Kernels for classification

Kernel regression (last class):

$$\hat{y} = \phi(\boldsymbol{x})^T \boldsymbol{w} \rightarrow \hat{y} = \sum_i \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i)$$

High-dimensional feature transformation

Kernel: measures similarity between x and x_i.

Kernels for binary classification (today):

Classification, after feature map:

$$\hat{y} = \text{sign}(\phi(\boldsymbol{x})^T \boldsymbol{w}) \qquad (1)$$

$\boldsymbol{w}$ depends on $\boldsymbol{x}_1, y_1, \boldsymbol{x}_2, y_2 \ldots$

Kernel methods – re-write above as:

$$\hat{y} = \text{sign}\left(\sum_i \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i)\right) \qquad (2)$$

weighted sum of similarities between feature vector and each training point

Representer Theorem: (1) and (2) are the same, when

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} ||\boldsymbol{\Phi w} - \boldsymbol{y}|| + \lambda ||\boldsymbol{w}||^2$$

$$\boldsymbol{w}^* = (\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda \boldsymbol{I})^{-1}\boldsymbol{\Phi}^T\boldsymbol{y}$$

[Kimeldorf 1970]

$$\boldsymbol{\alpha} = (K + \lambda \boldsymbol{I})^{-1}\boldsymbol{y}$$

where K has ℓ, m entry $K(\boldsymbol{x}_\ell, \boldsymbol{x}_m)$

Example of kernel classification

How do we predict class of $\boldsymbol{x}$?

$$\boldsymbol{x} = \begin{bmatrix} 0.3 \\ 0.1 \end{bmatrix}$$

$$K(\boldsymbol{x}, \boldsymbol{x}_i) = \exp\left(-||\boldsymbol{x} - \boldsymbol{x}_i||^2\right)$$

$$\hat{y} = \text{sign}\left(\sum_i \alpha_i \exp\left(-\left|\left|\begin{bmatrix} 0.3 \\ 0.1 \end{bmatrix} - \boldsymbol{x}_i\right|\right|^2\right)\right)$$