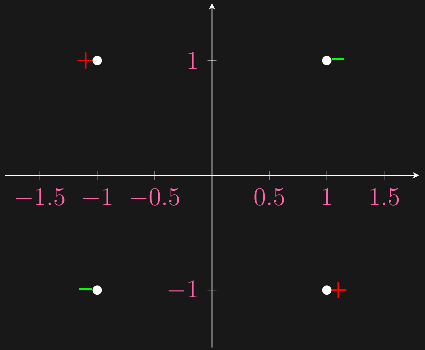
$$f(\mathbf{x}) := \langle \phi(\mathbf{x}), \mathbf{w} \rangle = \left\langle \phi(\mathbf{x}), \sum_{i=1}^{n} \alpha_i \mathsf{y}_i \phi(\mathbf{x}_i) \right\rangle = \sum_{i=1}^{n} \alpha_i \mathsf{y}_i k(\mathbf{x}, \mathbf{x}_i) \in \mathcal{H}_k$$

- Knowing the dual variable $\boldsymbol{\alpha}$, training set $\{\mathbf{x}_i, \mathsf{y}_i\}$ and the kernel k suffices!

Tradeoff

- Previously: training $O(nd)$ and testing $O(d)$
- Kernel (including the linear kernel $\langle \mathbf{x}, \mathbf{z} \rangle$): training $O(n^2 d)$ and testing $O(nd)$
- Managed to avoid explicit dependence on feature dimension (could even be ∞)
- At the price of n (the training set size) times slower, both in training and test
- Also necessary to store the training set (at least the support vectors)

Does It Work?



$$\phi(\mathbf{x}) = [x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1]$$

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

Crunch Crunch

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

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

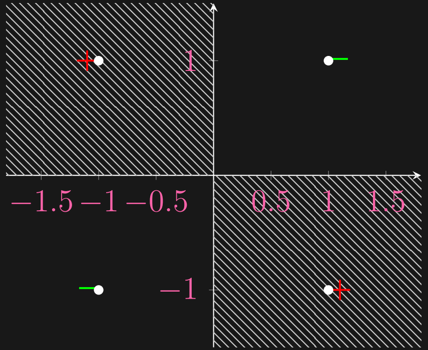
$$\begin{bmatrix} -1 & & & \\ & 1 & & \\ & & -1 & \\ & & & 1 \end{bmatrix} \underbrace{\begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix}}_{K} \begin{bmatrix} -1 & & & \\ & 1 & & \\ & & -1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3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

$$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \frac{1}{8}$$

$$\mathbf{w} = \sum_i \alpha_i \mathsf{y}_i \phi(\mathbf{x}_i) = [0, -\tfrac{1}{\sqrt{2}}, 0, 0, 0, 0]$$

$$\phi(\mathbf{x}) = [x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1]$$

$$f(\mathbf{x}) = \langle \phi(\mathbf{x}), \mathbf{w} \rangle = \sum_i \alpha_i \mathsf{y}_i k(\mathbf{x}, \mathbf{x}_i) = -x_1x_2$$



Logistic Regression Revisited

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \log(1+\exp(-\mathsf{y}_i \langle \mathbf{x}_i, \mathbf{w}\rangle)) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \log(1+\exp(-\mathsf{y}_i \langle \phi(\mathbf{x}_i), \mathbf{w}\rangle)) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

Theorem: Representer Theorem

The optimal $\mathbf{w} = \sum_{j=1}^{n} \alpha_j \mathsf{y}_j \phi(\mathbf{x}_j)$ for some $\boldsymbol{\alpha} \in \mathbb{R}^n$.

Orthogonal Decomposition

$$\mathbf{w} = \mathbf{w}^{\parallel} + \mathbf{w}^{\perp}$$

- $\mathbf{w}^{\parallel} \in \mathrm{span}\{\mathrm{y}_i \phi(\mathbf{x}_i) : i = 1, \ldots, n\}$
- Logistic loss only depends on $\mathbf{w}^{\parallel}$
- Regularizer is smaller if $\mathbf{w}^{\perp} = \mathbf{0}$
- Thus, $\mathbf{w} = \mathbf{w}^{\parallel} = \sum_j \alpha_j \mathrm{y}_j \phi(\mathbf{x}_j)$ for some $\boldsymbol{\alpha} \in \mathbb{R}^n$

Learning the Kernel

$$\min_{\mathbf{w}} \frac{1}{n} \sum_i \ell(\langle \phi(\mathbf{z}_i), \mathbf{w} \rangle) + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$$

- Can learn a positive combination of base kernels, with coefficient $\boldsymbol{\beta}$ learned simultaneously with $\mathbf{w}$

$$\min_{\mathbf{w},\boldsymbol{\beta}} \frac{1}{n} \sum_i \ell\left(\left\langle \bigoplus_p \beta_p \phi_p(\mathbf{z}_i), \mathbf{w} \right\rangle\right) + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$$

- Apply the representer theorem to plug in

$$\mathbf{w} = \sum_j \alpha_j \bigoplus_p \beta_p \phi_p(\mathbf{z}_j)$$

G. R. Lanckriet et al. "Learning the Kernel Matrix with Semidefinite Programming". *Journal of Machine Learning Research*, vol. 5 (2004), pp. 27–72.