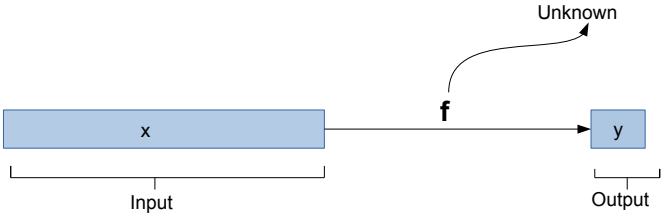
Deconstructing Gaussian Processes

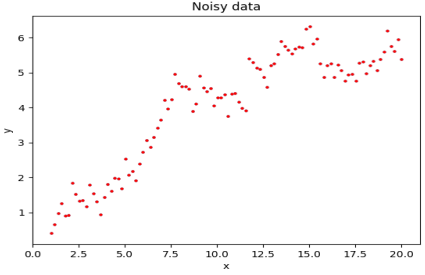
Vidhi Lalchand

June 11th, 2018

The Alan Turing Institute

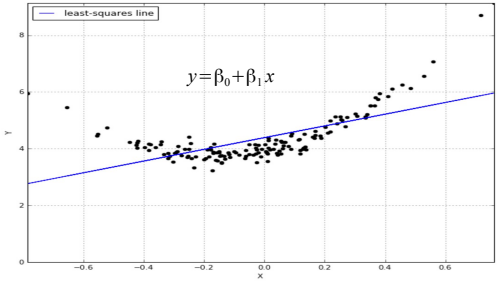
The Regression Problem

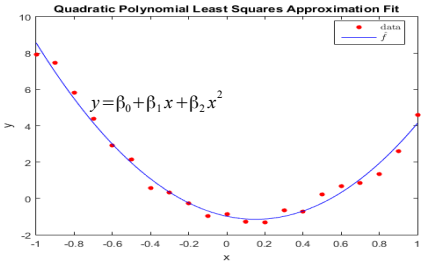


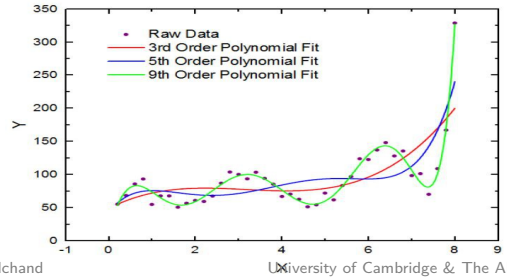


$$x \in \mathbb{R}^d$$
$$y \in \mathbb{R}$$
$$y = f(x) + \xi$$
$$\xi \in N(0, \sigma^2)$$

Traditional Parametric Approach







"Learning" applies to the parameters

$[\beta_0, \beta_1, \ldots, \beta_k]$

Gaussian processes (GPs) are a powerful **non-parametric** way to solve the regression problem.

Gaussian processes (GPs) are a powerful non-parametric way to solve the regression problem.

What is a non-parametric approach?

- We don't select the functional form of the model, ~~$y = \beta_0 + \beta_1 x$~~.
- It is *"letting the data speak for itself"* $\rightarrow$ the model becomes more complex as the size and the complexity of the data grow.
- The model structure (a.k.a functional form) and the parameters are both part of the *"learning"* in a non-parametric model.

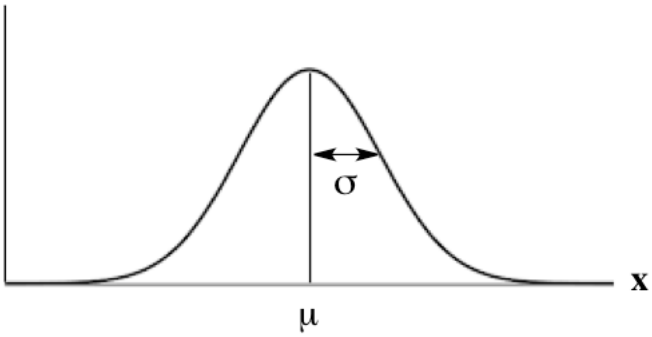
What is a non-parametric approach?

