Machine Learning: 708.063 (VO)

6. Kernel Methods

May 20, 2020

Introduction

- The linear parametric models, we have considered so far can be re-cast into an equivalent dual representation.
- It is based on a linear combination of kernel functions evaluated at the training data points.
- For a fixed nonlinear feature space mapping $\phi(\boldsymbol{x})$, the kernel function is given by

$$k(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^T \phi(\boldsymbol{x}'),$$

  which by definition is a symmetric function in its arguments.
- The concept of kernels has been introduce into the field of pattern recognition already in 1964 by Aizerman et al.
- The simple example of a kernel is given by the linear kernel $\phi(\boldsymbol{x}) = \boldsymbol{x}$ such that $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^T \boldsymbol{x}$.
- The concept of defining a kernel by means of an inner product in a feature space is also known as the kernel trick.
- Whenever an algorithm depends on inner products between features, they can be replaced by an equivalent kernel.
- Allows to solve problems with infinite-dimensional feature transforms $\phi(\boldsymbol{x})$.

Dual representation

- Many linear models for regression and classification can be reformulated in terms of a dual representation incorporating kernels.
- Let us consider a linear regression model minimizing a regularized least squares error function

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

- Can be written in more compact notation as

$$J(\boldsymbol{w}) = \frac{1}{2}\|\Phi\boldsymbol{w} - \boldsymbol{t}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2.$$

- The dual problem is given by

$$D(\boldsymbol{a}) = \frac{1}{2}\boldsymbol{a}^T K\boldsymbol{a} + \frac{\lambda}{2}\|\boldsymbol{a} + \boldsymbol{t}\|^2,$$

where

$$\boldsymbol{w} = -\frac{1}{\lambda}\Phi^T\boldsymbol{a}, \quad K = \Phi\Phi^T, \text{ with } K_{n,m} = \phi(\boldsymbol{x}_n)^T\phi(\boldsymbol{x}_m) = k(\boldsymbol{x}_n, \boldsymbol{x}_m).$$

Discussion

- The solution of the dual problem is given by

$$\boldsymbol{a} = -(I + \frac{1}{\lambda} K)^{-1} \boldsymbol{t}.$$

- Substituting back into the regression function gives

$$y(\boldsymbol{x}) = \phi(\boldsymbol{x})^T \boldsymbol{w} = \boldsymbol{k}(\boldsymbol{x})^T (\lambda I + K)^{-1} \boldsymbol{t},$$

  where $\boldsymbol{k}(\boldsymbol{x}) = (\boldsymbol{k}_1(\boldsymbol{x}), ..., \boldsymbol{k}_N(\boldsymbol{x}))^T$ with $\boldsymbol{k}_n(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}_n)$.
- Observe that the regression function is computed without the need to compute the feature transforms $\phi(\boldsymbol{x})$.
- The original primal problem is a least-squares problem of size $M \times M$.
- The dual problem is again a least squares problem but now of size $N \times N$.
- The problem with the smaller dimension should be considered for solution.

Constructing kernels

- One possibility is to compute kernels from the feature representations

$$k(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^T \phi(\boldsymbol{x}') = \sum_{i=1}^{M} \phi_i(\boldsymbol{x}) \phi_i(\boldsymbol{x}'),$$

- An alternative approach is to construct kernels directly, but we need to verify that the constructed kernels represent valid kernel functions.
- As an example, consider a kernel given by

$$k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^T \boldsymbol{y})^2.$$

- In dimension two it can be written as

$$k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^T \boldsymbol{y})^2 = (x_1 y_1 + x_2 y_2)^2 = (x_1^2 y_1^2 + 2 x_1 y_1 x_2 y_2 + x_2^2 y_2^2$$
$$(x_1^2, \sqrt{2} x_1 x_2, x_2^2)(y_1^2, \sqrt{2} y_1 y_2, y_2^2)^T = \phi(\boldsymbol{x})^T \phi(\boldsymbol{y}).$$

