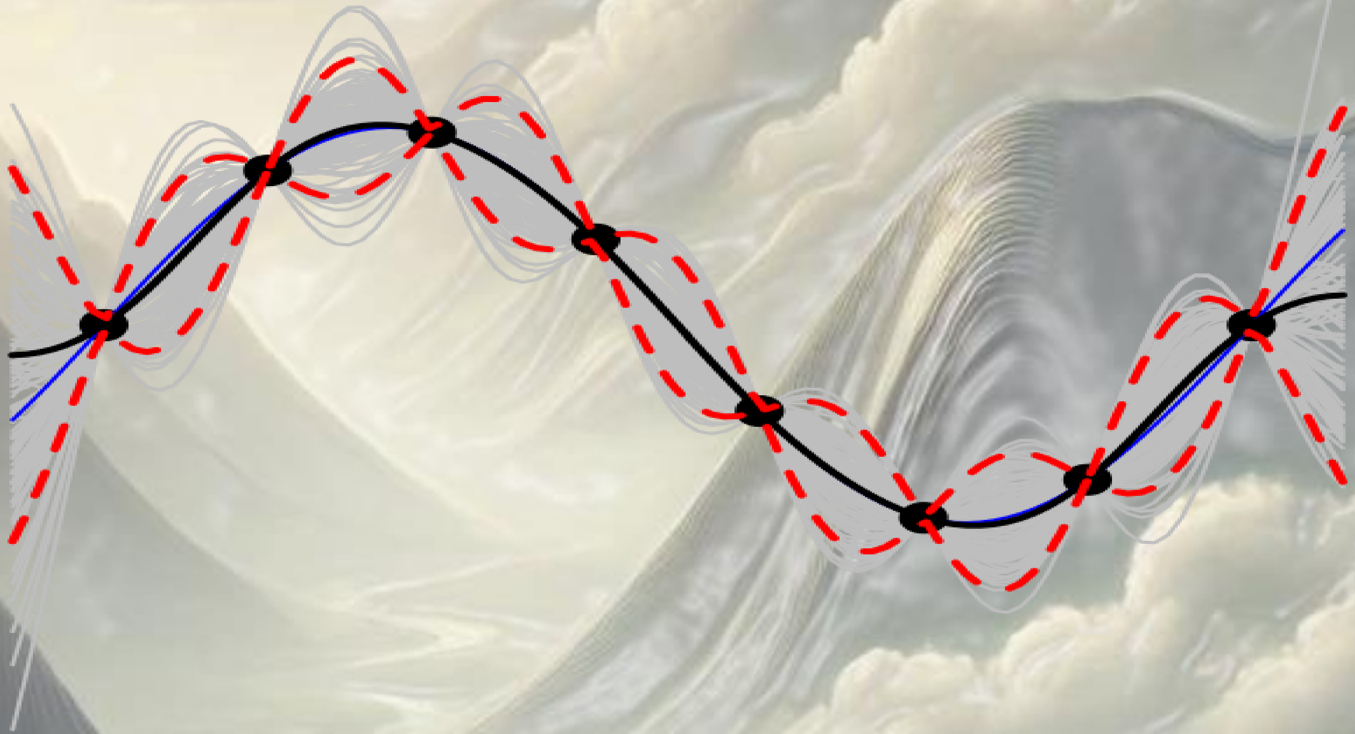
and why they may not be machine learning(?)

Gaussian Processes



Useful references:

Gaussian Processes for Machine Learning by Rasmussen and Williams @ www.GaussianProcess.org/gpml

What are Gaussian Processes?

A ***Gaussian process (GP)*** is a *probability distribution* over possible *functions*; any combination of these functions jointly Gaussian distributed

Key Points:

- GPs can be used as models for both regression and classification
- Since the model is a probability distribution (over a function space that is Gaussian distributed), we can calculate *means* AND *variances*
- The ***mean function*** is calculated from the posterior (recall Baye's) distribution of possible functions → regression predictions
- The ***variances*** provide native uncertainty measures on these predictions, which are typically absent in other ML models (e.g., ANNs)
- The model, as a posterior, can be updated based on new observations

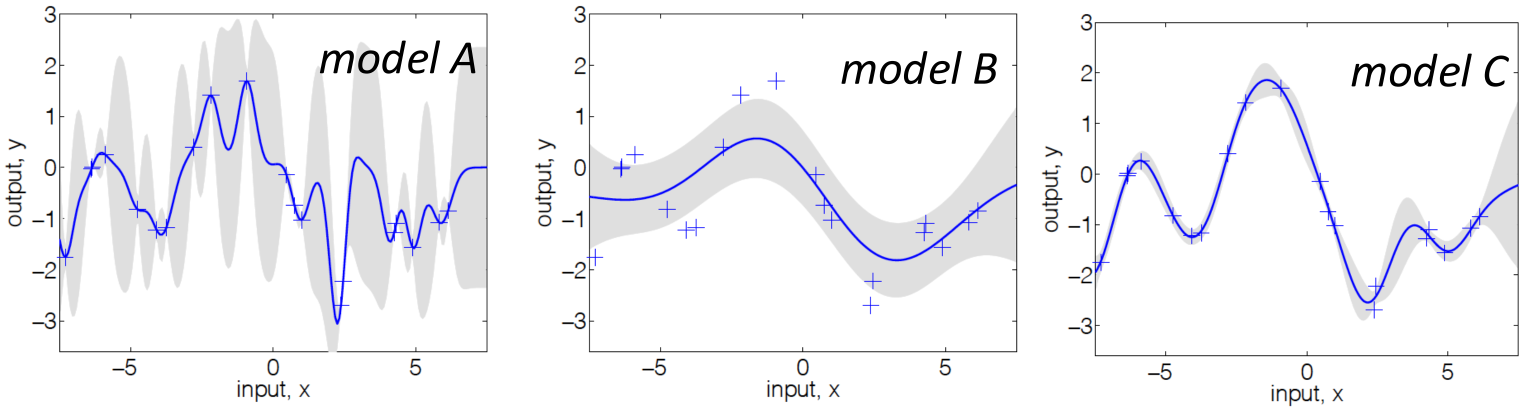
In "math" speak:

