ST308 - Lent term
Bayesian Inference

Kostas Kalogeropoulos

Gaussian Process Regression and Classification

Outline

Topics: Bayesian non-parametrics, Variance Kernels, Multivariate Normal Conditioning, Gaussian Process Posterior, Gaussian Process Prediction.

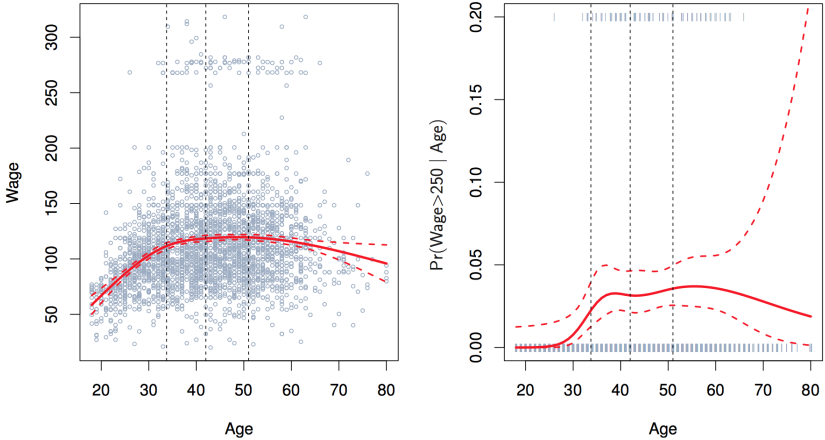
Outline

Bayesian Non-parametrics

There are two main areas in Bayesian Non-parametrics:

- Unknown distributions: Do not assume a specific distribution, instead use a mixture with potentially infinite components - Dirichlet process prior, we have seen finite mixture in week 7.
- Unknown functions in supervised learning: Do not assume a specific function between y and X instead perform Bayesian inference on the function - Gaussian process prior, today's topic.

Non-parametric regression / supervised learning



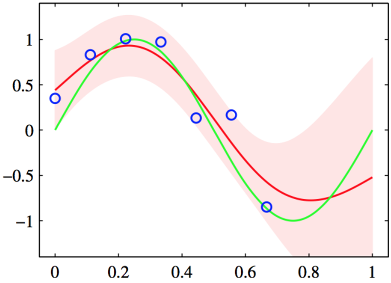
Non-parametric regression / supervised learning

Let $\mathcal{X}$ be a set and $\mathcal{F}$ be a set of functions over $\mathcal{X}$ (e.g., smooth functions). We observe $(x_1, y_i), \ldots, (x_n, y_n)$ ($x_i \in \mathcal{X}$, $y_i \in \mathbb{R}$) satisfying

$$y_i = f(x_i) + \varepsilon_i$$

where $f \in \mathcal{F}$, and $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)$ independent of $x = (x_1, \ldots, x_n)$.

We want to fit a smooth curve or surface through the data:



Putting a prior on the regression function

Non-parametric regression model:

$$y_i = f(x_i) + \varepsilon_i$$

Assume errors ε have density $\pi(\varepsilon)$, usually $N(0, \sigma^2 I_n)$. Then

$$y = (y_1, \ldots, y_n) | f, x, \sigma^2 \sim N(f,\ \sigma^2 I_n)$$

Bayes regression: assign prior $\pi(f)$ to f, and compute posterior

$$\pi(f|\mathbf{y}) = \frac{\pi(f) g(\varepsilon|f)}{\int_{\mathcal{F}} \pi(f) g(\varepsilon|f) df}$$

- f can be estimated by its posterior mean
- Credibility intervals for each $f(x)$ can be computed

Regression with a functional covariate

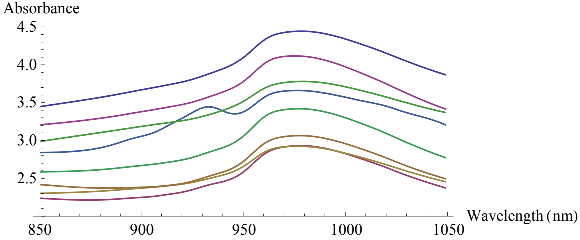


Figure: Sample of spectrometric curves

- $n = 215$ pieces of finely chopped meat. y_i is fat content
- $x_i = x_i(\cdot)$ is spectrometric curve ('functional' covariate)

