Kernel Methods

Chia-Ping Chen

Professor
National Sun Yat-sen University
Department of Computer Science and Engineering

Machine Learning

Main Points

- Dual Representation
- Constructing Kernels
- Radial Basis Functions
- Gaussian Processes
- Maximum Margin Classifiers
- Relevance Vector Machines

With a kernel method, the prediction for an unseen $\boldsymbol{x}$ is

$$y(\boldsymbol{x}) = \sum_n a_n k(\boldsymbol{x}_n, \boldsymbol{x})$$

- $k(\boldsymbol{x}, \boldsymbol{x}')$ is a **kernel function**
- n runs through a data set
- a_n are trainable parameters

Examples of Kernel Functions

- Gaussian kernel

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

- exponential kernel

$$k(x, x') = \exp(-\theta|x - x'|)$$

Consider a linear regression model $y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x})$. Let the model parameters $\boldsymbol{w}$ be trained with $\mathcal{D} = \{(\boldsymbol{x}_n, t_n)\}_{n=1}^N$ by minimizing

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

where $\lambda > 0$. The solution for $\boldsymbol{w}$ takes the form

$$\boldsymbol{w} = -\frac{1}{\lambda} \sum_{n=1}^{N} \left[\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n) - t_n \right] \boldsymbol{\phi}(\boldsymbol{x}_n) = \boldsymbol{\Phi}^T \boldsymbol{a}$$

where $\boldsymbol{\Phi}$ is the design matrix and $\boldsymbol{a}$ has components

$$a_n = -\frac{1}{\lambda} \left[\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n) - t_n \right]$$

With $\boldsymbol{w} = \boldsymbol{\Phi}^T \boldsymbol{a}$, the prediction function of linear regression is equivalent to a kernel machine

$$\begin{aligned} y(\boldsymbol{x}) &= \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) \\ &= \boldsymbol{a}^T \boldsymbol{\Phi} \boldsymbol{\phi}(\boldsymbol{x}) \\ &= \sum_{n=1}^{N} a_n \boldsymbol{\phi}(\boldsymbol{x}_n)^T \boldsymbol{\phi}(\boldsymbol{x}) \\ &= \sum_{n=1}^{N} a_n k(\boldsymbol{x}_n, \boldsymbol{x}) \end{aligned}$$

with the kernel function related to the basis functions by

$$k(\boldsymbol{x}_n, \boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x}_n)^T \boldsymbol{\phi}(\boldsymbol{x})$$

If we substitute $\boldsymbol{w} = \boldsymbol{\Phi}^T \boldsymbol{a}$, the objective function $J(\boldsymbol{w})$ becomes

$$\begin{aligned} J(\boldsymbol{a}) &= \frac{1}{2}\boldsymbol{a}^T\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{a} - \boldsymbol{a}^T\boldsymbol{\Phi}\boldsymbol{\Phi}^T\mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{t} + \frac{\lambda}{2}\boldsymbol{a}^T\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{a} \\ &= \frac{1}{2}\boldsymbol{a}^T\boldsymbol{K}\boldsymbol{K}\boldsymbol{a} - \boldsymbol{a}^T\boldsymbol{K}\mathbf{t} + \frac{1}{2}\mathbf{t}^T\mathbf{t} + \frac{\lambda}{2}\boldsymbol{a}^T\boldsymbol{K}\boldsymbol{a} \end{aligned}$$

where $\boldsymbol{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T$ is the Gram matrix. Note $\boldsymbol{K}$ is positive definite and

$$K_{nm} = \boldsymbol{\phi}(\boldsymbol{x}_n)^T\boldsymbol{\phi}(\boldsymbol{x}_m) = k(\boldsymbol{x}_n, \boldsymbol{x}_m)$$

