روش کرنل

Kernel Methods

Consider a dataset of DNA sequences over the alphabet
$\Sigma = \{A, C, G, T\}$

$\varphi(x) = \{P(A), P(C), P(G), P(T)\}, \qquad P(s) = \frac{n_s}{m}, \qquad |x| = m$

$$\mathbf{x} = ACAGCAGTA$$
$$\mathbf{y} = AGCAAGCGAG$$

$$\phi(\mathbf{x}) = (4/9, 2/9, 2/9, 1/9) = (0.44, 0.22, 0.22, 0.11)$$

$$\phi(\mathbf{y}) = (4/10, 2/10, 4/10, 0) = (0.4, 0.2, 0.4, 0)$$

$$\|\phi(\mathbf{x}) - \phi(\mathbf{y})\| = \sqrt{(0.44-0.4)^2 + (0.22-0.2)^2 + (0.22-0.4)^2 + (0.11-0)^2} = 0.22$$

$\mathbf{x} = (x_1, x_2)^T \in \mathbb{R}^2 \quad \phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)^T \in \mathbb{R}^3$

$$\mathbf{x} = (5.9, 3)^T$$

$$\phi(\mathbf{x}) = (5.9^2, 3^2, \sqrt{2}\cdot 5.9 \cdot 3)^T = (34.81, 9, 25.03)^T$$

$$\mathbf{K} = \begin{pmatrix} K(\mathbf{x}_1,\mathbf{x}_1) & K(\mathbf{x}_1,\mathbf{x}_2) & \cdots & K(\mathbf{x}_1,\mathbf{x}_n) \\ K(\mathbf{x}_2,\mathbf{x}_1) & K(\mathbf{x}_2,\mathbf{x}_2) & \cdots & K(\mathbf{x}_2,\mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ K(\mathbf{x}_n,\mathbf{x}_1) & K(\mathbf{x}_n,\mathbf{x}_2) & \cdots & K(\mathbf{x}_n,\mathbf{x}_n) \end{pmatrix}$$

where $K : \mathcal{I} \times \mathcal{I} \to \mathbb{R}$ is a *kernel function*

$$\boxed{K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)} \equiv \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$$

We shall see that many data mining methods can be kernelized

kernel trick, that is, show that the analysis task requires only dot products $\varphi(x_i)^T \varphi(x_j)$ in feature space

kernel methods allow much more flexibility, as we can just as easily perform non-linear analysis by employing nonlinear kernels, or we may analyze (non-numeric) complex objects without explicitly constructing the mapping $\varphi(x)$.

Kernel Matrix

The function K is called a positive semidefinite kernel if and only if it is symmetric: $K(\mathbf{x}_i, \mathbf{x}_j) = K(\mathbf{x}_j, \mathbf{x}_i)$

and the corresponding kernel matrix **K** for any subset $D \subset I$ is positive semidefinite, that is,

$$\mathbf{a}^T\mathbf{K}\mathbf{a} \geq 0, \text{ for all vectors } \mathbf{a} \in \mathbb{R}^n$$

$$\begin{aligned}
\mathbf{a}^T\mathbf{K}\mathbf{a} &= \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j K(\mathbf{x}_i, \mathbf{x}_j) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) \\
&= \left(\sum_{i=1}^{n} a_i \phi(\mathbf{x}_i)\right)^T \left(\sum_{j=1}^{n} a_j \phi(\mathbf{x}_j)\right) \\
&= \|\sum_{i=1}^{n} a_i \phi(\mathbf{x}_i)\|^2 \geq 0
\end{aligned}$$

Reproducing Kernel Map

Empirical Kernel Map

$$\phi(\mathbf{x}) = \mathbf{K}^{-1/2} \cdot \Big(K(\mathbf{x}_1, \mathbf{x}), K(\mathbf{x}_2, \mathbf{x}), \ldots, K(\mathbf{x}_n, \mathbf{x})\Big)^T \in \mathbb{R}^n$$