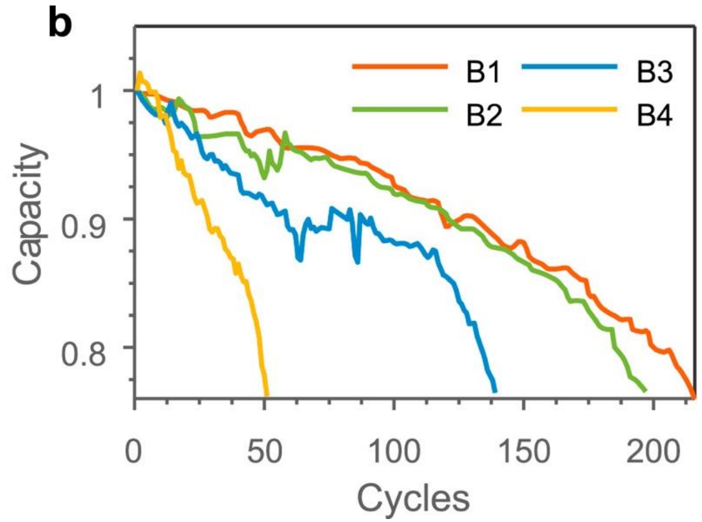
$$\boldsymbol{f}(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{x}), \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}'))$$

covariance/Kernel function that dictates model "smoothness"

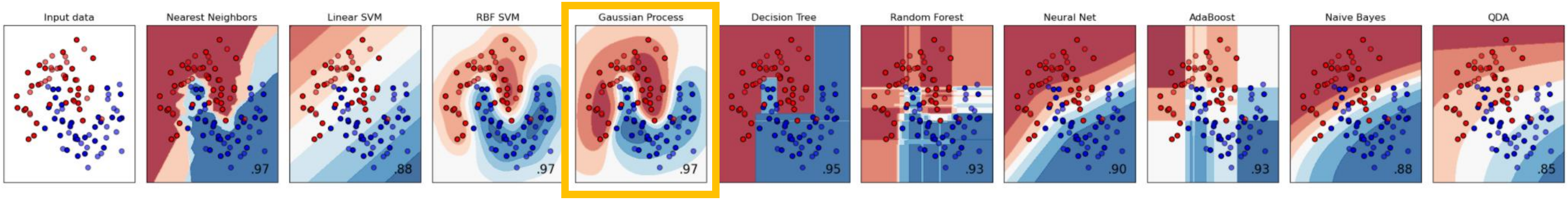
Why should we use them?

- Our observations may be "***noisy***," and it is important to capture that behavior
- They allow you to incorporate ***prior knowledge***
- They provide a reasonable framework to ***regulate model complexity*** and more reliably ***indicate model uncertainty***