Setting the gradient of $J(\boldsymbol{a})$ to zero, we obtain

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

The prediction is

$$y(\boldsymbol{x}) = \sum_{n=1}^{N} a_n k(\boldsymbol{x}_n, \boldsymbol{x}) = \boldsymbol{k}(\boldsymbol{x})^T(\boldsymbol{K} + \lambda \boldsymbol{I}_N)^{-1}\mathbf{t}$$

where $\boldsymbol{k}(\boldsymbol{x})$ has components

$$k_n(\boldsymbol{x}) = k(\boldsymbol{x}_n, \boldsymbol{x})$$

We invert an $N \times N$ matrix in determining $\boldsymbol{a}$ of a kernel machine

$$y(\boldsymbol{x}) = \sum_{n=1}^{N} a_n k(\boldsymbol{x}_n, \boldsymbol{x})$$

In comparison, we invert an $M \times M$ matrix in determining $\boldsymbol{w}$ of a linear regression model

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

Constructing Kernel Functions

In dual representation of linear regression, we have defined a kernel function as the inner product of the feature vectors

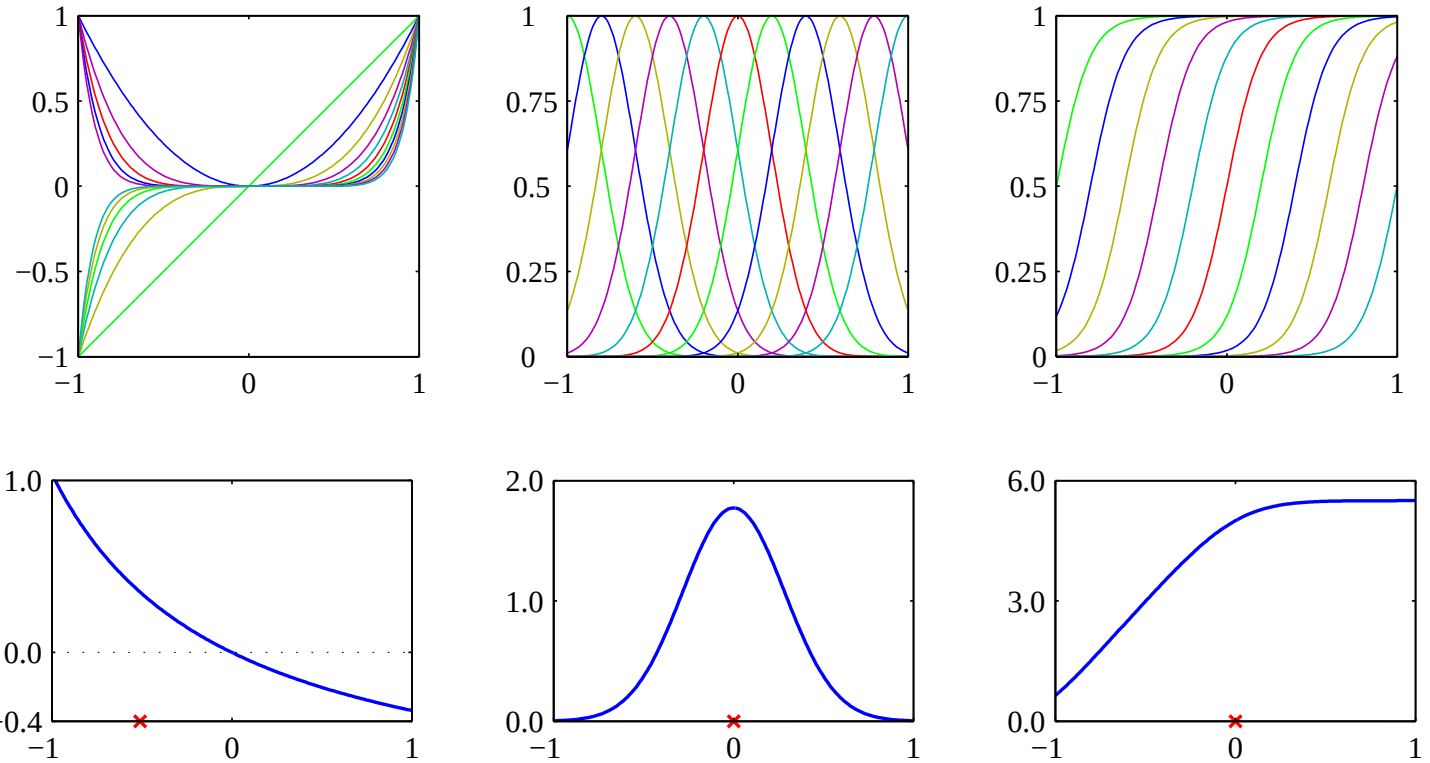
$$k(\boldsymbol{x}_n, \boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x}_n)^T \boldsymbol{\phi}(\boldsymbol{x})$$

