Gaussian Processes

* Motivation

1) $f(x) = \sum_{m=1}^{M} a_m \phi_m(x)$ in Bayesian framework

$M \to \infty \Rightarrow$ GP

2) Bayesian Neural Network

$p(\vec{\omega}|D) \propto p(D|\vec{\omega})\, p(\vec{\omega})$

↳ 1 hidden layer with H units

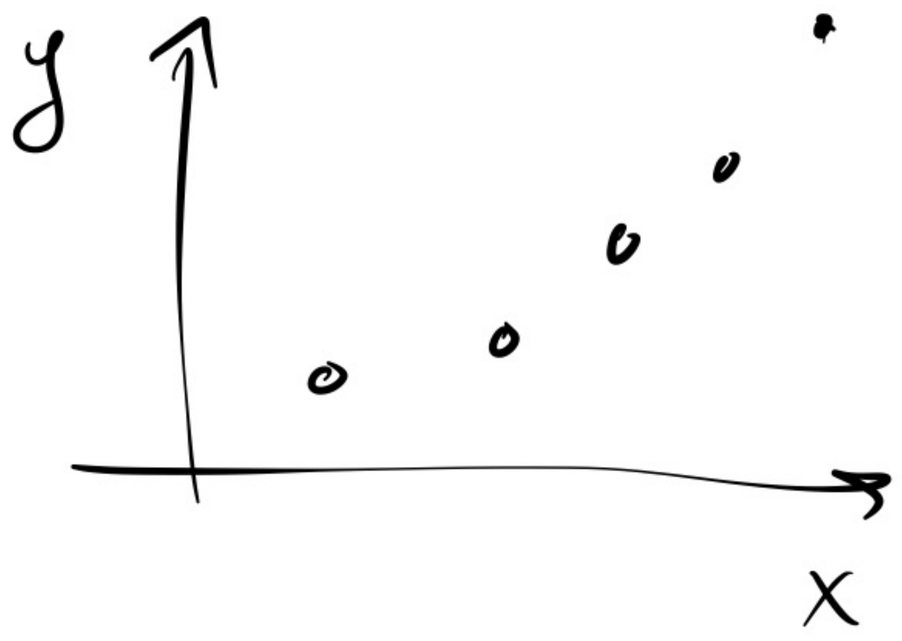
$H \to \infty \Rightarrow$ distribution of the output is GP

(Neal '96)

GP flexibility, expressiveness

↳ non-parametric

the complexity grows with size of dataset



Parametric approach

1 — assume $p(y)$

e.g.

$$p(y_i) = \mathcal{N}(y_i \mid f(x_i, \vec{w}), \sigma^2)$$

2 — $f(x_i, \vec{w})$ is given by a ML model

Non parametric

(previous lecture) ⇒ relax the 1st step

today ⇒ relax 2 step

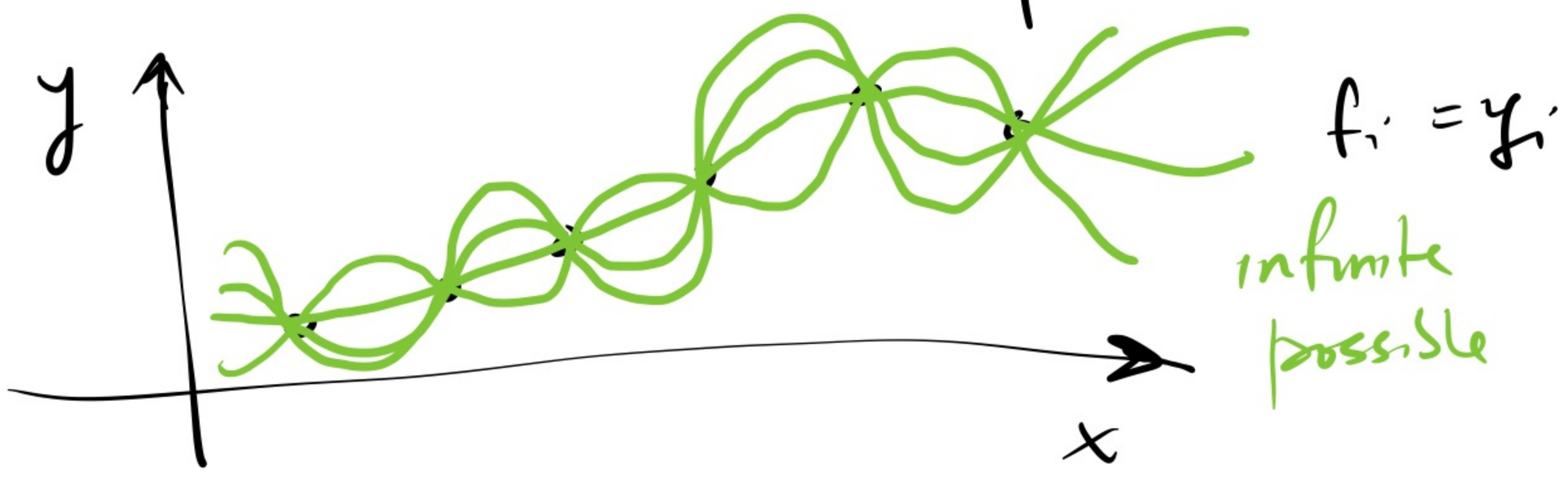
GP:

