CS 559: Machine Learning Fundamentals & Applications

Lecture 7: KNN, Kernel Method, and Gaussian Process

STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

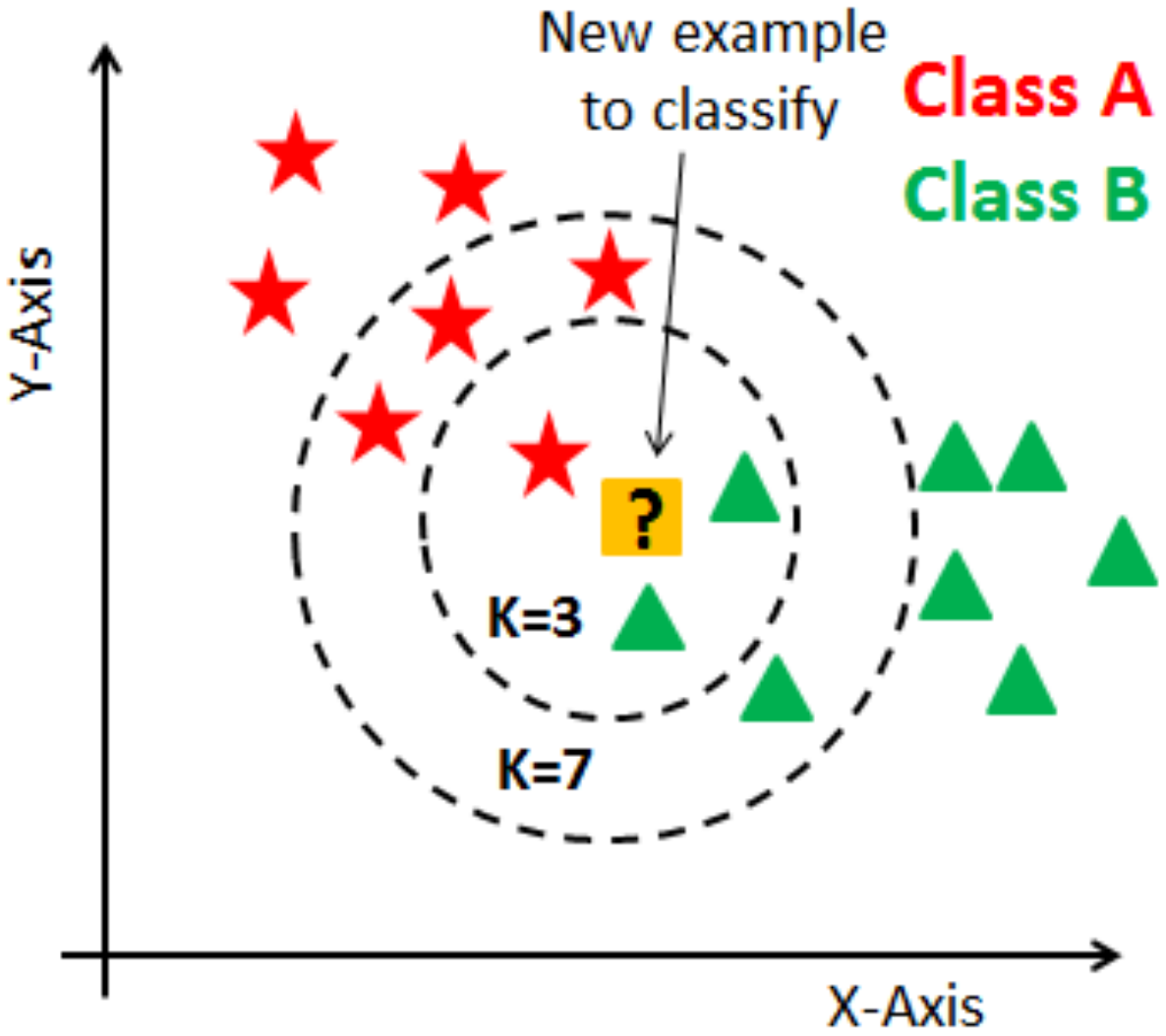
KNN: k-Nearest Neighbors

KNN: Non-parametric Models

Non-parametric model: Using instance-based learning, is characterizes by memorizing the training dataset. When the cost is 0 during the learning process, we call it lazy learning.

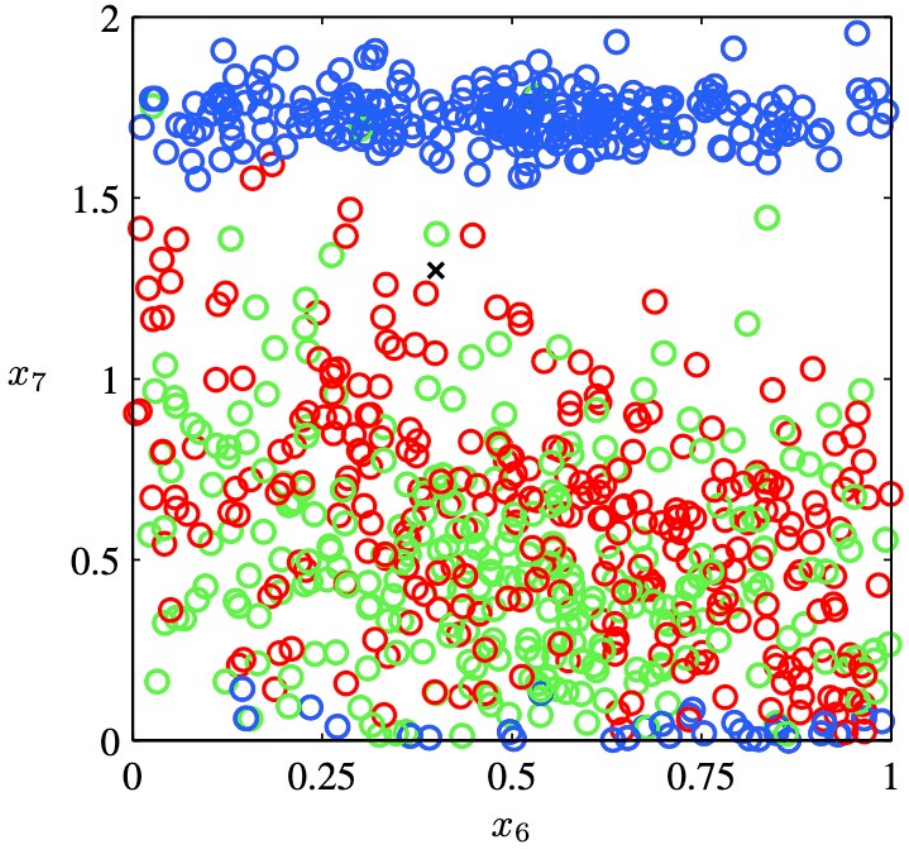
Typical non-parametric models are KNN, decision trees, random forest, and kernel SVM.