Richardson et al. *J. Power Sources* **(2017)**

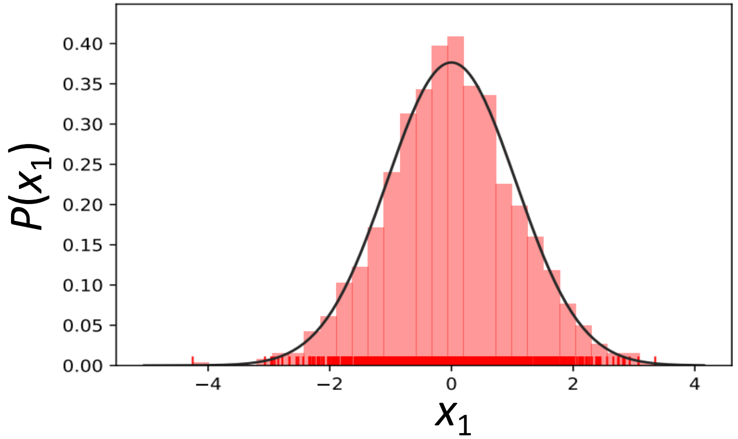


Some basic intuition underlying Gaussian Processes

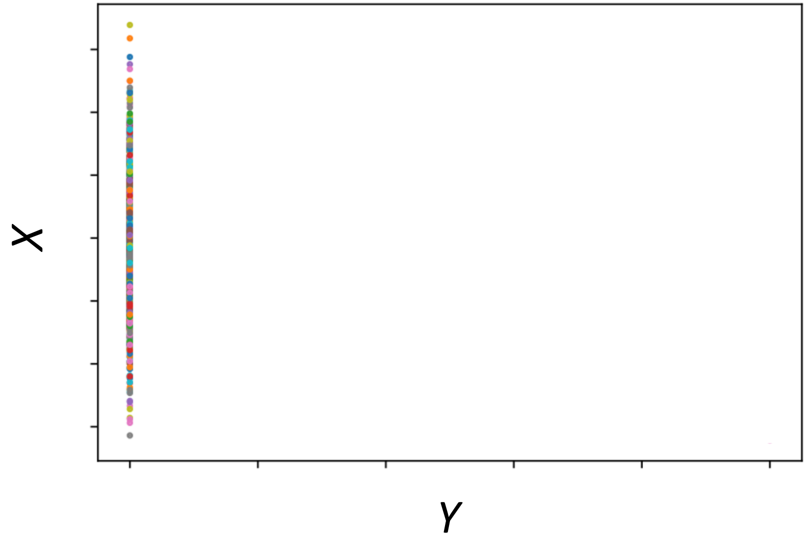
Suppose we have a random variable X_1 with distribution

$$P_{X_1}(x_1) \sim \mathcal{N}(\mu, \sigma^2)$$

We can generate samples from this distribution



and then project them onto a new axis

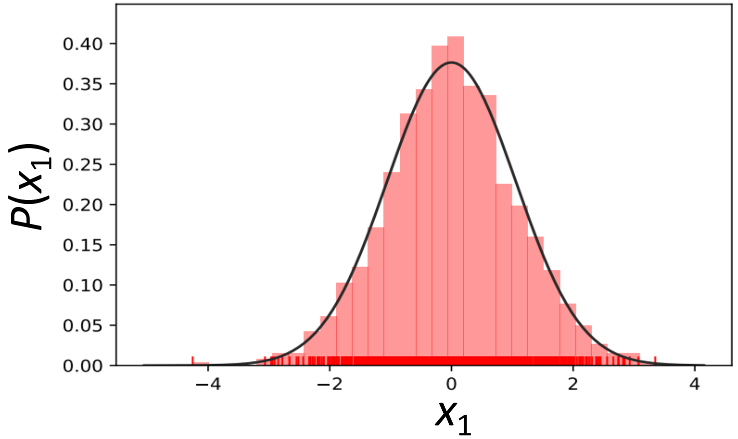


Some basic intuition underlying Gaussian Processes

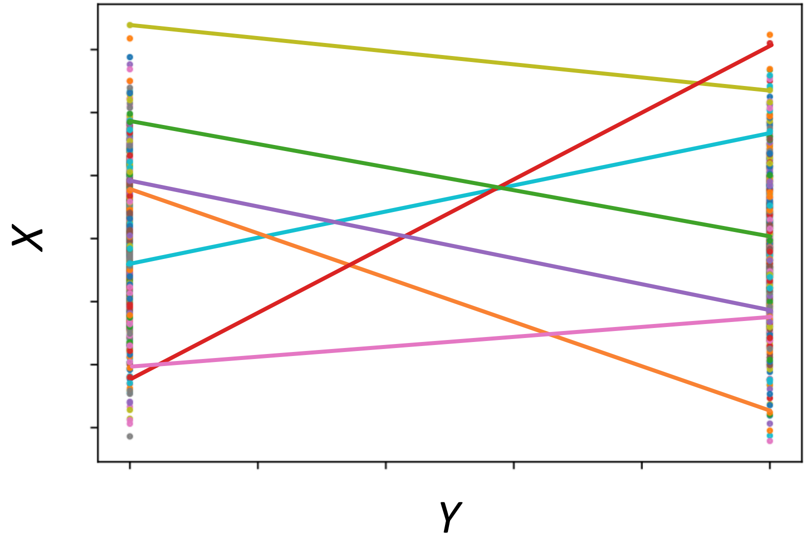
Suppose we have a random variable X_1 with distribution

$$P_{X_1}(x_1) \sim \mathcal{N}(\mu, \sigma^2)$$

We can generate samples from this distribution



and then project them onto a new axis



We can do this for another set of samples to obtain two independent Gaussian vectors

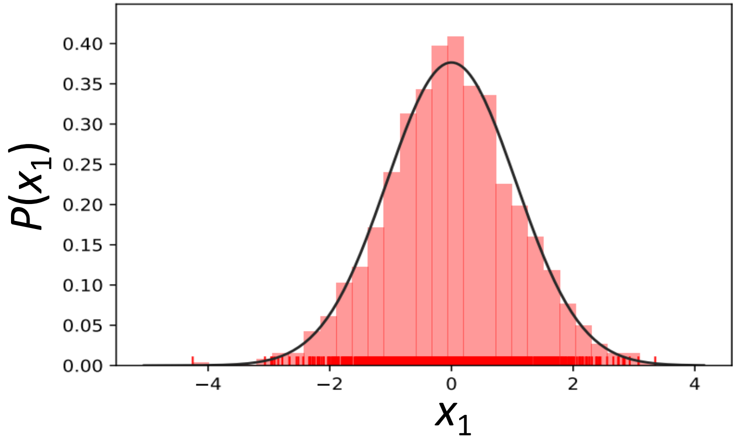
Then by connecting random points between the two vectors, we would get a set of linear functions