— Do assume a particular distrib. of your observable ⇒ Gaussian

— $f(x, \vec{w})$ ⇒ not fixed

knowing only values $f_i = f(x_i)$

of some finite # of points $\{x_i\}$



z is random variable

$f(x)$ as a random function

Task: define distrb over functions

mean function

$$m(\vec{x}) = \mathbb{E}[f(\vec{x})]$$

covariance of the function

$$\mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

$$\equiv k(x, x')$$

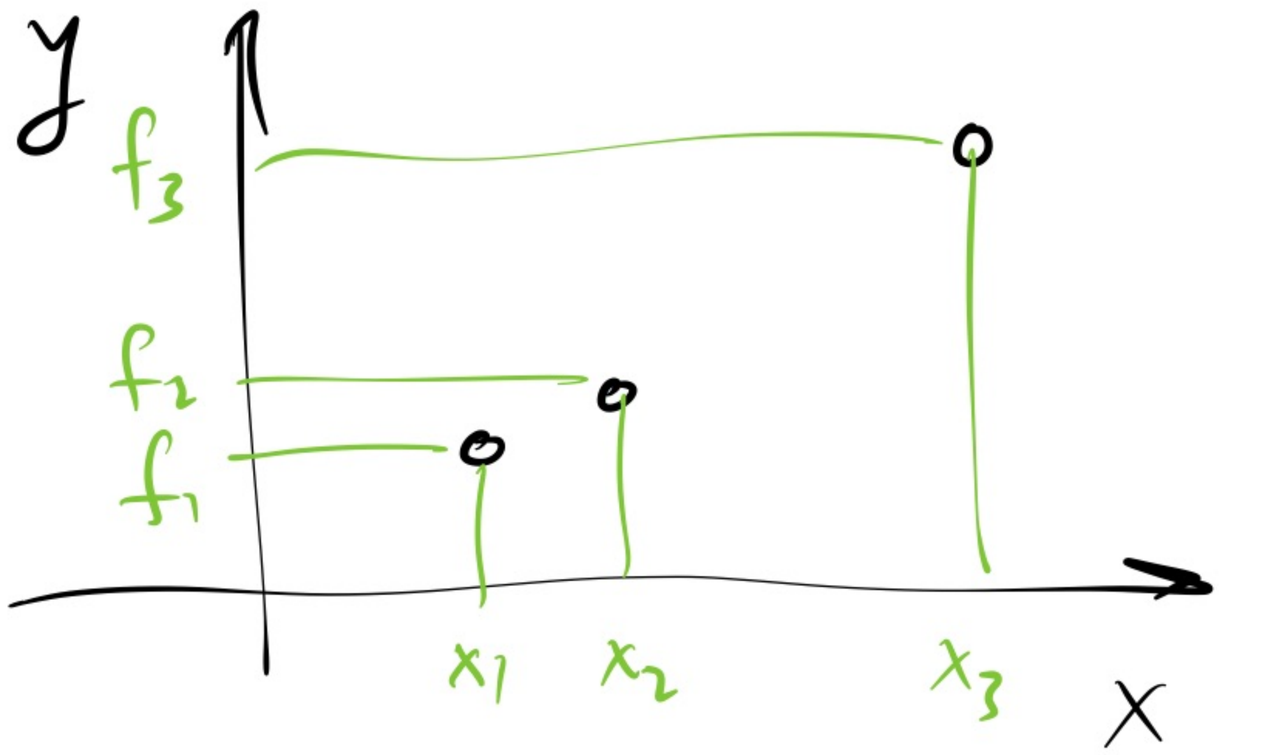
$f(x)$ is a GP

$$f(x) \sim GP(m(x), k(x, x'))$$

DEF: A GP is a collection of variables, any finite # of which have a joint Gaussian distrb.