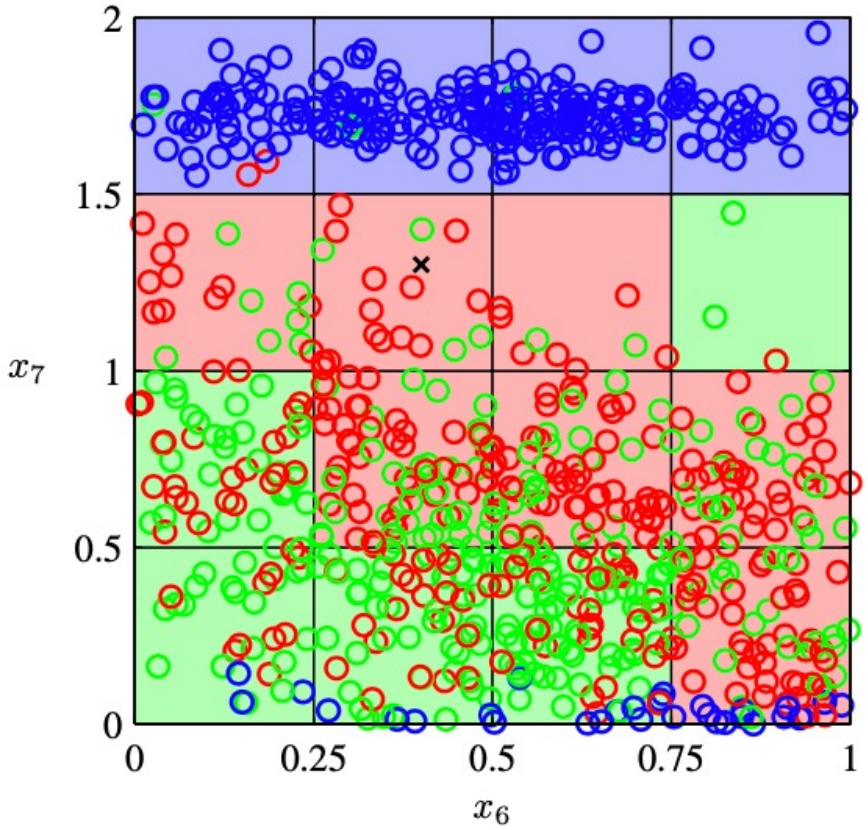
KNN



Curse of Dimensionality

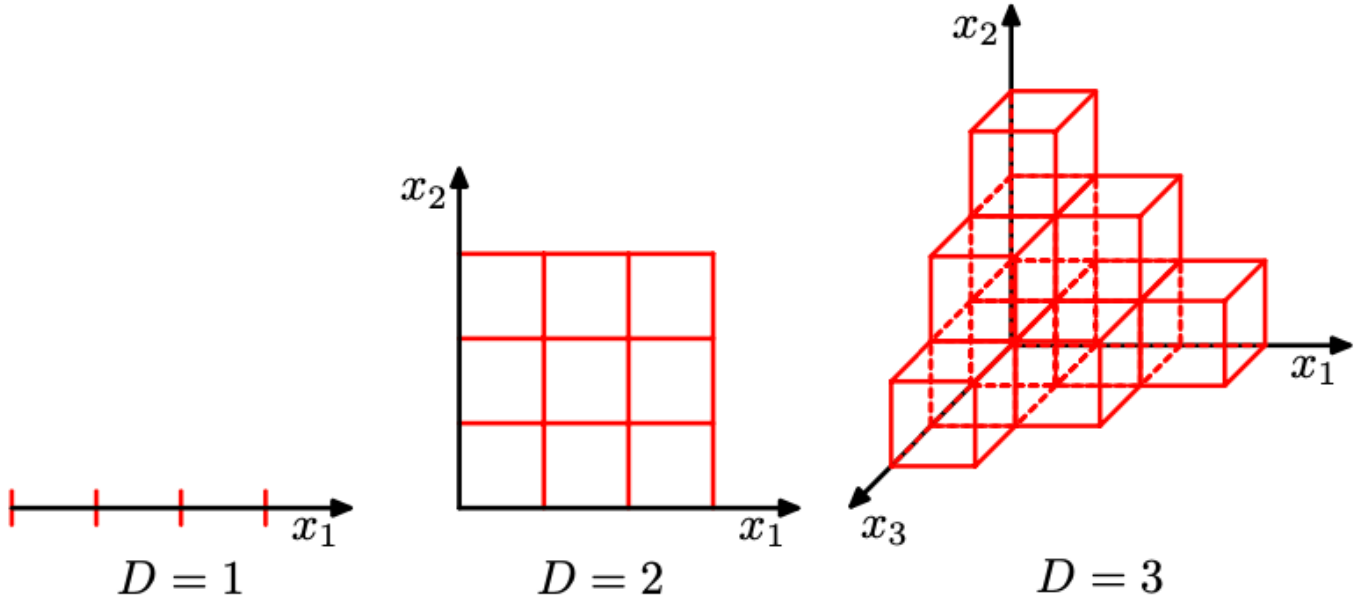
The KNN is very susceptible to overfitting due to the **curse of dimensionality**.





Curse of Dimensionality

- Consider if we are going to classify in a simple approach, there are several problems. The most severe is when we have a larger number of input variables – input spaces in higher dimensionality. The addition of one variable increases the dimension exponentially.
- The problem arises because we need an exponentially large quantity of training data in order to ensure that all cells are not empty.



Curse of Dimensionality

- For the simple consideration, let the model be $y(\boldsymbol{x}, \boldsymbol{w}) = \sum \boldsymbol{w}\boldsymbol{x}$,

$$y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D}\sum_{j=1}^{D} w_{ij} x_i x_j + \cdots.$$

- As D increases, the number of coefficients grows proportional to D^M.

- The KNN requires all points to be close to every dimension axis.
- An additional dimension needs to make the point to be closer to the new axis.

KNN – Pseudocode

KNN(X,k)

1. Load the training and test data
2. Choose the value of k
3. For each point in test data:
4. Find the Euclidean distance to all training data points.
5. Store the distances in a list and sort it.
6. Choose the first k points.
7. Assign a class to the test point based on the majority of classes present in the chosen points.
8. End

Gaussian Distribution

Conditional Gaussian Distribution

Suppose **x** is a D-dimensional vector with Gaussian distribution $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})$ and that partition into two disjoint subsets $\boldsymbol{x}_a$ (M components) and $\boldsymbol{x}_b$ (D-M components),

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_a \\ \boldsymbol{x}_b \end{pmatrix}$$

and

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}$$

where $\boldsymbol{\Sigma}_{ab} = \boldsymbol{\Sigma}_{ba}^{\boldsymbol{T}}$.
Let the *precision matrix* be

$$\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$$

so that

$$\begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}^{-1} = \begin{pmatrix} \boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ba} & \boldsymbol{\Lambda}_{bb} \end{pmatrix}.$$

Using the identity of a **portioned matrix**

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{pmatrix}^{-1} = \begin{pmatrix} \left(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}\right)^{-1} & -\left(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}\right)^{-1}\boldsymbol{B}\boldsymbol{D}^{-1} \\ -\boldsymbol{D}^{-1}\boldsymbol{C}\left(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}\right)^{-1} & \boldsymbol{D}^{-1}+\boldsymbol{D}^{-1}\boldsymbol{C}\left(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}\right)^{-1}\boldsymbol{B}\boldsymbol{D}^{-1} \end{pmatrix},$$

the precision matrix is equivalent to

$$\boldsymbol{\Lambda} = \begin{pmatrix} \left(\boldsymbol{\Sigma}_{aa}-\boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}\right)^{-1} & -\left(\boldsymbol{\Sigma}_{aa}-\boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}\right)^{-1}\boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1} \\ -\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}\left(\boldsymbol{\Sigma}_{aa}-\boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}\right)^{-1} & \boldsymbol{\Sigma}_{bb}^{-1}+\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}\left(\boldsymbol{\Sigma}_{aa}-\boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}\right)^{-1}\boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1} \end{pmatrix} \quad (Eq.\ 1)$$

Conditional Gaussian Distribution

By the definition, the conditional probability $p(\boldsymbol{x_a}|\boldsymbol{x_b})$ can be expressed as

$$p(\boldsymbol{x_a}|\boldsymbol{x_b}) = \frac{p(\boldsymbol{x_a} \cap \boldsymbol{x_b})}{p(\boldsymbol{x_a})}$$

evaluated from the joint distribution $p(\boldsymbol{x}) = p(\boldsymbol{x_a}, \boldsymbol{x_b})$ by fixing $\boldsymbol{x_b}$ to the observed value and normalizing the resulting expression to obtain a valid probability distribution over $\boldsymbol{x_a}$.

Instead, we can obtain the solution using the **quadratic form** in the exponent of the Gaussian distribution and **reinstating the normalization** at the end.

Starting with the Gaussian,

$$\mathcal{N}\left(\boldsymbol{x}\middle|\boldsymbol{\mu}_{(\boldsymbol{a}|\boldsymbol{b})}, \boldsymbol{\Sigma}_{(\boldsymbol{a}|\boldsymbol{b})}\right) = \frac{1}{(2\pi)^{\frac{D}{2}}\left|\boldsymbol{\Sigma}_{(\boldsymbol{a}|\boldsymbol{b})}\right|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\left(\boldsymbol{x}_{(\boldsymbol{a}|\boldsymbol{b})} - \boldsymbol{\mu}_{(\boldsymbol{a}|\boldsymbol{b})}\right)^T \boldsymbol{\Sigma}^{-1}_{(\boldsymbol{a}|\boldsymbol{b})}\left(\boldsymbol{x}_{(\boldsymbol{a}|\boldsymbol{b})} - \boldsymbol{\mu}_{(\boldsymbol{a}|\boldsymbol{b})}\right)\right\}.$$

The quadratic term, $-\frac{1}{2}\left(\boldsymbol{x}_{(\boldsymbol{a}|\boldsymbol{b})} - \boldsymbol{\mu}_{(\boldsymbol{a}|\boldsymbol{b})}\right)^T \boldsymbol{\Sigma}^{-1}_{(\boldsymbol{a}|\boldsymbol{b})}\left(\boldsymbol{x}_{(\boldsymbol{a}|\boldsymbol{b})} - \boldsymbol{\mu}_{(\boldsymbol{a}|\boldsymbol{b})}\right)$, in the **conditional probability** can be expressed as

$$-\frac{1}{2}\left(\boldsymbol{x}_{(\boldsymbol{a}|\boldsymbol{b})} - \boldsymbol{\mu}_{(\boldsymbol{a}|\boldsymbol{b})}\right)^T \boldsymbol{\Sigma}^{-1}_{(\boldsymbol{a}|\boldsymbol{b})}\left(\boldsymbol{x}_{(\boldsymbol{a}|\boldsymbol{b})} - \boldsymbol{\mu}_{(\boldsymbol{a}|\boldsymbol{b})}\right) = -\frac{1}{2}\boldsymbol{x}^T_{(\boldsymbol{a}|\boldsymbol{b})}\boldsymbol{\Sigma}^{-1}_{(\boldsymbol{a}|\boldsymbol{b})}\boldsymbol{x}_{(\boldsymbol{a}|\boldsymbol{b})} + \boldsymbol{x}^T_{(\boldsymbol{a}|\boldsymbol{b})}\boldsymbol{\Sigma}^{-1}_{(\boldsymbol{a}|\boldsymbol{b})}\boldsymbol{\mu}_{(\boldsymbol{a}|\boldsymbol{b})} + C \quad \textit{(Eq. 2)}$$

where the constant C contains the terms that independent of $\boldsymbol{x}$.
Then $\boldsymbol{\mu}$ can be obtained from $\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ term.

Conditional Gaussian Distribution

In the same manner, consider the **conditional distribution**. Eq. 2 can be expressed by following Eq. 1. as

$$= -\frac{1}{2}(\boldsymbol{x_a} - \boldsymbol{\mu_a})^T \boldsymbol{\Lambda}_{aa}(\boldsymbol{x_a} - \boldsymbol{\mu_a}) - \frac{1}{2}(\boldsymbol{x_a} - \boldsymbol{\mu_a})^T \boldsymbol{\Lambda}_{ab}(\boldsymbol{x_b} - \boldsymbol{\mu_b})$$
$$-\frac{1}{2}(\boldsymbol{x_b} - \boldsymbol{\mu_b})^T \boldsymbol{\Lambda}_{ba}(\boldsymbol{x_a} - \boldsymbol{\mu_a}) - \frac{1}{2}(\boldsymbol{x_b} - \boldsymbol{\mu_b})^T \boldsymbol{\Lambda}_{bb}(\boldsymbol{x_b} - \boldsymbol{\mu_b}).$$

which can be expand as

$$= -\frac{1}{2}\boldsymbol{x}_a^T \boldsymbol{\Lambda}_{aa}\boldsymbol{x}_a + \boldsymbol{x}_a^T \boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a - \frac{1}{2}\boldsymbol{x}_a^T \boldsymbol{\Lambda}_{ab}\boldsymbol{x}_b + \frac{1}{2}\boldsymbol{x}_a^T \boldsymbol{\Lambda}_{ab}\boldsymbol{\mu}_b + \frac{1}{2}\boldsymbol{x}_b^T \boldsymbol{\Lambda}_{ab}\boldsymbol{\mu}_a - \frac{1}{2}\boldsymbol{x}_b^T \boldsymbol{\Lambda}_{ba}\boldsymbol{x}_a + \frac{1}{2}\boldsymbol{x}_a^T \boldsymbol{\Lambda}_{ba}\boldsymbol{\mu}_b + \cdots$$