Thus, one way to define a kernel function is use the inner product of a feature space.

For one-dimensional input space, we have

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

where $\phi_i(x)$ are the basis functions.



Kernel: Direct Definition

Instead of defining kernel functions based on basis functions, we can also define kernel functions directly.

For example, we can define

$$k(\boldsymbol{x}, \boldsymbol{x}') = \left(\boldsymbol{x}^T \boldsymbol{x}'\right)^2$$

This is equivalent to a particular set of basis functions, since

$$\left(\boldsymbol{x}^T \boldsymbol{x}'\right)^2 = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{x}')$$

where

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

The point is that the feature space is **implicit** and not represented.

Kernels can be constructed from kernels in various ways.

$$
\begin{aligned}
k(\boldsymbol{x}, \boldsymbol{x}') &= ck_1(\boldsymbol{x}, \boldsymbol{x}') \\
k(\boldsymbol{x}, \boldsymbol{x}') &= f(\boldsymbol{x})k_1(\boldsymbol{x}, \boldsymbol{x}')f(\boldsymbol{x}') \\
k(\boldsymbol{x}, \boldsymbol{x}') &= q(k_1(\boldsymbol{x}, \boldsymbol{x}')) \\
k(\boldsymbol{x}, \boldsymbol{x}') &= \exp(k_1(\boldsymbol{x}, \boldsymbol{x}')) \\
k(\boldsymbol{x}, \boldsymbol{x}') &= k_1(\boldsymbol{x}, \boldsymbol{x}') + k_2(\boldsymbol{x}, \boldsymbol{x}') \\
k(\boldsymbol{x}, \boldsymbol{x}') &= k_1(\boldsymbol{x}, \boldsymbol{x}')k_2(\boldsymbol{x}, \boldsymbol{x}') \\
k(\boldsymbol{x}, \boldsymbol{x}') &= k_3(\boldsymbol{\phi}(\boldsymbol{x}'), \boldsymbol{\phi}(\boldsymbol{x}')) \\
k(\boldsymbol{x}, \boldsymbol{x}') &= \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}' \\
k(\boldsymbol{x}, \boldsymbol{x}') &= k_a(\boldsymbol{x}_a, \boldsymbol{x}'_a)k_b(\boldsymbol{x}_b, \boldsymbol{x}'_b) \\
k(\boldsymbol{x}, \boldsymbol{x}') &= k_a(\boldsymbol{x}_a, \boldsymbol{x}'_a)k_b(\boldsymbol{x}_b, \boldsymbol{x}'_b)
\end{aligned}
$$

where $c > 0$, $f(\cdot)$ is any function, $q(\cdot)$ is polynomial with non-negative coefficients, $\boldsymbol{\phi}(\cdot)$ is feature function, and $(\boldsymbol{x}_a, \boldsymbol{x}_b) = \boldsymbol{x}$.

Radial Basis Functions

Definition

A radial function is a function whose value at a point depends on the radial distance of the point from a center. That is

$$\phi(\boldsymbol{x}) = h(\|\boldsymbol{x} - \boldsymbol{\mu}\|)$$

where $\boldsymbol{\mu}$ is the center.