$\{f_i\}$

Our task : estimate generating function of data



Assume function is smooth

⇓

cov. matrix should reflect some measure of similarity

x' close x

$f(x')$ close $f(x)$

to be correlated

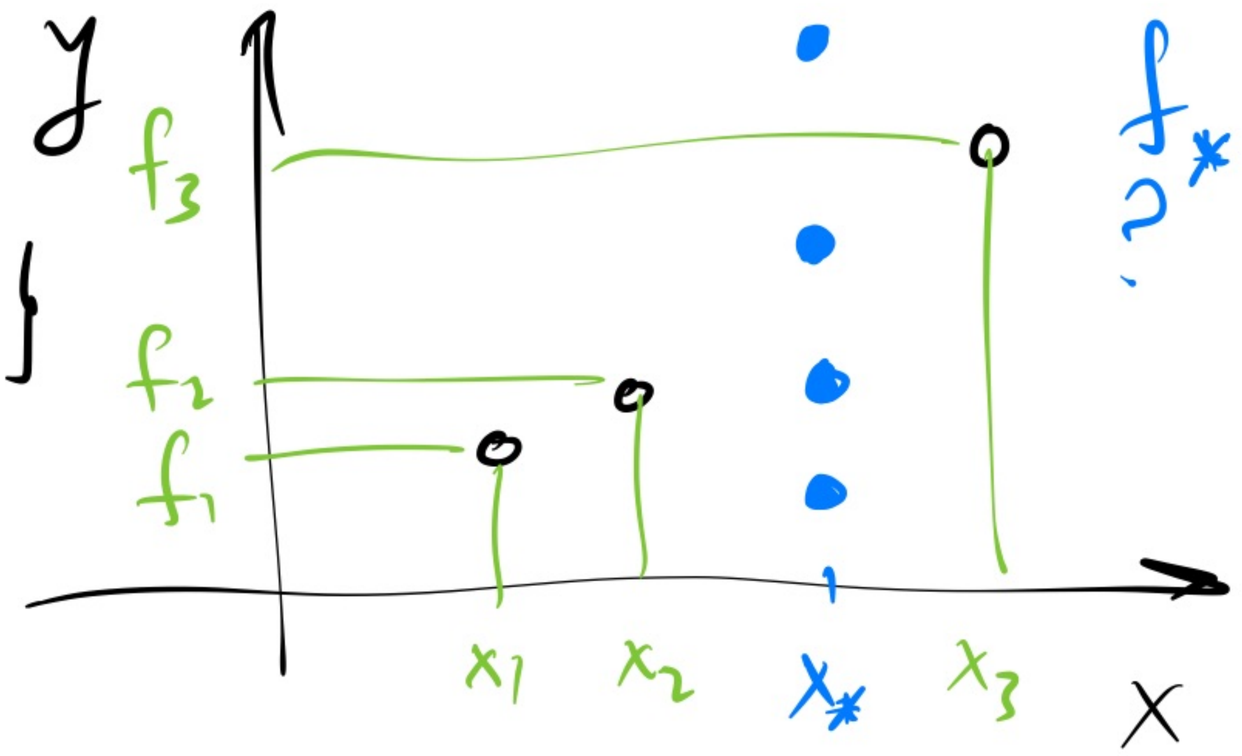
$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.7 & 0.2 \\ 0.7 & 1 & 0.5 \\ 0.2 & 0.5 & 1 \end{bmatrix} \right)$$

↓ kernel function

$$k_{ij} = e^{-\lambda |x_i - x_j|^2} = \begin{cases} 0 & |x_i - x_j| \to \infty \\ 1 & x_i = x_j \end{cases}$$

Let's evaluate f_*

Given $\vec{f} = \{f_1, f_2, f_3\}$



$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_* \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_{3\times3} & \vec{k}_*^{\,T} \\ \vec{k}_* & k_{**} \\ 3\times1 & 1\times1 \end{bmatrix} \right)$$