- Hence, the corresponding feature mapping takes the form

$$\phi(\boldsymbol{x}) = (x_1^2, \sqrt{2} x_1 x_2, x_2^2)^T$$

Admissible kernels

- A necessary and sufficient condition for a kernel being admissible is that the corresponding gram matrix with entries $K_{n,m}$ is positive semi-definite.
- A standard technique to construct admissible kernels is to consider operations that preserve the validity of a kernel.
- Let $k_1(\boldsymbol{x}, \boldsymbol{x}')$ and $k_2(\boldsymbol{x}, \boldsymbol{x}')$ be admissible kernels. Then

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

where $c > 0$, f is any function, q is a polynomial with nonnegative coefficients, $\phi(\boldsymbol{x})$ is a function from $\mathbb{R}^D$ to $\mathbb{R}^M$, $k_3(\boldsymbol{x}, \boldsymbol{x}')$ is an admissible kernel on $\mathbb{R}^M$, A is a symmetric and positive definite matrix, $\boldsymbol{x} = (\boldsymbol{x}_a, \boldsymbol{x}_b)$, and k_a, k_b are admissible kernels on the partitioned variables.

Common kernels

- The kernel $k(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^T \boldsymbol{x}')^M$ contains only monomials of order M.
- The kernel $k(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^T \boldsymbol{x}' + c)^M$ contains all terms up to order M
- Another commonly used kernel takes the form

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2/(2\sigma^2)\right),$$

  which is called a Gaussian kernel.
- This is easily shown to be an admissible kernel, as

$$\|\boldsymbol{x} - \boldsymbol{x}'\|^2 = \boldsymbol{x}^T \boldsymbol{x} + (\boldsymbol{x}')^T \boldsymbol{x}' - 2\boldsymbol{x}^T \boldsymbol{x}'.$$

- Hence, the Gaussian kernel can be written as the product

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\boldsymbol{x}^T \boldsymbol{x}/(2\sigma^2)) \exp(-(\boldsymbol{x}')^T \boldsymbol{x}'/(2\sigma^2)) \exp(\boldsymbol{x}^T \boldsymbol{x}'/\sigma^2),$$

  which defines a admissible kernel.

The Nadaraya-Watson model

- Suppose we have training data $\{\boldsymbol{x}_n, t_n\}$ and we consider a kernel density estimator to model the joint distribution

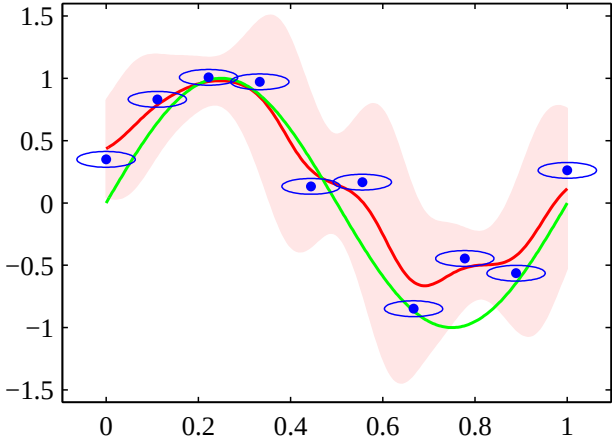
$$p(\boldsymbol{x}, t) = \frac{1}{N} \sum_{n=1}^{N} f(\boldsymbol{x} - \boldsymbol{x}_n, t - t_n),$$

  where $f(\boldsymbol{x}, t)$ is the kernel function centered at each data point $(\boldsymbol{x}_n, t_n)$.
- The regression function $y(\boldsymbol{x}) = \mathbb{E}[t|\boldsymbol{x}]$ can be computed as the conditional mean:

$$y(\boldsymbol{x}) = \sum_n k(\boldsymbol{x}, \boldsymbol{x}_n) t_n, \quad k(\boldsymbol{x}, \boldsymbol{x}_n) = \frac{g(\boldsymbol{x} - \boldsymbol{x}_n)}{\sum_m g(\boldsymbol{x} - \boldsymbol{x}_m)}, \quad g(\boldsymbol{x}) = \int_{-\infty}^{\infty} f(\boldsymbol{x}, t)\, \mathrm{d}t.$$

- This model is known as the Nadaraya-Watson model or kernel regression.
- It gives a higher weight to points $\boldsymbol{x}_n$ close to the new input $\boldsymbol{x}$.