(Eq. 3)

Conditional Gaussian Distribution

In the same manner, consider the **conditional distribution**. Eq. 2 can be expressed by following Eq. 1. as

$$= -\frac{1}{2}(\boldsymbol{x_a} - \boldsymbol{\mu_a})^T \boldsymbol{\Lambda}_{aa}(\boldsymbol{x_a} - \boldsymbol{\mu_a}) - \frac{1}{2}(\boldsymbol{x_a} - \boldsymbol{\mu_a})^T \boldsymbol{\Lambda}_{ab}(\boldsymbol{x_b} - \boldsymbol{\mu_b})$$
$$-\frac{1}{2}(\boldsymbol{x_b} - \boldsymbol{\mu_b})^T \boldsymbol{\Lambda}_{ba}(\boldsymbol{x_a} - \boldsymbol{\mu_a}) - \frac{1}{2}(\boldsymbol{x_b} - \boldsymbol{\mu_b})^T \boldsymbol{\Lambda}_{bb}(\boldsymbol{x_b} - \boldsymbol{\mu_b}).$$

which can be expand as

$$= -\frac{1}{2}\boldsymbol{x}_a^T \boldsymbol{\Lambda}_{aa}\boldsymbol{x}_a + \boldsymbol{x}_a^T \boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a - \frac{1}{2}\boldsymbol{x}_a^T \boldsymbol{\Lambda}_{ab}\boldsymbol{x}_b + \frac{1}{2}\boldsymbol{x}_a^T \boldsymbol{\Lambda}_{ab}\boldsymbol{\mu}_b + \frac{1}{2}\boldsymbol{x}_b^T \boldsymbol{\Lambda}_{ab}\boldsymbol{\mu}_a - \frac{1}{2}\boldsymbol{x}_b^T \boldsymbol{\Lambda}_{ba}\boldsymbol{x}_a + \frac{1}{2}\boldsymbol{x}_a^T \boldsymbol{\Lambda}_{ba}\boldsymbol{\mu}_b + \cdots$$

(Eq. 3)

The $\boldsymbol{x}_a$ second-order term in Eq. 3 shows that

$$\boldsymbol{\Sigma}_{(\boldsymbol{a}|\boldsymbol{b})} = \boldsymbol{\Lambda}_{\boldsymbol{aa}}^{-\boldsymbol{1}}.$$

(Eq. 4)

Conditional Gaussian Distribution

In the same manner, consider the **conditional distribution**. Eq. 2 can be expressed by following Eq. 1. as

$$= -\frac{1}{2}(\boldsymbol{x_a} - \boldsymbol{\mu_a})^T \boldsymbol{\Lambda}_{aa}(\boldsymbol{x_a} - \boldsymbol{\mu_a}) - \frac{1}{2}(\boldsymbol{x_a} - \boldsymbol{\mu_a})^T \boldsymbol{\Lambda}_{ab}(\boldsymbol{x_b} - \boldsymbol{\mu_b})$$
$$-\frac{1}{2}(\boldsymbol{x_b} - \boldsymbol{\mu_b})^T \boldsymbol{\Lambda}_{ba}(\boldsymbol{x_a} - \boldsymbol{\mu_a}) - \frac{1}{2}(\boldsymbol{x_b} - \boldsymbol{\mu_b})^T \boldsymbol{\Lambda}_{bb}(\boldsymbol{x_b} - \boldsymbol{\mu_b}).$$

which can be expand as

$$= -\frac{1}{2}\boldsymbol{x}_a^T \boldsymbol{\Lambda}_{aa}\boldsymbol{x}_a + \boldsymbol{x}_a^T \boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a - \frac{1}{2}\boldsymbol{x}_a^T \boldsymbol{\Lambda}_{ab}\boldsymbol{x}_b + \frac{1}{2}\boldsymbol{x}_a^T \boldsymbol{\Lambda}_{ab}\boldsymbol{\mu}_b + \frac{1}{2}\boldsymbol{x}_b^T \boldsymbol{\Lambda}_{ab}\boldsymbol{\mu}_a - \frac{1}{2}\boldsymbol{x}_b^T \boldsymbol{\Lambda}_{ba}\boldsymbol{x}_a + \frac{1}{2}\boldsymbol{x}_a^T \boldsymbol{\Lambda}_{ba}\boldsymbol{\mu}_b + \cdots$$

(Eq. 3)

The $\boldsymbol{x}_a$ second-order term in Eq. 3 shows that

$$\boldsymbol{\Sigma}_{(\boldsymbol{a}|\boldsymbol{b})} = \boldsymbol{\Lambda}_{\boldsymbol{aa}}^{-\boldsymbol{1}}.$$

(Eq. 4)

The terms are linear in $\boldsymbol{x}_a$ are

$$\boldsymbol{x}_a^T \boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a - \frac{1}{2}\boldsymbol{x}_a^T \boldsymbol{\Lambda}_{ab}\boldsymbol{x}_b + \frac{1}{2}\boldsymbol{x}_a^T \boldsymbol{\Lambda}_{ab}\boldsymbol{\mu}_b - \frac{1}{2}\boldsymbol{x}_b^T \boldsymbol{\Lambda}_{ba}\boldsymbol{x}_a + \frac{1}{2}\boldsymbol{x}_a^T \boldsymbol{\Lambda}_{ba}\boldsymbol{\mu}_b$$
$$= \boldsymbol{x}_a^T\{\boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{ba}(\boldsymbol{x}_b - \boldsymbol{\mu}_b)\}.$$

Conditional Gaussian Distribution

The comparison between (Eq. 2) and (Eq. 3) shows that

$$\boldsymbol{x}_{(\boldsymbol{a}|\boldsymbol{b})}^{T}\boldsymbol{\Sigma}_{(\boldsymbol{a}|\boldsymbol{b})}^{-1}\boldsymbol{\mu}_{(\boldsymbol{a}|\boldsymbol{b})} = \boldsymbol{x}_{a}^{T}\{\boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_{a} - \boldsymbol{\Lambda}_{ba}(\boldsymbol{x}_{b} - \boldsymbol{\mu}_{b})\}$$

$$\boldsymbol{\mu}_{(\boldsymbol{a}|\boldsymbol{b})} = \boldsymbol{\Sigma}_{(\boldsymbol{a}|\boldsymbol{b})}\{\boldsymbol{\Lambda}_{aa}\boldsymbol{\mu}_{a} - \boldsymbol{\Lambda}_{ba}(\boldsymbol{x}_{b} - \boldsymbol{\mu}_{b})\}$$

$$\boldsymbol{\mu}_{(\boldsymbol{a}|\boldsymbol{b})} = \boldsymbol{\mu}_{a} - \boldsymbol{\Lambda}_{aa}^{-1}\boldsymbol{\Lambda}_{ba}(\boldsymbol{x}_{b} - \boldsymbol{\mu}_{b})$$

(Eq. 5)

Using (Eq. 1), (Eq. 4) and (Eq. 5) can be expressed in terms of the mean ad covariance of the conditional distribution

$$\boldsymbol{\Sigma}_{(a|b)} = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}$$

$$\boldsymbol{\mu}_{(a|b)} = \boldsymbol{\mu}_{a} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\boldsymbol{x}_{b} - \boldsymbol{\mu}_{b}).$$

Sequential Learning

Considering the result of MLE for the mean,

$$\mu_{ML}^{(N)} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n = \frac{1}{N}\boldsymbol{x}_n + \frac{1}{N}\sum_{n=1}^{N-1} \boldsymbol{x}_n$$
$$= \frac{1}{N}\boldsymbol{x}_n + \frac{(N-1)}{N}\boldsymbol{\mu}_{ML}^{(N-1)}$$
$$= \boldsymbol{\mu}_{ML}^{(N-1)} + \frac{1}{N}\left(\boldsymbol{x}_N - \boldsymbol{\mu}_{ML}^{(N-1)}\right),$$

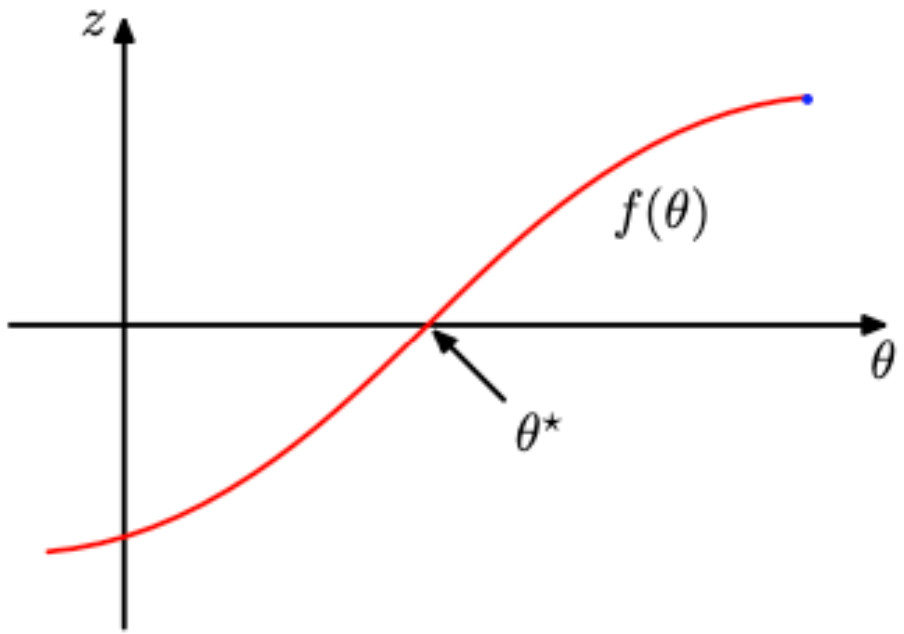
the contribution from data points gets smaller as N increases. However, this solution is the equivalent solution as a general MLE and we need a general derivable formulation.

Sequential Learning

The *Robbins-Monro* algorithm allows for the construction of a more general formulation of sequential learning.