Suppose we have input values $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ along with target values $\{t_1, \ldots, t_N\}$. For linear regression, we can define N basis functions each centered at an input point

$$f(\boldsymbol{x}) = \sum_{n=1}^{N} w_n h(\|\boldsymbol{x} - \boldsymbol{x}_n\|)$$

Then $\{w_1, \ldots, w_N\}$ can be found to fit every data point exactly, i.e.

$$t_i = \sum_{n=1}^{N} w_n h(\|\boldsymbol{x}_i - \boldsymbol{x}_n\|)$$

Suppose we have input values $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ along with target values $\{t_1, \ldots, t_N\}$. A special form of prediction based on kernel functions is

$$y(\boldsymbol{x}) = \sum_{n=1}^{N} k(\boldsymbol{x}, \boldsymbol{x}_n) t_n$$

In this form, the kernel function $k(\cdot, \cdot)$ has parameters which are decided by data.

Let $\mathcal{D} = \{(\boldsymbol{x}_1, t_1), \ldots, (\boldsymbol{x}_N, t_N)\}$ be a data set. Let the joint probability $p(\boldsymbol{x}, t)$ be estimated by

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

where $f(\boldsymbol{x}, t)$ is a component density function with

$$\iint f(\boldsymbol{x}, t) d\boldsymbol{x} dt = 1$$

and

$$\int t f(\boldsymbol{x}, t) dt = 0$$

The prediction of t given $\boldsymbol{x}$ with the minimum expected squared error is the conditional mean

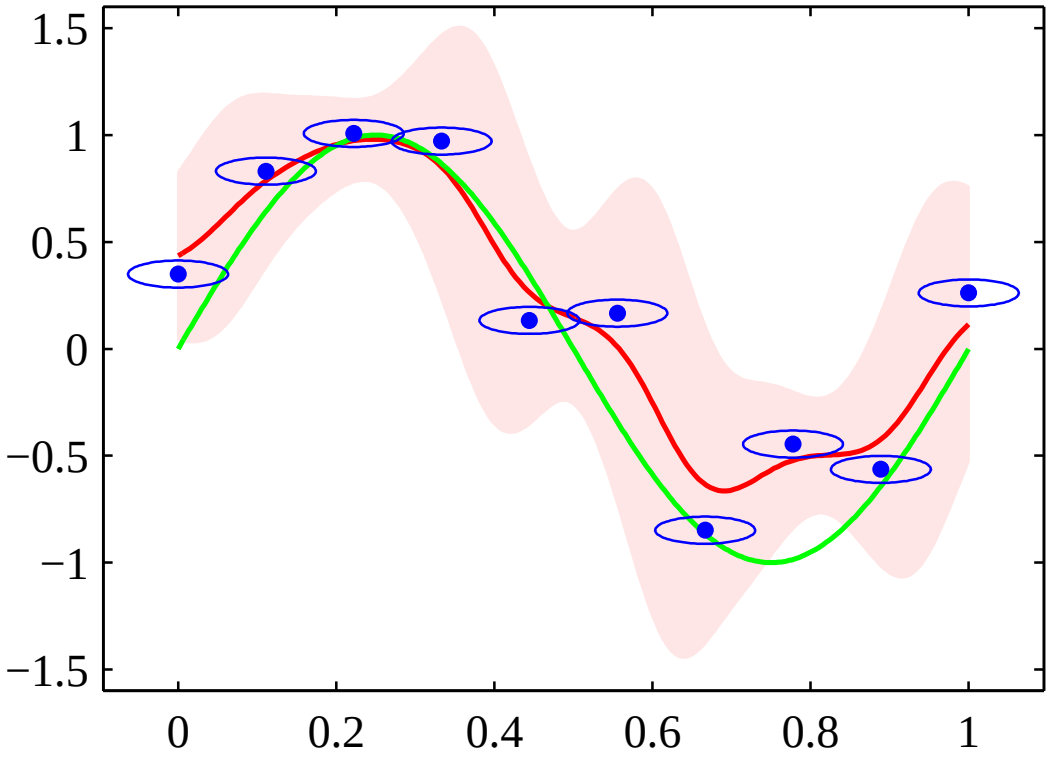
$$
\begin{aligned}
y(\boldsymbol{x}) = \mathbb{E}[t|\boldsymbol{x}] &= \int t p(t|\boldsymbol{x}) dt = \frac{\int t p(\boldsymbol{x}, t) dt}{\int p(\boldsymbol{x}, t) dt} \\
&= \frac{\sum_n \int t f(\boldsymbol{x} - \boldsymbol{x}_n, t - t_n) dt}{\sum_m \int f(\boldsymbol{x} - \boldsymbol{x}_m, t - t_m) dt} \\
&= \frac{\sum_n g(\boldsymbol{x} - \boldsymbol{x}_n) t_n}{\sum_m g(\boldsymbol{x} - \boldsymbol{x}_m)}, \quad g(\boldsymbol{x}) = \int f(\boldsymbol{x}, t) dt \\
&= \sum_n k(\boldsymbol{x}, \boldsymbol{x}_n) t_n
\end{aligned}
$$