Example



- Illustration of the Nadaraya-Watson kernel regression model.
- The original sine function is shown in green, and the conditional mean in red.
- The data points which define the centers of the Gaussians kernels in blue.
- The shaded area defines the region of plus-minus one standard deviation around the conditional mean.

Gaussian processes

- The framework of Gaussian processes is obtained by extending the role of kernels to probabilistic discriminative models.
- We will define a prior probability directly over the functions $y(\boldsymbol{x})$.
- As a motivation, we start by re-considering the linear regression model

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}),$$

together with a Gaussian prior

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|0, \alpha^{-1} I).$$

- The probability distribution on $\boldsymbol{w}$ also induces a probability distribution over the functions $y(\boldsymbol{x})$.
- Given data $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$, we consider the vector-valued function

$$\boldsymbol{y} = \Phi \boldsymbol{w},$$

where $\boldsymbol{y} = (y(\boldsymbol{x}_1), ..., y(\boldsymbol{x}_N))^T$ and the design matrix Φ has elements $\Phi_{nk} = \phi_k(\boldsymbol{x}_n)$.

Joint distribution

- Since $\boldsymbol{w}$ follows a Gaussian distribution, also $\boldsymbol{y}$ follows a Gaussian distribution with mean and covariance

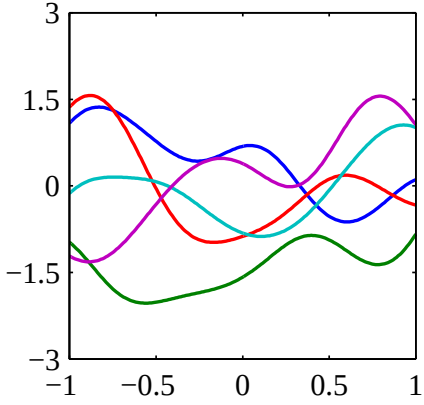
$$\mathbb{E}[\boldsymbol{y}] = \Phi\mathbb{E}[\boldsymbol{w}] = 0, \quad \text{cov}[\boldsymbol{y}] = \mathbb{E}[\boldsymbol{y}\boldsymbol{y}^T] = \Phi\mathbb{E}[\boldsymbol{w}\boldsymbol{w}^T]\Phi^T = \frac{1}{\alpha}\Phi\Phi^T = K,$$

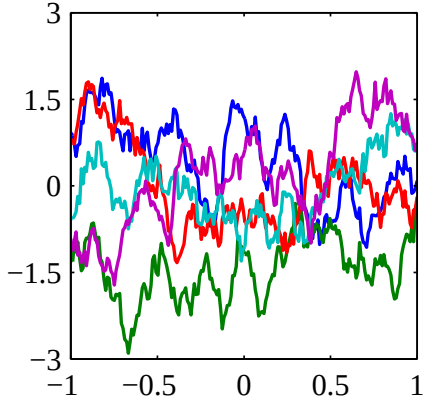
  where K is the Gram matrix with elements

$$K_{n,m} = k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \frac{1}{\alpha}\phi(\boldsymbol{x}_n)^T\phi(\boldsymbol{x}_m),$$

  and $k(\boldsymbol{x}, \boldsymbol{x}')$ is the kernel function.
- We see that a Gaussian process is defined as a distribution over functions $y(\boldsymbol{x})$ such that the set of functions evaluated at $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$ are jointly Gaussian.
- Instead of working with features $\phi(\boldsymbol{x}_n)$, one can also directly work with kernel functions.

Example





- Samples from a Gaussian process with left: a Gaussian kernel and right: an exponential kernel $k(x, x') = \exp(-\theta|x - x'|)$.

