Gaussian processes

Consider Bayesian regression for

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

Prior $p(\vec{w}|\alpha)$ can be thought of as inducing a prior over functions $y(\vec{x}, \vec{w})$. Given data, prior over weights 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 $y(\vec{x}) = \underbrace{\vec{w}^T}_{\text{dim. } M} \cdot \vec{\varphi}(\vec{x})$ with

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

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:

$$\vec{y} = \Phi \vec{w} \qquad \vec{y} = \underline{y_1 \dots y_N}$$

Φ ~ design matrix: $\Phi_{nk} = \varphi_k(\vec{x}_n)$

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

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

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

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

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

In general, a gaussian process is a prob. distribution over functions $y(\vec{x})$ s.t. if we take arbitrary points $\vec{x}_1, \ldots, \vec{x}_N \Rightarrow \vec{y} = \underline{y(\vec{x}_1) \ldots y(\vec{x}_N)}$ is such that $P(\vec{y})$ is gaussian.
" $P(y_1, \ldots, y_N)$

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

In most applications, we will assume $\mathbb{E}[\vec{y}] = 0$ 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 $\vec{y}$.

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

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

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

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

$$p(\vec{t}) = \int d\vec{y}\; p(\vec{t} | \vec{y})\, p(\vec{y}) \quad \boxed{=}$$

Once again,

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

$$\| \int d\vec{x}\, p(\vec{y} \mid \vec{x})\, p(\vec{x})$$

Here, $\begin{cases} \vec{y} \to \vec{t} \\ \vec{x} \to \vec{y} \end{cases} \Rightarrow \begin{cases} \vec{\mu} = \vec{0} \\ A\vec{x} + \vec{b} \Rightarrow A\vec{y} + \vec{b} = \vec{y}, \end{cases} \leftarrow \begin{cases} A = \mathbb{I}_N, \\ \vec{b} = \vec{0} \end{cases}$

$\Lambda^{-1} = K$

$L^{-1} = \beta^{-1}\mathbb{I}_N$

$$\boxed{=} \; \mathcal{N}(\vec{t} \mid \vec{0}, \underbrace{\beta^{-1}\mathbb{I}_N + K}_{\| \; G \text{ (or } G_N)})$$

s.t. $C(\vec{x}_n, \vec{x}_m) = \underbrace{k(\vec{x}_n, \vec{x}_m)}_{\substack{\text{uncertainty due to randomness in } \vec{y}}} + \underbrace{\beta^{-1}\delta_{nm}}_{\substack{\text{noise due to } \varepsilon_n}}$

Another kernel (popular in regression):

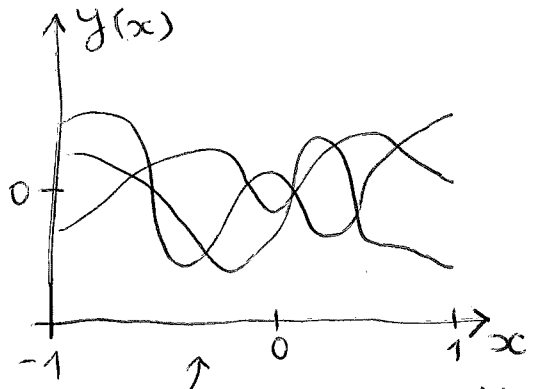
$$k(\vec{x}_n, \vec{x}_m) = \theta_0 e^{-\frac{\theta_1}{2}\|\vec{x}_n - \vec{x}_m\|^2} + \theta_2 + \theta_3 \vec{x}_n^T \vec{x}_m$$

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

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

θ_1

$(1, \underbrace{4}_{\theta_1}, 0, 0)$

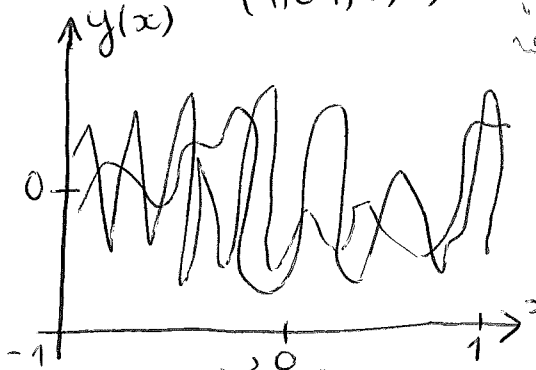


choose $\{\vec{x}_n\}_1^N$ equidistantly spaced in $[-1, 1]$

θ_1

$(1, \underbrace{64}_{\theta_1}, 0, 0)$

between neighboring $\vec{x}_n$ & $\vec{x}_m$ correlation is lower with higher θ_1



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

Now, in regression we are given

$\vec{t}_N = \underbrace{t_1, \ldots, t_N}$ along with $\underbrace{\vec{x}_1, \ldots, \vec{x}_N}$

& we want to know

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

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

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

$$C_{N+1} = \begin{pmatrix} C_N & \vec{k} \\ \vec{k}^T & c \end{pmatrix}$$

$\vec{k}$: $k(\vec{x}_n, \vec{x}_{N+1})$, $n = 1, \ldots, N$ — elements of N-dim vector

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

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

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

Likewise,

$$\vec{\mu} = \begin{pmatrix} \vec{\mu}_a \\ \vec{\mu}_b \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \overbrace{\Sigma_{aa}}^{M\times M} & \overbrace{\Sigma_{ab}}^{M\times(D-M)} \\ \underbrace{\Sigma_{ba}}_{(D-M)\times M} & \underbrace{\Sigma_{bb}}_{(D-M)\times(D-M)} \end{pmatrix} \Big\} D\times D$$