Some basic intuition underlying Gaussian Processes

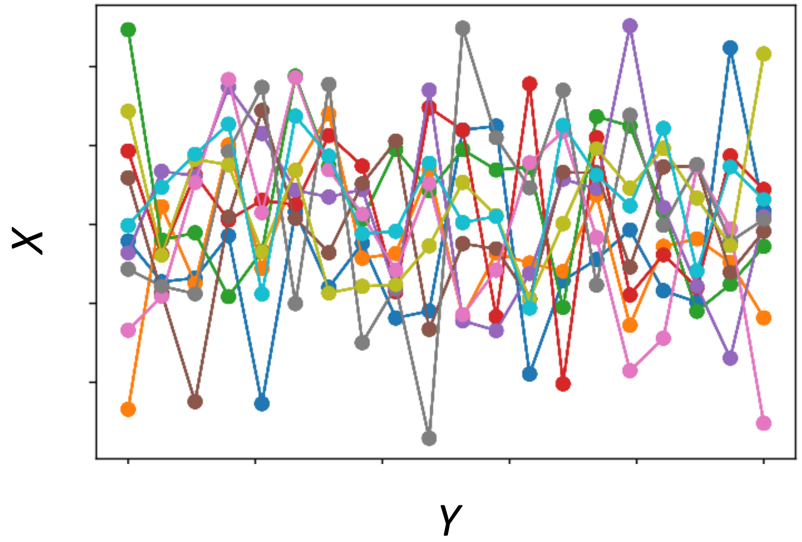
Suppose we have a random variable X_1 with distribution

$$P_{X_1}(x_1) \sim \mathcal{N}(\mu, \sigma^2)$$

We can generate samples from this distribution



and then project them onto a new axis



using more and more Gaussian vectors, we could represent more complex functions (showing only ten), but the resulting functions are inherently noisy!

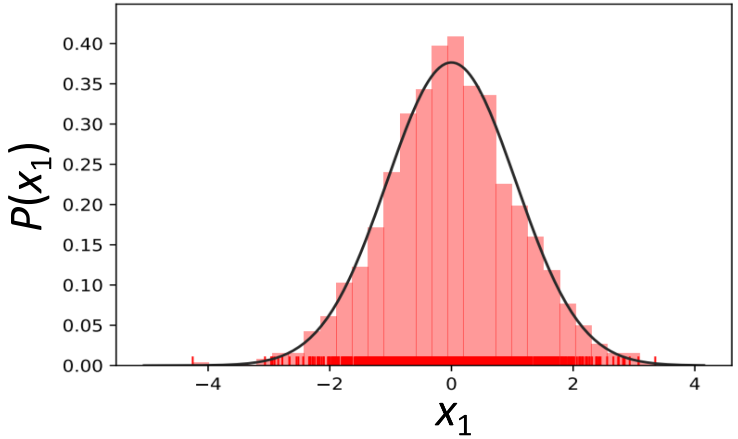
Why?

Some basic intuition underlying Gaussian Processes

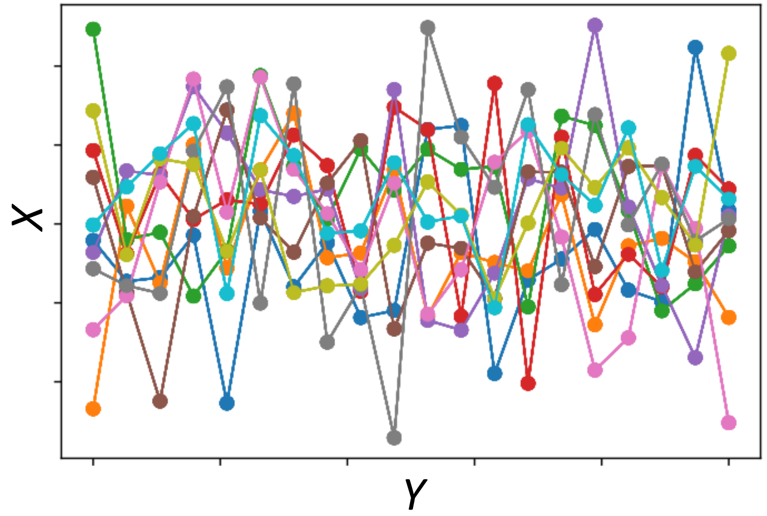
Suppose we have a random variable X_1 with distribution

