Kernels & the Dual Form

November 19, 2019

1 Introduction to Kernels

- One main idea behind using kernels is to go into a higher dimensional space where it might be easier to segment or analyze the structure of the data.
- Inner product: Let H be vector space over $\mathbb{R}$. A function $< \cdot, \cdot >_H$: $H \times H \to \mathbb{R}$ is an inner product in H if
  1. $\langle a_1 f_1 + a_2 f_2, g\rangle_H = a_1 \langle f_1, g\rangle_H + a_2 \langle f_2, g\rangle_H$
  2. $\langle f, g\rangle_H = \langle g, f\rangle_H$
  3. $\langle f, f\rangle_H \geq 0$ and $\langle f, f\rangle_H = 0 \iff f = 0$
- Essentially, a Hilbert space is a (complete metric) space where an inner product is defined.
- **Kernel:** Let X be a non-empty set. A function $k : X \times X \to \mathbb{R}$ is called a kernel if there exists an $\mathbb{R}$-Hilbert space and a map $\phi : X \to H$ such that $\forall x, x' \in X$,

$$k(x, x') = \langle \phi(x), \phi(x')\rangle \tag{1}$$

- A very common kernel to use is the radial basis function kernel:

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \left\|\mathbf{x} - \mathbf{x}'\right\|^2) \tag{2}$$

So, is this an inner product in some space? Consider the one-dimensional case with $\gamma = 1$, $k(x, x' = \exp(-(x - x')^2)$ *using Taylor Series expansion*

$$\begin{aligned} k(x, x') &= \exp(-\left\|x - x'\right\|^2) && (3) \\ &= \exp(-x^2) \exp(-x'^2) \exp(2xx') && (4) \end{aligned}$$

Recall: The Taylor series expansion of $f(x)$ around a is $f(x) = f(a) + f'(a)(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \frac{f^{(3)}(a)}{3!}(x - a)^3 + ... + \frac{f^{(n)}(a)}{n!}(x - a)^n + ...$
So, the Taylor series expansion of $\exp(x)$ around 0 is $\left[1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + ...\right]$

$$k(x, x') = \exp(-x^2)\exp(-x'^2)\sum_{k=0}^{\infty}\frac{2^k x^k x'^k}{k!} \text{ using Taylor Series expansion} \tag{5}$$

So, how does this show that the radial basis function is an inner product in some space?

- You can construct kernels from other kernels (e.g. sum of two kernels is a kernel, product of two kernels is a kernel)

2 The Kernel Trick

- We introduced *kernel functions* and mentioned the *kernel trick*
- What is a kernel? $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T\phi(\mathbf{x}')$, an inner product of the feature space mapping of $\mathbf{x}$ and $\mathbf{x}'$
- Easiest example: linear kernel, $\phi(\mathbf{x}) = \mathbf{x}$ so, $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$
- More interesting example: $\phi(\mathbf{x}) = \left[x_1^2, \sqrt{2}x_1x_2, x_2^2\right]$ where $\mathbf{x} = [x_1, x_2]$
- Recall: the motivation is to have a non-linear mapping into a feature space (usually, a higher dimensional space) where the data is hopefully easier to classify and analyze
- When your method can make use of the *kernel trick*, you do not need to explicitly deal with the feature space mapping. You only need to deal with kernels in the original space.
- *What is the kernel trick?* Only operate on kernel functions, i.e., the inner products and skip the feature space representation directly
- : Consider:

$$J(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(\mathbf{w}^T\phi(\mathbf{x}_n) - t_n\right)^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} \tag{6}$$

- We've seen this before, remember? We want to minimize $J(\mathbf{w})$ with respect to $\mathbf{w}$:

$$\frac{\partial J}{\partial \mathbf{w}} = \sum_{n=1}^{N}\left(\mathbf{w}^T\phi(\mathbf{x}_n) - t_n\right)\phi(\mathbf{x}_n) + \lambda\mathbf{w} = 0 \tag{7}$$

$$\mathbf{w} = -\frac{1}{\lambda}\sum_{n=1}^{N}\left(\mathbf{w}^T\phi(\mathbf{x}_n) - t_n\right)\phi(\mathbf{x}_n) \tag{8}$$

- Lets call the following a_n: $-\frac{1}{\lambda}\left(\mathbf{w}^T\phi(\mathbf{x}_n) - t_n\right) = a_n$

- So,

$$\mathbf{w} = \sum_{n=1}^{N} a_n \phi(\mathbf{x}_n) = \Phi^T \mathbf{a} \tag{9}$$

where Φ is the *design matrix* whose n^{th} row is given by $\phi(\mathbf{x}_n)$

- So, we now can use $\mathbf{w} = \Phi^T \mathbf{a}$ to rewrite $J(\mathbf{w})$:

$$J(\mathbf{w}) = \frac{1}{2}\left(\mathbf{w}^T\Phi^T - \mathbf{t}\right)\left(\mathbf{w}^T\Phi^T - \mathbf{t}\right)^T - \frac{\lambda}{2}\mathbf{w}^t\mathbf{w} \tag{10}$$

$$= \frac{1}{2}\mathbf{w}^T\Phi^T\Phi\mathbf{w} - \mathbf{w}^T\Phi^T\mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{t} + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} \tag{11}$$