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

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

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

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

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

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

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

Finally, note that

$$m(\vec{x}_{N+1}) = \sum_{n,n'} k_n C^{-1}_{N,nn'} t_{n'} =$$

$$= \sum_n \Big(\underbrace{\sum_{n'} C^{-1}_{N,nn'} t_{n'}}_{\text{“}a_n\text{”}} \Big) \underbrace{k_n}_{\text{“}k(\vec{x}_n, \vec{x}_{N+1})\text{”}} = \sum_n a_n k(\vec{x}_n, \vec{x}_{N+1}).$$

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

$$C_{nm} = \underbrace{K_{nm}}_{\frac{1}{\alpha}\varphi_j(\vec{x}_n)\cdot\varphi_j(\vec{x}_m)} + \beta^{-1}\delta_{nm} = \frac{1}{\alpha}\varphi_{nj}\underbrace{\varphi_{mj}}_{\varphi^T_{jm}} + \beta^{-1}\delta_{nm}, \quad \text{or}$$

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

Then $m(\vec{x}_{N+1}) = \alpha^{-1}\underbrace{\vec{\varphi}^T(\vec{x}_{N+1})\cdot\vec{\varphi}(\vec{x}_n)}\times$

$\times(\alpha^{-1}\varphi\varphi^T + \beta^{-1}\mathbb{I}_N)^{-1}\vec{t} \equiv$ $\varphi_j(\vec{x}_{N+1})\underbrace{\varphi_{nj}}_{\varphi^T_{jn}}$

$\equiv \alpha^{-1}\vec{\varphi}^T(\vec{x}_{N+1})\underbrace{\varphi^T(\alpha^{-1}\varphi\varphi^T + \beta^{-1}\mathbb{I}_N)^{-1}\vec{t}}$

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

Indeed, $(\mathbb{I}+AB)^{-1}A(\mathbb{I}+BA) = (\mathbb{I}+AB)^{-1}(\mathbb{I}+AB)A = A$,

as expected, from the RHS: $A(\mathbb{I}+BA)^{-1}(\mathbb{I}+BA) = A$.

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

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

Then $m(\vec{x}_{N+1}) = \overbrace{\alpha\beta S_N \Phi^T \vec{t}}^{\alpha^{-1}\vec{\varphi}^T(\vec{x}_{N+1})} = \beta\vec{\varphi}^T(\vec{x}_{N+1})S_N\Phi^T\vec{t}$ (*)

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

(3.58) → $\vec{\varphi}^T(\vec{x})\cdot\vec{m}_N = \beta\vec{\varphi}^T(\vec{x})S_N\Phi^T\vec{t}$, same as (*).

Similarly, one can show that

$$\sigma^2(\vec{x}_{N+1}) = \alpha^{-1}\vec{\varphi}(\vec{x}_{N+1})^T\cdot\vec{\varphi}(\vec{x}_{N+1}) + \beta^{-1} -$$

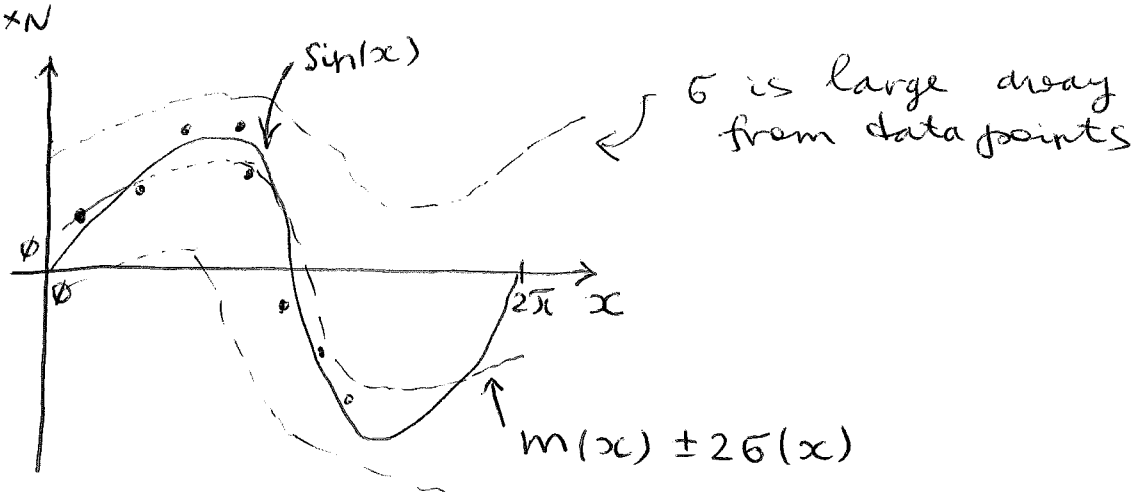
$$- \alpha^{-2}\vec{\varphi}(\vec{x}_{N+1})^T\Phi^T[\alpha^{-1}\Phi\Phi^T + \beta^{-1}\mathbb{I}_N]^{-1}\Phi\vec{\varphi}(\vec{x}_{N+1}) \quad (=)$$

$$(=)\ \beta^{-1} + \vec{\varphi}^T(\vec{x}_{N+1})S_N\vec{\varphi}(\vec{x}_{N+1}).$$

↑ skipped intermediate steps

↑ same as (3.59)

Note that we need to invert $\underbrace{C_N}_{N\times N}$ 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

($\vec{t}$: dataset, $\vec{\theta}$: 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

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

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

Automatic relevance determination

Consider $k(\vec{x},\vec{x}') = \theta_0 e^{-\frac{1}{2}\sum_{i=1}^{D}(x_i - x_i')^2\eta_i}$ ($D \leftarrow$ #dims in input space)

As η_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:

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