Consider a pair of random variables θ and z in a joint distribution $p(z, \theta)$.
The conditional expectation, $\mathbb{E}[z|\theta]$, defines a deterministic function $f(\theta)$ that is given by

$$f(\theta) = \mathbb{E}[z|\theta] = \int zp(z|\theta)dz\,.$$



Sequential Learning

The goal is to find the root θ^*: $f(\theta^*) = 0$.

Suppose we have a large data set of observation.
We can find the regression model directly but the sequential estimation scheme for θ^* is not easy.

The following conditions are given by Robbins and Monro.
1. The conditional variance of z is finite: $\mathbb{E}[(z - f)^2|\theta] < \infty$.
2. A sequence of successive estimates of θ^* is defined as $\theta^{(N)} = \theta^{(N-1)} - a_{N-1}z\left(\theta^{(N-1)}\right)$.
3. The coefficients $\{a_N\}$ satisfy the conditions.

$$\lim_{N\to\infty} a_N = 0$$

$$\sum_{N=1}^{\infty} a_N = \infty$$

$$\sum_{N=1}^{\infty} a_N^2 < \infty$$

The 2nd condition shows that the sequence will converge to the root with probability 1.
The 3-1 condition shows that the successive correlations decrease in magnitude so that the process can converge to a limiting value. The 3-2 condition ensures the algorithm does not converge short of the root. The 3-3 condition ensures that the accumulated noise has finite variance and does not spoil convergence.

Sequential Learning

By the definition, the MLE solution θ_{ML} is a stationary point of the negative log-likelihood function and satisfies

$$\frac{\partial}{\partial\theta}\left\{-\frac{1}{N}\sum_{n=1}^{N}\ln p(x_n|\theta)\right\}\Bigg|_{\theta_{ML}} = 0.$$

By taking the limit $N \to \infty$, the equation above becomes

$$-\lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}\frac{\partial}{\partial\theta}\ln p(x_n|\theta) = \mathbb{E}_x\left[-\frac{\partial}{\partial\theta}\ln p(x_n|\theta)\right].$$

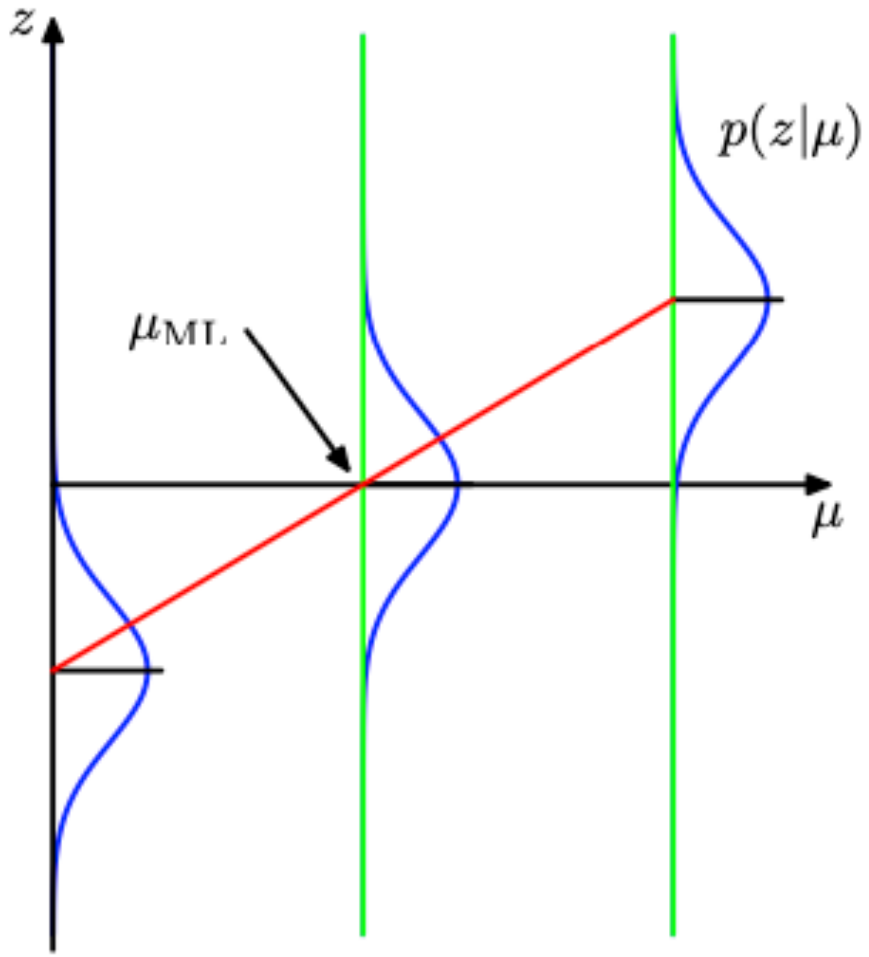
We can apply the Robbins-Monro procedure,

$$\theta^{(N)} = \theta^{(N-1)} - a_{N-1}\frac{\partial}{\partial\theta^{N-1}}[-\ln p(x_N|\theta^{N-1})].$$

Since $\theta^{(N)} = \mu_{ML}^{(N)}$,

$$z = \frac{\partial}{\partial\mu_{ML}}[-\ln p(x|\mu_{ML},\sigma^2)] = -\frac{1}{\sigma^2}(x-\mu_{ML}).$$

The Gaussian z distribution has the mean of $-(\mu-\mu_{ML})/\sigma^2$.
This allows us to choose the coefficients a_N in the form σ^2/N.



Kernel Method

Dual Presentation

Many linear parametric models can be re-cast into an equivalent *dual representation* that based on linear combinations of a *kernel function* evaluated at the training data points.

For models are based on a fixed nonlinear *feature space* mapping $\phi(\boldsymbol{x})$, the kernel function is given by

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

Dual Presentation

Consider a linear regression model

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

where $\lambda \geq 0$. Setting $\nabla J(\boldsymbol{w}) = 0$, the solution of $\mathbf{w}$ is

$$\boldsymbol{w} = -\frac{1}{\lambda}\sum_{n=1}^{N}\{\boldsymbol{w}^T\phi(\boldsymbol{x_n}) - t_n\}\phi(\boldsymbol{x_n}) = \sum_{n=1}^{N} a_n\phi(\boldsymbol{x_n}) = \boldsymbol{\Phi}^T\boldsymbol{a}$$

where $\boldsymbol{a} = (a_1, \ldots, a_N)^T$ and a_n is defined as

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

The least-squares algorithm can be reformulated using a *dual representation* with the parameter vector $\mathbf{w}$ by substituting $\boldsymbol{w} = \boldsymbol{\Phi}^T\boldsymbol{a}$ into $J(\boldsymbol{w})$:

$$J(\boldsymbol{a}) = \frac{1}{2}\boldsymbol{a}^T\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{a} - \boldsymbol{a}^T\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}^T\boldsymbol{t} + \frac{\lambda}{2}\boldsymbol{a}^T\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{a}$$

(Eq. 6)

where $\boldsymbol{t} = (t_1, \ldots, t_N)^T$.

Dual Presentation

Eq. 6 can be simplified using the *Gram* matrix $\boldsymbol{K} = \boldsymbol{\Phi}\boldsymbol{\Phi}^{\boldsymbol{T}}$ which is an $N \times N$ symmetric matrix with elements

$$K_{nm} = \phi(\boldsymbol{x_n})^T \phi(\boldsymbol{x_m}) = k(\boldsymbol{x_n}, \boldsymbol{x_m})$$

where $k(x, x')$ is the *kernel function*:

$$J(\boldsymbol{a}) = \frac{1}{2}\boldsymbol{a^T K K a} - \boldsymbol{a^T K t} + \frac{1}{2}\boldsymbol{t^T t} + \frac{\lambda}{2}\boldsymbol{a^T K a}.$$

(Eq. 7)

Setting the gradient of $J(\boldsymbol{a})$ w.r.t. to $\boldsymbol{a}$ to 0,

$$\nabla_{\boldsymbol{a}} J(\boldsymbol{a}) = 0 = \boldsymbol{K K a} - \boldsymbol{K t} + \lambda \boldsymbol{K a}$$
$$\boldsymbol{a} = (\boldsymbol{K} + \lambda \boldsymbol{I_N})^{-1} \boldsymbol{t}.$$

(Eq. 8)

Dual Presentation

Substituting Eq. 8 back to the linear model, the prediction for a new **x** can be found as

