Machine learning lecture slides

COMS 4771 Fall 2020

Regression III: Kernels

Outline

- Dual form of ridge regression
- Examples of kernel trick
- Kernel methods

Linear algebraic identity

- Let $A = \frac{1}{\sqrt{n}} \begin{bmatrix} \leftarrow & x_1^\mathsf{T} & \rightarrow \\ & \vdots & \\ \leftarrow & x_n^\mathsf{T} & \rightarrow \end{bmatrix} \in \mathbb{R}^{n \times d}$ and $b = \frac{1}{\sqrt{n}} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$
- Linear algebraic identity: for any $A \in \mathbb{R}^{n \times d}$ and any $\lambda > 0$,

$$(\underbrace{A^\mathsf{T} A + \lambda I}_{d \times d})^{-1} A^\mathsf{T} = A^\mathsf{T} (\underbrace{A A^\mathsf{T} + \lambda I}_{n \times n})^{-1}.$$

  - Check: multiply both sides by $A^\mathsf{T} A + \lambda I$ and "factor".

Alternative (dual) form for ridge regression (1)

- Implications for ridge regression

$$\hat{w} = A^\mathsf{T} \underbrace{(AA^\mathsf{T} + \lambda I)^{-1} b}_{=:\sqrt{n}\hat{\alpha}} = \sqrt{n} A^\mathsf{T} \hat{\alpha} = \sum_{i=1}^{n} \hat{\alpha}_i x_i.$$

- Matrix $AA^\mathsf{T} = \frac{1}{n} K$, where $K \in \mathbb{R}^{n \times n}$ is the *Gram matrix*

$$K_{i,j} = x_i^\mathsf{T} x_j.$$

- Prediction with $\hat{w}$ on new point x:

$$x^\mathsf{T} \hat{w} = \sum_{i=1}^{n} \hat{\alpha}_i \cdot x^\mathsf{T} x_i$$

Alternative (dual) form for ridge regression (2)

- Therefore, can "represent" predictor via data points $x_1, \ldots, x_n$ and $\hat{\alpha}$.
  - Similar to nearest neighbor classifier, except also have $\hat{\alpha}$
  - To get $\hat{\alpha}$: solve linear system involving K (and not A directly)
  - To make prediction on x: iterate through the x_i to compute inner products with x; take appropriate weighted sum of results
- When is this a good idea?

Quadratic expansion

- Suppose we want to do feature expansion to get all quadratic terms in $\varphi(x)$

$$\varphi(x) = (1, \underbrace{\sqrt{2}x_1, \ldots, \sqrt{2}x_d}_{\text{linear terms}}, \underbrace{x_1^2, \ldots, x_d^2}_{\text{squared terms}}, \underbrace{\sqrt{2}x_1x_2, \ldots, \sqrt{2}x_1x_d, \ldots, \sqrt{}}_{\text{cross terms}}$$

- This feature expansion has $1 + 2d + \binom{d}{2} = \Theta(d^2)$ terms
  - Explicitly computing $\varphi(x)$, $\varphi(x')$, and then $\varphi(x)^\mathsf{T}\varphi(x')$ would take $\Theta(d^2)$ time.

- "Kernel trick": can compute $\varphi(x)^\mathsf{T}\varphi(x')$ in $O(d)$ time:

$$\varphi(x)^\mathsf{T}\varphi(x') = (1 + x^\mathsf{T}x')^2.$$

- Similar trick for cubic expansion, quartic expansion, etc.

Gaussian kernel

- For any $\sigma > 0$, there is an infinite-dimensional feature expansion $\varphi \colon \mathbb{R}^d \to \mathbb{R}^\infty$ such that

$$\varphi(x)^\mathsf{T}\varphi(x') = \exp\left(-\frac{\|x - x'\|_2^2}{2\sigma^2}\right),$$

  which can be computed in $O(d)$ time.
- Called *Gaussian kernel* or *Radial Basis Function (RBF) kernel* (with bandwidth σ).

- Feature expansion for $d = 1$ and $\sigma = 1$ case:

$$\varphi(x) = e^{-x^2/2}\left(1, x, \frac{x^2}{\sqrt{2!}}, \frac{x^3}{\sqrt{3!}}, \dots\right).$$

Kernels

- A *positive definite kernel* $\mathsf{k} \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a symmetric function satisfying the following property: For any n, and any $x_1, \ldots, x_n \in \mathcal{X}$, the $n \times n$ matrix whose (i, j)-th entry is $\mathsf{k}(x_i, x_j)$ is positive semidefinite.

- **Theorem**: For any positive definite kernel k, there exists a feature map $\varphi \colon \mathcal{X} \to H$ such that $\varphi(x)^\top \varphi(x') = \mathsf{k}(x, x')$ for all $x, x' \in \mathcal{X}$.
  - Here, H is a special kind of inner product space called the *Reproducing Kernel Hilbert Space (RKHS)* corresponding to k.

- Algorithmically, we don't have to worry about what φ is. Instead, just use k.