$$P_{X_1}(x_1) \sim \mathcal{N}(\mu, \sigma^2)$$

We can generate samples from this distribution



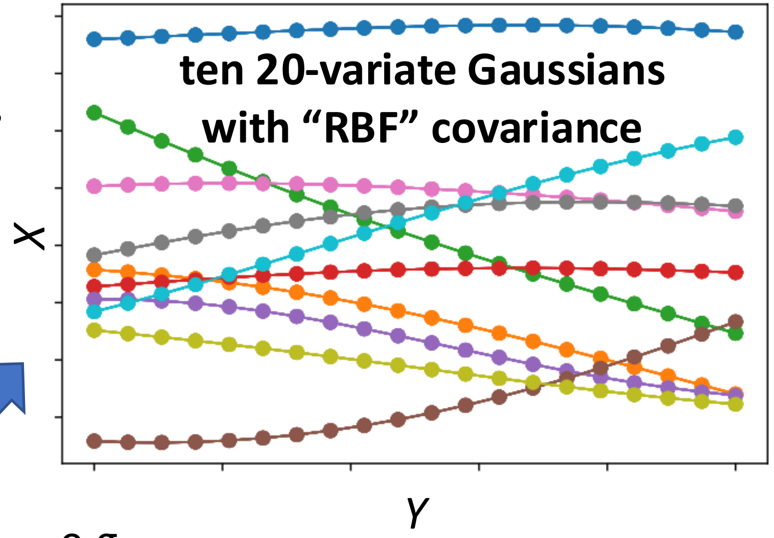
and then project them onto a new axis



using more and more Gaussian vectors, we could represent more complex functions (showing only ten), but the resulting functions are inherently noisy!

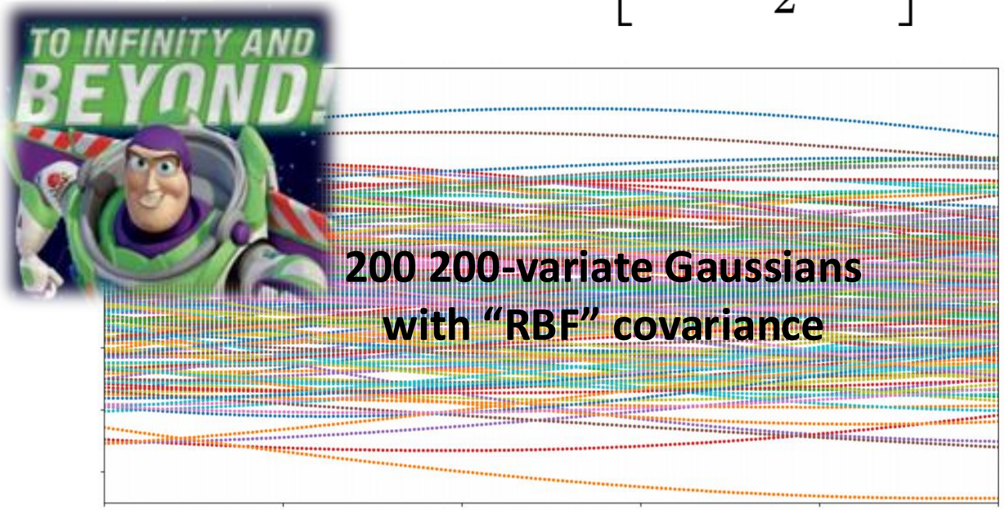
...because the Gaussian vectors are *independent*

using multivariate Gaussians with true covariance enables more smoothly varying functions

