Bayesian Linear Reg. w/ features

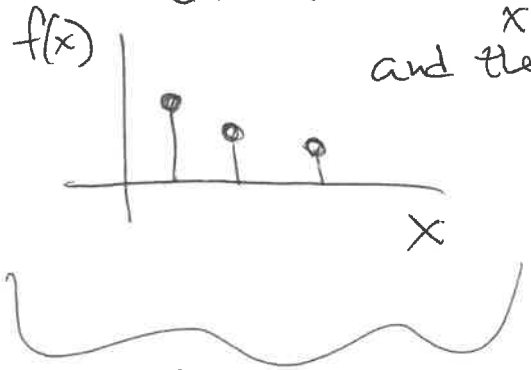
$\phi(x) = \begin{bmatrix} x^2 \\ x \\ 1 \end{bmatrix}$. $w \sim N(0, \Sigma_p)$ ← noise w/ variance σ_n^2

$y_n \leftarrow \underbrace{w^T \phi(x_n)}_{f(x_n)} + \varepsilon_n$

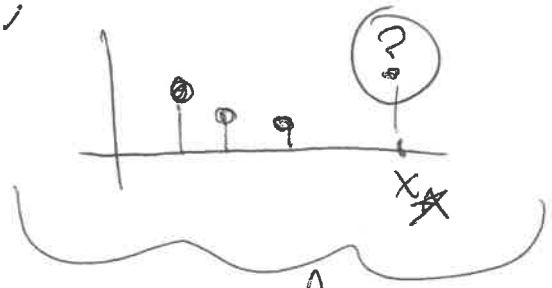
Day 2 01

Two ways to write predictive distrib.

Given 3 data points, what is function value at a new point x^*?



observed



goal

Look at posterior predictive (formulas from R&W)

$$f^* | x^*, X, y \sim N\left(\frac{1}{\sigma_n^2} \phi_*^T A^{-1} \Phi y \,,\; \phi_*^T A^{-1} \phi_* \right)$$

$$A = \frac{1}{\sigma_n^2} \Phi \Phi^T + \Sigma_p^{-1}$$

above is feature view

needs to invert $F \times F$ matrix A

F = # feature dims in $\phi(x)$, $\Phi = [\phi(x_1) \cdots \phi(x_N)]$

$$f^* | x^*, X, y \sim N\left(k(x_*, X)^T \left[k(X,X) + \sigma_n^2 I_N \right]^{-1} y \,,\; k(x_*, x_*) - k(x_*, X) \left[k(X,X) + \sigma_n^2 I_N \right]^{-1} k(X, x_*) \right)$$

above is the kernel view

needs to invert $N \times N$ matrix

$\sigma^2 I + k(X,X)$

Kernel Functions

must be
(1) symmetric
(2) pos. semi-definite

Suppose $x \in \mathbb{R}^2 = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

$$k(x,z) = (1 + x^T z)^2$$

$$= (1 + x_1 z_1 + x_2 z_2)^2$$

$$= 1 + x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 z_1 + 2x_2 z_2 + x_1 x_2 z_1 z_2$$

Can we write as a dot-product $\phi(x)\phi(z)$?

Yes, if $\phi(x) = \begin{bmatrix} 1 \\ x_1^2 \\ x_2^2 \\ \sqrt{2}\, x_1 \\ \sqrt{2}\, x_2 \\ x_1 x_2 \end{bmatrix}$

$\phi(x)$ has size 6

so $\phi^T \phi$ has cost $O(6)$

but $k(x,z)$ costs only $O(2)$

$$k(x,z) = \exp\{-(x-z)^2\}$$

$$= \exp\{-x^2 - z^2 + 2xz\}$$

$$= \exp\{-x^2\} \exp\{-z^2\} \sum_{k=0}^{\infty} \frac{2^k}{k!} x^k z^k$$

$$= \sum_{k=0}^{\infty} \left(\sqrt{\frac{2^k}{k!}}\, e^{-x^2}\, x^k\right)\left(\sqrt{\frac{2^k}{k!}}\, e^{-z^2}\, z^k\right)$$

