Kernel functions (Quick recap)

Feature Mapping

$$\underline{\tilde{x}} = \underline{\phi}(\underline{x})$$

$\underline{\tilde{x}}$: vector in the transformed space e.g., 3-D

$\underline{\phi}$: vector function

$\underline{x}$: vector in the original space e.g., 2-D

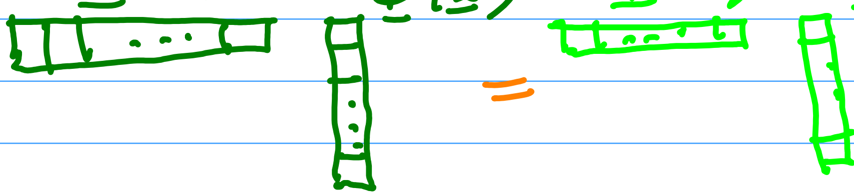
Kernel function:

$$\underbrace{k(\underline{x}, \underline{x}')}_{\text{scalar}} \triangleq \underline{\phi}^T(\underline{x})\,\underline{\phi}(\underline{x}')$$

$\underline{x}, \underline{x}'$: two vectors in the original space

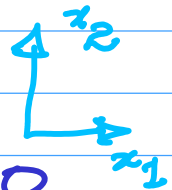
$\underline{x}$: first, $\underline{x}'$: second (original space)

$$\underbrace{\underline{\phi}^T(\underline{x})\,\underline{\phi}(\underline{x}') = \underline{\phi}^T(\underline{x}')\,\underline{\phi}(\underline{x})}_{\text{symmetric}}$$



Some common examples of Kernels

1) **linear kernel** $\underline{\phi}(\underline{x}) = \underline{x}$ itself.



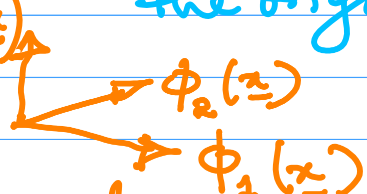
e.g., linear decision boundary $\underbrace{\underline{w}^T \underline{x} + b = 0}_{\text{linearity in the original space}}$

In the transformed domain

$$\underline{\tilde{w}}^T \underline{\phi}(\underline{x}) + b = 0$$

$\underline{\tilde{w}}$: appropriate dimension

linearity in the transformed space.



(for a linear kernel: the transformation is an identity transformation)

② **Stationary kernel** $k(\underline{x}, \underline{x}') = k(\underline{x} - \underline{x}')$

invariant to translations in the input pattern space.

e.g., music information retrieval
male, female voice ('pitch transposition')

②a **RBF** $k(\underline{x}, \underline{x}') = k(\| \underline{x} - \underline{x}' \|)$
(Radial Basis Function)

Key point: '**Kernel Trick**'

(*) Kernel function: Why? Feature transform -
decision boundary: linear in a higher dimension, or at least the linear decision boundary in the transformed space could give a better separation as compared to the original space.

(*) Computation: 'trick': make computations in the lower dimensional space itself.

Dual Representations (Regression)

Earlier example: SVM: classification

(*) Many linear models for classification and regression: which can be reformulated in terms of a dual representation in which kernels arise naturally.

Regularised Linear Regression

$$J(\underline{w}) = \frac{1}{2} \sum_{i=1}^{N} \{ \underbrace{\underline{w}^T \Phi(\underline{x})}_{\text{model (linear)}} - \underbrace{t_i}_{\text{target value}} \}^2 + \underbrace{\frac{\lambda}{2} \underline{w}^T \underline{w}}_{\text{Regulariser}}$$

"cosmetic purposes"

"cosmetic purposes"

function, to minimise

in $\Phi(x)$ space

difference b/w the target and the model

"fidelity" term

Summation: for all training data points

+/− discrepancies b/w the model and the target are treated in the same way

Lagrange multiplier

Regulariser → drive the system to favour low weights, unless it is supported by the data (fidelity)

$$\boxed{\text{Recap} \quad \frac{\partial (\underline{x}^T \underline{a})}{\partial \underline{x}} = \underline{a} \quad \equiv \quad \frac{\partial (\underline{a}^T \underline{x})}{\partial \underline{x}} = \underline{a}}$$

optimum ⟶ minimum

$$\frac{\partial J(\underline{w})}{\partial \underline{w}} = 0$$

$\lambda \geq 0$ Lagrange Multiplier

$$\frac{\partial J(\underline{w})}{\partial \underline{w}} = 0 \Rightarrow \frac{1}{\cancel{2}} \times \cancel{2} \sum_{i=1}^{N} \left\{ \underline{w}^T \underline{\phi}(\underline{x}_i) - t_i \right\} \underline{\phi}(\underline{x}_i) + \frac{\lambda}{\cancel{2}} \cancel{2}\, \underline{w} = 0$$

