

Gaussian processes

Consider Bayesian regression for

$$y(\vec{x}, \vec{w}) = \vec{w}^T \cdot \vec{g}(\vec{x})$$

Prior  $p(\vec{w}|\alpha)$  can be thought of as inducing a prior <sup>over weights</sup> over functions  $y(\vec{x}, \vec{w})$ .  
 Given data, prior becomes posterior, leading to another distribution over functions  $y(\vec{x}, \vec{w})$ .  
Idea: dispense with parameterizing through  $\vec{w}$  altogether and instead define distributions over functions directly.

—○—  
 Consider  $y(\vec{x}) = \underbrace{\vec{w}^T}_{\text{dim. } M} \cdot \vec{g}(\vec{x})$  with

prior  $p(\vec{w}) = \mathcal{N}(\vec{w} | \vec{0}, \alpha^{-1} \mathbb{I})$

For a given  $\vec{w}$  sampled from the prior, we get a specific  $y(\vec{x})$ . Thus we're sampling in the space of functions  $y(\vec{x})$ . In practice, we're interested in  $y(\vec{x}_1) \dots y(\vec{x}_N)$  for some dataset.

Denoting  $y_n = y(\vec{x}_n)$ , we get:

$$\vec{y} = \Phi \vec{w} \quad \vec{y} = \overline{y_1 \dots y_N}$$

↘ design matrix:  $\Phi_{nk} = g_k(\vec{x}_n)$

What is  $p(\vec{y})$ ?

$\vec{y}$  is a linear combination of gaussian-distributed  $\vec{w}$ 's  $\Rightarrow$  itself a gaussian.

$$\text{Then } \begin{cases} E[\vec{y}] = \Phi E[\vec{w}] = 0, \\ \text{Cov}[\vec{y}] = E[\underbrace{\vec{y} \cdot \vec{y}^T}_{N \times N \text{ matrix}}] = \Phi E[\vec{w} \cdot \vec{w}^T] \Phi^T \ominus \end{cases}$$

$$\ominus \frac{1}{2} \underbrace{\Phi \Phi^T}_{N \times N} \equiv K \leftarrow \text{gram matrix with } K_{nm} = \underbrace{k(\vec{x}_n, \vec{x}_m)}_{\text{kernel f'n}} = \frac{1}{2} \vec{y}^T(\vec{x}_n) \cdot \vec{y}(\vec{x}_m)$$

$$\text{So, } P(\vec{y}) = \mathcal{N}(\vec{y} | 0, \underline{\underline{K}})$$

In general, a gaussian process is a prob. distribution over functions  $y(\vec{x})$  s.t. if we take arbitrary points  $\vec{x}_1, \dots, \vec{x}_N \Rightarrow \vec{y} = \underline{y(\vec{x}_1) \dots y(\vec{x}_N)}$  is such

that  $P(\vec{y})$  is gaussian.  
"  $P(y_1, \dots, y_N)$

Gaussian process is an example of a stochastic process which induces a joint prob. distribution for  $y(\vec{x}_1) \dots y(\vec{x}_N)$ .

In most applications, we will assume  $E[\vec{y}] = \vec{0}$  for any  $\vec{y}$  (similar to choosing unbiased priors for  $\vec{w}$ 's), and the covariance matrix is given by  $E[y(\vec{x}_n) y(\vec{x}_m)] = k(\vec{x}_n, \vec{x}_m)$

The kernel function can be chosen directly rather than constructed through  $\vec{g}$ .

For example,  $k(\vec{x}, \vec{x}') = e^{-\frac{\|\vec{x} - \vec{x}'\|^2}{2\sigma^2}}$  is a common choice, but not the only one:

$k(\vec{x}, \vec{x}') = e^{-\theta \|\vec{x} - \vec{x}'\|}$  is another.

