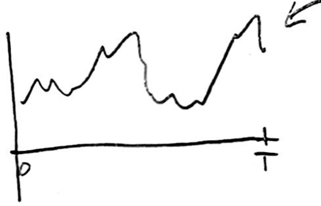
Computational Statistics

Oct 13, 2021

Gaussian Processes (G.P.s)

A G.P. or any other stochastic process is just the generalization of a random variable to a random function.

If f is a stoch. proc., then it is usally indexed by another variable, t:



one realization of f on the interval $[0,T]$

$\Rightarrow$ $f(t)$ is a scalar quantity, it is a random variable

f is a random function

---

obviously, we can then talk about $E(f) = \mu$ ← mean function

$$\mathrm{Cov}(f(t), f(s)) = k(s,t)$$

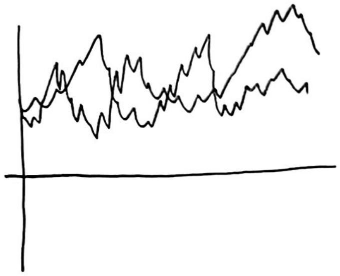
↑ covariance kernel function

The two important quantities are

- the distribution of $f(t)$
- the joint distribution of $f(t)$, $f(s)$

Most "common" example of a stochastic process?

Brownian Motion:



two realizations of B

B is defined via the following construction:

- $B(0)=0$
- B is "almost surely continuous"
- If $[a,b]\cap[c,d]=\phi$, then $B(b)-B(a)$ is independent of $B(d)-B(c)$
- $B(t)-B(s)\sim N(0,t-s)$ for $0\le s\le t$

$\Rightarrow$ $B(t)\sim N(0,t)$

Another way to think about Brownian Motion (or the Wiener Process) is as an integral:

Let W be another stochastic process called white noise

$W(t)\sim N(0,1)$ for all t.

(i.e. not continuous anywhere)

then $\underline{B(t)=\int_0^t W(\tau)\,d\tau}$ ← adding up many IID increments of Normal r.v.'s

Both B and W are examples of Gaussian Processes.

The definition of a general G.P. is the following:

f is a Gaussian Process if and only if $f(t_1), \dots, f(t_n)$, where $t_1, \dots, t_n$ is any collection of points, is a multivariate Normal [illegible] random vector.

(Best reference: Gaussian Processes for Machine Learning by Rasmussen & Williams)

I.e: for any collection $t_1, \dots, t_n$, $f(t_1) \dots f(t_n)$ follows a multivariate normal distribution.

Ex: Brownian Motion

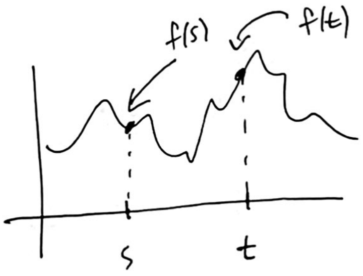
$$B(t) \sim N(0, t)$$

$$\mathrm{cov}(B(t), B(s)) = ?$$

A special case that we will restrict ourselves to is that where the covariance structure is determined explicitly by a covariance kernel / function, $k = k(s,t)$

$$\Rightarrow \quad \mathrm{cov}\big(f(t), f(s)\big) = k(s,t)$$

Graphically:



mean function

$$\mathbb{E}(f(s)) = m(s)$$

$$\mathrm{Var}(f(s)) = k(s,s)$$

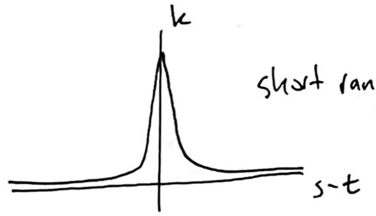
$$\Rightarrow f(s) \sim N(m(s), k(s,s))$$

$$k(s,t) = \mathbb{E}\Big(\big(f(s) - m(s)\big)\big(f(t) - m(t)\big)\Big)$$

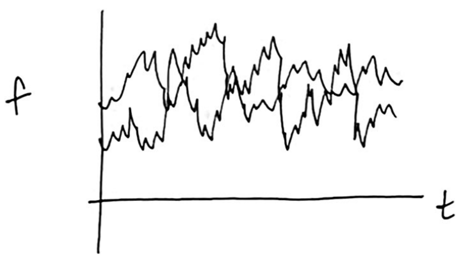
What dos the covariance function control?