Mercer Kernel Map

Because K is a symmetric positive semidefinite matrix, it has real and non-negative eigenvalues, and it can be decomposed as follows:

$$\mathbf{K} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \qquad \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$$

$$\begin{aligned}
\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) &= \lambda_1 u_{1i} u_{1j} + \lambda_2 u_{2i} u_{2j} \cdots + \lambda_n u_{ni} u_{nj} \\
&= \sum_{k=1}^{n} \lambda_k u_{ki} u_{kj}
\end{aligned}$$

$$\mathbf{U} = \begin{pmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \\ | & | & & | \end{pmatrix} \qquad \mathbf{\Lambda} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}$$

$$\phi(\mathbf{x}_i) = \sqrt{\mathbf{\Lambda}}\mathbf{U}_i = \left(\sqrt{\lambda_1}\, u_{1i}, \sqrt{\lambda_2}\, u_{2i}, \ldots, \sqrt{\lambda_n}\, u_{ni}\right)^T$$

We now consider two of the most commonly used vector kernels in practice.

Polynomial Kernel

The *homogeneous polynomial kernel*

$$\boxed{K_q(\mathbf{x},\mathbf{y}) = \phi(\mathbf{x})^T\phi(\mathbf{y}) = (\mathbf{x}^T\mathbf{y})^q}$$

linear (with q = 1) and quadratic (with q = 2) kernels

The *inhomogeneous polynomial kernel*

$$\boxed{K_q(\mathbf{x},\mathbf{y}) = \phi(\mathbf{x})^T\phi(\mathbf{y}) = (c+\mathbf{x}^T\mathbf{y})^q}$$

$$= \sum_{k=0}^{q} \binom{q}{k} c^{q-k} \left(\mathbf{x}^T\mathbf{y}\right)^k$$

Gaussian Kernel

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

Norm of a Point

$$\boxed{\|\phi(\mathbf{x})\|^2 = \phi(\mathbf{x})^T\phi(\mathbf{x}) = K(\mathbf{x},\mathbf{x})}$$

$$\|\phi(\mathbf{x})\| = \sqrt{K(\mathbf{x},\mathbf{x})}.$$

Distance between Points

$$\boxed{\begin{aligned}\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 &= \|\phi(\mathbf{x}_i)\|^2 + \|\phi(\mathbf{x}_j)\|^2 - 2\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)\\ &= K(\mathbf{x}_i,\mathbf{x}_i) + K(\mathbf{x}_j,\mathbf{x}_j) - 2K(\mathbf{x}_i,\mathbf{x}_j)\end{aligned}}$$

$$\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\| = \sqrt{K(\mathbf{x}_i,\mathbf{x}_i) + K(\mathbf{x}_j,\mathbf{x}_j) - 2K(\mathbf{x}_i,\mathbf{x}_j)}$$

Mean in Feature Space

$$\boldsymbol{\mu}_\phi = \frac{1}{n}\sum_{i=1}^{n}\phi(\mathbf{x}_i)$$

$$\|\boldsymbol{\mu}_\phi\|^2 = \boldsymbol{\mu}_\phi^T\boldsymbol{\mu}_\phi = \left(\frac{1}{n}\sum_{i=1}^{n}\phi(\mathbf{x}_i)\right)^T\left(\frac{1}{n}\sum_{j=1}^{n}\phi(\mathbf{x}_j)\right) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$$

$$\boxed{\|\boldsymbol{\mu}_\phi\|^2 = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}K(\mathbf{x}_i,\mathbf{x}_j)}$$

Total Variance in Feature Space

$$\boxed{\begin{aligned}\|\phi(\mathbf{x}_i) - \boldsymbol{\mu}_\phi\|^2 &= \|\phi(\mathbf{x}_i)\|^2 - 2\phi(\mathbf{x}_i)^T\boldsymbol{\mu}_\phi + \|\boldsymbol{\mu}_\phi\|^2\\ &= K(\mathbf{x}_i,\mathbf{x}_i) - \frac{2}{n}\sum_{j=1}^{n}K(\mathbf{x}_i,\mathbf{x}_j) + \frac{1}{n^2}\sum_{a=1}^{n}\sum_{b=1}^{n}K(\mathbf{x}_a,\mathbf{x}_b)\end{aligned}}$$