- We don't select the functional form of the model, ~~$y = \beta_0 + \beta_1 x$~~.
- It is *"letting the data speak for itself"* $\rightarrow$ the model becomes more complex as the size and the complexity of the data grow.
- The model structure (a.k.a functional form) and the parameters are both part of the *"learning"* in a non-parametric model.

What is a non-parametric approach?

- We don't select the functional form of the model, ~~$y = \beta_0 + \beta_1 x$~~.
- It is *"letting the data speak for itself"* $\rightarrow$ the model becomes more complex as the size and the complexity of the data grow.
- The model structure (a.k.a functional form) and the parameters are both part of the *"learning"* in a non-parametric model.

What is a Gaussian Process (GP)?

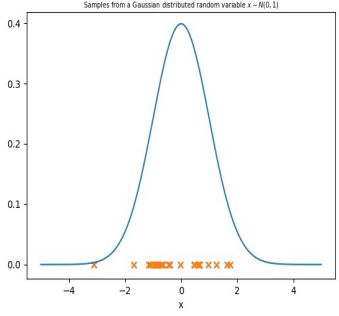


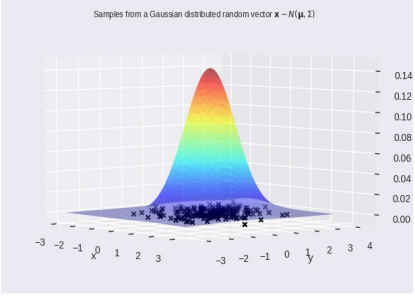
The gaussian (normal) distribution $x \sim \mathcal{N}(\mu, \sigma^2)$

What is a Gaussian Process (GP)?

A Gaussian process is a generalization of a Gaussian distribuiton

- A sample from a univariate Gaussian distribution $\rightarrow$ scalar value $x \sim \mathcal{N}(\mu, \sigma^2)$
- A sample from a bi-variate Gaussian distribution $\rightarrow$ vector, $(x, y) \sim \mathcal{N}(\mu, \Sigma)$
- A sample from a $k-$dimesional Gaussian distribution $\rightarrow$ vector of size k. Eg: $[x_1, x_2, ..., x_k]$.

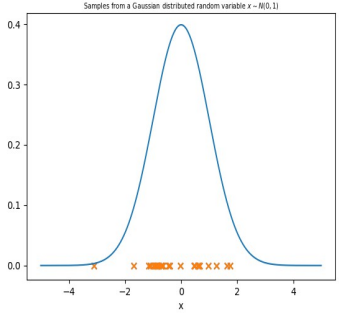


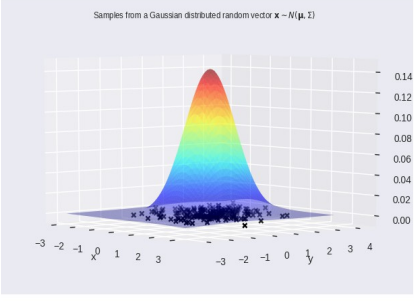


What is a Gaussian Process (GP)?

A Gaussian process is a generalization of a Gaussian distribuiton

- A sample from a univariate Gaussian distribution $\rightarrow$ scalar value $x \sim \mathcal{N}(\mu, \sigma^2)$
- A sample from a bi-variate Gaussian distribution $\rightarrow$ vector, $(x, y) \sim \mathcal{N}(\mu, \Sigma)$
- A sample from a $k-$dimesional Gaussian distribution $\rightarrow$ vector of size k. Eg: $[x_1, x_2, ..., x_k]$.

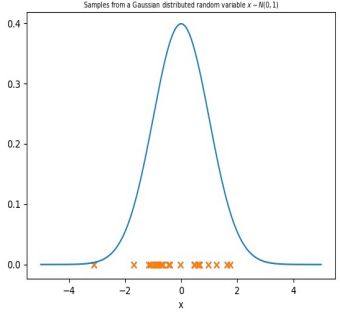


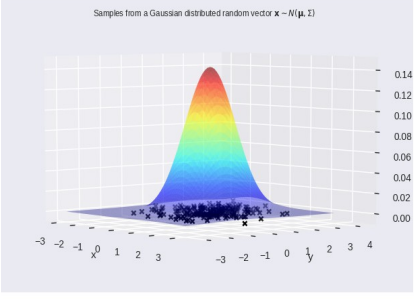


What is a Gaussian Process (GP)?