Gaussian process for regression

- We now extend the idea of Gaussian processes to regression under the noise model

$$t_n = y_n + \varepsilon_n.$$

- We assume that ε_n is a Gaussian distributed random variable, therefore

$$p(t_n|y_n) = \mathcal{N}(t_n|y_n, \beta^{-1}).$$

- The joint distribution of the target vector $\boldsymbol{t} = (t_1, ..., t_N)^T$ conditioned on the values $\boldsymbol{y} = (y_1, ..., y_N)^T$ is therefore given by an isotropic Gaussian of the form

$$p(\boldsymbol{t}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{t}|\boldsymbol{y}, \beta^{-1} I_N),$$

  where I_N is the N-dimensional unit matrix.
- From the definition of a Gaussian process, the marginal distribution $p(\boldsymbol{y})$ is given by a zero-mean Gaussian with covariance defined by the Gram matrix, that is

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|0, K),$$

  where K is again the Gram matrix.

The marginal distribution

- The marginal distribution is obtained by integrating the joint distribution $p(\boldsymbol{y}, \boldsymbol{t}) = p(\boldsymbol{t}|\boldsymbol{y})p(\boldsymbol{y})$ with respect to $\boldsymbol{y}$:

$$p(\boldsymbol{t}) = \int p(\boldsymbol{t}|\boldsymbol{y})p(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} = \mathcal{N}(\boldsymbol{t}|0, C),$$

where the covariance matrix C has elements

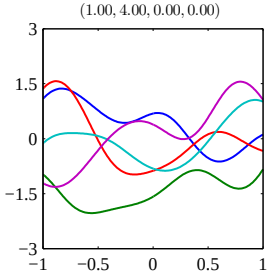
$$C(\boldsymbol{x}_n, \boldsymbol{x}_m) = k(\boldsymbol{x}_n, \boldsymbol{x}_m) + \beta^{-1}\delta_{nm}.$$

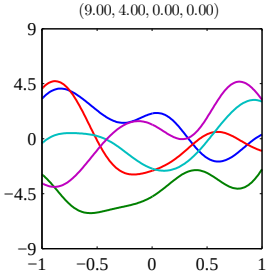
- This covariance reflects the fact that the Gaussian nature of the functions $\boldsymbol{y}(\boldsymbol{x})$ are independent from the Gaussian noise assumption on ε.
- A widely used kernel function for Gaussian process regression is given by

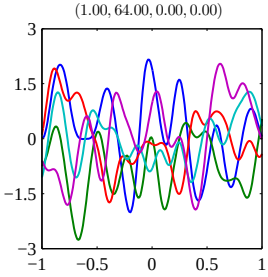
$$k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \theta_0 \exp\left(-\frac{\theta_1}{2}\|\boldsymbol{x}_n - \boldsymbol{x}_m\|^2\right) + \theta_2 + \theta_3 \boldsymbol{x}_n^T \boldsymbol{x}_m,$$

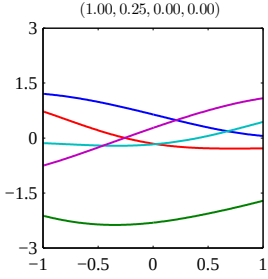
which is a combination of a Gaussian kernel, a constant term and a linear term.

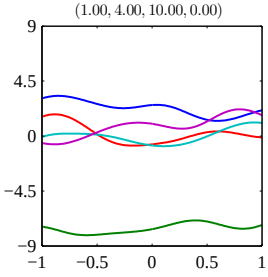
Examples of Gaussian process priors

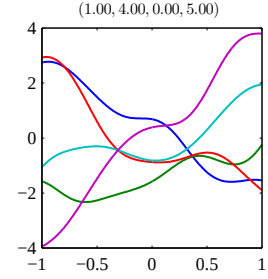




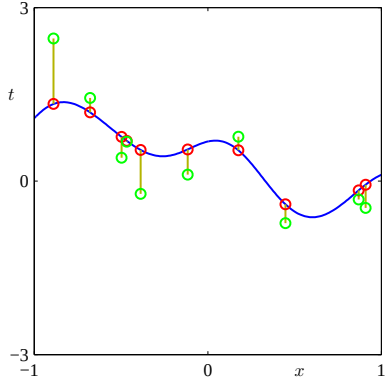








Sampling from a Gaussian process



- The blue curve shows a sample function from the Gaussian process prior $p(\mathbf{y})$.
- The red points are obtained by evaluating the sampled function values y_n on a set of input vectors x_n.
- The green points are obtained by adding independent Gaussian noise to the y_n.

