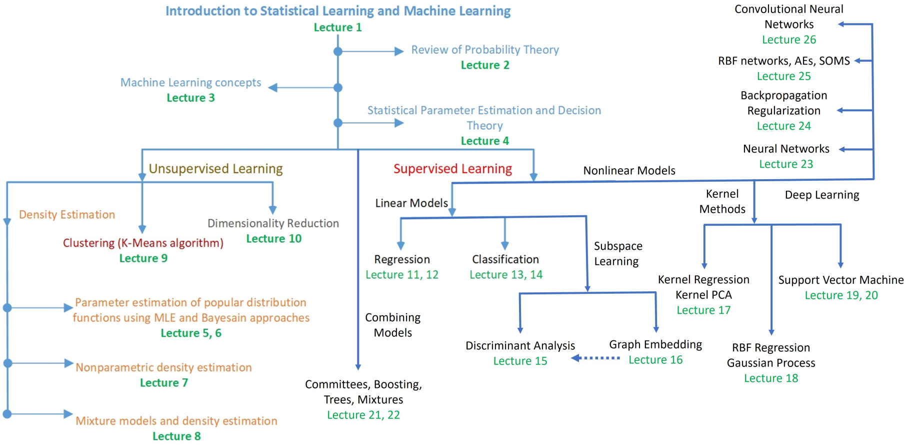
Statistical Learning and Machine Learning

Lecture 17 - Kernel Methods 1

November 1, 2021

Course overview and where do we stand



Why kernel methods?

Main idea in kernel methods:

- The prediction of the model is based on a linear combination of dot-products between data points:

$$\kappa(\mathrm{x}, \mathrm{x}') = \phi(\mathrm{x})^T \phi(\mathrm{x}') \tag{1}$$

  where $\phi(\mathrm{x})$ is an (a-priori defined) (non)linear transformation of the data points x.
- Observation: $\kappa(\cdot, \cdot)$ is a (scalar) value, and this is not depending on the dimensionality of x. $\kappa(\cdot, \cdot)$ is called *kernel function*.
- Whenever a method can be expressed using (only) dot-products between data points, a so-called *dual representation* of the method is defined using $\kappa(\cdot, \cdot)$
- The dot-product between the data points can be replaced by any nonlinear function (satisfying the properties of kernel functions).
- Using a nonlinear kernel function, the nonlinear counterpart of the linear method is defined.

Kernel Regression

Regularized sum-of-squares error:

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

where t_n is the target value for $\phi(\mathrm{x}_n)$ and $\lambda \geq 0$.

Using $\Phi = [\phi(\mathrm{x}_1)^T, \ldots, \phi(\mathrm{x}_N)^T] \in \mathbb{R}^{N\times|\phi|}$ and $\mathrm{t} = [t_1, \ldots, t_N]^T \in \mathbb{R}^N$:

$$\mathcal{J}(\mathrm{w}) = \frac{1}{2}\|\Phi\mathrm{w} - \mathrm{t}\|^2 + \frac{\lambda}{2}\mathrm{w}^T\mathrm{w} \tag{3}$$

Kernel Regression

Setting $\theta \mathcal{J}(\mathrm{w})/\theta \mathrm{w} = 0$, we get:

$$\mathrm{w} = -\frac{1}{\lambda}\sum_{n=1}^{N}\Big(\mathrm{w}^T\phi(\mathrm{x}_n) - t_n\Big)\phi(\mathrm{x}_n) = \sum_{n=1}^{N}\alpha_n\phi(\mathrm{x}_n) = \Phi^T\boldsymbol{\alpha} \qquad (4)$$

where $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_N]^T$:

$$\alpha_n = -\frac{1}{\lambda}\Big(\mathrm{w}^T\phi(\mathrm{x}_n) - t_n\Big). \qquad (5)$$

Thus, we can obtain a *dual representation* for linear regression, leading to a non-linear version of regression.