Aim $p(f_* \mid \vec{f})$

Property

$$\vec{f} = \{f_1, f_2, \ldots, f_N\}$$

$$p(\vec{f}) = N(\vec{f} \mid \vec{\mu}, \Sigma)$$

$$p(\underbrace{f_1, \ldots, f_k}_{\vec{f}_1} \mid \underbrace{f_{k+1}, \ldots, f_N}_{\vec{f}_2}) \Rightarrow \text{Gaussian}$$

with mean $\mu_{1/2}$

Cov $\Sigma_{1/2}$

$$p(\vec{f}) = N\left(\begin{bmatrix} \vec{\mu}_1 \\ \vec{\mu}_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right)$$

$$\left\{ \begin{aligned} \mu_{1|2} &= \vec{\mu_1} + \Sigma_{12}\Sigma_{22}^{-1}(\vec{f_2} - \vec{\mu_2}) \\ \Sigma_{1|2} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \end{aligned} \right\} \quad (I)$$

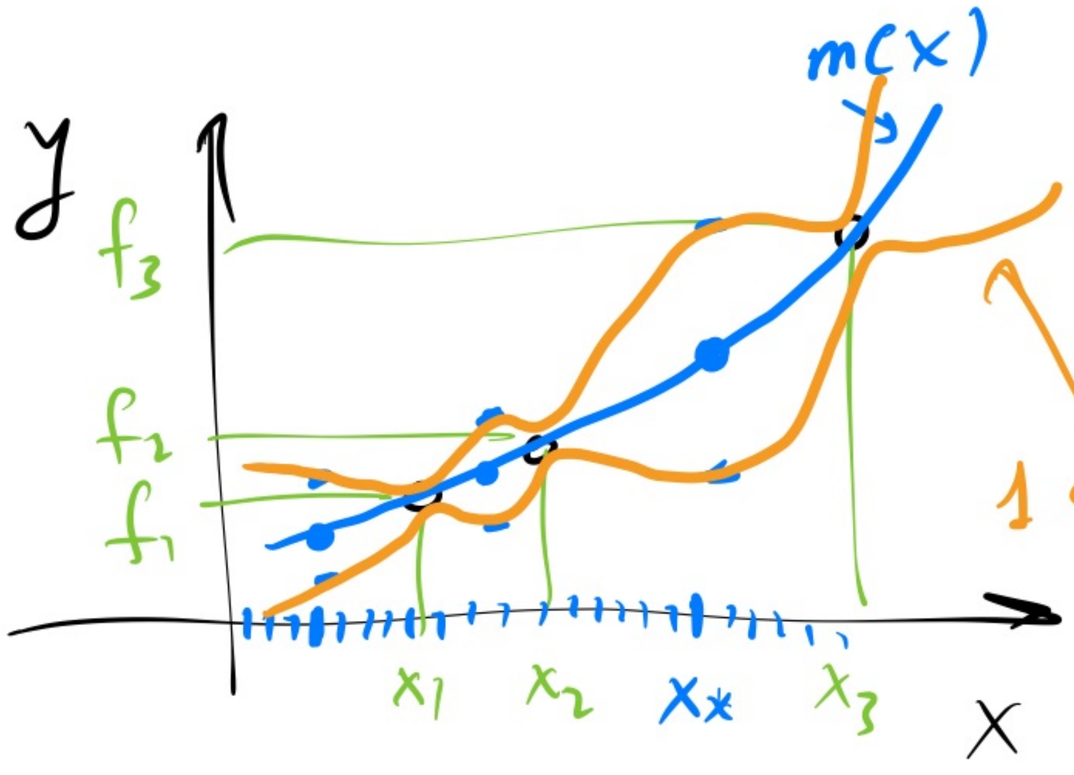
Using (I)

$p(f_* | \vec{f}) \Rightarrow$ Gaussian with

$$\mathbb{E}[f_* | \vec{f}] = \vec{k}_*^T K^{-1} \cdot \vec{f}$$

Variance $\sigma_*^2 = k_{**} - \vec{k}_*^T \cdot K^{-1} \cdot \vec{k}_*$

mean and variance of a point in the test set



Go do the same for many x_*

Summary

- Propose a GP prior
  - create a grid $\{x_i\}_{i=1}^{N}$
  - compute the matrix $K_{N\times N}$ (covariance)

$$k_{ij} = \exp\left\{-\frac{1}{2}(x_i - x_j)^2\right\}$$

  - set values of the function $f_i \equiv f(x_i)$
    (sample functions from the prior)

$$\text{sample } \vec{f} \sim N(\vec{0}, K) = p(\vec{f})$$

how to do ↗

if $f \sim N(\mu, \sigma^2)$

sampling from ↑ the same as from

$$f \sim \mu + \sigma N(0,1)$$

For multivariate Gaussian

$$\vec{f} \sim N(\vec{\mu}, K)$$

$$K = L \cdot L^T \Rightarrow \text{Cholesky decomposition}$$

L: real - lower triangular matrix with positive diag entries

$$\vec{f} \sim \vec{\mu} + L \cdot N(\vec{0}, \mathbb{1})$$

- Compute $p(f_* | \vec{y})$

Build the Gaussian joint

$$p\begin{pmatrix} \vec{y} \\ \vec{f}_* \end{pmatrix} = N\left(\begin{bmatrix} \vec{0}_N \\ \vec{0}_M \end{bmatrix}, \begin{bmatrix} K & K_* \\ K_*^T & K_{**} \end{bmatrix} \right)$$