A: The <u>smoothness</u> of the G.P.

<u>Ex1</u> $k(s,t) = e^{-(s-t)^2/.0001}$

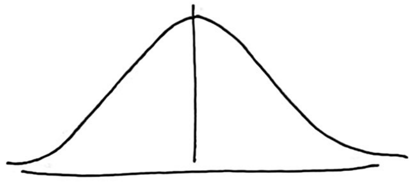


short range correlations

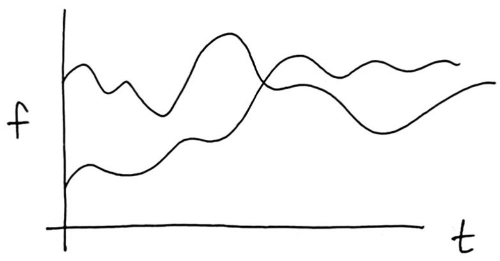


$\Rightarrow$ f(t), f(s) for large t-s, are basically independant

<u>Ex2</u> $k(s,t) = e^{-(s-t)^2/10000}$



long-range correlations



$\Rightarrow$ f(t) and f(s) are highly dependent for large t-s

---

<u>Recall</u>: Multivariate Normal Random Vectors

$\vec{X} \in \mathbb{R}^k \sim N(\vec{m}, C)$ if its density is

$$f(x_1, \dots, x_k) = f(\vec{x}) = \frac{1}{(2\pi)^{k/2}} \frac{1}{\sqrt{\det C}} e^{-\frac{1}{2}(\vec{x}-\vec{m})^T C^{-1}(\vec{x}-\vec{m})}$$

$\vec{m} \in \mathbb{R}^k$ $\qquad \mathbb{E}(\vec{X}) = \vec{m}$

$C \in \mathbb{R}^{k \times k}$ $\qquad Var(\vec{X}) = C$

In particular, C must be symmetric positive definite

(if only semi-definite, then this means that at least one X_j has zero variance)

What does this imply about the function k?

- k must be symmetric: $k(s,t) = k(t,s)$

- k must be a positive kernel:

$$\iint \varphi(x)\, k(x,y)\, \varphi(y)\, dx\,dy > 0$$

(as an inner product $(\varphi, K\varphi) > 0$,

when $K\varphi(x) = \int k(x,y)\varphi(y)\,dy$

(compare with matrix version.)

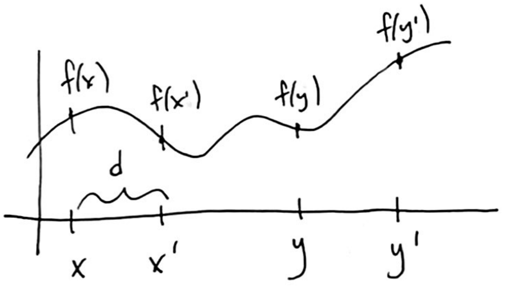
For a particular set of points $x_1 \dots x_n$, call $\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$,

the matrix $K(\vec{x}, \vec{x})$ with entries $K_{ij} = k(x_i, x_j)$ is

called the <u>Gram Matrix</u>.

Other useful classifications of Gaussian Processes:

① Translation invariance: $k(x,x') = k(x-x')$ (also called a stationary process)



$$\mathrm{cov}(f(x), f(x')) = \mathrm{cov}(f(y), f(y'))$$

Useful when modeling time dependent signals with time dependent correlations.

(2) Isotropic : $k(x,x') = k(|x-x'|)$

Ex: $k(x,x') = e^{-|x-x'|}$

$k(x,x') = a\, e^{-|x-x'|^2/b}$

Positive definite translation invariant covariance kernels have a nice one-to-one correspondence with "spectral densities":

Thm (Bochner's Thm) A stationary covariance kernel

$k = k(x-x') = k(\tau)$ can be written as

$$k(\tau) = \underbrace{\int \overbrace{S(s)}^{\text{spectral density, or power spectrum of } k.} e^{2\pi i s \tau}\, ds}_{\text{Inverse Fourier Transform}}$$

where S is a positive function i.e. $S(s) > 0$, if k is a positive definite kernel.