Kernel Regression

If we substitute $\mathrm{w} = \Phi^T \boldsymbol{\alpha}$ in $\mathcal{J}(\mathrm{w})$:

$$\mathcal{J}(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T \Phi\Phi^T \Phi\Phi^T \boldsymbol{\alpha} - \boldsymbol{\alpha}\Phi\Phi^T \mathrm{t} + \frac{1}{2}\mathrm{t}^T\mathrm{t} + \frac{\lambda}{2}\boldsymbol{\alpha}^T \Phi\Phi^T \boldsymbol{\alpha} \tag{6}$$

We define the *Gram* matrix (or *kernel matrix*) $\mathrm{K} = \Phi\Phi^T \in \mathbb{R}^{N\times N}$ having elements:

$$K_{nm} = \phi(\mathrm{x}_n)^T \phi(\mathrm{x}_m) = \kappa(\mathrm{x}_n, \mathrm{x}_m). \tag{7}$$

Using K, $\mathcal{J}(\boldsymbol{\alpha})$ becomes:

$$\mathcal{J}(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T \mathrm{K}\mathrm{K}\boldsymbol{\alpha} - \boldsymbol{\alpha}^T \mathrm{K}\mathrm{t} + \frac{1}{2}\mathrm{t}^T\mathrm{t} + \frac{\lambda}{2}\boldsymbol{\alpha}^T \mathrm{K}\boldsymbol{\alpha} \tag{8}$$

Setting $\theta\mathcal{J}(\boldsymbol{\alpha})/\theta\boldsymbol{\alpha} = 0$:

$$\boldsymbol{\alpha} = (\mathrm{K} + \lambda \mathrm{I}_N)^{-1}\mathrm{t}. \tag{9}$$

Kernel Regression

For a data point x_* we have:

$$y(\mathrm{x}_*) = \mathrm{w}^T \phi(\mathrm{x}_*) = \boldsymbol{\alpha}^T \Phi \phi(\mathrm{x}_*) = \mathrm{k}(\mathrm{x}_*)^T \left(\mathrm{K} + \lambda \mathrm{I}_N\right)^{-1} \mathrm{t} \qquad (10)$$

where $\mathrm{k}(\mathrm{x}_*) \in \mathbb{R}^N$ has elements equal to $k_n(\mathrm{x}_*) = \kappa(\mathrm{x}_n, \mathrm{x}_*)$.

Note that above we used $|\phi|$ to denote the dimensionality of the vectors $\phi(\mathrm{x}_n)$. Since $|\phi|$ is 'hidden' in the dot-product, it can take any value (even infinity).

Constructing kernels

We can define a kernel function by:

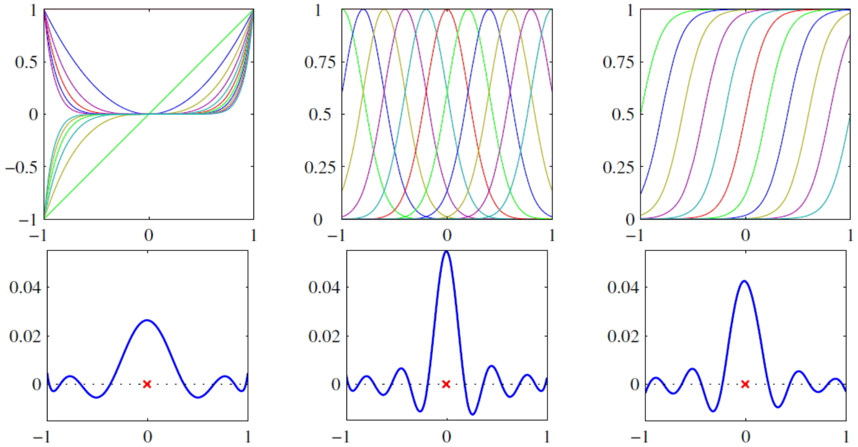
- using a nonlinear function $\phi(\mathrm{x})$ to calculate:

$$\kappa(\mathrm{x}, \mathrm{x}') = \phi(\mathrm{x})^T \phi(\mathrm{x}') \tag{11}$$