Making predictions

- Based on a training data set $\boldsymbol{t}_N = (t_1, ..., t_N)^T$ and corresponding input vectors $(\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$ the goal is to predict the target variable t_{N+1} for a new input vector $\boldsymbol{x}_{N+1}$.
- For this, we need to evaluate the predictive distribution

$$p(t_{N+1}|\boldsymbol{t})$$

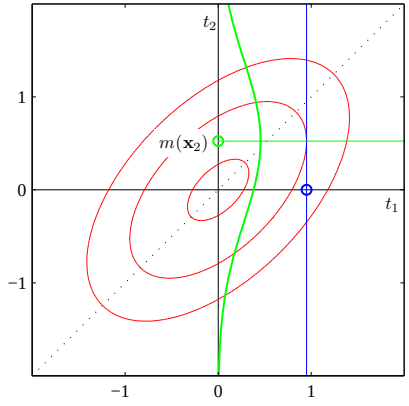
  where the dependence on the $\boldsymbol{x}_n$ is omitted.
- We start by writing down the marginal distribution

$$p(\boldsymbol{t}_{N+1}) = \mathcal{N}(\boldsymbol{t}_{N+1}|0, C_{N+1}), \quad C_{N+1} = \begin{pmatrix} C_N & \boldsymbol{k} \\ \boldsymbol{k}^T & c \end{pmatrix},$$

  where $\boldsymbol{k} = (k(\boldsymbol{x}_1, \boldsymbol{x}_{N+1}), ..., k(\boldsymbol{x}_N, \boldsymbol{x}_{N+1}))^T$ and $c = k(\boldsymbol{x}_{N+1}, \boldsymbol{x}_{N+1}) + \beta^{-1}$.
- The predictive distribution $p(t_{N+1}|\boldsymbol{t})$ is a conditional Gaussian and is computed as

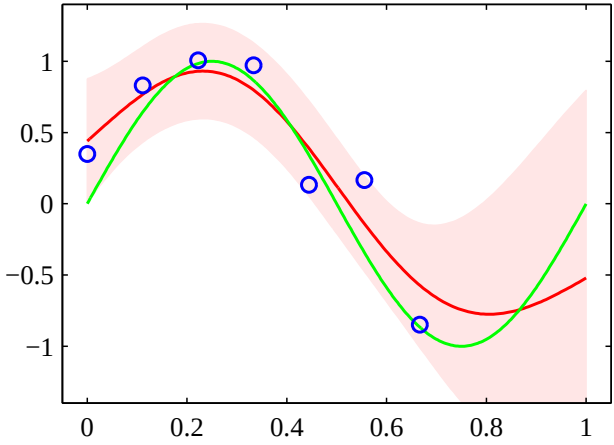
$$p(t_{N+1}|\boldsymbol{t}) = \mathcal{N}(t_{N+1}| \underbrace{\boldsymbol{k}^T C_N^{-1} \boldsymbol{t}}_{m(\boldsymbol{x}_{N+1})}, \underbrace{c - \boldsymbol{k}^T C_N^{-1} \boldsymbol{k}}_{\sigma^2(\boldsymbol{x}_{N+1})})$$

Example



- The horizontal axis is t_1, the vertical axis is t_2.
- The red ellipses are the contours of the joint distribution $p(t_1, t_2)$.
- The blue line t_1 is the training data point
- The conditional distribution $p(t_2|t_1)$ is shown as the green curve.

Gaussian process regression



- The green curve is the true sinusoidal function.
- The blue points are the noisy data points (x_n, t_n).
- The red line is the mean of the Gaussian process.
- The shaded regions corresponds to plus-minus one standard deviation around the mean.