Therefore by Fourier Inversion,

$$S(s) = \underbrace{\int k(\tau)\, e^{-2\pi i s \tau}\, d\tau}_{\text{Fourier Transform.}}$$

Ex: $k(0) = Var(f(x), f(x))$

$= \int S(s)\, e^{2\pi i s \cdot 0}\, ds = \int S(s)\, ds \Rightarrow S$ is integrable $\in L^1$

Gaussian Process Regression

Noise Free :

Ground truth : $y = f(x)$, a deterministic function

Model : $$y = \underset{\hookrightarrow GP(0,k)}{f(x)}$$

Training data : $(x_1, y_1) \dots (x_n, y_n)$

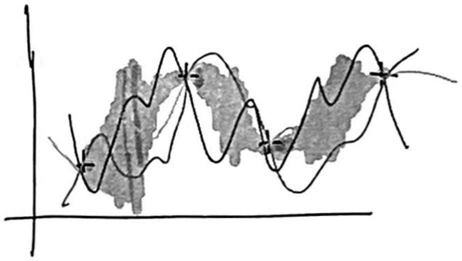
Predictions : $(x_1^*, y_1^*) \dots (x_m^*, y_m^*)$

Joint distribution of $\vec{y}, \vec{y}_*$ ~ multivariate normal distribution

$$\begin{pmatrix} \vec{y} \\ \vec{y}_* \end{pmatrix} \sim N\left(\vec{0}, \begin{pmatrix} K(\vec{x},\vec{x}) & K(\vec{x},\vec{x}_*) \\ K(\vec{x}_*,\vec{x}) & K(\vec{x}_*,\vec{x}_*) \end{pmatrix} \right)$$

Gram Matrics

We want to compute the distribution of $\vec{y}_* | \vec{x}_*, \vec{x}, \vec{y}$ ← posterior, conditioned on the observed data.



+ : observed data points.

In this setup, draws from the posterior have to pass through the observed data.

It can be shown that the conditional distribution is

$$\vec{y}_* | \vec{x}_*, \vec{x}, \vec{y} \sim N\left(K(\vec{x}_*,\vec{x}) K(\vec{x},\vec{x})^{-1} \vec{y} ,\ K(\vec{x}_*,\vec{x}_*) - K(\vec{x}_*,\vec{x}) K(\vec{x},\vec{x})^{-1} K(\vec{x},\vec{x}_*) \right)$$

Exercise To prove this to yourself.

An analogous calculation:

Solve
$$\begin{pmatrix} A & C \\ C^T & B \end{pmatrix}\begin{pmatrix} \vec{x} \\ \vec{y} \end{pmatrix} = \begin{pmatrix} \vec{a} \\ \vec{b} \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} A - CB^{-1}C^T & 0 \\ C^T & B \end{pmatrix}\begin{pmatrix} \vec{x} \\ \vec{y} \end{pmatrix} = \begin{pmatrix} \vec{a} - CB^{-1}\vec{b} \\ \vec{b} \end{pmatrix}$$

With indpendent noise

dterministic function

Observe $y = f(x) + \epsilon$

$\sim N(0, \sigma^2)$

Modl: $y = f(x) + \epsilon$

Gaussian process $GP(0,k)$

Normal random variable, also known as white noise, also a Gaussian process.

$\Rightarrow$ y is a Gaussian Proess with

$$\mathrm{cov}(y_i, y_j) = k(x_i, x_j) + \sigma^2 \delta_{ij}$$

Kronecker dlta function,

$\delta_{ij} = 1$ if $i=j$

0 otherwise.

$$\mathrm{Cov}(\vec{y}) = K(\vec{x}, \vec{x}) + \sigma^2 I$$

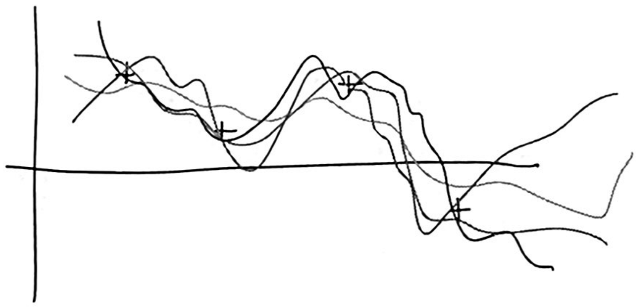
So the joint distribution of the training data $\vec{x}, \vec{y}$ with the predicted $\vec{x}_*, \vec{y}_*$ is

