Hw 5 → Fri

Final → 2 weeks, practice exam posted

COMPSCI 688: Probabilistic Graphical Models

Lecture 22: Gaussian Processes

Dan Sheldon

Manning College of Information and Computer Sciences
University of Massachusetts Amherst

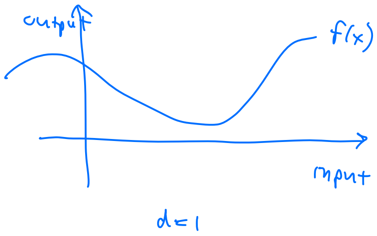
Partially based on materials by Benjamin M. Marlin (marlin@cs.umass.edu) and Justin Domke (domke@cs.umass.edu)

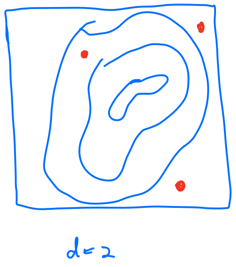
Overview

Gaussian Processes

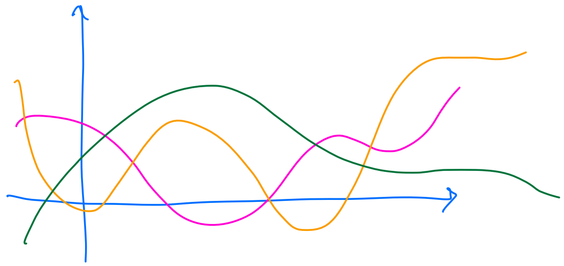
GPs = distributions over *functions*

Function $f : \mathbb{R}^d \to \mathbb{R}$



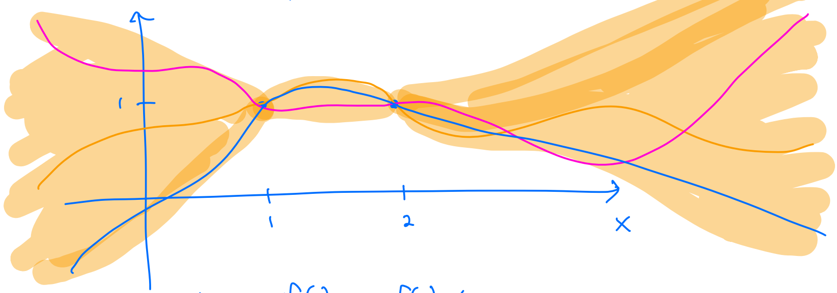


Distribution over functions — $p(f)$ ("prior")



Why? Model an unknown function. Compute “posterior” $p(f|\cdots)$ conditioned on some observed values.

$p(f \mid f(1)=1, f(2)=1)$



observe $f(1)=1$, $f(2)=1$

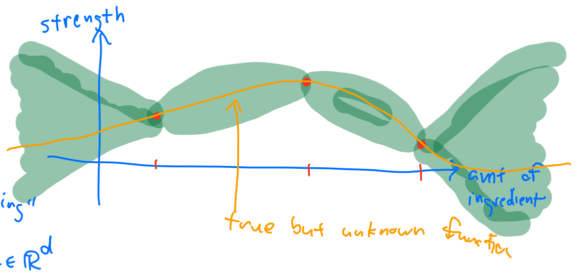
Demo

- prior samples
- posterior samples
- posterior mean and variance

UCB – “upper confidence bound”

Applications

— Gaussian Process



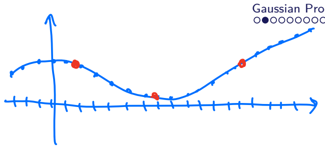
- Bayesian optimization
- Spatial statistics — “Kriging”
- Machine learning

$x \in \mathbb{R}^d$
want to estimate function f
to predict on new values by observing
output on training examples

GPs closely related to neural nets

Gaussian Processes

How to build a distribution over functions?



- **Key idea**: we only ever query a function at a finite number of points
- For any fixed $x^{(1)}, \dots, x^{(n)}$, model $f(x^{(1)}), \dots, f(x^{(n)})$ as jointly Gaussian

$$\begin{bmatrix} f(x^{(1)}) \\ f(x^{(2)}) \\ \vdots \\ f(x^{(n)}) \end{bmatrix} \sim \mathcal{N}(0, \underset{\overset{\uparrow}{n\times n}}{\Sigma}) = \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{bmatrix} k(x^{(1)},x^{(1)}) & k(x^{(1)},x^{(2)}), & \dots & , k(x^{(1)},x^{(n)}) \\ \vdots & \ddots & & \\ \vdots & & \ddots & \\ k(x^{(n)},x^{(1)}) & & & k(x^{(n)},x^{(n)}) \end{bmatrix}\right)$$

What Σ? Needs to work for *any* set of input points.

Covariance Function

Take $\Sigma_{ij} = k(x^{(i)}, x^{(j)})$ where

$$k(x, x') := \text{Cov}(f(x), f(x'))$$

is a **covariance function** or **kernel function**

- Specifies covariance between outputs $f(x)$, $f(x')$ for any inputs x, x'
- Example: $k(x,x') = \exp(-\frac{1}{2}(x-x')^2)$ — squared exponential
- must lead to positive semidefinite matrices — look up common covariance functions

Gaussian Process

This construction is a *Gaussian process* or "GP".