$\vec{y}$: observations, size N

$\vec{f}_*$: predictions for test data, size M

K: $N \times N$

K_*: $N \times M$

K_*^T: $M \times N$

K_{**}: $M \times M$

$$\mathbb{E}[\vec{f}_* | \vec{y}] = K_*^T \cdot K^{-1} \cdot \vec{y} = \vec{\mu}_*$$

$$\text{Cov}[\vec{f}_* | \vec{y}] = K_{**} - K_*^T \cdot K^{-1} \cdot K_*$$

$$\Sigma_* = L_* L_*^T$$

- Draw predictors

$$\vec{f}_*^{(s)} = \vec{\mu}_* + L_* \cdot \vec{\varepsilon}^{(s)}$$

sample s

$$\vec{\varepsilon}^{(s)} \sim \mathcal{N}(\vec{0}, \mathbb{1})$$

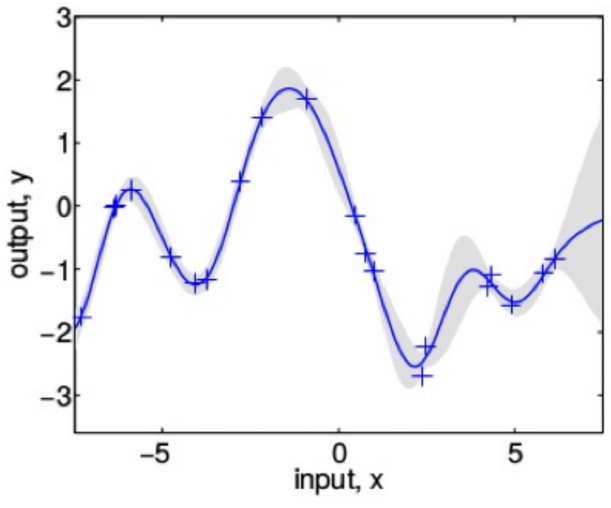
---

kernel function may have hyperparameters

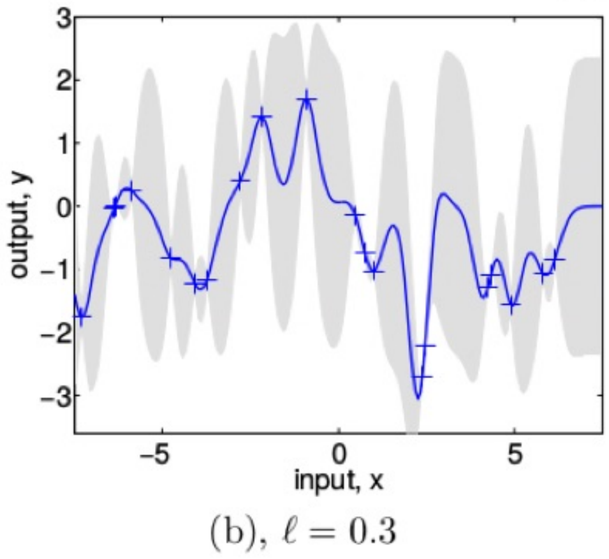
$$k(x, x') = \sigma_f^2 \exp\left\{ -\frac{1}{2\ell^2} (x - x')^2 \right\}$$

a priori should be optimised

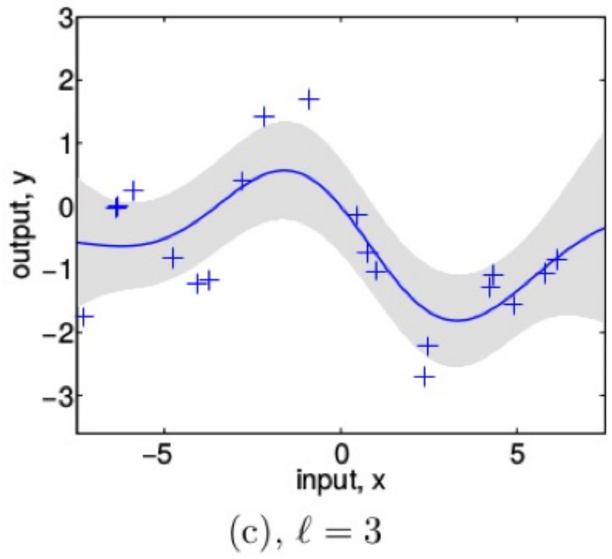
predictions smoother or not depending on