$$\begin{pmatrix} \vec{y} \\ \vec{y}_* \end{pmatrix} \sim N\left(\vec{0}, \begin{pmatrix} K(\vec{x},\vec{x}) + \sigma^2 I & K(\vec{x},\vec{x}_*) \\ K(\vec{x}_*,\vec{x}) & K(\vec{x}_*,\vec{x}_*) \end{pmatrix}\right)$$

Same calculation to compute posterior distribution:

$$\vec{y}_* \mid \vec{x}_*, \vec{x}, \vec{y} \sim N\Big(K(\vec{x}_*, \vec{x})\left(K(\vec{x},\vec{x}) + \sigma^2 I\right)^{-1} \vec{y} \,,$$

$$K(\vec{x}_*, \vec{x}_*) - K(\vec{x}_*, \vec{x})\left(K(\vec{x},\vec{x}) + \sigma^2 I\right)^{-1} K(\vec{x}, \vec{x}_*)\Big)$$



+ observations

$y = f(x) + \epsilon$

Draws from posterior do not pass through data.

One last comment:

The Bayesian "predictor" or "estimator" at the points $\vec{x}_*$ is:

$$\hat{y}_* = K(\vec{x}_*, \vec{x})\left(K(\vec{x},\vec{x}) + \sigma^2 I\right)^{-1} \vec{y}$$

$$= \sum \alpha_i K(\vec{x}_*, x_i) \quad \leftarrow \text{a linear combination of covariance kernels.}$$

$\uparrow$

$\alpha_i =$ i^{th} entry of $\left(K(\vec{x},\vec{x}) + \sigma^2 I\right)^{-1} \vec{y}$

Computational Considerations

From the previous formulae, it is clear that dense linear algebra is needed to model with GPs:

- evaluating density $\alpha \frac{1}{\sqrt{\det C}} e^{-\vec{x}^T C^{-1} \vec{x}}$

- mean in regression: $K(\vec{x}_*, \vec{x}) \underbrace{(\sigma^2 I + K(\vec{x}, \vec{x}))^{-1}} \vec{y}$

- cov in regression: $\underbrace{\qquad}$

These are $O(N^3)$ operations when applied to N "training" points.

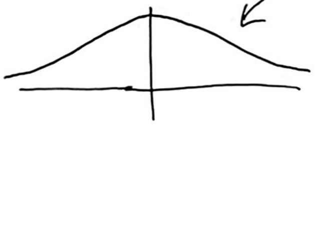
Methods to Circumvent

① low-rank approximations

Often the matrix $K(\vec{x}, \vec{x})$ is of approximate numerical low rank: i.e. there exist $U, V \in \mathbb{R}^{N \times r}$ s.t.

$$\| K - UV^T \|_2 < \varepsilon \quad \text{when } \varepsilon \text{ is chosen to be small}$$

Qualitatively, this occurs when the covariance function is very flat → "every row looks the same"



Gaussian with large variance

(a) If $K \approx UV^T$, then

$(\sigma^2 I + UV^T)^{-1}$ can be computed in $O(Nr^2)$ time using the Woodbury matrix identity.

(b) $\det(I + UV^T)$ can be computed in ~~[illegible]~~ $O(Nr^2 + r^3)$ time using the Sylvester (or Weinstein-Aronszajn) matrix identity:

$$\det(I + UV^T) = \det(I + V^T U)$$

To compute $\det A$, best to compute eigenvalues when A is SPD.

To obtain the low-rank factorization $K \approx UV^T$, ~~one~~ options include:

- randomized compression
  - Bad Method: randomly pick rows/columns (cross-approximation)
  - Good Method: Compute random "projections" $B = K\Omega_1$ and $A = K^T\Omega_2$ to find column/row space bases (More on this later in the semester)
- Return to the continuous problem:

  Write $k(x,y) \approx \sum_{n=1}^{r} \varphi_n(x)\varphi_n(y)\lambda_n \leftarrow$ continuous version of SVD or eigen-decomposition.