where

$$
k(\boldsymbol{x}, \boldsymbol{x}_n) = \frac{g(\boldsymbol{x} - \boldsymbol{x}_n)}{\sum_m g(\boldsymbol{x} - \boldsymbol{x}_m)}
$$

Discussion and Example

- Nadaraya-Watson model leads to a kernel regression
- The kernel function is a radial function
- Example: isotropic Gaussian component density



Gaussian Processes

Stochastic Process and Gaussian Process

- A **stochastic process** or **random process** is a collection of indexed random variables.
- A stochastic process is characterized by the joint distribution of any finite set of random variables in the process.
- A **Gaussian process** is a stochastic process such that the joint distribution of any finite set of random variables is Gaussian.

Specification of a Gaussian Process

A Gaussian process is completely specified by the means and covariances of the random variables in the process.

- This is because a Gaussian distribution is specified by a mean vector and a covariance matrix.
- Let $y(t)$ be a Gaussian process. Then it is specified by

$$\mu(t) = \mathbb{E}[y(t)]$$

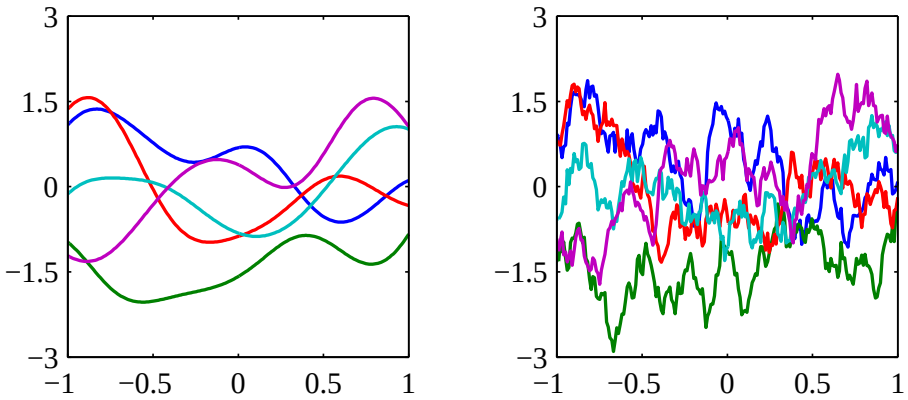
and

$$k(t, t') = \text{cov}(y(t), y(t'))$$

Examples of Gaussian Process

A zero-mean Gaussian process is specified by a **kernel function** for the covariance

$$k(t, t') = \mathbb{E}[y(t)y(t')]$$



Gaussian processes with Gaussian (left) and exponential kernel.

A linear regression function in which the parameters have a Gaussian distribution is a Gaussian process.

