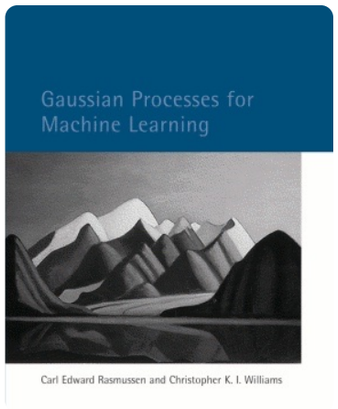
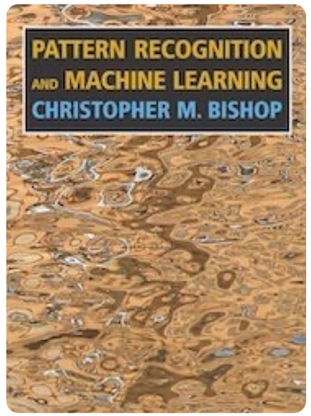
Gaussian Processes for Regression

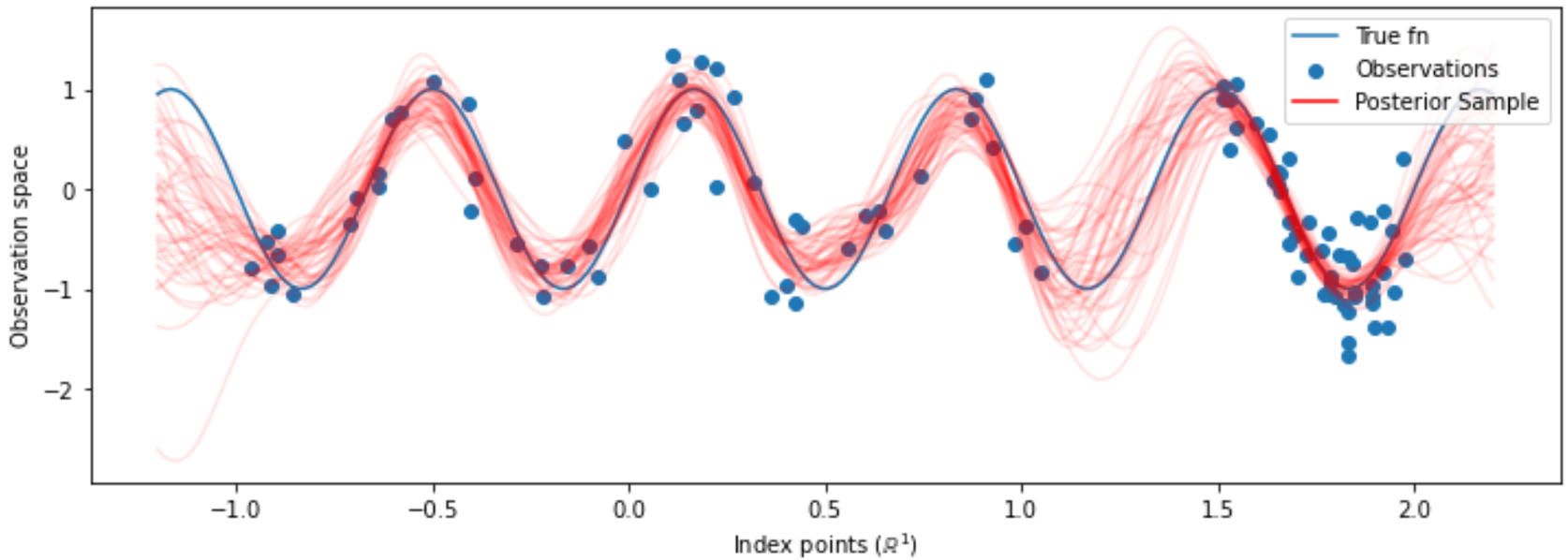
References

- Gaussian processes for ML (MIT Press 2006) [RW]
- PRML section 6.4
- Tensorflow probability





Gaussian Processes for Regression



Quantify the uncertainty (or uncertainty quantify)

Gaussian Process to describe a distribution over functions.