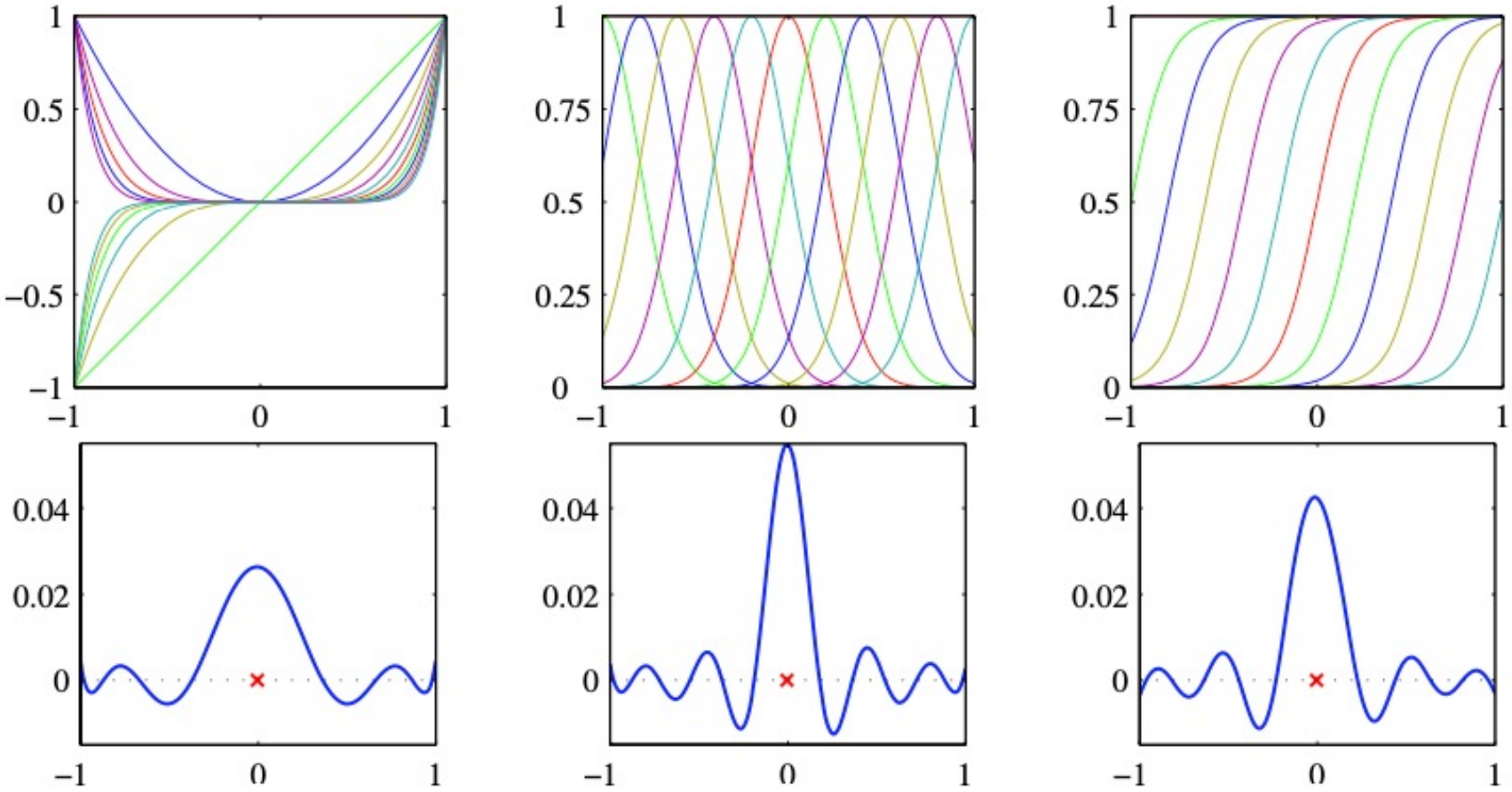
$$y(\boldsymbol{x}) = \boldsymbol{w}^{\boldsymbol{T}}\phi(\boldsymbol{x}) = \boldsymbol{a}^{\boldsymbol{T}}\boldsymbol{\Phi}\phi(\boldsymbol{x}) = \boldsymbol{k}(\boldsymbol{x})^{T}(\boldsymbol{K} + \lambda \boldsymbol{I}_{\boldsymbol{N}})^{-\mathbf{1}}\boldsymbol{t} \quad \textit{(Eq. 9)}$$

where the vector $\boldsymbol{k}(\boldsymbol{x})$ has elements $k_n(\boldsymbol{x}) = k(\boldsymbol{x}_{\boldsymbol{n}}, \boldsymbol{x})$.

Eq. 9 expresses the solution completely in $k(\boldsymbol{x}, \boldsymbol{x}')$ where **a** can be expressed as a linear combination of $\phi(\boldsymbol{x})$. This will allow us to recover the solution in terms of **w**.

Kernel Construction

To construct the valid kernel functions, we can map a feature space $\phi(\boldsymbol{x})$ and find the corresponding kernel validity.



Kernel Construction

In 1-D input space, the kernel function is defined as

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

Or we can construct kernel functions directly using the valid kernel that is a scalar product in some feature space.

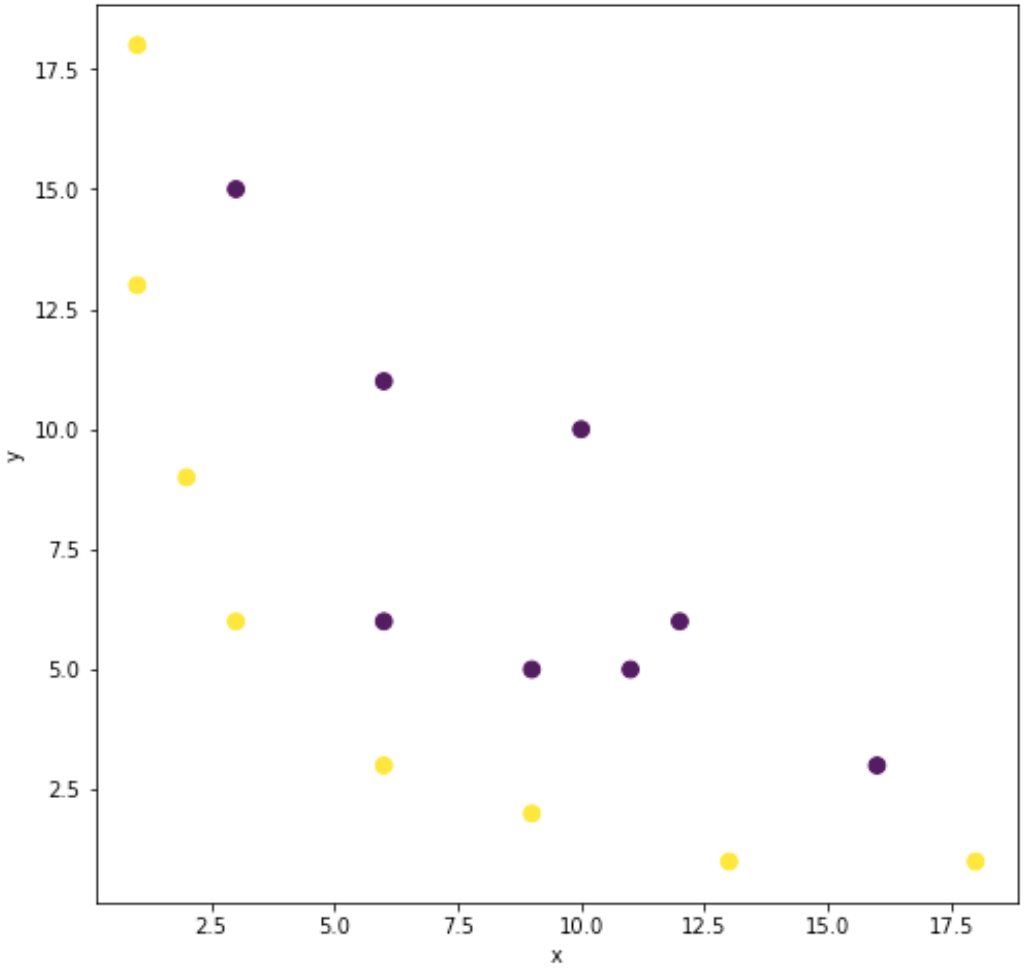
For example, consider the following

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

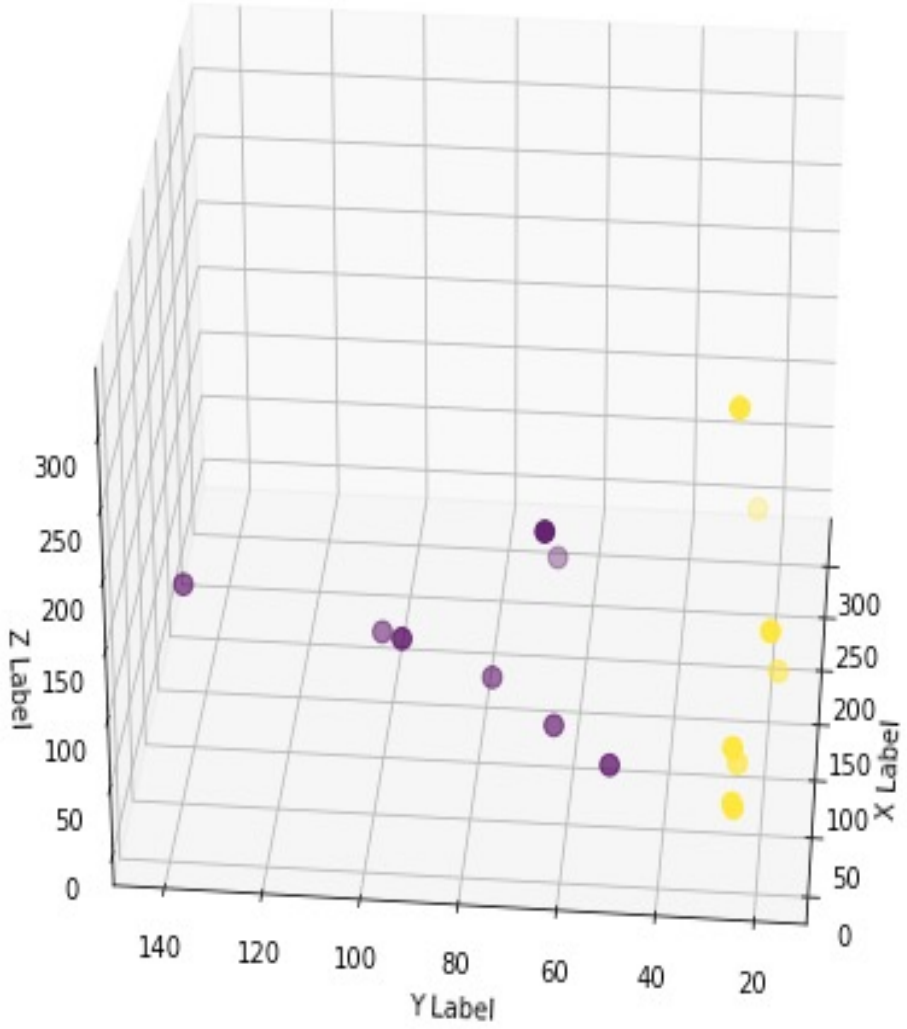
Let the input space in 2-D be $\boldsymbol{x} = (x_1, x_2)$.
If we expand out the terms and identify the corresponding nonlinear feature mapping