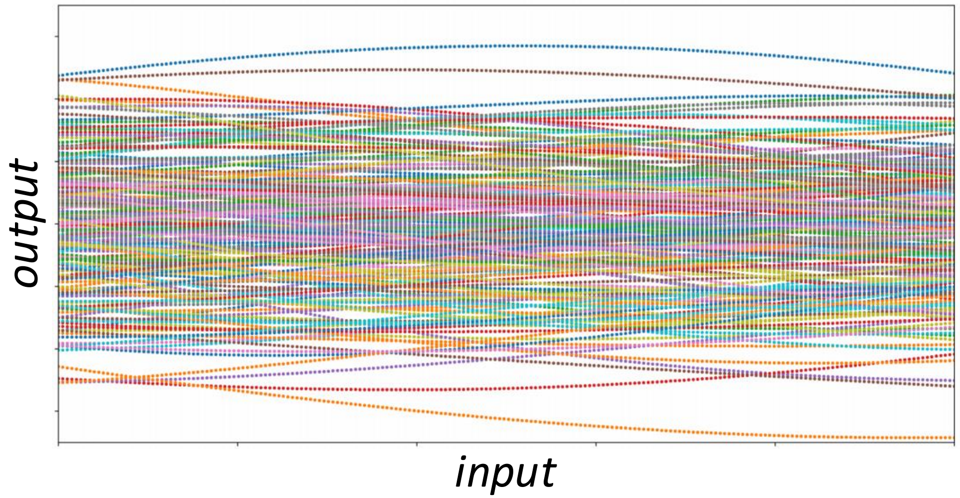


e.g.,

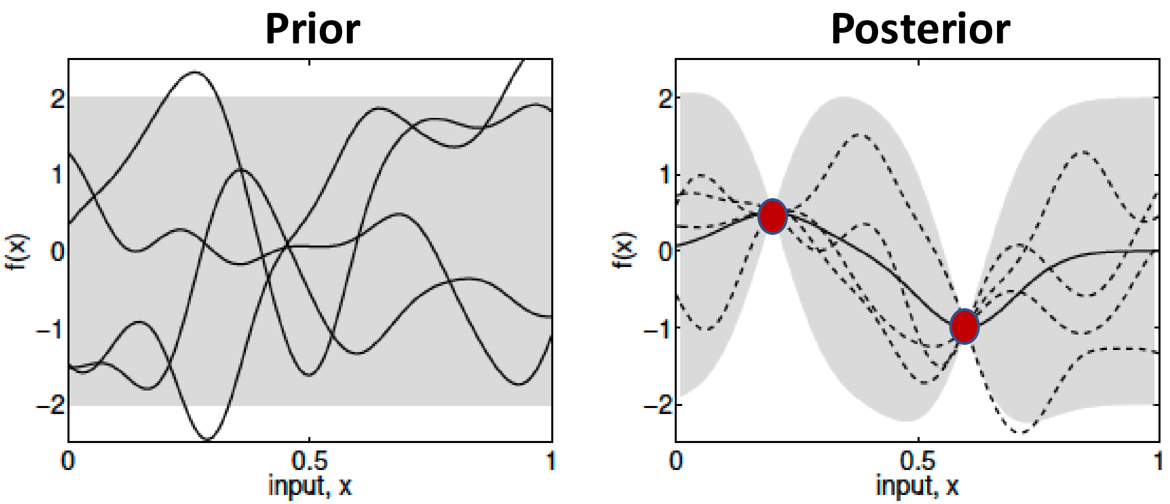
$$K(x_i, x_j) = \exp\left[-\frac{(x_i - x_j)^2}{2}\right]$$



Some basic intuition underlying Gaussian Processes



- If the dimensionality of the ***multivariate Gaussians*** (MVG) becomes infinite → a continuous function
- With an infinite number of such functions, we could make predictions at any point
- Functions as MVGs is our initial assumption (***prior***)
- The functions shown provide expected outputs as a function of inputs *without having observed any data*



- As data is collected, we can downselect from infinite functions to only the functions that describe the data/observations → ***posterior*** (kind of like wavefunctions in quantum mechanics, if that helps)
- As we add more data, the current posterior can be used as the prior to obtain a new posterior

Theoretical Development of GPs for Regression

A ***Gaussian process (GP)*** is a *probability distribution* over possible *functions*; any combination of these functions jointly Gaussian distributed

$$f(\boldsymbol{x}) \sim \mathcal{GP}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$$

this is a stochastic function, i.e., a collection of random variables

$$\mu(\boldsymbol{x}) = \mathbb{E}\left[f(\boldsymbol{x})\right]$$

its distribution is characterized by its mean and covariance function

$$k(\boldsymbol{x}, \boldsymbol{x}') = \mathrm{Cov}\left[f(\boldsymbol{x}), f(\boldsymbol{x}')\right] = \mathbb{E}\left[(f(\boldsymbol{x}) - \mu(\boldsymbol{x}))(f(\boldsymbol{x}') - \mu(\boldsymbol{x}'))\right]$$

A simple example $f(\boldsymbol{x}) = \phi(\boldsymbol{x})^T \boldsymbol{w}$

weighting coefficients BUT $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_p)$

some vector field, e.g., $\phi(\boldsymbol{x})^T = (\sum_i x_i, \sum_i x_i^2, \ldots, \sum_i x_i^m)$

for such a process, it is easy to show that $\mu(\boldsymbol{x}) = 0;\; k(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^T \boldsymbol{\Sigma}_p \phi(\boldsymbol{x}')$

that is just to say that f(x) and f(x’) are indeed jointly Gaussian with the above mean and covariance → any number of input points will be jointly Gaussian

Theoretical Development of GPs for Regression

Let's suppose we have a dataset now $\mathcal{D} := \{(\boldsymbol{x}_i, f_i = f(\boldsymbol{x}_i)) \text{ for } i = 1, \ldots, n\}$

$\boldsymbol{f}, \boldsymbol{x}$ **training** $\boldsymbol{f}_*, \boldsymbol{x}_*$ **test**

These two sets should be jointly distributed according to our Gaussian prior as

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k(\boldsymbol{x}, \boldsymbol{x}) & k(\boldsymbol{x}, \boldsymbol{x}_*) \\ k(\boldsymbol{x}_*, \boldsymbol{x}) & k(\boldsymbol{x}_*, \boldsymbol{x}_*) \end{bmatrix}\right)$$

prior distribution

Now, to get the posterior distribution, we must restrict our prior distribution to only the set of functions that "agree" with our observations/training data (i.e., we should make our prior distribution conditional on our observations)

Theoretical Development of GPs for Regression

Let's suppose we have a dataset now $\mathcal{D} := \{(\boldsymbol{x}_i, f_i = f(\boldsymbol{x}_i)) \text{ for } i = 1, \ldots, n\}$

$\boldsymbol{f}, \boldsymbol{x}$ ***training*** $\boldsymbol{f}_*, \boldsymbol{x}_*$ ***test***