Def A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution [RW]

$\mathcal{X}$: any set (index)

A random process $X(t)$ is a GP indexed by $\mathcal{X}$ if for any $\{t_1, .. t_n\} \subset \mathcal{X}$, a random vector formed by $X(t_1) .. X(t_n)$ is jointly Gaussian (or $(X(t_1) .. X(t_n))^T$ is a multivariate Gaussian)

Specification of GP

GP can be specified by

- mean function $\mu: \mathcal{X} \to \mathbb{R}$ s.t $\mu(t) = \mathbb{E}[X(t)]$
- covariance function $k: \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ s.t

$$k(t,s) = \text{cov}[X(t), X(s)] \quad \text{a.k.a kernel function}$$

The kernel function is required only to be symmetric and positive definite.

$$X(t) \sim GP(\mu(t), k(t,s)) \qquad t, s \in \mathcal{X}$$

Example D- dim multivariate Gaussian distribution is a GP indexed by $\{1, 2, \ldots D\}$

See PRML section 2.3.2

Example $X(t) := t\,x$ where $x = \mathcal{N}(x \mid 0, 1)$ and $t \in \mathbb{R}$

Then $X(t)$ is a GP indexed by $\mathbb{R}$

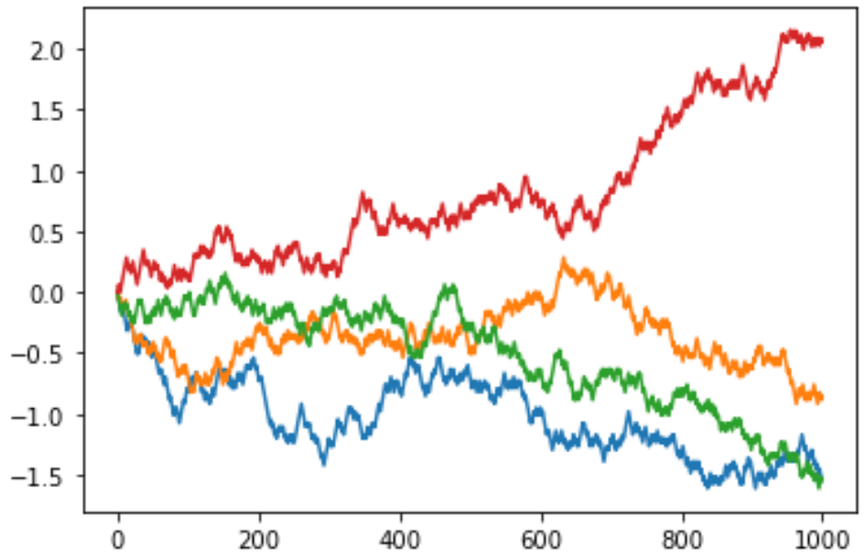
$t_1, \ldots t_n$, $X(t_1), \ldots X(t_n)$

$$\mu(t) = \mathbb{E}[X(t)] = t\,\mathbb{E}[x] = 0$$

$$k(t,s) = \text{cov}[X(t), X(s)] = \mathbb{E}[tx \cdot sx] = ts\,\mathbb{E}[x^2] = ts$$

Def A stochastic (random) process $\{X(t), t \geq 0\}$ is said to be a Brownian motion (Wiener process) if

- $X(0) = 0$ $\quad (X_t(\omega), \omega \in \Omega)$
- $\{X(t), t \geq 0\}$ has stationary and independent increments
- For $t > 0$, $X(t)$ is normally distributed with mean 0 and variance t.
- With probability 1, $t \longmapsto X(t)$ is continuous.