$$\Rightarrow \underline{w} = \left(-\frac{1}{\lambda}\right) \sum_{i=1}^{N} \underbrace{\left\{ \underline{w}^T \underline{\phi}(\underline{x}_i) - t_i \right\}}_{a_i} \underline{\phi}(\underline{x}_i)$$

$$a_i \triangleq \left(-\frac{1}{\lambda}\right) \left\{ \underline{w}^T \underline{\phi}(\underline{x}_i) - t_i \right\}$$

$$\Rightarrow \underbrace{\underline{w} = \sum_{i=1}^{N} a_i\, \underline{\phi}(\underline{x}_i)}_{\text{inner product representation}}$$

$$\underbrace{\left[\underline{\phi}(\underline{x}_1)\ \underline{\phi}(\underline{x}_2) \cdots \underline{\phi}(\underline{x}_N) \right]}_{N} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix}}_{\underline{a}}$$

M M M

Φ : transpose of the "Design Matrix" Φ

$$\Phi \text{ was } \begin{array}{c} \\ N \end{array} \begin{bmatrix} \underline{\phi}^T(\underline{x}_1) \\ \vdots \\ \underline{\phi}^T(\underline{x}_N) \end{bmatrix} \quad \begin{array}{c} M \\ \\ M \end{array}$$

N M Φ

$$\Rightarrow \boxed{\underline{w} = \Phi^T \underline{a}}$$

Reformulate the problem in terms of $\underline{a}$, instead of $\underline{w}$

Substitute $\underline{w} = \Phi^T \underline{a}$ into the expression for $J(\underline{w})$ to try to eliminate w altogether, and attempt to replace $J(\underline{w})$ with an expression which involves $\underline{a}$ alone, at the optimum.

$$J(\underline{a}) = \frac{1}{2}\sum_{i=1}^{N}\left\{(\Phi^T\underline{a})^T \underline{\phi}(\underline{x}_i) - t_i\right\}^2 + \frac{\lambda}{2}(\Phi^T\underline{a})^T(\Phi^T\underline{a})$$

$$= \frac{1}{2}\sum_{i=1}^{N}\left\{(\underline{a}^T\Phi)\underline{\phi}(\underline{x}_i)\right\}^2 - \frac{1}{2}2\sum_{i=1}^{N}(\underline{a}^T\Phi)\underline{\phi}(\underline{x}_i)t_i$$

$$+ \frac{1}{2}\sum_{i=1}^{N} t_i^2 + \underbrace{\frac{\lambda}{2}\underline{a}^T\Phi\Phi^T\underline{a}}$$

fourth term, is in its final form (we will see this later!)

The third term $= \frac{1}{2}\sum_{i=1}^{N} t_i^2$ we write this as an inner product

$$= \frac{1}{2}\left[t_1\ t_2\ \cdots\ t_N\right]\begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix} = \frac{1}{2}\underline{t}^T\underline{t}$$

The second term $= -\sum_{i=1}^{N}\boxed{\underline{a}^T\Phi}\,\underline{\phi}(\underline{x}_i)t_i$

$$= -(\underline{a}^T\Phi)\sum_{i=1}^{N} t_i\,\underline{\phi}(\underline{x}_i)$$

$$= -(\underline{a}^T\Phi)\underbrace{\left[\underline{\phi}(\underline{x}_1)\ \underline{\phi}(\underline{x}_2)\cdots\underline{\phi}(\underline{x}_N)\right]}_{\Phi^T}\underbrace{\begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}}_{\underline{t}} = -\underline{a}^T\Phi\Phi^T\underline{t}$$

Basic Philosophy

(*) In some cases, formulating an original (primal) problem completely in terms of other variables (dual) is possible. There is no guarantee that given a primal problem, it should be possible to formulate a dual one in terms of another variable

(*) Even if one is able to formulate a dual problem, there is no guarantee that the dual problem may have a 'better' solution: better in term of the computational complexity, attractiveness in terms of a kernel trick.