Formally, a GP is a distribution over an infinite set of random variables (the values of $f(x)$ for *all* x), where the joint distribution of any finite subset is multivariate Gaussian.

A GP is specified by the covariance function $k(x, x')$. (We assume without loss of generality the mean is zero.) → can add a non-random function $\mu(x)$

We often write $f(x) \sim \text{GP}(0, k(x, x'))$ or $f \sim \text{GP}(0, k)$.

$f \sim GP(\mu, k)$

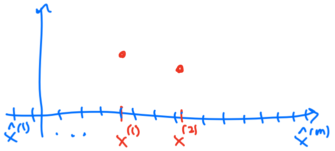
$\mu(x)$

$k(x, x')$

Demo

Demo: sampling from prior

Conditioning



How can we compute $p(f|\cdots)$ given some observations? Setup:

- Training inputs $x^{(1)}, \ldots, x^{(n)}$
- Test inputs $\hat{x}^{(1)}, \ldots, \hat{x}^{(m)}$
- Have joint Gaussian distribution over training and test *outputs*

$$\underbrace{f(x^{(1)}), \ldots, f(x^{(n)}),}_{\mathbf{f}\text{:observed}} \quad \underbrace{f(\hat{x}^{(1)}), \ldots, f(\hat{x}^{(m)})}_{\hat{\mathbf{f}}\text{:unobserved}}$$

$k(x, x') = k(x', x)$

$x^{(1)}$
$\vdots$
$x^{(n)}$
$\hat{x}^{(1)}$
$\vdots$
$\hat{x}^{(m)}$

entries $k(x^{(i)}, \hat{x}^{(j)})$

$$\begin{matrix}\text{train}\\ \text{test}\end{matrix}\begin{bmatrix}\mathbf{f}\\ \hat{\mathbf{f}}\end{bmatrix} \sim \mathcal{N}\left(0, \begin{array}{ccc} \text{train} & \text{test} & \\ K_{XX} & K_{X\hat{X}} & \text{train} \\ K_{\hat{X}X} & K_{\hat{X}\hat{X}} & \text{test}\end{array}\right)$$

Notation:

- Matrices $X \in \mathbb{R}^{n\times d}$ and $\hat{X} \in \mathbb{R}^{m\times d}$ of training and test inputs
- Training covariance matrix $K_{X,X} \in \mathbb{R}^{n\times n}$ — entries $k(x^{(i)}, x^{(j)})$ for all (i, j)
- Test covariance matrix $K_{\hat{X}\hat{X}} \in \mathbb{R}^{m\times m}$
- Train-test covariance matrix $K_{X\hat{X}} \in \mathbb{R}^{n\times m}$

Want $p(\hat{\mathbf{f}}|\mathbf{f}) \implies$ Gaussian conditioning

Gaussian Conditioning

$z \sim \mathcal{N}(0, \Sigma)$

Suppose

joint $$\begin{bmatrix} z_a \\ z_b \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix}\right)$$

Then

"Schur complement"

conditional $$p(z_b|z_a) = \mathcal{N}(z_b|\underbrace{\Sigma_{ba}\Sigma_{aa}^{-1}z_a}_{\mu_{b|a}}, \underbrace{\Sigma_{bb} - \Sigma_{ba}\Sigma_{aa}^{-1}\Sigma_{ab}}_{\Sigma_{b|a}})$$

GP Conditioning

For the GP model, Gaussian conditioning gives

$$p(\hat{\mathbf{f}}|\mathbf{f}) = \mathcal{N}(\underbrace{K_{\hat{X}X}K_{XX}^{-1}\mathbf{f}}_{\mu_{\hat{f}|f}}, \underbrace{K_{\hat{X}\hat{X}} - K_{\hat{X}X}K_{XX}^{-1}K_{X\hat{X}}}_{\Sigma_{\hat{f}|f}})$$

Demo: GP conditioning

Noisy Observations

We usually don't get to observe output values exactly. In *GP regression* we observe noisy outputs for each training input:

$$y^{(i)} = f(x^{(i)}) + \underset{\uparrow\,\text{meas. error}}{\epsilon^{(i)}}, \quad \epsilon^{(i)} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2).$$

We want $p(\overset{f}{\overset{\downarrow}{\hat{\mathbf{f}}}}|\mathbf{y})$ where $\mathbf{y}$ is a vector of the $y^{(i)}$ values. The joint distribution is

$$\begin{bmatrix} f \\ \hat{f} \end{bmatrix} \sim N(\cdots, \cdots)$$

$$\begin{bmatrix} \mathbf{y} \\ \hat{\mathbf{f}} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K_{XX} + \sigma^2 I & K_{X\hat{X}} \\ K_{\hat{X}X} & K_{\hat{X}\hat{X}} \end{bmatrix}\right)$$

Gaussian conditioning now gives

$$p(\hat{\mathbf{f}}|\mathbf{y}) = \mathcal{N}(K_{\hat{X}X}(K_{XX} + \sigma^2 I)^{-1}\mathbf{y},\, K_{\hat{X}\hat{X}} - K_{\hat{X}X}(K_{XX} + \sigma^2 I)^{-1}K_{X\hat{X}})$$

Demo: Gaussian conditioning with noisy observations