For $t_1 < t_2 < \cdots < t_n$, joint density $X(t_1), \ldots, X(t_n)$

$X(t_1)$, $X(t_2) - X(t_1)$, $\ldots$, $X(t_n) - X(t_{n-1})$ are independent and

$X(t_k) - X(t_{k-1}) \sim N(0, t_k - t_{k-1})$

Eg. Brownian motion $\{B(t), t \geq 0\}$ is a GP indexed by $\mathbb{R}^+$

Each of $B(t_1) \ldots B(t_n)$ can be expressed as a linear combination of independent normal r.v $B(t_1)$, $B(t_2) - B(t_1) \ldots, B(t_n) - B(t_{n-1})$

Gaussian Processes for Regression (GPR)

Notations

$\mathcal{X}$: index set of GP

Z : collection of random variables indexed by $\mathcal{X}$

$$Z(x) \sim GP(\mu(x), K(x, x')) \qquad x, x' \in \mathcal{X}$$

For any finite set $x_1, x_2, \ldots x_n \in \mathcal{X}$,

$$(z(x_1), \ldots z(x_n))^T = \mathcal{N}(\mu_n, K)$$

where $\mu_n = (\mu(x_1), \mu(x_2), \ldots \mu(x_n))^T$, $K_{ij} = \kappa(x_i, x_j)$

$$= \text{cov}[z(x_i), z(x_j)]$$

index set of GP $\approx$ domain of function

GPR: nonparametric Bayesian regression method using the properties of GP

Gaussian Processes for Regression

Goal: estimate an unknown function f with noisy observations $\{t_1, \dots t_N\}$ of f at points $\{x_1, \dots x_N\}$ with error bars (U.Q.)

Idea: GP as defining a distribution over functions and inference taking place directly in the space of functions.

We dispense with the parametric model and instead define a prior distribution over functions directly.

Linear regression revisited

Training data set $D = (\mathbf{x}_n, t_n)_{n=1 \dots N}$

$\mathbf{x}$: D-dim input vector, t : target value

$y(\mathbf{x})$: prediction value at $\mathbf{x}$

predictive distribution

Goal : find $p(t^* \mid \mathbf{x}^*, \mathbf{X}, \mathbf{t})$ at some new input $\mathbf{x}^*$

Fix basis functions $\phi_1, \dots \phi_{M-1}$ or feature map $\Phi(\mathbf{x})$

Linear model

$$y(\mathbf{x}, \mathbf{w}) := \mathbf{w}^T \Phi(\mathbf{x})$$ weight vector $\mathbf{w} \in \mathbb{R}^M$

Model assumption

$$t \sim \mathcal{N}(t \mid y(\mathbf{x}, \mathbf{w}), \beta^{-1}) \qquad \text{precision} \quad \beta > 0$$

prior over $\mathbf{w}$

$$p(\mathbf{w} \mid \alpha) = \mathcal{N}(\mathbf{w} \mid \mathbf{0}, \alpha^{-1} \mathbf{I})$$

posterior over $\mathbf{w}$

$$p(\mathbf{w} \mid \mathbf{t}) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}_N, \mathbf{S}_N)$$

$$\mathbf{m}_N = \beta \mathbf{S}_N \boldsymbol{\Phi}^T \mathbf{t}$$

$$\mathbf{S}_N^{-1} = \alpha \mathbf{I} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}$$

Example

Let $w \sim \mathcal{N}(w \mid 0, \alpha^{-1} I)$ be as a prior.

For $n = 1, 2, \ldots N$, let $y_n := y(x_n, w) = w^T \phi(x_n)$. Then y_n can be seen a 1-dim random variable (1-dim Gaussian)

Let $y := (y_1 \ldots y_N)^T$. Then

$$y = \underset{N \times M}{\Phi}\, w$$

$M \times 1$

is the N-dim random vector where $\Phi_{nk} = \phi_k(x_n)$.

Φ is called design matrix so that

$$\Phi = \begin{pmatrix} \phi_0(x_1) & \phi_1(x_1) & \cdots & \phi_{M-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & & \phi_{M-1}(x_2) \\ \vdots & \vdots & & \\ \phi_0(x_N) & & & \phi_{M-1}(x_N) \end{pmatrix} = \begin{pmatrix} \Phi(x_1)^T \\ \Phi(x_2)^T \\ \vdots \\ \Phi(x_N)^T \end{pmatrix}$$

Since y is a linear combination of w (Gaussian dist.), y is N-dim Gaussian. (y is determined by the first and the second moments)

$$\mathbb{E}[y] = \mathbb{E}[\Phi w] = \Phi \mathbb{E}[w] = 0$$

$$\mathrm{Cov}[y] = \mathbb{E}[y y^T] = \Phi \underbrace{\mathbb{E}[w w^T]}_{\alpha^{-1} I} \Phi^T =: K$$

where $\mathbf{K}$ is the gram matrix

$$K_{nm} = K(\mathbf{x}_n, \mathbf{x}_m) = \frac{1}{\alpha} \boldsymbol{\Phi}(\mathbf{x}_n)^T \boldsymbol{\Phi}(\mathbf{x}_m)$$

and $K(\mathbf{x}, \mathbf{x}')$ is the kernel function.