### Gaussian processes for regression

Consider  $t_n = \underbrace{y_n}_{y(\vec{x}_n)} + \underbrace{\epsilon_n}_{\text{noise}}$

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

Then  $p(\vec{\epsilon} | \vec{y}) = \underbrace{\mathcal{N}(\vec{\epsilon} | \vec{y}, \beta^{-1} \mathbb{I}_N)}_{\text{product of } N \text{ Gaussians}}$

Prior  $p(\vec{y}) = \mathcal{N}(\vec{y} | \vec{0}, K)$  then yields

$$p(\vec{\epsilon}) = \int d\vec{y} p(\vec{\epsilon} | \vec{y}) p(\vec{y}) \quad \square$$

Once again,

$$\begin{cases} p(\vec{x}) = \mathcal{N}(\vec{x} | \vec{\mu}, \Lambda^{-1}) \\ p(\vec{y} | \vec{x}) = \mathcal{N}(\vec{y} | A\vec{x} + \vec{b}, L^{-1}) \end{cases} \Rightarrow p(\vec{y}) = \mathcal{N}(\vec{y} | A\vec{\mu} + \vec{b}, L^{-1} + A\Lambda^{-1}A^T)$$

Here,  $\begin{cases} \vec{y} \rightarrow \vec{t} \\ \vec{x} \rightarrow \vec{y} \end{cases} \Rightarrow \begin{cases} \vec{\mu} = \vec{0} \\ A\vec{x} + \vec{b} = \vec{y} \end{cases} \Rightarrow \begin{cases} \Lambda^{-1} = K \\ A\vec{y} + \vec{b} = \vec{y} \\ L^{-1} = \beta^{-1} \mathbb{I}_N \end{cases}$

$$\square \mathcal{N}(\vec{t} | \vec{0}, \beta^{-1} \mathbb{I}_N + K)$$

"G, s.t.  $C(\vec{x}_n, \vec{x}_m) = k(\vec{x}_n, \vec{x}_m) + \beta^{-1} \delta_{nm}$   
(or  $G_N$ )

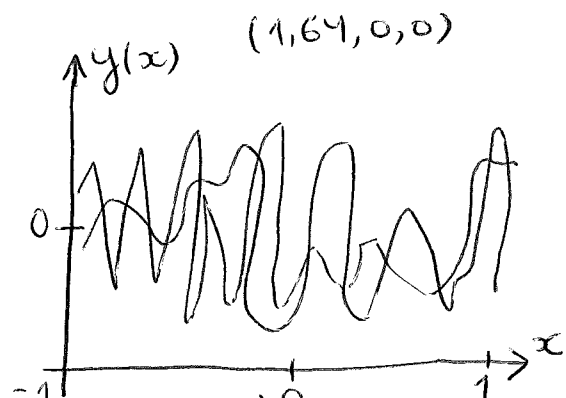
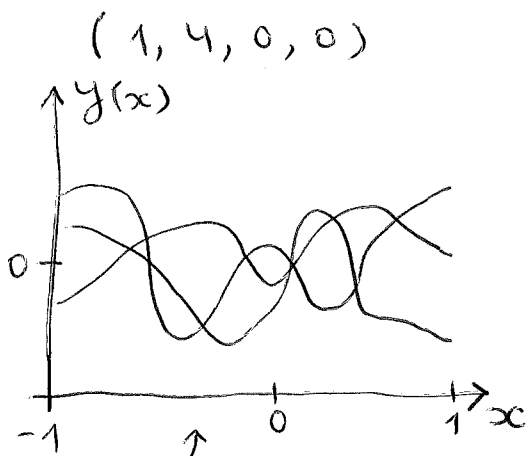
noise due to  $\vec{t}_n$   
uncertainty due to randomness in  $\vec{y}$

Another kernel (popular in regression):

$$k(\vec{x}_n, \vec{x}_m) = \theta_0 e^{-\frac{\theta_1}{2} \|\vec{x}_n - \vec{x}_m\|^2} + \theta_2 + \theta_3 \vec{x}_n^T \vec{x}_m$$

$(\theta_0, \dots, \theta_3)$  are hyperparameters

$p(\vec{y})$  can be used to generate prior functions, e.g.:



choose  $\{\vec{x}_n\}_1^N$  equidistantly spaced in  $[-1, 1]$

generate  $\vec{y}$  from  $\mathcal{N}(\vec{y} | \vec{0}, K)$ , plot & interpolate

Now, in regression we are given

$$\vec{t}_N = \overbrace{t_1, \dots, t_N} \text{ along with } \overbrace{\vec{x}_1, \dots, \vec{x}_N}$$