- using a set of basis functions $\phi_j(\mathrm{x})$, $j = 1, \ldots, M$ and calculate:

$$\kappa(\mathrm{x}, \mathrm{x}') = \sum_{j=1}^{M} \phi_j(\mathrm{x})^T \phi_j(\mathrm{x}') \tag{12}$$

Constructing kernels



Top: Basis functions using polynomials, Gaussians and logistic sigmoids
Bottom: corresponding $\kappa(x, x')$ plotted as a function of x for $x' = 0$.

Constructing kernels

Alternatively we can construct a kernel function (expressing similarity) directly, like:

$$\kappa(\mathrm{x}, \mathrm{z}) = \left(\mathrm{x}^T \mathrm{z}\right)^2. \tag{13}$$

However, not any function is a kernel function. It needs to correspond to a dot-product:

$$\begin{aligned} \kappa(\mathrm{x}, \mathrm{z}) &= \left(\mathrm{x}^T \mathrm{z}\right)^2 = (x_1 z_1 + x_2 z_2)^2 \\ &= x_1^2 z_1^2 + 2 x_1 z_1 x_2 z_2 + x_2^2 z_2^2 \\ &= (x_1^2, \sqrt{2} x_1 x_2, x_2^2)(z_1^2, \sqrt{2} z_1 z_2, z_2^2) \\ &= \phi(\mathrm{x})^T \phi(\mathrm{z}) \end{aligned} \tag{14}$$

A *necessary and sufficient condition* for a kernel function $\kappa(\mathrm{x}, \mathrm{x}')$ to be a valid kernel function is that the corresponding K is positive semi-definite for any choices of $\mathrm{x}_n, \; n = 1, \ldots, N$.

Constructing kernels

Techniques for Constructing New Kernels.

Given valid kernels $k_1(\mathbf{x}, \mathbf{x}')$ and $k_2(\mathbf{x}, \mathbf{x}')$, the following new kernels will also be valid:

$$
\begin{aligned}
k(\mathbf{x}, \mathbf{x}') &= c k_1(\mathbf{x}, \mathbf{x}') \\
k(\mathbf{x}, \mathbf{x}') &= f(\mathbf{x}) k_1(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') \\
k(\mathbf{x}, \mathbf{x}') &= q\left(k_1(\mathbf{x}, \mathbf{x}')\right) \\
k(\mathbf{x}, \mathbf{x}') &= \exp\left(k_1(\mathbf{x}, \mathbf{x}')\right) \\
k(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}') \\
k(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') k_2(\mathbf{x}, \mathbf{x}') \\
k(\mathbf{x}, \mathbf{x}') &= k_3\left(\boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}')\right) \\
k(\mathbf{x}, \mathbf{x}') &= \mathbf{x}^{\mathrm{T}} \mathbf{A} \mathbf{x}' \\
k(\mathbf{x}, \mathbf{x}') &= k_a(\mathbf{x}_a, \mathbf{x}_a') + k_b(\mathbf{x}_b, \mathbf{x}_b') \\
k(\mathbf{x}, \mathbf{x}') &= k_a(\mathbf{x}_a, \mathbf{x}_a') k_b(\mathbf{x}_b, \mathbf{x}_b')
\end{aligned}
$$

where $c > 0$ is a constant, $f(\cdot)$ is any function, $q(\cdot)$ is a polynomial with nonnegative coefficients, $\boldsymbol{\phi}(\mathbf{x})$ is a function from $\mathbf{x}$ to $\mathbb{R}^M$, $k_3(\cdot, \cdot)$ is a valid kernel in $\mathbb{R}^M$, $\mathbf{A}$ is a symmetric positive semidefinite matrix, $\mathbf{x}_a$ and $\mathbf{x}_b$ are variables (not necessarily disjoint) with $\mathbf{x} = (\mathbf{x}_a, \mathbf{x}_b)$, and k_a and k_b are valid kernel functions over their respective spaces.

