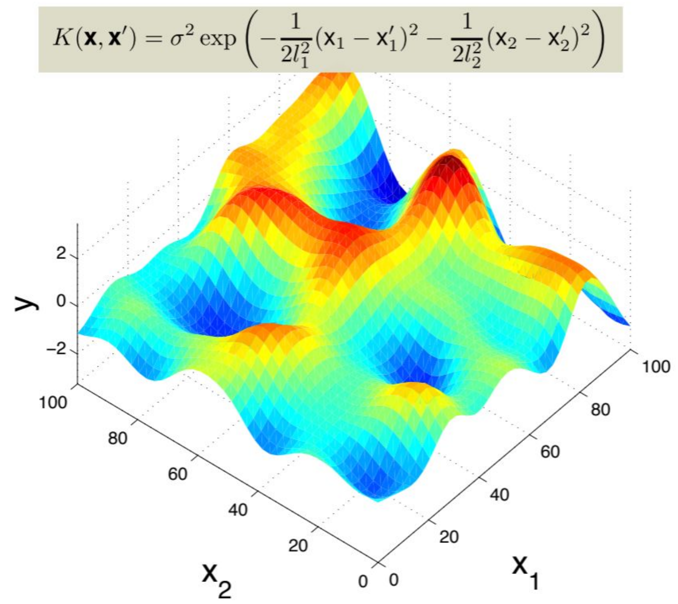


Gaussian Process Cheat Sheet

V2020.12.03

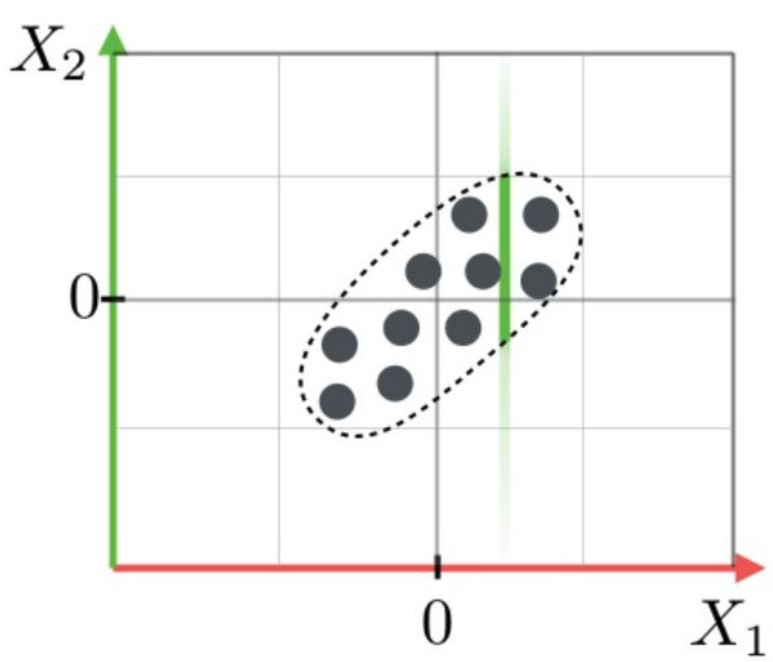
(Dr Yan Xu)

Summary

- ➔ Supervised learning method
- ➔ Non-linear smooth representation
- ➔ Bayesian (prior/posterior: Gaussian)
- ➔ Measure of Similarity (Kernel Functions)
- ➔ Non-parametric (excluding Kernels)
- ➔ Regression (strength) and Classification
- ➔ Small Problem (<10K records)