- Plug in for $\mathbf{w} = \Phi^T \mathbf{a}$

$$= \frac{1}{2}\left(\Phi^T\mathbf{a}\right)^T\Phi^T\Phi\Phi^T\mathbf{a} - \left(\Phi^T\mathbf{a}\right)^T\Phi^T\mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{t} + \frac{\lambda}{2}\left(\Phi^T\mathbf{a}\right)^T\Phi^T\mathbf{a} \tag{12}$$

$$= \frac{1}{2}\mathbf{a}\Phi\Phi^T\Phi\Phi^T\mathbf{a} - \mathbf{a}\Phi\Phi^T\mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{t} + \frac{\lambda}{2}\mathbf{a}\Phi\Phi^T\mathbf{a} \tag{13}$$

- Let $\mathbf{K} = \Phi\Phi^T$ be the *Gram Matrix*. Note that the Gram Matrix is symmetric
- $K_{nm} = \phi(\mathbf{x}_n)^T\phi(\mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m)$

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}\mathbf{K}\mathbf{K}\mathbf{a} - \mathbf{a}\mathbf{K}\mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{t} + \frac{\lambda}{2}\mathbf{a}\mathbf{K}\mathbf{a} \tag{14}$$

$$\frac{\partial J(\mathbf{a})}{\partial \mathbf{a}} = \mathbf{K}\mathbf{K}\mathbf{a} - \mathbf{K}\mathbf{t} + \lambda\mathbf{K}\mathbf{a} \tag{15}$$

$$\mathbf{a} = \left(\mathbf{K} + \lambda\mathbf{I}\right)^{-1}\mathbf{t} \tag{16}$$

- Since $\mathbf{w}^T = \mathbf{a}^T\Phi$,

$$y(\mathbf{x}) = \mathbf{a}^T\Phi\phi(\mathbf{x}) \tag{17}$$

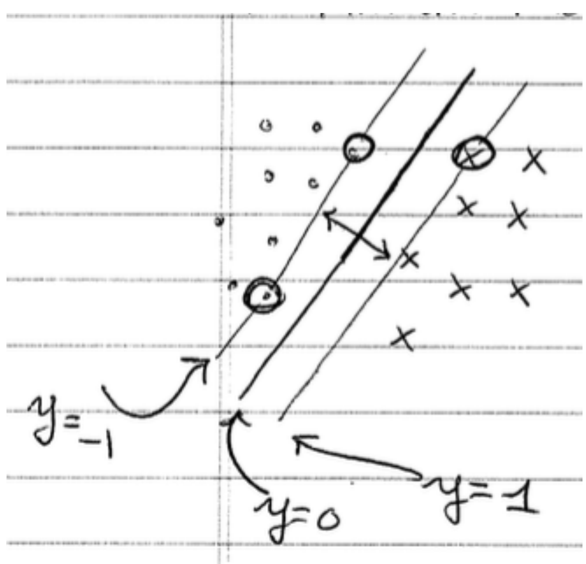
$$= \mathbf{K}(\mathbf{x})^T\left(\mathbf{K} + \lambda\mathbf{I}\right)^{-1}\mathbf{t} \tag{18}$$

where $\mathbf{K}(\mathbf{x}) = \Phi\phi(\mathbf{x})$

- The Dual formulation shows you do not need to deal with the feature space mapping at all
- *Mercer's Theorem* - 1980 - said if you have $\mathbf{K}(\mathbf{x}, \mathbf{y})$ and it is a positive definite matrix, then, there is an equivalent $\phi(\mathbf{x})^T\phi(\mathbf{y})^T$ in some Hilbert space (i.e., a vector space where an inner product is defined - for our purposes)
- *What's the big deal?* Well, the feature space mapping can be infinite dimensional (e.g., RBF kernel). So, you can do analysis in an infinite dimensional feature space while only needing to compute kernel functions.

3 Introduction to Support Vector Machines

- SVMs are Maximum Margin Classifiers
- Two class classification problems: $y(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b$



- We cannot directly compute $\phi(\mathbf{x})$ for all mappings, we want to use a kernel trick. So, we have to write up a dual representation of the problem in terms of K matrices
- We will start with the case where the $\phi(x)$ are linearly separable in the kernel space
- Note: the SVM finds a linear decision boundary in the feature space. But since we can do non-linear transformations to get to the feature space, the descision boundary can be non-linear in the feature space.