$e^x = 1 + x + \frac{x^2}{2!} + \cdots$

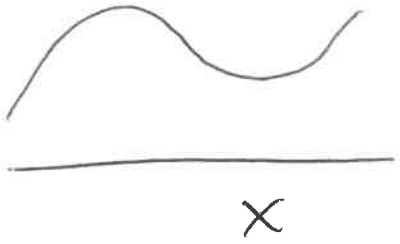
Taylor on e^{2xz}

inner product

Distrib. of Function Evaluations

Day 2 03

How do we represent functions?



Algebra

$f(x) = x^2$

Table (each input has one output)

x	$f(x)$
1.1	4.38
2.1	2.16
3.3	8.11

most useful for GPs

GP says: Given any finite set of inputs $x_1 \cdots x_N$, we can write $f(x_1) \cdots f(x_N)$ as a vector and ask what is joint distrib. of this vector

$$\vec{f} \mid \vec{x} \sim N(m(\vec{x}), k(\vec{x}, \vec{x}))$$

$m(x)$ can be any function $x \to \mathbb{R}$

$k(\cdot, \cdot)$ needs to specify valid covariance

must be positive semi-definite & symmetric

$k(X,X)$ is $N \times N$ valid kernel matrix if { all eigenvals ≥ 0 ; invertible (if > 0) } ↘ implies

Exercise:

With neighbor, come up with Python pseudocode to sample function values from a GP w/

mean $m(x) = 1$

kernel $k(x,x') = \delta(x,x')$

Useful Gaussian Properties

Consider a joint distribution that is Gaussian

$$\begin{bmatrix} x_A \\ x_B \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix}, \begin{bmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{bmatrix} \right)$$

Marginalization yields Gaussians

$$x_A \sim N(\mu_A, \Sigma_{AA}) \qquad p(x_A) = \int p(x_A, x_B)\, dx_B$$

$$x_B \sim N(\mu_B, \Sigma_{BB})$$

integral "marginalizes away" x_B

Conditioning yields Gaussians

$$x_A | x_B \sim N\left(\mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}(x_B - \mu_B),\ \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}\right)$$

Summation yields Gaussians if x_A same size as x_B

$$x_A + x_B \sim N(\mu_A + \mu_B,\ \Sigma_{AA} + \Sigma_{BB})$$

Big Idea:

if you know joint distr. of $\begin{bmatrix} x_A \\ x_B \end{bmatrix}$ is Gaussian,

there are super easy formulas to look up

for conditional $p(x_A | x_B)$

marginal $p(x_A)$

or sum $p(x_A + x_B)$

} all Gaussian

Weights space

prior is on weight vector of size D (# input dims / features)

Func space

prior is directly on values of function

GP prediction

easier than Linear regr. to derive

$$\begin{bmatrix} \vec{y} \\ \vec{y}_* \end{bmatrix} \Big| \, x, x^* \sim \mathcal{N}\left(\vec{0}, \begin{bmatrix} \sigma^2 + k(x,x) & k(x,x^*) \\ k(x^*,x) & \sigma^2 + k(x^*,x^*) \end{bmatrix} \right)$$

assumes $m(x) = 0$. For general case, see R&W.

Use conditioning rule from useful Gaussian properties

$$\vec{y}_* \mid y, x, x^* \sim \mathcal{N}(\mu^*, \Sigma^*)$$

$$\mu^* = k(x_*, x)\left[k(x,x) + \sigma^2 I_N\right]^{-1} \vec{y}$$

$$\Sigma^* = k(x_*, x_*) + \sigma^2 I - k(x_*, x)^T \left[k(x,x) + \sigma^2\right]^{-1} k(x, x_*)$$