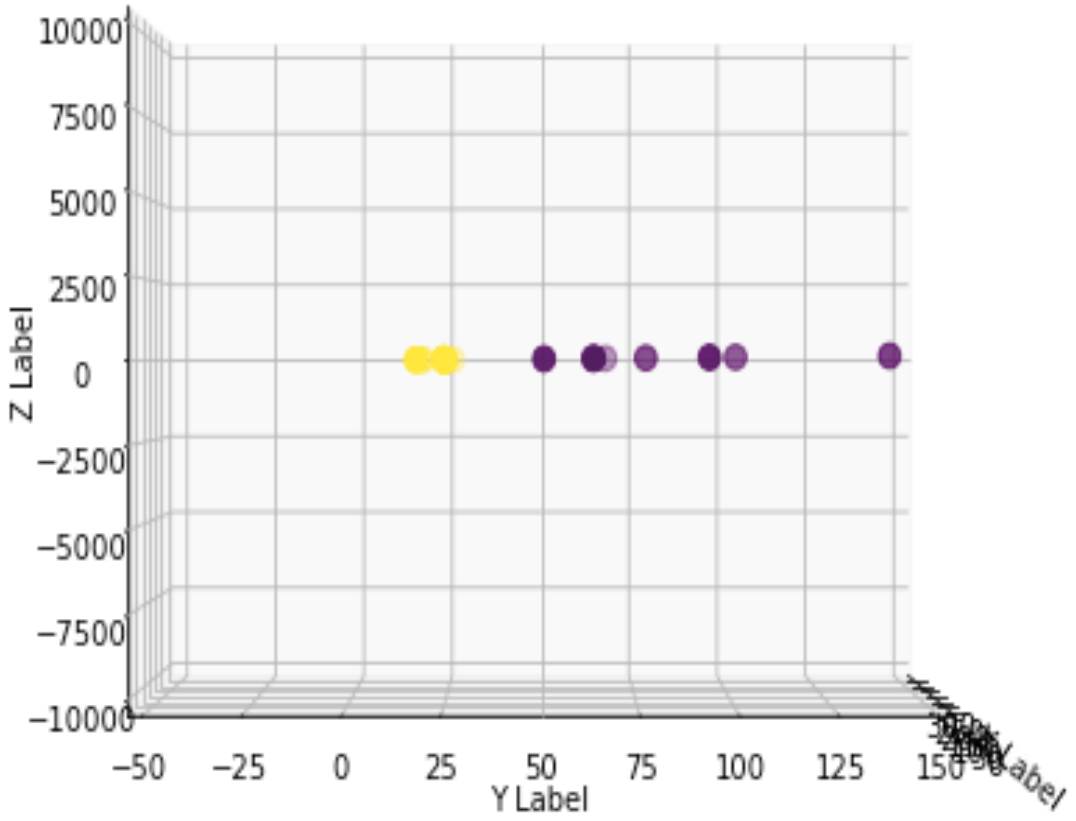
$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^T \boldsymbol{z})^2 &= (x_1 z_1 + x_2 z_2)^2 \\ &= x_1^2 z_1^2 + 2x_1 z_1 x_2 z_2 + x_2^2 z_2^2 \\ &= \left(x_1^2, \sqrt{2} x_1 x_2, x_2^2\right)\left(z_1^2, \sqrt{2} z_1 z_2, z_2^2\right)^T \\ &= \phi(\boldsymbol{x})^T \phi(\boldsymbol{z}). \end{aligned}$$

Kernel Construction



Kernel Construction





Kernel Construction

As long as the Gram matrix **K** has elements $k(\boldsymbol{x}_n, \boldsymbol{x}_m)$ are positive semidefinite for all possible choices of the set $\{x_n\}$, the validity of kernel is confirmed.

Positive semidefinite matrices are the scalars $x^T Mx$ and $x^* Mx$ are positive or zero where x^* is a conjugate transpose of x.

Check if matrix **A** is a positive definite.

$$\boldsymbol{A} = \begin{bmatrix} 9 & -15 \\ -15 & 25 \end{bmatrix}$$

Let $\boldsymbol{x}$ be a 2×1 vector, $\boldsymbol{x} = [x_1, x_2]$.

$$\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{bmatrix} 9 & -15 \\ -15 & 25 \end{bmatrix} [x_1, x_2]$$
$$= 9x_1^2 - 15x_1x_2 - 15x_1x_2 + 25x_2^2$$
$$= (3x_1 - 5x_2)^2$$

If $\boldsymbol{x} = [5,3]$, $\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} = 0$.
If $\boldsymbol{x} = [5,2]$, $\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} = 25$.

Kernel Construction

Another commonly used kernel function is a "Gaussian" kernel also known as RBF and is defined as

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

where $|\boldsymbol{x} - \boldsymbol{x}'|^2 = \boldsymbol{x}^T\boldsymbol{x} + (\boldsymbol{x}')^T\boldsymbol{x}' - 2\boldsymbol{x}^T\boldsymbol{x}'$ and this gives

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

$$= \exp\left\{-\frac{1}{2\sigma^2}\big(k(\boldsymbol{x}, \boldsymbol{x}) + k(\boldsymbol{x}', \boldsymbol{x}') - 2k(\boldsymbol{x}, \boldsymbol{x}')\big)\right\}.$$

Gaussian Processes

Gaussian Processes

By extending the roles of kernels to probabilistic discriminative models, we can see how kernels arise in a Bayesian setting and lead to the framework of Gaussian processes.

Recall linear regression model $y(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x})$.

- A prior distribution over **w** induced a corresponding prior distribution over functions $y(\boldsymbol{x}, \boldsymbol{w})$.
- Then the posterior distribution over **w** is evaluated and obtained the corresponding posterior distribution over regression functions that predict new input data **x**.

In the Gaussian process, we define the prior probability distribution over functions directly.

- for a finite training set, we only need to consider the values of the function at the discrete set of input values.

Linear Regression Revisit

Consider a model $y(\boldsymbol{x}) = \boldsymbol{w}^{\boldsymbol{T}}\phi(\boldsymbol{x})$ and a prior distribution over **w** given by

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

Since we are interested in the function evaluation at specific value of **x**, we are also interested in the joint distribution of the function values $y(\boldsymbol{x}_1), \ \ldots, y(\boldsymbol{x}_N)$,

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

The mean and covariance of **y** are

$$\mathbb{E}[\boldsymbol{y}] = \boldsymbol{\Phi}\mathbb{E}[\boldsymbol{w}] = 0$$

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

where **K** is the Gram matrix with elements

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

Gaussian Processes

In general, a GP is defined as a probability distribution over functions $y(\boldsymbol{x})$ s.t. the set of $y(\boldsymbol{x})$ values are evaluated at an arbitrary set of points **x** jointly have a Gaussian distribution.

The stochastic process is the joint distribution over N variables $y_1, \ldots, y_N$ that are specified by the mean and the covariance.

- We do not know the mean of $y(\boldsymbol{x})$.
- But we can take it to be zero symmetrically.
- This allows the mean of the prior over weight values $p(\boldsymbol{w}|\alpha)$ be zero in the basis function viewpoint.
- Then, we give the covariance of $y(\boldsymbol{x})$ at any two values of **x** that is given by the kernel function $\mathbb{E}[y(\boldsymbol{x}_n), y(\boldsymbol{x}_m)] = k(\boldsymbol{x}_n, \boldsymbol{x}_m)$.

The **Ornstein-Uhlenbeck process** allows defining the kernel function directly.

Gaussian Process for regression

Let the observed target values be

$$t_n = y_n + \epsilon_n$$

where $y_n = y(\boldsymbol{x}_n)$ and ϵ_n is the random noise variable chosen i.i.d,

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

The joint distribution of $\mathbf{t}$ conditioned on $\mathbf{y}$ is an isotropic Gaussian form

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

The marginal distribution $p(\boldsymbol{y})$ is given by a Gaussian

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

The marginal distribution $p(\boldsymbol{t})$ conditioned on inputs is

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

where the covariance matrix $\mathbf{C}$ has elements

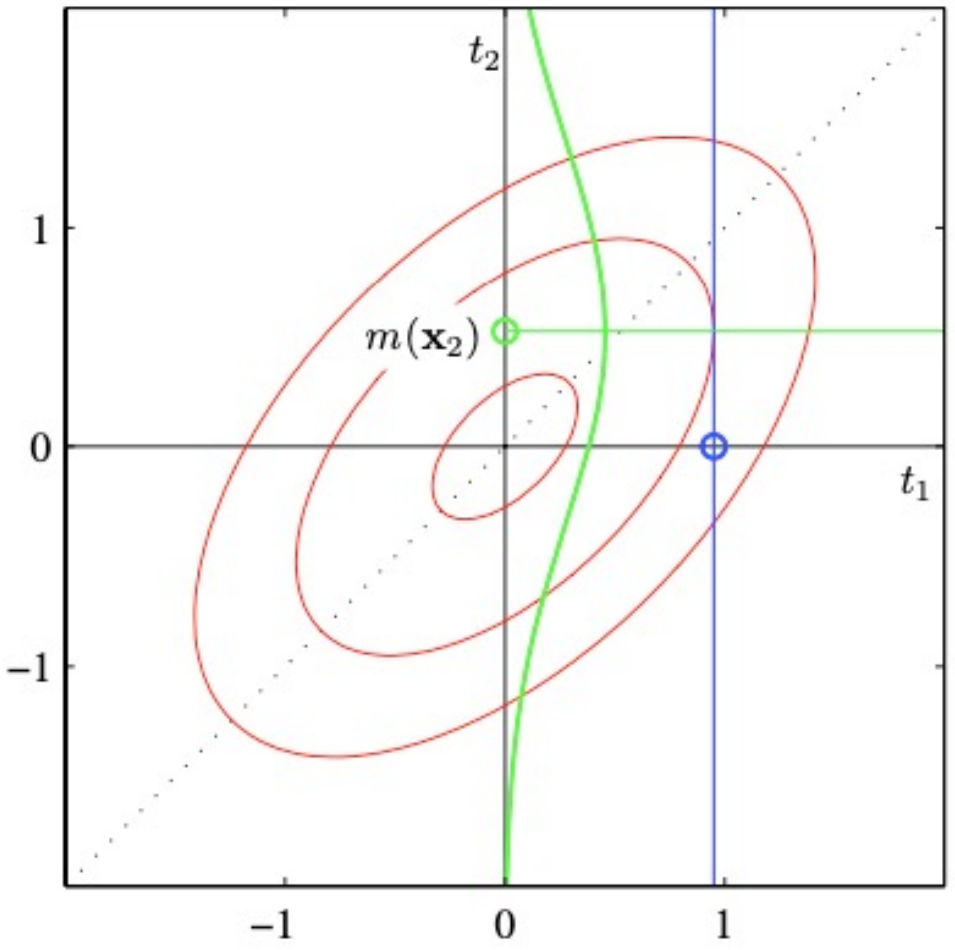
$$\boldsymbol{C} = \begin{pmatrix} k(\boldsymbol{x}_n, \boldsymbol{x}_n) + \beta^{-1} & k(\boldsymbol{x}_n, \boldsymbol{x}_m) \\ k(\boldsymbol{x}_m, \boldsymbol{x}_n) & k(\boldsymbol{x}_m, \boldsymbol{x}_m) + \beta^{-1} \end{pmatrix}$$

$$C(\boldsymbol{x_n}, \boldsymbol{x_m}) = k(\boldsymbol{x_n}, \boldsymbol{x_m}) + \beta^{-1}\delta_{nm}.$$

(Eq. 10)

Gaussian Processes

Suppose the training data set $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)^T$ and $\boldsymbol{t}_N = (t_1, \ldots, t_N)^T$.
If the new data has the input vector $\boldsymbol{x}_{N+1}$ with the target variable t_{N+1}, the predictive distribution $p(t_{N+1}|\boldsymbol{t}_N)$.