Constructing kernels

Commonly used kernel functions:

- Polynomial kernel of order M:

$$\kappa(\mathrm{x},\mathrm{x}') = \left(\mathrm{x}^T\mathrm{x}' + c\right)^M \tag{15}$$

- Gaussian kernel based on Euclidean distance:

$$\kappa(\mathrm{x},\mathrm{x}') = exp\left(-\frac{\|\mathrm{x}-\mathrm{x}'\|^2}{2\sigma^2}\right) \tag{16}$$

- Gaussian kernel based on distance function $D(\cdot,\cdot)$:

$$\kappa(\mathrm{x},\mathrm{x}') = exp\left(-\frac{D(\mathrm{x},\mathrm{x}')^2}{2\sigma^2}\right) \tag{17}$$

Kernel Principal Component Analysis

Consider a *centered* data set $\mathcal{D} = \{\phi(\mathrm{x}_1), \dots, \phi(\mathrm{x}_N)\}$ (which means $\sum_n \phi(\mathrm{x}_n) = 0$).

Reminder: The data covariance matrix C is:

$$\mathrm{C} = \frac{1}{N} \sum_{n=1}^{N} \phi(\mathrm{x}_n)\phi(\mathrm{x}_n)^T \tag{18}$$

and its eigenvectors are given by solving for $\mathrm{C}\mathrm{v}_i = \lambda \mathrm{v}_i$.

Using the above:

$$\begin{aligned} \lambda_i \mathrm{v}_i &= \frac{1}{N} \sum_{n=1}^{N} \phi(\mathrm{x}_n)\Big(\phi(\mathrm{x}_n)^T \mathrm{v}_i\Big) \\ \mathrm{v}_i &= \sum_{n=1}^{N} \alpha_{in}\phi(\mathrm{x}_n) \end{aligned} \tag{19}$$

Kernel Principal Component Analysis

Substituting v_i to C:

$$\frac{1}{N}\sum_{n=1}^{N}\phi(\mathrm{x}_n)\phi(\mathrm{x}_n)^T\sum_{m=1}^{N}\alpha_{im}\phi(\mathrm{x}_m)=\lambda_i\sum_{n=1}^{N}\alpha_{in}\phi(\mathrm{x}_n) \tag{20}$$

Left-multiplying with $\phi(\mathrm{x}_l)$ and re-arranging the summations:

$$\begin{aligned}\frac{1}{N}\sum_{n=1}^{N}\kappa(\mathrm{x}_l,\mathrm{x}_n)\sum_{m=1}^{N}\alpha_{im}\kappa(\mathrm{x}_n,\mathrm{x}_m) &= \lambda_i\sum_{n=1}^{N}\alpha_{in}\kappa(\mathrm{x}_l,\mathrm{x}_n) \qquad (21)\\ \mathrm{K}^2\boldsymbol{\alpha}_i &= \lambda_i N\mathrm{K}\boldsymbol{\alpha}_i \qquad (22)\end{aligned}$$

Thus, we can calculate $\boldsymbol{\alpha}_i,\ i=1,\ldots,D'$, by solving the eigenanalysis problem:

$$\mathrm{K}\boldsymbol{\alpha}_i=\lambda_i N\boldsymbol{\alpha}_i \tag{23}$$

Kernel Principal Component Analysis

Normalization condition:

$$1 = \mathrm{v}_i^T \mathrm{v}_i = \boldsymbol{\alpha}_i^T \mathrm{K} \boldsymbol{\alpha}_i = \lambda_i N \boldsymbol{\alpha}_i^T \boldsymbol{\alpha}_i \tag{24}$$