Recap:

$$J(\underline{w}) = \frac{1}{2}\sum_{i=1}^{N}\{\underline{w}^T\underline{\phi}(\underline{x}) - t_i\}^2 + \frac{\lambda}{2}\underline{w}^T\underline{w}$$

$$\frac{\partial J(\underline{w})}{\partial \underline{w}} = 0 \xrightarrow{\text{gave}} \underline{w} = \Phi^T\underline{a}, \quad a_i \triangleq \frac{1}{\lambda}\left[\underline{w}^T\underline{\phi}(\underline{x}_i) - t_i\right]$$

Put in the original expression

$$J(\underline{a}) = \text{1st term} + \text{2nd term} + \text{3rd term} + \text{4th term}$$

$$\frac{1}{2}\sum_{i=1}^{N}\{\underline{a}^T\Phi\,\underline{\phi}(\underline{x}_i)\}^2 \qquad -\underline{a}^T\Phi\Phi^T\underline{t} \qquad \frac{1}{2}\underline{t}^T\underline{t} \qquad \frac{\lambda}{2}\underline{a}^T\Phi\Phi^T\underline{a}$$

s^Ts or s^2 will not give a 'nice' expression

trick: the square of a scalar can be written as ss^T

$$= \frac{1}{2}\sum_{i=1}^{N}\{\underline{a}^T \Phi\, \underline{\phi}(\underline{x}_i)\}\{\underline{a}^T \Phi\, \underline{\phi}(\underline{x}_i)\}^T$$

$$= \frac{1}{2}\sum_{i=1}^{N}\underline{a}^T \Phi\, \underbrace{\underline{\phi}(\underline{x}_i)\,\underline{\phi}^T(\underline{x}_i)\,\Phi^T}\,\underline{a}$$

Take the parts not involved in the summation, outside

$$= \frac{1}{2}\underline{a}^T \Phi \underbrace{\left\{\sum_{i=1}^{N}\underline{\phi}(\underline{x}_i)\underline{\phi}^T(\underline{x}_i)\right\}}_{\text{consider this summation alone}}\Phi^T \underline{a}$$

write as an inner product in one of the two possible ways

$$\underbrace{\underbrace{\left[\underline{\phi}(\underline{x}_1)\ \underline{\phi}(\underline{x}_2)\ \cdots\ \underline{\phi}(\underline{x}_N)\right]}_{\Phi^T}\ \underbrace{\begin{bmatrix}\underline{\phi}^T(\underline{x}_1)\\ \underline{\phi}^T(\underline{x}_2)\\ \vdots \\ \underline{\phi}^T(\underline{x}_N)\end{bmatrix}}_{\Phi}}_{\Phi^T\Phi}$$

$\Rightarrow$ the first term $= \frac{1}{2}\underline{a}^T(\Phi\Phi^T)(\Phi\Phi^T)\underline{a}$

The complete expression at the optimal value becomes

$$J(\underline{a}) = \frac{1}{2}\underline{a}^T(\Phi\Phi^T)(\Phi\Phi^T)\underline{a} - \underline{a}^T(\Phi\Phi^T)\underline{t}$$
$$+ \frac{1}{2}\underline{t}^T\underline{t} + \frac{\lambda}{2}\underline{a}^T(\Phi\Phi^T)\underline{a}$$

We define the Gram Matrix $K \triangleq \Phi \Phi^T$

What is this?

$$\underset{N\times M}{\Phi}\ \underset{M\times N}{\Phi^T} = \underset{N\times M}{\begin{bmatrix} \underline{\phi}^T(\underline{x}_1) \\ \underline{\phi}^T(\underline{x}_2) \\ \vdots \\ \underline{\phi}^T(\underline{x}_N) \end{bmatrix}} \underset{M\times N}{\left[\underline{\phi}(\underline{x}_1)\ \underline{\phi}(\underline{x}_2) \dots \underline{\phi}(\underline{x}_N) \right]}$$

$N\times N$

Now, what is $K(i,j)$? i'th row $\times$ j'th column

$$K(i,j) = \underbrace{\underset{1\times M}{\underline{\phi}^T(\underline{x}_i)}\ \underset{M\times 1}{\underline{\phi}(\underline{x}_j)}}$$

this is symmetric, scalar!

$= k(\underline{x}_i, \underline{x}_j)$ the kernel function

$$K(i,j) = \underline{\phi}^T(\underline{x}_i)\,\underline{\phi}(\underline{x}_j) = k(\underline{x}_i, \underline{x}_j)$$

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

[dual]

optimisation theory $\rightarrow$ $\dfrac{\partial J(\underline{a})}{\partial \underline{a}} = 0$

Use result: for a quadratic form

$$\frac{\partial}{\partial \underline{a}}(\underline{a}^T K \underline{a}) = 2K\underline{a}$$

$$\frac{\partial J(\underline{a})}{\partial \underline{a}} = \frac{1}{2}\cdot 2KK\underline{a} - K\underline{t} + \frac{\lambda}{2}\cdot 2K\underline{a} = 0$$

$$\Rightarrow K\underline{t} = K(K + \lambda I_N)\underline{a}$$

assume K to be invertible $\quad \underline{a} = (K + \lambda I_N)^{-1}\underline{t}$

[please go to p 4]