Consider $y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x})$ with $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}; \mathbf{0}, \alpha^{-1}\boldsymbol{I})$.

- For any $\mathbf{y} = (y(\boldsymbol{x}_1), \ldots, y(\boldsymbol{x}_N))^T$, we have $\mathbf{y} = \boldsymbol{\Phi}\boldsymbol{w}$. As $\mathbf{y}$ is a linear function of $\boldsymbol{w}$, it is Gaussian.
- Thus $y(\boldsymbol{x})$ is a Gaussian process.
- The mean and kernel (covariance) functions of $y(\boldsymbol{x})$ are

$$\mu(\boldsymbol{x}) = \mathbb{E}[y(\boldsymbol{x})] = 0$$
$$k(\boldsymbol{x}, \boldsymbol{x}') = \mathbb{E}[y(\boldsymbol{x})y(\boldsymbol{x}')] = \alpha^{-1}\boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\phi}(\boldsymbol{x}')$$

How do we fit a data set to a Gaussian process?

Let $\mathcal{D} = \{(\boldsymbol{x}_n, t_n)\}_{n=1}^N$ be a data set.

- We assume a zero-mean Gaussian process $y(\boldsymbol{x})$ with kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$.
- We assume noises on the target values $t_n = y(\boldsymbol{x}_n) + \epsilon_n$, where ϵ_n are i.i.d. Gaussian noises with variance β^{-1}. That is

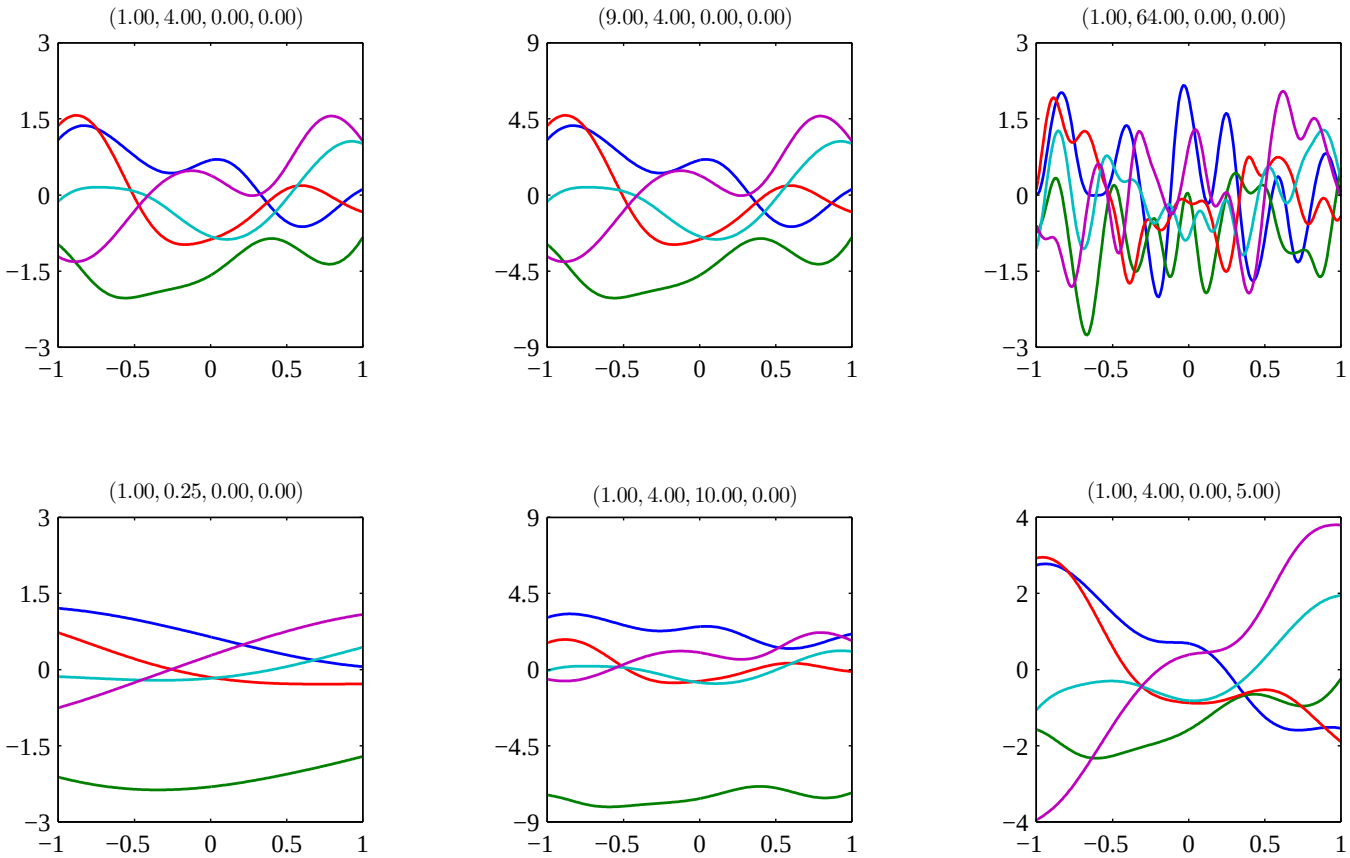
$$p(t_n|y_n) = \mathcal{N}(t_n|y_n, \beta^{-1}),\ y_n = y(\boldsymbol{x}_n)$$