Gaussian Processes

The joint distribution $p(\boldsymbol{t}_{N+1})$ where the target vector is $\boldsymbol{t}_{N+1} = (t_1, \ldots, t_N, t_{N+1})^T$ can be expressed as

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

where $\boldsymbol{C}_{N+1}$ is the covariance matrix with elements given by (Eq. 10).
Note that the joint distribution is Gaussian, we can apply the Gaussian distribution discussed above.

The covariance matrix partition is then

$$\boldsymbol{C}_{N+1} = \begin{pmatrix} \boldsymbol{C}_N & k(\boldsymbol{x}_n, \boldsymbol{x}_{N+1}) \\ k^T(\boldsymbol{x}_n, \boldsymbol{x}_{N+1}) & k(\boldsymbol{x}_{N+1}, \boldsymbol{x}_{N+1}) + \beta^{-1} \end{pmatrix}$$

for $n = 1, \ldots, N$.

Using the result from (Eq. 5), the mean and covariance of the conditional distribution of $p(t_{N+1}|\boldsymbol{t})$ are

$$m(\boldsymbol{x}_{N+1}) = \boldsymbol{k}^T \boldsymbol{C}_N^{-1} \boldsymbol{t}$$
$$\sigma^2(\boldsymbol{x}_{N+1}) = k(\boldsymbol{x}_{N+1}, \boldsymbol{x}_{N+1}) + \beta^{-1} - \boldsymbol{k}^T \boldsymbol{C}_N^{-1} \boldsymbol{k}.$$

Learning the hyperparameters

The hypermeters' learning is based on the evaluation of likelihood function $p(\boldsymbol{t}|\theta)$ and θ can be estimated by maximizing the log likelihood function.

The maximization of the log likelihood can be done by using the gradient-based optimization algorithms.

Start with the standard form for a multivariate Gaussian distribution

$$\ln p(\boldsymbol{t}|\boldsymbol{\theta}) = -\frac{1}{2}\ln|\boldsymbol{C}_N| - \frac{1}{2}\boldsymbol{t}^T\boldsymbol{C}_N^{-1}\boldsymbol{t} - \frac{N}{2}\ln(2\pi)$$

and set the gradient w.r.t. θ_i equal to 0 for the θ_i estimation.

Learning the hyperparameters

The derivative of the inverse of a matrix $\boldsymbol{A}^{-1}$ is

$$\frac{\partial}{\partial x}\boldsymbol{A}^{-1} = -\boldsymbol{A}^{-1}\left(\frac{\partial \boldsymbol{A}}{\partial x}\right)\boldsymbol{A}^{-1}.$$

(Eq. 11)

Using (Eq. 11), the gradient of $\boldsymbol{C}_N^{-1}$ is

$$\frac{\partial}{\partial \theta_i}\boldsymbol{C}_N^{-1} = -\boldsymbol{C}_N^{-1}\left(\frac{\partial \boldsymbol{C}_N}{\partial \theta_i}\right)\boldsymbol{C}_N^{-1}.$$

(Eq. 12)

Using the derivative of log-**A**,

$$\frac{\partial}{\partial x}\ln|\boldsymbol{A}| = Tr\left(\boldsymbol{A}^{-1}\frac{\partial \boldsymbol{A}}{\partial x}\right),$$

the derivative of $\ln|\boldsymbol{C}_N|$ is

$$\frac{\partial}{\partial \theta_i}\ln|\boldsymbol{C}_N| = Tr\left(\boldsymbol{C}_N^{-1}\frac{\partial \boldsymbol{C}_N}{\partial \theta_i}\right).$$

(Eq. 13)

Combining (Eq. 12) and (Eq. 13), the derivation of the Gaussian distribution is

$$\frac{\partial}{\partial \theta_i}\ln p(\boldsymbol{t}|\boldsymbol{\theta}) = -\frac{1}{2}Tr\left(\boldsymbol{C}_N^{-1}\frac{\partial \boldsymbol{C}_N}{\partial \theta_i}\right) + \frac{1}{2}\boldsymbol{t}^T\boldsymbol{C}_N^{-1}\left(\frac{\partial \boldsymbol{C}_N}{\partial \theta_i}\right)\boldsymbol{C}_N^{-1}\boldsymbol{t}.$$

Gaussian Processes

Because $\ln p(\boldsymbol{t}|\boldsymbol{\theta})$ is generally a nonconvex function, there can be multiple maxima.

The estimation is straightforward:
1. a prior over $\boldsymbol{\theta}$
2. log posterior via gradient-based methods.

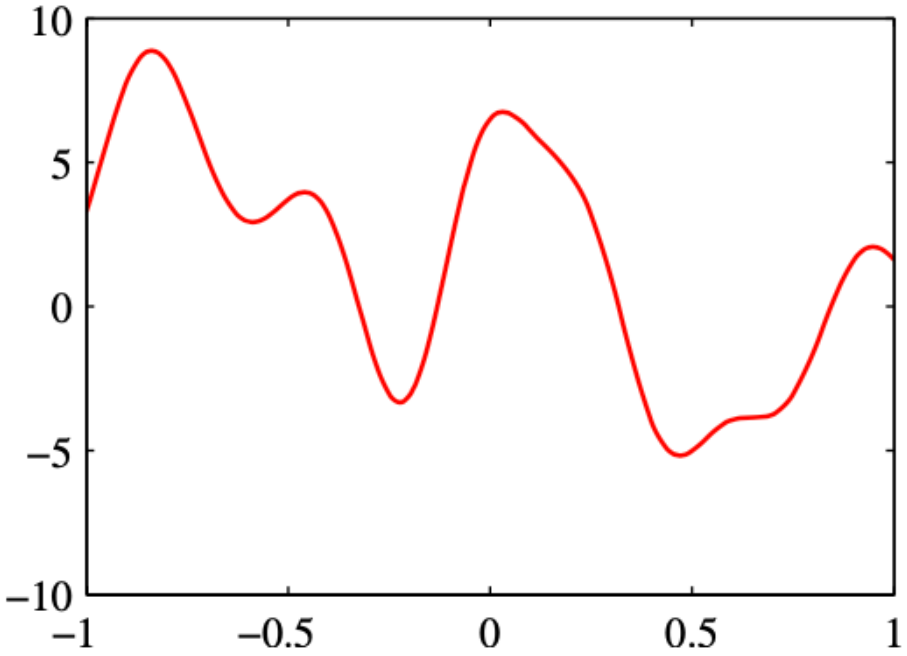
While it is a Bayesian treatment, finding the exact marginalization is not possible. Instead, we need to approximate the marginalization.

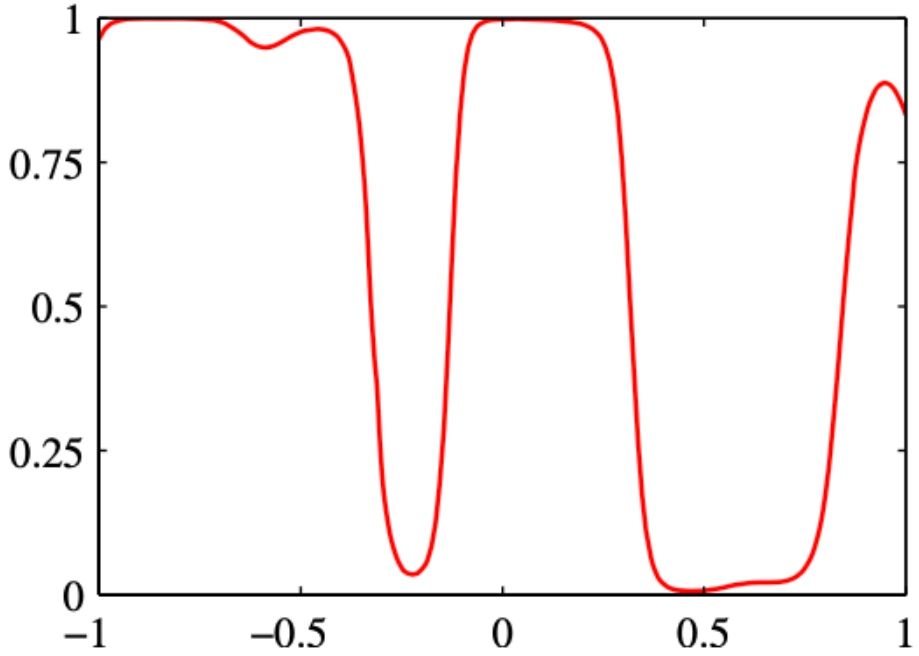
Gaussian Process for Classification

Consider the binary class problem with a target variable $t \in \{0,1\}$.

Define a Gaussian process over a function $a(\boldsymbol{x})$ and transform the function using a sigmoid function, $y = \sigma(a)$.
Obtain a non-Gaussian stochastic process over functions $y(\boldsymbol{x})$ where $y \in (0,1)$.
The Bernoulli distribution over the target variable t is given as

$$p(t|a) = \sigma(a)^t\big(1 - \sigma(a)\big)^{1-t}.$$





Gaussian Process for Classification

Similar to the regression problem, we introduce a Gaussian process prior over the vector $\boldsymbol{a}_{N+1}$ and a non-Gaussian process over $\boldsymbol{t}_{N+1}$.