& we want to know

$$p(t_{N+1} | \vec{t}_N, \underbrace{\vec{x}_1, \dots, \vec{x}_{N+1}}_{\text{omit for brevity}})$$

Define  $\vec{t}_{N+1} = \overbrace{t_1, \dots, t_{N+1}}$ , then

$$p(\vec{t}_{N+1}) = \mathcal{N}(\vec{t}_{N+1} | \vec{0}, \underbrace{C_{N+1}}_{(N+1) \times (N+1)})$$

$$C_{N+1} = \begin{pmatrix} C_N & \vec{k} \\ \vec{k}^T & C \end{pmatrix} \begin{matrix} \left[ \begin{matrix} k(\vec{x}_n, \vec{x}_{N+1}) \\ n=1, \dots, N \end{matrix} \right] \text{ elements of} \\ \text{N-dim vector} \end{matrix}$$

"  $k(\vec{x}_{N+1}, \vec{x}_{N+1}) + \beta^{-1}$

~~o~~

Suppose we have  $\mathcal{N}(\vec{x} | \vec{\mu}, \Sigma)$ .

Consider  $\vec{x} = \begin{pmatrix} \vec{x}_a \\ \vec{x}_b \end{pmatrix} \begin{matrix} \left. \begin{matrix} \{ M \\ \{ D-M \end{matrix} \right\} \end{matrix} \right\} D \text{ components}$

likewise,  $\vec{\mu} = \begin{pmatrix} \vec{\mu}_a \\ \vec{\mu}_b \end{pmatrix}, \Sigma = \begin{pmatrix} \underbrace{\Sigma_{aa}}_{M \times M} & \underbrace{\Sigma_{ab}}_{M \times (D-M)} \\ \underbrace{\Sigma_{ba}}_{(D-M) \times M} & \underbrace{\Sigma_{bb}}_{(D-M) \times (D-M)} \end{pmatrix} \left. \begin{matrix} \{ \\ \{ \end{matrix} \right\} D \times D$

Note that  $\Sigma^T = \Sigma$  gives  $\Sigma_{aa}^T = \Sigma_{aa},$

$\Sigma_{bb}^T = \Sigma_{bb}, \Sigma_{ba}^T = \Sigma_{ab}.$

It can be shown ((2.81) & (2.82)) that

$$p(\vec{x}_a | \vec{x}_b) = \mathcal{N}(\vec{x}_a | \vec{\mu}_{ab}, \Sigma_{ab}), \text{ where}$$

$$\begin{cases} \vec{\mu}_{ab} = \vec{\mu}_a + \Sigma_{ab} \Sigma_{bb}^{-1} (\vec{x}_b - \vec{\mu}_b), \\ \Sigma_{ab} = \Sigma_{aa} - \Sigma_{ab} \Sigma_{bb}^{-1} \Sigma_{ba}. \end{cases}$$

Then  $p(t_{N+1} | \underbrace{\vec{t}}_{\vec{t}_N}) = \mathcal{N}(t_{N+1} | m(\vec{x}_{N+1}), \sigma^2(\vec{x}_{N+1}))$ , where

$$\begin{cases} \vec{x}_a \rightarrow t_{N+1} \\ \vec{x}_b \rightarrow \vec{t}_N \end{cases} \Rightarrow \begin{cases} \vec{\mu}_a \rightarrow \vec{0}, \Sigma_{aa} \rightarrow c \\ \vec{\mu}_b \rightarrow \vec{0}, \Sigma_{bb} \rightarrow C_N \\ \Sigma_{ba} \rightarrow \vec{k}, \Sigma_{ab} \rightarrow \vec{k}^T \end{cases}$$

$$\begin{cases} m(\vec{x}_{N+1}) = \underbrace{\vec{k}^T}_{N \text{ vector}} \cdot \underbrace{C_N^{-1}}_{N \text{ vector}} \vec{t}, \\ \sigma^2(\vec{x}_{N+1}) = c - \vec{k}^T C_N^{-1} \vec{k}. \end{cases} \Leftarrow \text{Gaussian process regression}$$