(a), $\ell = 1$



(b), $\ell = 0.3$



(c), $\ell = 3$

(Rasmussen et al., "GP for ML")

⇓

should learn the correct values of the hyperparameters

you maximise the evidence as a function of the hyperparams.

In general noisy data

$$y = f(\vec{x}) + \varepsilon$$

$$\varepsilon \hookrightarrow N(0, \sigma^2)$$

The prior covariance is changed as

$$\vec{f} \sim N\left(\vec{0}, K + \sigma^2 \mathbb{1}\right)$$

and the predictions

mean: $K_*^T \left(K + \sigma^2 \mathbb{1}\right)^{-1} \vec{y}$

Cov: $K_{**} - K_*^T \left(K + \sigma^2 \mathbb{1}\right)^{-1} \cdot K_*$

Classification

$$p(C_k | \vec{x})$$

1-to-k true labels ?

The likelihood

$$p(D|\vec{w}) = \prod_{i=1}^{N} \prod_{k=1}^{K} p(C_k|\vec{x}_i, \vec{w})^{t_{ik}}$$

propose prior $p(\vec{w})$

$\Rightarrow$ Posterior $p(\vec{w}|D)$

The prediction distrib.

$$p(C_k(\vec{x}_*, D) = \int d\vec{w}\; p(C_k|\vec{x}_*, \vec{w})\; p(\vec{w}|D)$$

output of classifier

In GP:

$\Rightarrow \{f_i\}^{\in \mathbb{R}} \quad -\infty < f_i < \infty$

$\Downarrow$

$[0,1]$ { 'squash' f_i in a bounded function

$\sigma(f(\vec{x}))$ $\Rightarrow$ sigmoid $[0,1]$

Binary:

$\Downarrow$ $p(C=1|\vec{x})$ in binary classif.

Predictive more complicated:

$$p(y_* = c_1 \mid \vec{x}_*, D)$$

$$= \int \sigma(f_*)\, p(f_* \mid \vec{x}_*, D)\, df_*$$

↳ latent variable

$$p(f_* \mid \vec{x}_*, D) = \int p(f_* \mid \vec{x}_*, \vec{f})\; p(\vec{f} \mid \vec{y})\, d\vec{f}$$

the conditional as above

posterior of $\vec{f}$ latent functions

$$p(\vec{f} \mid \vec{y}) = \frac{p(\vec{y} \mid \vec{f})\; p(\vec{f} \mid X)}{p(\vec{y})}$$

GP prior

untractable

Likelihood (non-Gaussian)

A common choice of likelihood

$$\log p(y_i \mid f_i) = -\log\left(1 + e^{-y_i f_i}\right)$$

$$p(y_i \mid f_i) = \frac{1}{1 + e^{-y_i f_i}}$$

This is reasonable

	$f_i \to \infty$	$f_i \to -\infty$
$y_i = +1$	$p(y_i \mid f_i) \to 1$ high likelihood	$p \to 0$
$y_i = -1$	$p \to 0$ low likelihood	$p \to 1$

For multi-class classif.

instead of 1 latent function

$$\vec{f} = \{f_1, \dots, f_N\}$$

↳ # datapoints

⇓

1 latent func. per class, K

$$\{\vec{f}_k\} \qquad k = 1, \dots, K$$

$$F = \begin{bmatrix} f_{11} & f_{12} \dots & f_{1K} \\ f_{21} & f_{22} \dots & f_{2K} \\ \vdots & & \\ f_{N1} & f_{N2} \dots & f_{NK} \end{bmatrix}$$

⇓ assumed uncorrelated

A popular likelihood

$$p(y_{ik} \mid \vec{f}_i) = \frac{e^{f_{ik}}}{\sum_{k'} e^{f_{ik'}}}$$

↘ Softmax

Another choice

$$p(y_{ik} \mid \vec{f}_i) = \prod_{k' \neq k} \theta(f_{ik} - f_{ik'})$$