In general, GP is defined as a probability distribution over functions $y(x)$ s.t the set of values of $y(x)$ evaluated at an arbitrary set $x_1 \ldots x_N$ jointly have a Gaussian Distribution.

A key point about Gaussian processes is that the joint distribution over N variables $y_1 \dots y_N$ is specified completly by the second-order statistics.

① Since we have not any knowledge about the mean of $y(\mathbf{x})$, we take it to be zero. This is equivalent to choosing the mean vector of prior $p(\mathbf{w}|\alpha)$ to be 0

② The specification of GP is then completed by giving the covarience of $y(\mathbf{x})$ evaluated at any two values of $\mathbf{x}$ with kernel $\mathbb{E}[y(\mathbf{x}_n), y(\mathbf{x}_m)] = k(\mathbf{x}_n, \mathbf{x}_m)$.

Sketch of GPR

Let $f(x)$ ($y(x)$) be a zero mean GP indexed by its domain space.

I.e.

$$\begin{pmatrix} y(x_1) \\ \vdots \\ y(x_N) \end{pmatrix} \sim \mathcal{N}\left(0, \underset{\mathbb{K}}{\begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_N) \\ \vdots & \ddots & \\ k(x_N, x_1) & \cdots & k(x_N, x_N) \end{bmatrix}} \right)$$

and

$$\begin{pmatrix} y(x_1) \\ \vdots \\ y(x_N) \\ y(x^*) \end{pmatrix} \sim \mathcal{N}\left(0, \underset{\mathbb{K}^*}{\begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_N) & k(x_1, x^*) \\ \vdots & \ddots & & \vdots \\ k(x_N, x_1) & \cdots & k(x_N, x) & \vdots \\ k(x^*, x_1) & \cdots & k(x^*, x) & k(x^*, x^*) \end{bmatrix}} \right)$$

Assume $t = y + \varepsilon_n$ with $\varepsilon_{noise} \sim \mathcal{N}(0, \beta^{-1})$

$\Rightarrow$

$$\begin{pmatrix} t_1 \\ \vdots \\ t_N \end{pmatrix} \sim \mathcal{N}(0, K + \beta^{-1} I_N)$$

and

$$\begin{pmatrix} t_1 \\ \vdots \\ t_N \\ t^* \end{pmatrix} \sim \mathcal{N}(0, K^* + \beta^{-1} I_{N+1})$$

Since $p(t, t^*)$ is Gaussian so is $p(t^* | t)$

$$p(t^* | t) = \mathcal{N}(t^* | m(x^*), \sigma^2(x^*))$$

//

GPR in detail

We shall consider the noise on the observed target value

Assumption : Gaussian noisy model.

$$t_n = y_n + \varepsilon_n \qquad n = 1, 2, \ldots N$$

where ε_n followed zero mean Gaussian with precision β.

So for an N-dim vector $\mathbf{y} := (y_1 \ldots y_N)^T$, the probability distribution over $\mathbf{t} := (t_1, \ldots t_N)^T$ can be expressed as

$$p(\mathbf{t} \mid \mathbf{y}) = \mathcal{N}(\mathbf{t} \mid \mathbf{y}, \beta^{-1} I_N)$$

where I_N is the $N \times N$ identity matrix.