- The parameters in the kernel function and the precision β can be learned from $\mathcal{D}$.

Parametric Kernel Function

One widely used parametric kernel function is

$$k(\boldsymbol{x}, \boldsymbol{x}') = \theta_0 \exp\left\{-\frac{\theta_1}{2}\|\boldsymbol{x} - \boldsymbol{x}'\|^2\right\} + \theta_2 + \theta_3 \boldsymbol{x}^T \boldsymbol{x}'$$



Probability of a Vector of Targets

Denote $\mathbf{y} = [y_1, \ldots, y_N]^T$ and $\mathbf{t} = [t_1, \ldots, t_N]^T$. We have

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

and

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

Thus

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

where

$$\boldsymbol{C} = \boldsymbol{K} + \beta^{-1}\boldsymbol{I}_N$$

Prediction

Given a Gaussian process $y(\boldsymbol{x})$ and the parameter β of Gaussian noise on targets, how do we predict t for input $\boldsymbol{x}$?

- The joint probability of $\mathbf{t}_{N+1} = \left[\mathbf{t}_N, t\right]^T$ is

$$p(\mathbf{t}_{N+1}) = \mathcal{N}(\mathbf{t}_{N+1}; \mathbf{0}, \boldsymbol{C}_{N+1})$$

- Note $\boldsymbol{C}_{N+1} = \begin{pmatrix} \boldsymbol{C}_N & \mathbf{k} \\ \mathbf{k}^T & c \end{pmatrix}$ where $\mathbf{k} = [k_1, \ldots, k_N]^T$ with $k_n = k(\boldsymbol{x}_n, \boldsymbol{x})$ and $c = k(\boldsymbol{x}, \boldsymbol{x}) + \beta^{-1}$.
- It follows that $p(t|\mathbf{t}_N) = \mathcal{N}(t; m(\boldsymbol{x}), \sigma^2(\boldsymbol{x}))$, where $m(\boldsymbol{x}) = \mathbf{k}^T \boldsymbol{C}_N^{-1} \mathbf{t}_N$ and $\sigma^2(\boldsymbol{x}) = c - \mathbf{k}^T \boldsymbol{C}_N^{-1} \mathbf{k}$.
- Thus, the optimal prediction is a kernel regression function

$$m(\boldsymbol{x}) = \sum_{n=1}^{N} a_n k(\boldsymbol{x}_n, \boldsymbol{x}), \ \boldsymbol{a} = \boldsymbol{C}_N^{-1} \mathbf{t}_N$$