↑ depend on  $\vec{x}_{N+1}$  through  $c$  &  $\vec{k}$

Note that since  $p(\vec{t}) = \mathcal{N}(\vec{t} | \vec{0}, C)$ ,

$C$  must be positive-definite,

with eigenvalues  $\underbrace{\lambda_i}_{\text{eigenvalues of } K} + \underbrace{\beta^{-1}}_{> 0}$

Then  $\lambda_i \geq 0 \Rightarrow K$  is pos.-semidefinite, just as before

Finally, note that

$$\begin{aligned}
 m(\vec{x}_{N+1}) &= \sum_{n, n'} k_n C_{N, nn'}^{-1} t_{n'} = \\
 &= \sum_n \left( \sum_{n'} C_{N, nn'}^{-1} t_{n'} \right) \underbrace{k_n}_{k(\vec{x}_n, \vec{x}_{N+1})} = \sum_n a_n k(\vec{x}_n, \vec{x}_{N+1}).
 \end{aligned}$$

The above results are valid for any kernel  $f'n$ . If this kernel is explicitly rewritten through the basis  $f$ 's, we obtain:

$$C_{nm} = \underbrace{K_{nm}}_{\frac{1}{\alpha} \psi_j(\vec{x}_n) \cdot \psi_j(\vec{x}_m)} + \beta^{-1} \delta_{nm} = \frac{1}{\alpha} \phi_{nj} \underbrace{\phi_{mj}}_{\phi_{jm}^T} + \beta^{-1} \delta_{nm}, \text{ or}$$

$$C = \frac{1}{\alpha} \phi \phi^T + \beta^{-1} \mathbb{I}_N$$

$$\begin{aligned}
 \text{Then } m(\vec{x}_{N+1}) &= \alpha^{-1} \underbrace{\vec{\psi}_j^T(\vec{x}_{N+1}) \cdot \vec{\psi}_j(\vec{x}_n)}_{\psi_j(\vec{x}_{N+1}) \phi_{nj}} \times \\
 &\times (\alpha^{-1} \phi \phi^T + \beta^{-1} \mathbb{I}_N)^{-1} \vec{t} \Leftrightarrow \\
 &\Leftrightarrow \alpha^{-1} \vec{\psi}_j^T(\vec{x}_{N+1}) \phi^T (\alpha^{-1} \phi \phi^T + \beta^{-1} \mathbb{I}_N)^{-1} \vec{t}
 \end{aligned}$$

$$\text{Use } \underbrace{(\mathbb{I} + AB)^{-1} A}_{A + B^{-1}} = \underbrace{A(\mathbb{I} + BA)^{-1}}_{A + B^{-1}} \text{ to get:}$$

$$\Phi^T (\alpha^{-1} \Phi \Phi^T + \beta^{-1} \mathbb{I}_N)^{-1} =$$

$$= (\alpha^{-1} \Phi^T \Phi + \beta^{-1} \underbrace{\mathbb{I}_M}_{M})^{-1} \Phi^T = \alpha \beta \underbrace{(\beta \Phi^T \Phi + \alpha \mathbb{I}_M)^{-1}}_{S_N \text{ from (3.54)}} \Phi^T.$$

Then  $m(\bar{x}_{N+1}) = \sqrt{\alpha^{-1} \bar{\Psi}^T(\bar{x}_{N+1})} \underbrace{\alpha \beta S_N \Phi^T \bar{\epsilon}}_{(*)} = \beta \bar{\Psi}^T(\bar{x}_{N+1}) S_N \Phi^T \bar{\epsilon}$  (\*)

Recall that the mean of predictive distribution  $p(t|\bar{x})$  is given by ~~...~~

(3.58)  $\rightarrow \bar{\Psi}^T(\bar{x}) \cdot \bar{m}_N = \beta \bar{\Psi}^T(\bar{x}) S_N \Phi^T \bar{\epsilon}$ , same as (\*).