A Gaussian process is a generalization of a Gaussian distribuiton

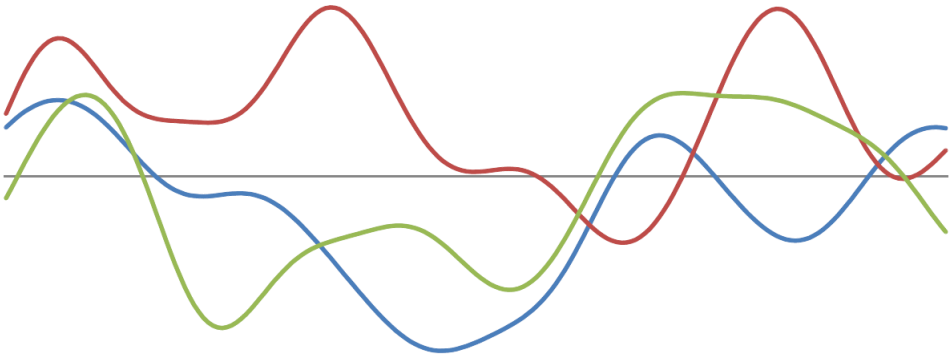
- A sample from a univariate Gaussian distribution $\rightarrow$ scalar value $x \sim \mathcal{N}(\mu, \sigma^2)$
- A sample from a bi-variate Gaussian distribution $\rightarrow$ vector, $(x, y) \sim \mathcal{N}(\mu, \Sigma)$
- A sample from a $k-$dimesional Gaussian distribution $\rightarrow$ vector of size k. Eg: $[x_1, x_2, ..., x_k]$.





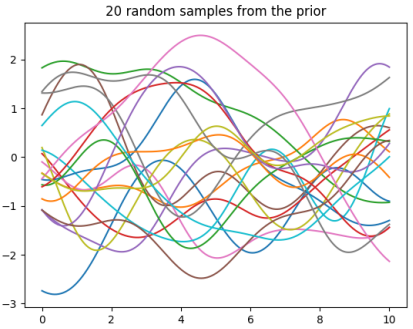
What is a Gaussian Process (GP)?

A sample from a GP is a function rather than a vector of finite length!.



What is a Gaussian Process (GP)?

Gaussian Distribution $\rightarrow$ random variables (scalars, vectors).
Gaussian Process $\rightarrow$ random functions (infinite length as defined on a continuous



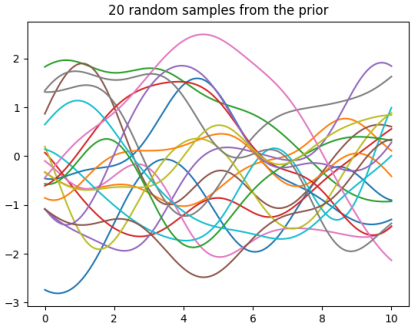
GP defines
a distribution over space of functions.

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

$m(x)$ - mean function.
$k(x, x')$ - kernel/covariance function

The kernel / covariance function

The **shape, smoothness and periodicity** of the functions is tied to the covariance or kernel function $k(x, x')$



$$k(x_i, x_j) = \exp -\frac{(x_i - x_j)^2}{2l^2} \tag{1}$$

Interpretation of functions $f(x)$ in GP world

When we think of a 'function' in a mathematical sense we immediately try to think of a parametric form. For example, $5x - 2, x^2, 3x^3 - x, e^x$.