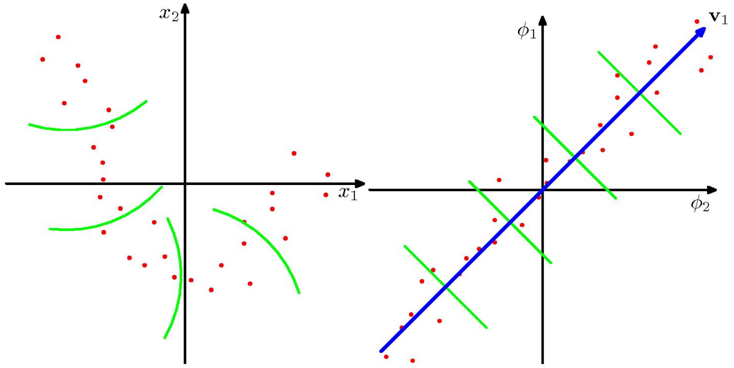
Thus, we need to normalize $\boldsymbol{\alpha}_i$ with a factor of $\sqrt{\lambda_i N}$.

A data point x_* is mapped to the kernel PCA space by:

$$\mathrm{y}_* = \mathrm{A}^T \mathrm{k}(\mathrm{x}_*), \tag{25}$$

where $\mathrm{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}'_D] \in \mathbb{R}^{N \times D'}$ and $\mathrm{k}(\mathrm{x}_*) \in \mathbb{R}^N$ has elements equal to $k_n(\mathrm{x}_*) = \kappa(\mathrm{x}_n, \mathrm{x}_*)$.

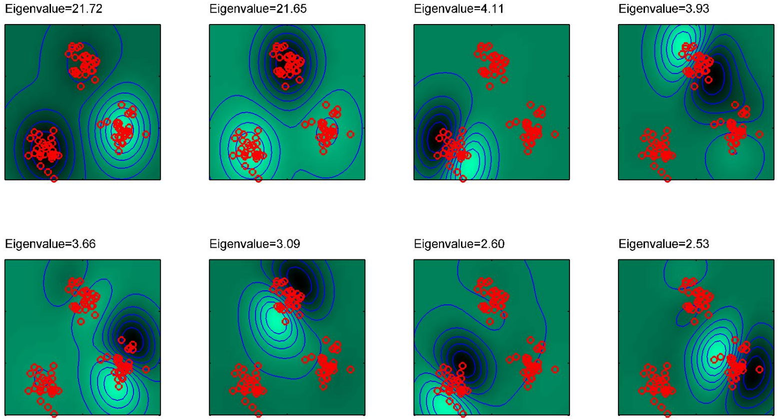
KPCA: Examples



Left: Original data

Right: Transformed data using a $\phi(\mathrm{x})$, and the two principal axes

KPCA: Examples



Kernel PCA with a Gaussian kernel. The contours are lines along which the projection onto the corresponding principal component is constant.

Centering the kernel matrix K

In order to use a *centered* kernel matrix K, we define:

$$\tilde{\phi}(\mathrm{x}_n) = \phi(\mathrm{x}_n) - \frac{1}{N}\sum_{l=1}^{N}\phi(\mathrm{x}_l) \tag{26}$$

and

$$\begin{aligned}\tilde{K}_{nm} &= \tilde{\phi}(\mathrm{x}_n)^T\tilde{\phi}(\mathrm{x}_m) \\ &= \kappa(\mathrm{x}_n,\mathrm{x}_m) - \frac{1}{N}\sum_{l=1}^{N}\kappa(\mathrm{x}_l,\mathrm{x}_m) \\ &- \frac{1}{N}\sum_{l=1}^{N}\kappa(\mathrm{x}_n,\mathrm{x}_l) + \frac{1}{N^2}\sum_{j=1}^{N}\sum_{l=1}^{N}\kappa(\mathrm{x}_j,\mathrm{x}_l)\end{aligned} \tag{27}$$

Thus:

$$\tilde{\mathrm{K}} = \mathrm{K} - 1_N\mathrm{K} - \mathrm{K}1_N + 1_N\mathrm{K}1_N \tag{28}$$

where $1_N \in \mathbb{R}^{N\times N}$ has elements equal to $1/N$.