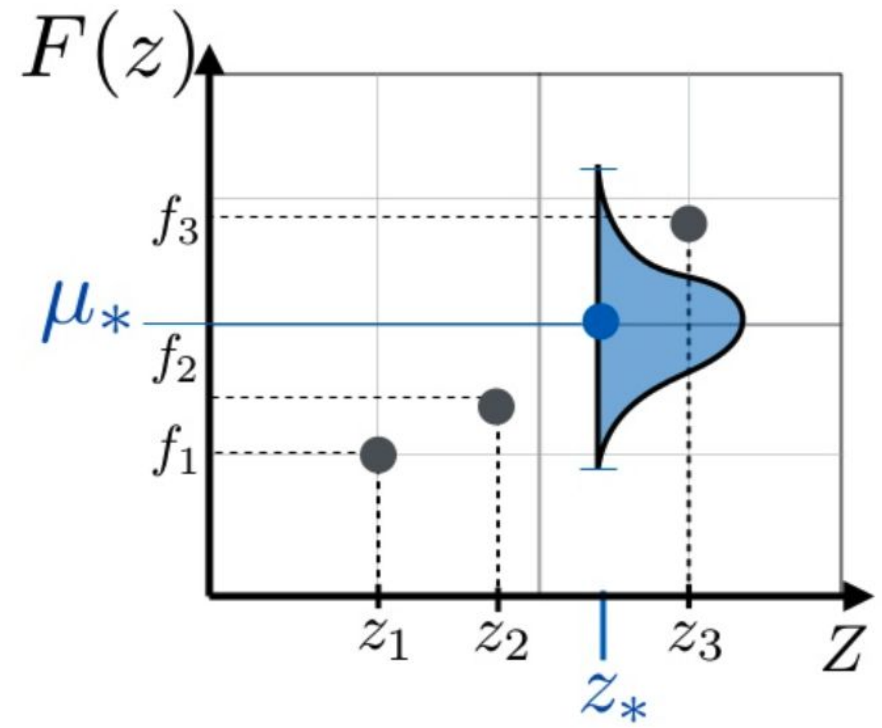
Multivariate Gaussian

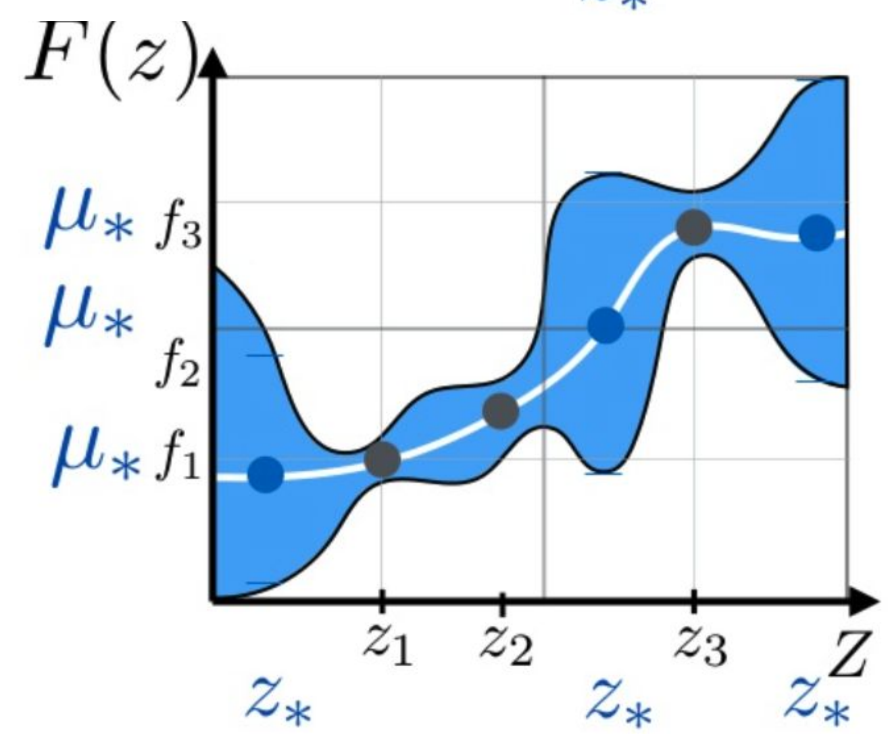
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \right)$$



Multivariate Gaussian (Continued)

➔ Conditional Distribution Prediction





Gaussian Process (Bayesian)

- Prior:

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}, \mathbf{K})$$

where $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ and $\boldsymbol{\mu} = (m(\mathbf{x}_1), \ldots, m(\mathbf{x}_N))$.

  - k() is a positive definite kernel
  - It is common that: m(x)=0, since GP is flexible enough to model the mean

- Posterior:

$$\begin{aligned} p(\mathbf{f}_*|\mathbf{X}_*, \mathbf{X}, \mathbf{f}) &= \mathcal{N}(\mathbf{f}_*|\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*) \\ \boldsymbol{\mu}_* &= \boldsymbol{\mu}(\mathbf{X}_*) + \mathbf{K}_*^T \mathbf{K}^{-1}(\mathbf{f} - \boldsymbol{\mu}(\mathbf{X})) \\ \boldsymbol{\Sigma}_* &= \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{K}_* \end{aligned}$$

- It is common that: u(x)=0

Gaussian Process (Kernel)

➔ Squared Exponential Kernel (or RBF)

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

➔ RBF Kernel with Gaussian Noise

$$\kappa_y(x_p, x_q) = \sigma_f^2 \exp(-\frac{1}{2\ell^2}(x_p - x_q)^2) + \sigma_y^2 \delta_{pq}$$

➔ kernel parameters:

- *l* : horizontal scale (critical)
- $\square_f$: vertical scale
- $\square_y$: observe noise variance

➔ kernel parameters estimation (MLE)

- non-convex (multiple local min)
- with constraints, e.g. >= 0

➔ Default kernel for GP and SVM

Computational Algorithm (Regression)

Algorithm 15.1: GP regression

$\mathbf{L} = \text{cholesky}(\mathbf{K} + \sigma_y^2 \mathbf{I})$;
$\boldsymbol{\alpha} = \mathbf{L}^T \setminus (\mathbf{L} \setminus \mathbf{y})$;
$\mathbb{E}[f_*] = \mathbf{k}_*^T \boldsymbol{\alpha}$;
$\mathbf{v} = \mathbf{L} \setminus \mathbf{k}_*$;
$\text{var}[f_*] = \kappa(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{v}^T \mathbf{v}$;
$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^T \boldsymbol{\alpha} - \sum_i \log L_{ii} - \frac{N}{2}\log(2\pi)$

source

Python Libraries

- ➔ sklearn
- ➔ gpytorch

Github Example

- ➔ regression example

Known Issues

- ➔ Computationally Slow
- ➔ Overfitting
- ➔ *Not Recommended in Production*