$$\begin{aligned}\sigma_\phi^2 &= \frac{1}{n}\sum_{i=1}^{n}\|\phi(\mathbf{x}_i) - \boldsymbol{\mu}_\phi\|^2\\ &= \frac{1}{n}\sum_{i=1}^{n}\left(K(\mathbf{x}_i,\mathbf{x}_i) - \frac{2}{n}\sum_{j=1}^{n}K(\mathbf{x}_i,\mathbf{x}_j) + \frac{1}{n^2}\sum_{a=1}^{n}\sum_{b=1}^{n}K(\mathbf{x}_a,\mathbf{x}_b)\right)\\ &= \frac{1}{n}\sum_{i=1}^{n}K(\mathbf{x}_i,\mathbf{x}_i) - \frac{2}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}K(\mathbf{x}_i,\mathbf{x}_j) + \frac{n}{n^3}\sum_{a=1}^{n}\sum_{b=1}^{n}K(\mathbf{x}_a,\mathbf{x}_b)\end{aligned}$$

$$\boxed{\sigma_\phi^2 = \frac{1}{n}\sum_{i=1}^{n}K(\mathbf{x}_i,\mathbf{x}_i) - \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}K(\mathbf{x}_i,\mathbf{x}_j)}$$

Centering in Feature Space

$$\bar{\phi}(\mathbf{x}_i) = \phi(\mathbf{x}_i) - \boldsymbol{\mu}_\phi$$

$$\begin{aligned}
\bar{K}(\mathbf{x}_i, \mathbf{x}_j) &= \bar{\phi}(\mathbf{x}_i)^T \bar{\phi}(\mathbf{x}_j) \\
&= (\phi(\mathbf{x}_i) - \boldsymbol{\mu}_\phi)^T (\phi(\mathbf{x}_j) - \boldsymbol{\mu}_\phi) \\
&= \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) - \phi(\mathbf{x}_i)^T \boldsymbol{\mu}_\phi - \phi(\mathbf{x}_j)^T \boldsymbol{\mu}_\phi + \boldsymbol{\mu}_\phi^T \boldsymbol{\mu}_\phi \\
&= K(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{n} \sum_{k=1}^{n} \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_k) - \frac{1}{n} \sum_{k=1}^{n} \phi(\mathbf{x}_j)^T \phi(\mathbf{x}_k) + \|\boldsymbol{\mu}_\phi\|^2 \\
&= K(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{n} \sum_{k=1}^{n} K(\mathbf{x}_i, \mathbf{x}_k) - \frac{1}{n} \sum_{k=1}^{n} K(\mathbf{x}_j, \mathbf{x}_k) + \frac{1}{n^2} \sum_{a=1}^{n} \sum_{b=1}^{n} K(\mathbf{x}_a, \mathbf{x}_b)
\end{aligned}$$

$$\boxed{\bar{\mathbf{K}} = \mathbf{K} - \frac{1}{n}\mathbf{1}_{n\times n}\mathbf{K} - \frac{1}{n}\mathbf{K}\mathbf{1}_{n\times n} + \frac{1}{n^2}\mathbf{1}_{n\times n}\mathbf{K}\mathbf{1}_{n\times n} = \left(\mathbf{I} - \frac{1}{n}\mathbf{1}_{n\times n}\right)\mathbf{K}\left(\mathbf{I} - \frac{1}{n}\mathbf{1}_{n\times n}\right)}$$

Normalizing in Feature Space

$$\phi_n(\mathbf{x}_i)^T \phi_n(\mathbf{x}_j) = \frac{\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)}{\|\phi(\mathbf{x}_i)\| \cdot \|\phi(\mathbf{x}_j)\|} = \cos\theta$$

$$\boxed{\mathbf{K}_n(\mathbf{x}_i, \mathbf{x}_j) = \frac{\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)}{\|\phi(\mathbf{x}_i)\| \cdot \|\phi(\mathbf{x}_j)\|} = \frac{K(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{K(\mathbf{x}_i, \mathbf{x}_i) \cdot K(\mathbf{x}_j, \mathbf{x}_j)}}}$$

Kernels for Complex Objects