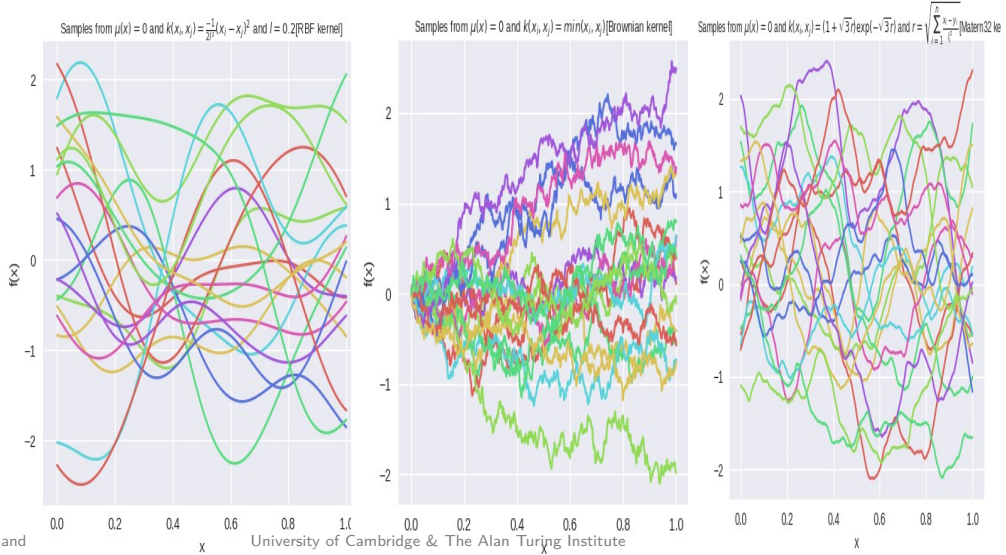
But in GP world there is a fundamental shift in thinking about functions. **We completely abandon the parametric form viewpoint**.

Instead GPs represent functions $f(x)$ obliquely (but rigorously) by selecting the covariance function $k(x, x')$.

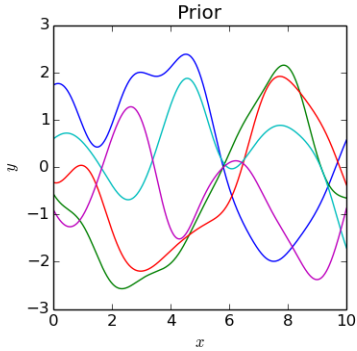
$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & \dots & \dots & k(x_n, x_n) \end{bmatrix}_{n \times n} \tag{2}$$

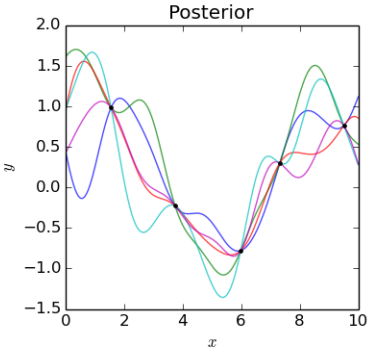
Visualizing the space defined by a GP?

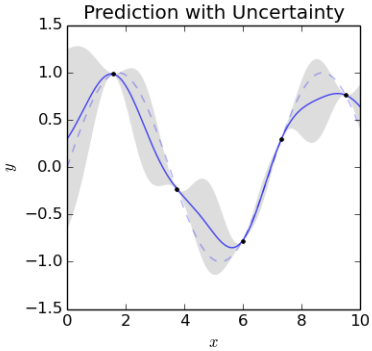
The covariance function controls how the functions in our "space" of functions look.



Gaussian Process $\rightarrow$ Regression







$p(f) \sim \mathcal{N}(0, K(X, X))$ [We get some new inputs X_*] $p(f^*|f) \sim \mathcal{N}(\mu_*, \Sigma_*)$

The Math

The central question is how do we mathematically get from the **prior** $\rightarrow$ **posterior.** We have a distribution over functions f, we define as:

$$f \sim \mathcal{N}(0, K(X, X))$$

Given some new inputs X_* we want to find f_*, probabilistically we want $p(f_*|f)$.