Similarly, one can show that

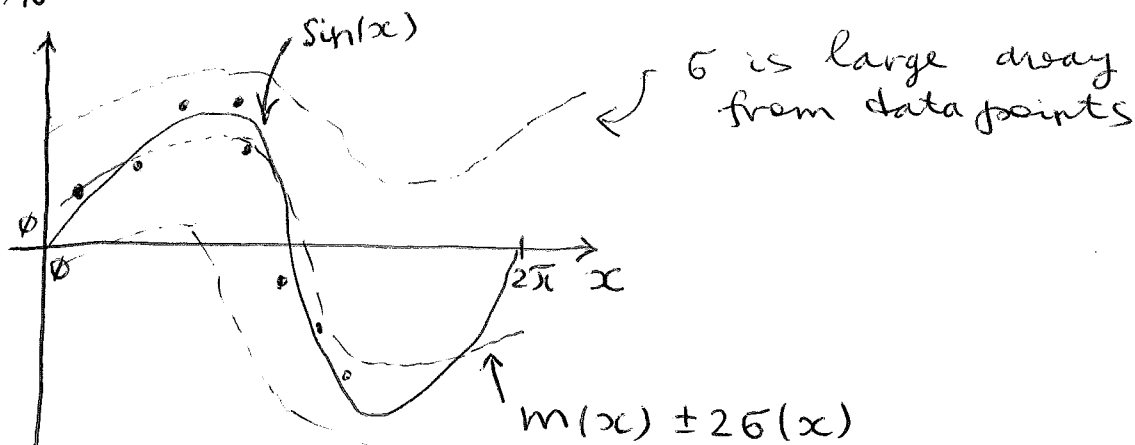
$$\sigma^2(\bar{x}_{N+1}) = \alpha^{-1} \bar{\Psi}(\bar{x}_{N+1})^T \cdot \bar{\Psi}(\bar{x}_{N+1}) + \beta^{-1} - \alpha^{-2} \bar{\Psi}(\bar{x}_{N+1})^T \Phi^T [\alpha^{-1} \Phi \Phi^T + \beta^{-1} \mathbb{I}_N]^{-1} \Phi \bar{\Psi}(\bar{x}_{N+1}) \quad (\ominus)$$

$$\ominus \beta^{-1} + \bar{\Psi}^T(\bar{x}_{N+1}) S_N \bar{\Psi}(\bar{x}_{N+1}).$$

$\uparrow$  skipped intermediate steps       $\uparrow$  same as (3.59)

Note that we need to invert  $C_N$  rather than  $S_N$  in the kernel method.

$\underbrace{C_N}_{N \times N}$       rather than       $\underbrace{S_N}_{M \times M}$       in the kernel method.





## Hyperparameters

Kernel f's often depend on a few parameters.

Consider  $p(\vec{t} | \vec{\theta})$  joint prob. of the dataset  
dataset      hyperprms

We can maximize it by e.g. conjugate gradients to obtain  $\vec{\theta}_{ML}$ :

$$\log p(\vec{t} | \vec{\theta}) = -\frac{1}{2} \vec{t}^T C_N^{-1} \vec{t} - \frac{1}{2} \log |C_N| - \frac{N}{2} \log(2\pi)$$

We also need

$$\frac{\partial}{\partial \theta_i} \log p(\vec{t} | \vec{\theta}) = \frac{1}{2} \vec{t}^T C_N^{-1} \frac{\partial C_N}{\partial \theta_i} C_N^{-1} \vec{t} - \frac{1}{2} \text{Tr} \left( C_N^{-1} \frac{\partial C_N}{\partial \theta_i} \right)$$

Use  $\frac{\partial}{\partial x} A^{-1} = -A^{-1} \frac{\partial A}{\partial x} A^{-1}$ ,  
 $\frac{\partial}{\partial x} \log |A| = \text{Tr} \left( A^{-1} \frac{\partial A}{\partial x} \right)$ .

## Automatic relevance determination

$D \leftarrow$  #dims in input space

Consider  $k(\vec{x}, \vec{x}') = \theta_0 e^{-\frac{1}{2} \sum_{i=1}^D (x_i - x'_i)^2 \eta_i}$

As  $\eta_i$  becomes small, the kernel becomes less sensitive to the choice of the variable  $x_i \Rightarrow x_i$  can be discarded. Allows to rank all input vars in terms of their usefulness for predicting t.

Can be used instead of (6.63) as well:

$$k(\vec{x}, \vec{x}') = \theta_0 e^{-\frac{1}{2} \sum_{i=1}^D (x_i - x'_i)^2 \eta_i} + \theta_2 + \theta_3 \sum_{i=1}^D x_i x'_i$$

=