These two sets should be jointly distributed according to our Gaussian prior as

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k(\boldsymbol{x}, \boldsymbol{x}) & k(\boldsymbol{x}, \boldsymbol{x}_*) \\ k(\boldsymbol{x}_*, \boldsymbol{x}) & k(\boldsymbol{x}_*, \boldsymbol{x}_*) \end{bmatrix}\right)$$

prior distribution

from math review

$$p(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{bmatrix}\right)$$

$$p(\boldsymbol{x} \mid \boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}_{x \mid y}, \boldsymbol{\Sigma}_{x \mid y})$$

$$\boldsymbol{\mu}_{x \mid y} = \boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1} (\boldsymbol{y} - \boldsymbol{\mu}_y)$$

$$\boldsymbol{\Sigma}_{x \mid y} = \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy} \boldsymbol{\Sigma}_{yy}^{-1} \boldsymbol{\Sigma}_{yx}$$

$$\boldsymbol{f}_* \mid \boldsymbol{f}, \boldsymbol{x}, \boldsymbol{x}_* \sim \mathcal{N}\big(k(\boldsymbol{x}_*, \boldsymbol{x}) k(\boldsymbol{x}, \boldsymbol{x})^{-1} \boldsymbol{f}, \; k(\boldsymbol{x}_*, \boldsymbol{x}_*) - k(\boldsymbol{x}_*, \boldsymbol{x}) k(\boldsymbol{x}, \boldsymbol{x})^{-1} k(\boldsymbol{x}, \boldsymbol{x}_*)\big)$$

easy to just sample function values from the posterior by computing the mean and covariance in the above!

Theoretical Development of GPs for Regression

Let's suppose we have a dataset now $\mathcal{D} := \{(\boldsymbol{x}_i, f_i = f(\boldsymbol{x}_i)) \text{ for } i = 1, \dots, n\}$

$$\boldsymbol{f}, \boldsymbol{x} \qquad\qquad \boldsymbol{f}_*, \boldsymbol{x}_*$$

training ***test***

In the prior example, our observations were treated as Noise-free; otherwise…

$$y = f(\boldsymbol{x}) + \varepsilon \qquad \varepsilon \sim \mathcal{N}(0, \sigma_n^2) \qquad \operatorname{cov}(\boldsymbol{y}, \boldsymbol{y}) = k(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 \boldsymbol{I}$$

This would slightly modify our joint distribution as

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f}_* \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} k(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 \boldsymbol{I} & k(\boldsymbol{x}, \boldsymbol{x}_*) \\ k(\boldsymbol{x}_*, \boldsymbol{x}) & k(\boldsymbol{x}_*, \boldsymbol{x}_*) \end{bmatrix}\right)$$

Then, the posterior is obtained by again conditioning on our observations

these are governing equations

$$\boldsymbol{f}_* | \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{x}_* \sim \mathcal{N}(\bar{\boldsymbol{f}}_*, \operatorname{cov}(\boldsymbol{f}_*, \boldsymbol{f}_*))$$

$$\bar{\boldsymbol{f}}_* := \mathbb{E}\left[\boldsymbol{f}_* | \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{x}_*\right] = k(\boldsymbol{x}_*, \boldsymbol{x}) \left[k(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 \boldsymbol{I}\right]^{-1} \boldsymbol{y}$$

$$\mathbb{V}\left[\boldsymbol{f}_*\right] = \operatorname{cov}(\boldsymbol{f}_*, \boldsymbol{f}_*) = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - k(\boldsymbol{x}_*, \boldsymbol{x}) \left[k(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 \boldsymbol{I}\right]^{-1} k(\boldsymbol{x}, \boldsymbol{x}_*)$$

Log Marginal Likelihood

"marginal likelihood" or "evidence"

$$p(\boldsymbol{y}|\boldsymbol{x}) = \int p(\boldsymbol{y}|\boldsymbol{f},\boldsymbol{x})p(\boldsymbol{f}|\boldsymbol{x})d\boldsymbol{f}$$

→ *how likely would we observe the data that we have given the data originates from **f**(**x**)*

Assuming a Gaussian process, we can show that

$$\log p(\boldsymbol{y}|\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{y}^T(\boldsymbol{K}+\sigma_n^2\boldsymbol{I})^{-1}\boldsymbol{y} - \frac{1}{2}\log|\boldsymbol{K}+\sigma_n^2\boldsymbol{I}| - \frac{n}{2}\log 2\pi$$

"log marginal likelihood""

- All the terms necessary to evaluate this are also used to obtain the mean function
- Useful metric for evaluating different models
- Hyperparameter optimization based on minimizing log likelihood

Basic Algorithm Outline

1. Define data/models: $\boldsymbol{x}, \boldsymbol{y}, k, \sigma_n^2, \boldsymbol{x}_*$

2. Perform Cholesky decomposition $\boldsymbol{L} := \text{cholesky}(k(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 \boldsymbol{I})$

3. Compute predictive mean function $\boldsymbol{\alpha} := \boldsymbol{L}^T \backslash (\boldsymbol{L} \backslash \boldsymbol{y})$

$$\bar{f}_* := k(\boldsymbol{x}, \boldsymbol{x}_*)^T \boldsymbol{\alpha}$$

4. Compute predictive variance

$$\boldsymbol{v} := \boldsymbol{L} \backslash k(\boldsymbol{x}, \boldsymbol{x}_*)$$

$$\mathbb{V}[f_*] = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{v}^T \boldsymbol{v}$$

5. Compute log marginal likelihood

$$\log p(\boldsymbol{y}|\boldsymbol{x}) = -\frac{1}{2} \boldsymbol{y}^T \boldsymbol{\alpha} - \sum_i \log L_{ii} - \frac{n}{2} \log 2\pi$$