Kernel ridge regression (1)

- Training data $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathbb{R}$
- Ridge regression with feature map φ: minimize

$$\frac{1}{n} \sum_{i=1}^{n} (\varphi(x_i)^\mathsf{T} w - y_i)^2 + \lambda \|w\|_2^2$$

- Compute the $n \times n$ *kernel matrix* K where

$$K_{i,j} = \mathsf{k}(x_i, x_j).$$

- Letting $w = \sum_{i=1}^{n} \alpha_i \varphi(x_i)$ for $\alpha = (\alpha_1, \ldots, \alpha_n)$, ridge regression objective is equivalent to

$$\frac{1}{n} \|K\alpha - y\|_2^2 + \lambda \alpha^\mathsf{T} K \alpha$$

where $y = (y_1, \ldots, y_n) \in \mathbb{R}^n$.

Kernel ridge regression (2)

- Minimizer wrt α is solution $\hat{\alpha}$ to linear system of equations

$$(K + n\lambda I)\alpha = y.$$

- Return predictor that is represented by $\hat{\alpha} \in \mathbb{R}^n$ and $x_1, \dots, x_n$
  - To make prediction on new $x \in \mathcal{X}$: output

$$\sum_{i=1}^{n} \hat{\alpha}_i \cdot \mathsf{k}(x, x_i).$$

- Inductive bias:

$$\begin{aligned} |\hat{w}^{\mathsf{T}}\varphi(x) - \hat{w}^{\mathsf{T}}\varphi(x')| &\leq \|\hat{w}\|_2 \cdot \|\varphi(x) - \varphi(x')\|_2 \\ &= \sqrt{\hat{\alpha}^{\mathsf{T}} K \hat{\alpha}} \cdot \|\varphi(x) - \varphi(x')\|_2 \end{aligned}$$

Kernel methods

- Many methods / algorithms can be "kernelized" into *kernel methods*
  - E.g., nearest neighbor, PCA, SVM, gradient descent, ...
- "Spectral regularization" with kernels: solve $g(K/n)\alpha = y/n$ for α.

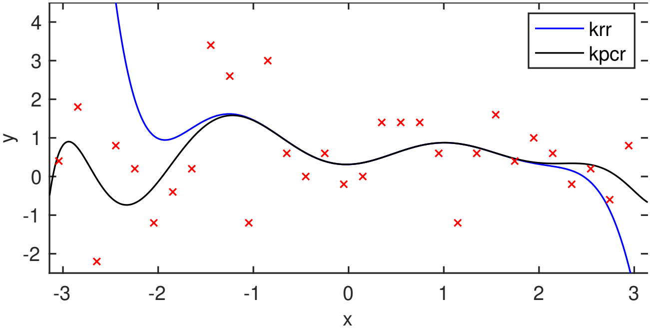

Figure 1: Polynomial kernel with Kernel Ridge Regression and Kernel PCR

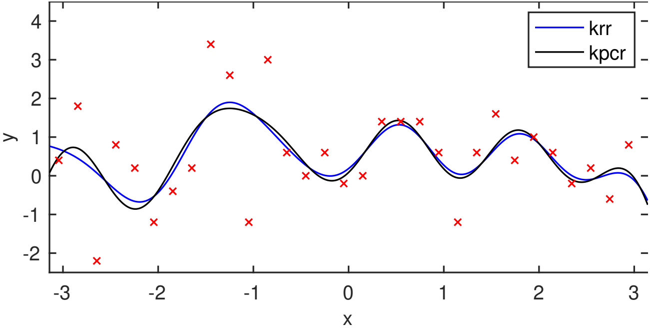

Figure 2: RBF kernel with Kernel Ridge Regression and Kernel PCR

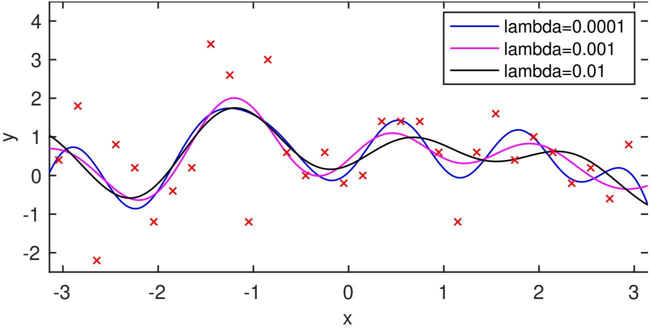

Figure 3: RBF kernel with Kernel PCR

New kernels from old kernels

- Suppose k_1 and k_2 are positive definite kernel functions.
- Is $\mathsf{k}(x, x') = \mathsf{k}_1(x, x') + \mathsf{k}_2(x, x')$ a positive definite kernel function?
- Is $\mathsf{k}(x, x') = a\,\mathsf{k}_1(x, x')$ (for $a \geq 0$) a positive definite kernel function?
- Is $\mathsf{k}(x, x') = \mathsf{k}_1(x, x')\,\mathsf{k}_2(x, x')$ a positive definite kernel function?

Postscript

- Problem with kernel methods when n is large
  - Kernel matrix K is of size n^2
  - Time for prediction generally $\propto n$
- Some possible solutions:
  - Nystrom approximations
  - Find other ways to make $\hat{\alpha}$ sparse
  - Random Fourier features