The Gaussian process prior for $\boldsymbol{a}_{N+1}$:

$$p(\boldsymbol{a}_{N+1}) = \mathcal{N}(\boldsymbol{a}_{N+1}|\boldsymbol{0}, \boldsymbol{C}_{N+1})$$

The covariance matrix is

$$C(\boldsymbol{x}_n, \boldsymbol{x}_m) = k(\boldsymbol{x}_n, \boldsymbol{x}_m) + \nu\delta_{nm}$$

where ν is a fixed noise term.

The predictive distribution is

$$p(t_{N+1} = 1|\boldsymbol{t}_N) = \int p(t_{N+1} = 1|a_{N+1})p(a_{N+1}|\boldsymbol{t}_N)da_{N+1}$$

where $p(t_{N+1} = 1|a_{N+1}) = \sigma(a_{N+1})$.

Gaussian Process for Classification

The posterior distribution over a_{N+1} using Bayes' theorem given by

$$p(a_{N+1}|\boldsymbol{t}_N) = \int p\left(a_{(N+1)}, \boldsymbol{a}_N \middle| \boldsymbol{t}_N\right) d\boldsymbol{a}_N = \frac{1}{p(\boldsymbol{t}_N)} \int p(a_{N+1}, \boldsymbol{a}_N) p(\boldsymbol{t}_N | a_{N+1}, \boldsymbol{a}_N) d\boldsymbol{a}_N$$

$$\frac{1}{p(\boldsymbol{t}_N)} \int p(a_{N+1}|\boldsymbol{a}_N) p(\boldsymbol{a}_N) p(\boldsymbol{t}_N|\boldsymbol{a}_N) d\boldsymbol{a}_N = \int p(a_{N+1}|\boldsymbol{a}_N) p(\boldsymbol{a}_N|\boldsymbol{t}_N) d\boldsymbol{a}_N$$

where $p(\boldsymbol{t}_N | a_{N+1}, \boldsymbol{a}_N) = p(\boldsymbol{t}_N|\boldsymbol{a}_N)$.

The conditional distribution $p(a_{N+1}|\boldsymbol{a}_N)$ is obtained by invoking the result seem from the regression,

$$p(a_{N+1}|\boldsymbol{a}_N) = \mathcal{N}\left(a_{N+1} \middle| \boldsymbol{k}^T \boldsymbol{C}_N^{-1} \boldsymbol{a}_N, c - \boldsymbol{k}^T \boldsymbol{C}_N^{-1} \boldsymbol{k}\right).$$

Gaussian Process for Classification

The prior $p(\boldsymbol{a}_N)$ is given by a zero-mean Gaussian process with $\boldsymbol{C}_N$ and the data term is given by:

$$p(\boldsymbol{t}_N|\boldsymbol{a}_N) = \prod_{n=1}^{N} \sigma(a_n)^{t_n}\big(1-\sigma(a_n)\big)^{(1-t_n)} = \prod_{n=1}^{N} e^{a_n t_n}\sigma(-a_n)\,.$$

The approximation can be obtained by Taylor expanding the log of $p(\boldsymbol{t}_N|\boldsymbol{a}_N)$, $\Psi(\boldsymbol{a}_N)$, is

$$\Psi(\boldsymbol{a}_N) = \ln p(\boldsymbol{a}_N) + \ln p(\boldsymbol{t}_N|\boldsymbol{a}_N) =$$

$$-\frac{1}{2}\boldsymbol{a}_N^T\boldsymbol{C}_N^{-1}\boldsymbol{a}_N - \frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{C}_N| + \boldsymbol{t}_N^T\boldsymbol{a}_N - \sum_{n=1}^{N}\ln(1+e^{a_n}) + const.$$

To find the mode of the posterior distribution, we need to evaluate the gradient of $\Psi(\boldsymbol{a}_N)$ that is given by

$$\nabla\Psi(\boldsymbol{a}_N) = \boldsymbol{t}_N - \boldsymbol{\sigma}_N - \boldsymbol{C}_N^{-1}\boldsymbol{a}_N$$

where $\boldsymbol{\sigma}_N$ is a vector with elements $\sigma(a_n)$.

Gaussian Process for Classification

The second derivative of $\Psi(\boldsymbol{a}_N)$ is

$$\nabla\nabla\Psi(\boldsymbol{a}_N) = -\boldsymbol{W}_N - \boldsymbol{C}_N^{-1}$$

where $\boldsymbol{W}_N$ is a diagonal matrix with elements $\sigma(a_n)\big(1-\sigma(a_n)\big)$ in the range of (1, ¼) and a positive definite matrix. Since C_N^{-1} is also positive definite matrix, the Hessian matrix $-\nabla\nabla\Psi(\boldsymbol{a}_N)$ is also positive definite.

Using the Newton-Raphson formula, the iterative update equation is

$$\boldsymbol{a}_N^{new} = \boldsymbol{C}_N(\boldsymbol{I} + \boldsymbol{W}_N\boldsymbol{C}_N)^{-1}\{\boldsymbol{t}_N - \boldsymbol{\sigma}_N + \boldsymbol{W}_N\boldsymbol{a}_N\}.$$

At the mode, the gradient $\nabla\Psi(\boldsymbol{a}_N)$ will vanish and $\boldsymbol{a}_N^*$ will satisfy

$$\boldsymbol{a}_N^* = \boldsymbol{C}_N(\boldsymbol{t}_N - \boldsymbol{\sigma}_N).$$

Once the mode of the posterior is found, the Hessian matrix can be evaluated:

$$\boldsymbol{H} = -\nabla\nabla\Psi(\boldsymbol{a}_n) = \boldsymbol{W}_N + \boldsymbol{C}_N^{-1}.$$

This defines the Gaussian approximation of the posterior distribution $p(a_N|\boldsymbol{t}_N)$ given by

$$q(\boldsymbol{a}_N) = \mathcal{N}(\boldsymbol{a}_N|\boldsymbol{a}_N^*, \boldsymbol{H}^{-1}).$$

Gaussian Process for Classification

The combination of $p(a_{N+1}|\boldsymbol{a}_N)$ and $q(\boldsymbol{a}_N)$ can obtain the result

$$\mathbb{E}[a_{N+1}|\boldsymbol{t}_N] = \boldsymbol{k}^T(\boldsymbol{t}_N - \boldsymbol{\sigma}_N)$$
$$var[a_{N+1}|\boldsymbol{t}_N] = c - \boldsymbol{k}^T(\boldsymbol{W}_N^{-1} + \boldsymbol{C}_N)^{-1}\boldsymbol{k}.$$

Then the determination of $\boldsymbol{\theta}$ of the covariance function can be found by maximizing the likelihood $p(\boldsymbol{t}_N|\boldsymbol{\theta})$:

$$p(\boldsymbol{t}_N|\boldsymbol{\theta}) = \int p(\boldsymbol{t}_N|\boldsymbol{a}_N)p(\boldsymbol{a}_N|\boldsymbol{\theta})d\boldsymbol{a}_N.$$

The approximation of log likelihood function is

$$\ln p(\boldsymbol{t}_N|\boldsymbol{\theta}) = \Psi(\boldsymbol{a}_N^*) - \frac{1}{2}\ln\left|\boldsymbol{W}_N + \boldsymbol{C}_N^{-1}\right| + \frac{N}{2}\ln(2\pi)$$

where $\Psi(\boldsymbol{a}_N^*) = \ln p(\boldsymbol{a}_N^*|\boldsymbol{\theta}) + \ln p(\boldsymbol{t}_N|\boldsymbol{a}_N^*)$.

Gaussian Process for Classification

The derivative w.r.t. θ_j is then

$$\frac{\partial \ln p(\boldsymbol{t}_N|\boldsymbol{\theta})}{\partial \theta_j} = \frac{1}{2}\boldsymbol{a}_N^{*\,-1}\boldsymbol{C}_N^{-1}\left(\frac{\partial \boldsymbol{C}_N}{\partial \theta_j}\right)\boldsymbol{C}_N^{-1}\boldsymbol{a}_N^* - \frac{1}{2}Tr\left\{(\boldsymbol{I}+\boldsymbol{C}_N\boldsymbol{W}_N)^{-1}\boldsymbol{W}_N\left(\frac{\partial \boldsymbol{C}_N}{\partial \theta_j}\right)\right\}.$$

This leaves the following contribution to the derivative w.r.t. a component θ_j of $\boldsymbol{\theta}$

$$-\frac{1}{2}\sum_{n=1}^{N}\frac{\partial \ln|W_N + C_N^{-1}|}{\partial a_n^*}\left(\frac{\partial a_n^*}{\partial \theta_j}\right) = -\frac{1}{2}\sum_{n=1}^{N}[(I+C_NW_N)^{-1}C_N]_{nn}\sigma_n^*(1-\sigma_n^*)(1-2\sigma_n^*)\left(\frac{\partial a_n^*}{\partial \theta_j}\right)$$

where $\sigma_n^* = \sigma(a_n^*)$.

The differentiation of $\boldsymbol{a}_N^* = \boldsymbol{C}_N(\boldsymbol{t}_N - \boldsymbol{\sigma}_N)$ w.r.t. θ_j gives

$$\frac{\partial a_n^*}{\partial \theta_j} = \frac{\partial \boldsymbol{C}_N}{\partial \theta_j}(\boldsymbol{t}_N - \boldsymbol{\sigma}_N) - \boldsymbol{C}_N\boldsymbol{W}_N\left(\frac{\partial a_n^*}{\partial \theta_j}\right).$$

Rearranging then gives

$$\frac{\partial a_n^*}{\partial \theta_j} = (\boldsymbol{I}+\boldsymbol{W}_N\boldsymbol{C}_N)^{-1}\left(\frac{\partial \boldsymbol{C}_N}{\partial \theta_j}\right)(\boldsymbol{t}_N - \boldsymbol{\sigma}_N).$$

The evaluation of the gradient of the log likelihood function can be used to determine the value for $\boldsymbol{\theta}$.

Gaussian Process for Classification