6. Return mean, variance, log marginal likelihood

Basic Algorithm Outline

1. Define data/models: $\boldsymbol{x}, \boldsymbol{y}, k, \sigma_n^2, \boldsymbol{x}_*$

2. Perform Cholesky decomposition $\boldsymbol{L} := \mathrm{cholesky}(k(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 \boldsymbol{I})$

Go to Jupyter Notebook

5. Compute log marginal likelihood

$$\log p(\boldsymbol{y}|\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{y}^T\boldsymbol{\alpha} - \sum_i \log L_{ii} - \frac{n}{2}\log 2\pi$$

6. Return mean, variance, log marginal likelihood

Gaussian Distribution

Gaussian distribution is the most well-studied distribution in science, engineering, and ML

- *ubiquity (often stemming from* ***central limit theorem****)*
- *computationally convenient*

<u>univariate</u>

$$p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

<u>multivariate</u>

$$p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{m/2}} |\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right)$$

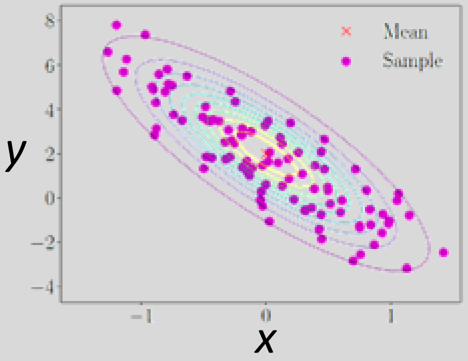
$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$ for short

standard normal distribution → ***0*** mean and $\boldsymbol{I}_n$ as the covariance matrix

<u>**Basic tenant of Gaussian distributions**</u>**:** *doing things with Gaussians results in Gaussians*

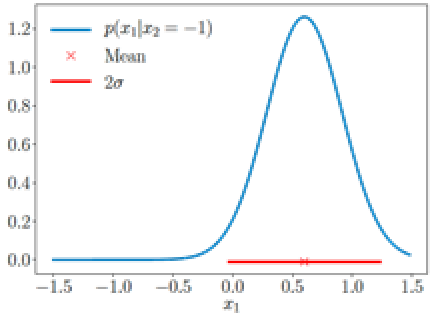
Gaussian Distribution

Consider the joint distribution over *X* and *Y*:



$$p(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy} \end{bmatrix} \right)$$

- *Conditionals of Gaussians are Gaussians*



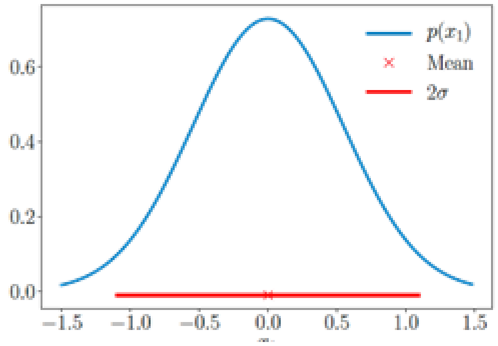
$$p(\boldsymbol{x} \,|\, \boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{\mu}_{x\,|\,y}, \, \boldsymbol{\Sigma}_{x\,|\,y}\right)$$

$$\boldsymbol{\mu}_{x\,|\,y} = \boldsymbol{\mu}_x + \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_y)$$

$$\boldsymbol{\Sigma}_{x\,|\,y} = \boldsymbol{\Sigma}_{xx} - \boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}$$

- *Marginals over Gaussians are Gaussians*

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{y})\mathrm{d}\boldsymbol{y} = \mathcal{N}\left(\boldsymbol{x} \,|\, \boldsymbol{\mu}_x, \, \boldsymbol{\Sigma}_{xx}\right)$$



- *The product of Gaussians is Gaussian*

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{a}, \boldsymbol{A})\mathcal{N}(\boldsymbol{x}|\boldsymbol{b}, \boldsymbol{B}) \rightarrow c\mathcal{N}(\boldsymbol{x}|\boldsymbol{c}, \boldsymbol{C})$$

$$\boldsymbol{C} = (\boldsymbol{A}^{-1} + \boldsymbol{B}^{-1})^{-1}$$

$$\boldsymbol{c} = \boldsymbol{C}(\boldsymbol{A}^{-1}\boldsymbol{a} + \boldsymbol{B}^{-1}\boldsymbol{b}) \quad c = \mathcal{N}(\boldsymbol{a}|\boldsymbol{b}, \boldsymbol{A} + \boldsymbol{B})$$

- *The sum over independent Gaussian variables is… Gaussian*

$$p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y})$$

$$p(\boldsymbol{x} + \boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu}_x + \boldsymbol{\mu}_y, \boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_y)$$

- *Any linear transformation of Gaussians is Gaussian* $p(a\boldsymbol{x} + b\boldsymbol{y}) = \mathcal{N}\left(a\boldsymbol{\mu}_x + b\boldsymbol{\mu}_y, \, a^2\boldsymbol{\Sigma}_x + b^2\boldsymbol{\Sigma}_y\right)$