From the definition of GP, (prior dist. over y)

$$p(y) = \mathcal{N}(y \mid 0, K) \qquad \mathcal{N}\text{-dim Gaussian}$$

The kernel function determining the gram matrix K is typically chosen to express the property that

$$x_n \sim x_m \Rightarrow y(x_n) \text{ and } y(x_m) \text{ are strongly correlated}$$

One widely used kernel function for GPR is given by

$$k(x_n, x_m) := \theta_0 \exp\left\{ -\frac{\theta_1}{2} \| x_n - x_m \|^2 \right\} + \theta_2 + \theta_3 x_n^T x_m$$

Now we consider the distribution over t. By PRML section 2.3.3 (linear Gaussian model), we obtain

$$p(t) = \int p(t|y)\, p(y)\, dy = \mathcal{N}(t \mid 0, C) \qquad (6.61)$$

where $C = K + \beta^{-1} I_N$ with $C(x_n, x_m) = k(x_n, x_m) + \beta^{-1} \delta_{nm}$

Using Gaussian process viewpoint, we shall build a model of the joint probability distribution over sets of data points.

Goal : Given training data points, make prediction of target value for new input

x^* : some new input, t^* : its target value

$(\mathbf{X}, \mathbf{t})$ set of training data

Find the predictive distribution $p(t^* | \mathbf{t}) = p(t^* | x^*, \mathbf{X}, \mathbf{t})$

Let $\mathbf{t}^* := (t_1, \ldots t_N, t^*)^T$ ($N+1$ – dim random vector)

To find $p(t^* \mid \mathbf{t})$, we first consider $p(\mathbf{t}^*)$. From (6.61),

$$p(\mathbf{t}^*) = \mathcal{N}(\mathbf{t}^* \mid \mathbf{0}, \mathbf{C}^*) \qquad N+1 \text{ – dim Gaussian}$$

The covariance matrix $\mathbf{C}^*$ can be decomposed as

$$\mathbf{C}^* = \begin{pmatrix} \overset{N\times N}{\mathbf{C}} & \mathbf{k} \\ \mathbf{k}^T & c \end{pmatrix} \qquad (N+1) \times (N+1)$$

where $\mathbf{k}$ is the N-dim vector whose n^{th} element is $k(x_n, x^*)$ and $c = k(x^*, x^*) + \beta^{-1}$.

Since the conditional Gaussian distribution is again a Gaussian, $p(t^* \mid \mathbf{t})$ is Gaussian. $\mathbf{t}^* = \begin{pmatrix} \mathbf{t} \\ t^* \end{pmatrix}$ $N+1$ dim

From (2.81) and (2.82),

$$p(t^* \mid \mathbf{t}) = \mathcal{N}(t^* \mid m(x^*), \sigma^2(x^*))$$

where

$$m(x^*) = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t}$$

$$\sigma^2(x^*) = c - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k}$$

) depending on x^*

The expectation of predictive distribution can be written as

$$m(x^*) = \sum_{n=1}^{N} a_n \, k(x_n, x^*)$$

where a_n is the n^{th} component of $C^{-1}t$

Remark :

- Computational costs of GPR : the invertion of $N \times N$ matrix requires $O(N^3)$ once. For each new point, GPR has the cost $O(N^2)$ from the matrix multiplication

GPR in the real world (learning for hyperparameters)

Hyperparameter $\theta := \{\theta_0, \theta_1, \theta_2, \theta_3\}$ and β

Log likelihood function

$$\ln p(t|\theta) = -\frac{1}{2}\ln|C| - \frac{1}{2}t^T C^{-1} t - \frac{N}{2}\ln(2\pi)$$

$$\frac{\partial}{\partial\theta_i}\ln p(t|\theta) = -\frac{1}{2}\left(C^{-1}\frac{\partial C}{\partial\theta_i}\right) + \frac{1}{2}t^T C^{-1}\frac{\partial C}{\partial\theta_i}C^{-1}t$$

Generally, $\ln p(t|\theta)$ is not convex.

Tensorflow Probability

Exponentiated Quadratic

$$k(x_n, x_m) := \theta_0^2 \exp\left(\frac{\|x_n - x_m\|^2}{-2\theta_1^2}\right)$$

θ_0 : amplitude

θ_1 : length scale

$1/\beta$: observation noise variance.

$$k(x_n, x_m) := \theta_0 \exp\left\{-\frac{\theta_1}{2}\|x_n - x_m\|^2\right\} + \theta_2 + \theta_3 x_n^T x_m$$