

★ Kernels

⇒ We have a problem in which input is x .

⇒ We consider performing regression with the features x, x^2 & x^3 to obtain a cubic function.

⇒ To distinguish between these two set of variables, we'll call the "original" input value the **input attributes** of a problem.

⇒ When that is mapped to some new set of quantities that are passed to the learning algorithm, we'll call those new quantities the **input features**.

⇒ We will also let ϕ denote the **feature mapping**, which maps from the attributes to the features.

$$\phi(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}$$

⇒ Rather than applying SVMs using the original input attributes x , we may instead want to learn using some features $\phi(x)$

↳ To do so, we simply need to go over our previous algorithm, & replace x everywhere in it with $\phi(x)$

⇒ Since the algorithm can be written entirely in terms of the inner products $\langle x, z \rangle$, this means that we could replace all those inner product with $\langle \phi(x), \phi(z) \rangle$.

⇒ Specifically, given a feature mapping ϕ , we define the ~~spanning~~ corresponding **Kernel** to be:

$$K(x, z) = \phi(x)^T \phi(z)$$

⇒ Then, everywhere we previously had $\langle x, z \rangle$ in our algorithm, we could simply replace it with $K(x, z)$, and our algorithm would now be learning using the features ϕ .

⇒ Let's see an example. Suppose $x, z \in \mathbb{R}^n$ and consider

$$K(x, z) = (x^T z)^2$$

$$K(x, z) = \left(\sum_{i=1}^n x_i z_i \right) \left(\sum_{j=1}^n x_j z_j \right)$$

$$= \sum_{i=1}^n \sum_{j=1}^n x_i x_j z_i z_j$$

$$= \sum_{i,j=1}^n (x_i x_j) (z_i z_j)$$

⇒ Thus, we see that $k(x, z) = \phi(x)^T \phi(z)$, where the feature mapping ϕ is given by: (for $n=3$)

$$\phi(x) = \begin{bmatrix} x_1 x_1 \\ x_1 x_2 \\ x_1 x_3 \\ x_2 x_1 \\ x_2 x_2 \\ x_2 x_3 \\ x_3 x_1 \\ x_3 x_2 \\ x_3 x_3 \end{bmatrix}$$

⇒ Note that whereas calculating the high-dimensional $\phi(x)$ requires $O(n^2)$ time, finding $K(x, z)$ takes only $O(n)$ time.

⇒ For a related kernel, also consider

$$K(x, z) = (x^T z + c)^2$$

⇒ This corresponds to feature mapping:

$$\phi(x) = \begin{bmatrix} x_1 x_1 \\ x_1 x_2 \\ x_1 x_3 \\ x_2 x_1 \\ x_2 x_2 \\ x_2 x_3 \\ x_3 x_1 \\ x_3 x_2 \\ x_3 x_3 \\ \sqrt{c} x_1 \\ \sqrt{c} x_2 \\ \sqrt{c} x_3 \\ c \end{bmatrix} \quad \text{(for } n=3\text{)}$$

⇒ More broadly the kernel $K(x, z) = (x^T z + c)^d$ corresponds to a feature mapping to an $\binom{m+d}{d}$ feature space.

↳ Corresponding of all monomials of the form $x_{i_1} x_{i_2} \dots x_{i_k}$ that are up to order d .

⇒ Intuitively:

- If $\phi(x)$ & $\phi(z)$ are close together, then we might expect $K(x, z) = \phi(x)^T \phi(z)$ to be large.
- If $\phi(x)$ & $\phi(z)$ are far apart (say nearly orthogonal to each other) then $K(x, z) = \phi(x)^T \phi(z)$ will be small.

⇒ So, we can think of $K(x, z)$ as some measure of how similar are $\phi(x)$ and $\phi(z)$ or of how similar are x & z .

⇒ Given some function K , how can we tell if it's a valid kernel?

⇒ Suppose for now that K is indeed a valid kernel corresponding to some feature mapping ϕ .

- Now consider a finite set of m points $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$

- Let a square, $m \times m$ matrix K be defined so that (i, j) entry is given by $K_{ij} = K(x^{(i)}, x^{(j)})$

- This matrix is called **Kernel Matrix**.

⇒ Now, if K is a valid kernel, then $K_{ij} = K_{ji}$ and hence K must be symmetric.

\Rightarrow Let $\phi_k(x)$ denote the k^{th} coordinate of the vector $\phi(x)$, we find that for any vector z we have

$$\begin{aligned} z^T K z &= \sum_i \sum_j z_i k_{ij} z_j \\ &= \sum_i \sum_j z_i \phi(x^{(i)})^T \phi(x^{(j)}) z_j \\ &= \sum_i \sum_j z_i \sum_k \phi_k(x^{(i)}) \phi_k(x^{(j)}) z_j \\ &= \sum_k \sum_i \sum_j z_i \phi_k(x^{(i)}) \phi_k(x^{(j)}) z_j \\ &= \sum_k \left(\sum_i z_i \phi_k(x^{(i)}) \right)^2 \geq 0 \end{aligned}$$

\Rightarrow Since z was arbitrary, this shows that K is Positive Semi-definite.

\Rightarrow If K is a valid Kernel, then the corresponding Kernel matrix $K \in \mathbb{R}^{n \times n}$ is symmetric positive Semidefinite.

\hookrightarrow This turns out to be not only a necessary, but also a sufficient condition for K to be a valid Kernel.
(also called a Mercer Kernel)

$$K(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right)$$

{ Gaussian Kernel }

Theorem (Mercer): Let $K: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ be given. Then for K to be a valid (Mercer) Kernel, it is necessary and sufficient that for any $\{x^{(1)} \dots x^{(m)}\}$, ($m < \infty$), the corresponding Kernel matrix is symmetric positive semi-definite.

* Regularization and the non-separable case

- \Rightarrow The derivation of the SVM as presented so far assumed that the data is linearly separable.
- \Rightarrow While mapping data to a high-dimensional feature space via ϕ does generally increase the likelihood that the data is separable, but we can't guarantee that it always will be so.
- \Rightarrow It is not clear that finding a separating hyperplane is exactly what we'd want to do, since that might be susceptible to outliers.
- \Rightarrow To make the algorithm work for non-linearly separable datasets as well as be less sensitive to outliers, we reformulate our optimization (using L_1 regularization) as follows:

$$\min_{\gamma, \omega, b} \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^m \xi_i$$

$$\text{s.t. } y^{(i)} (\omega^T x^{(i)} + b) \geq 1 - \xi_i, \quad i = 1, \dots, m$$

$$\xi_i \geq 0 \quad i = 1, \dots, m$$