Step 1: Joint Gaussian in function space

$$\begin{aligned}(f, f_*) &\sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \\ &\sim \mathcal{N}\left(0, \begin{bmatrix} K & K_*^T \\ K_* & K_{**} \end{bmatrix}\right)\end{aligned}$$

Step 2: Deriving Conditional from Joint

$$P(f_*|f) \sim N(K_* K^{-1} f, K_{**} - K_* K^{-1} K_*^T) \tag{3}$$

In Summary

Key benefits

- They provide an intrinsic measure of uncertainty in predictions.
- The use of kernel functions provides additional flexibility.
- Mathematically tractable $\rightarrow$ derive posterior distribution in closed form.

Key Drawbacks

- Expensive to train as the complexity scales in $O(N^3)$.
- No rigorous way of choosing the kernel function which is the key ingredient in the inference.

Trivia: The idea of using a gaussian process for regression originally appeared in geostatistics, and was called *kriging*.

Real World Applications

- Particle Physics $\rightarrow$ to model densities of mass of a particle.
- Finance, econometrics $\rightarrow$ as a tool for timeseries forecasting.
- They are used in oceanography, metallurgy, terrain modelling and ....

Real World Applications

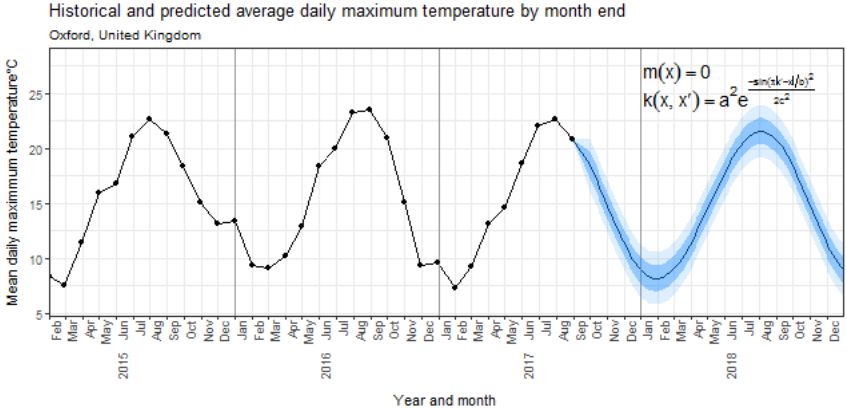
- Particle Physics $\rightarrow$ to model densities of mass of a particle.
- Finance, econometrics $\rightarrow$ as a tool for timeseries forecasting.
- They are used in oceanography, metallurgy, terrain modelling and ....

Real World Applications

- Particle Physics $\rightarrow$ to model densities of mass of a particle.
- Finance, econometrics $\rightarrow$ as a tool for timeseries forecasting.
- They are used in oceanography, metallurgy, terrain modelling and ....

Real World Applications

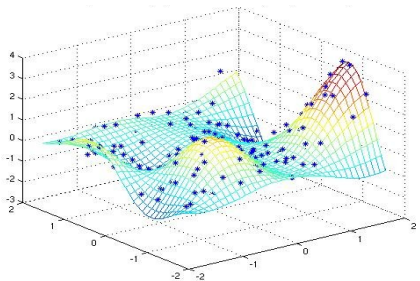
- Particle Physics $\rightarrow$ to model densities of mass of a particle.
- Finance, econometrics $\rightarrow$ as a tool for timeseries forecasting.
- They are used in oceanography, metallurgy, terrain modelling and ....



Challenges

- Selecting the right kernel function for the data.
- Making them scale to bigger datasets using sparse approaches.

Example of GP Regression in higher dimensions.



Appendix: Deriving Conditional from Joint

Lemma: The property of conditional distributions of Gaussian distributed random variables states that:

$$A_1 \sim N(\mu_1, \Sigma_1) \tag{4}$$

$$A_2 \sim N(\mu_2, \Sigma_2) \tag{5}$$

$$(A_1, A_2) \sim N\left((\mu_1, \mu_2), \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right) \tag{6}$$

$$A_1|A_2 \sim N(\mu_3, \Sigma_3) \tag{7}$$

where,

$$\mu_3 = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(A_2 - \mu_2) \tag{8}$$

$$\Sigma_3 = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \tag{9}$$

(Proof for conditioning is easily available on the internet, $A_2|A_1$ can be evaluated using symmetry)