Regression model

$$y_i = f(x_i) + \varepsilon_i$$

$$\uparrow$$

a curve

Prior: a Gaussian process needed over the curves x_i

Effect of treatment on cow growth

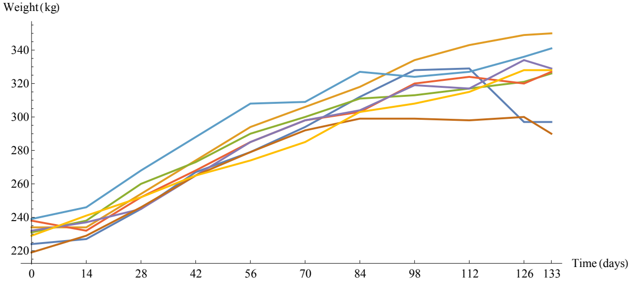
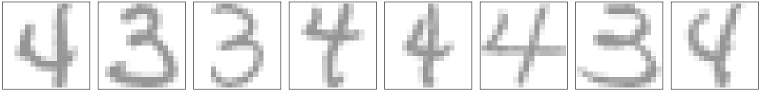


Figure: Sample of eight growth curves of cows

- 60 cows (30 get treatment A, 30 treatment B).
- How does treatment affect growth?
- y_{it}: weight of cow i at time t

Handwritten digit recognition



USPS data. Training/test sample: 7291/2007 handwritten digits

Multidimensional response:

$$y_{ij} = \begin{cases} 1 & \text{picture } i \text{ represents digit } j \\ 0 & \text{otherwise} \end{cases}$$

Covariate x_i is a picture.

Regression model:

$$y_{ij} = f(j, x_i) + \varepsilon_{ij} \qquad f \in \mathcal{F}$$

Here, the digit j acts like a nominal level covariate, and x_i is a ‘picture type’ covariate.

Outline

Gaussian processes

Definition

Let $\mathcal{X}$ be a set. A random function $f : \mathcal{X} \to \mathbb{R}$ is called a *Gaussian process* if for any $x_1, \dots, x_n \in \mathcal{X}$, $[f(x_1), \dots, f(x_n)]$ has a multivariate normal distribution.

Gaussian processes are characterized by

- Mean $f_0(\cdot)$.
- Covariance kernel $K(\cdot, \cdot)$.

If f is a Gaussian process with mean f_0 and covariance kernel K, then

$$[f(x_1), \dots, f(x_n)]^\top \sim N(f_0, K)$$

where $f_0 = [f_0(x_1), \dots, f_0(x_n)]$, and K is the $n \times n$ matrix with elements $K(x_i, x_j)$.

Covariance kernels

A covariance kernel is a positive definite function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, i.e.,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} K(x_i, x_j)\alpha_i\alpha_j \geq 0$$

for all $x_1, \ldots, x_n \in \mathcal{X}$ and all scalars $\alpha_1, \ldots, \alpha_n$.

Examples:

- The *linear* kernel $K(x, x') = \langle x, x' \rangle$
- The *squared exponential* kernel

$$K(x, x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$$

- The *fractional Brownian motion* kernel (FBM) with Hurst coefficient α $(0 < \alpha < 1)$: $K(x, x') = \frac{1}{2}\left(\|x\|^{2\alpha} + \|x'\|^{2\alpha} - \|x - x'\|^{2\alpha}\right)$

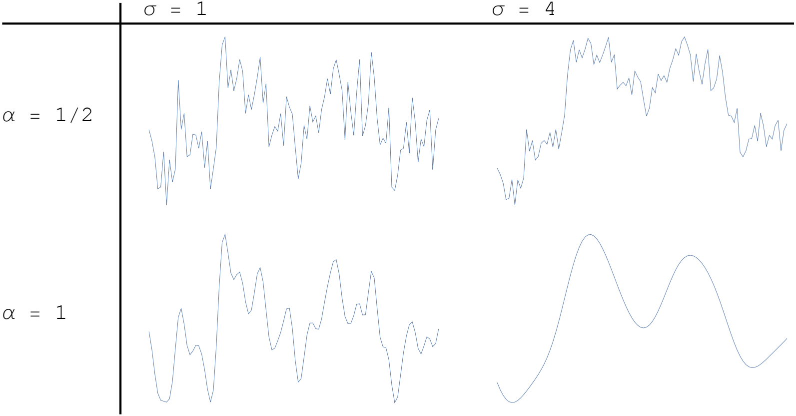
What do sample paths look like?

Let’s look at one fractional Brownian motion (FBM) and exponential process paths, with different values of the hyper-parameters.

Higher dimensions are also important, but cannot be visualised easily.

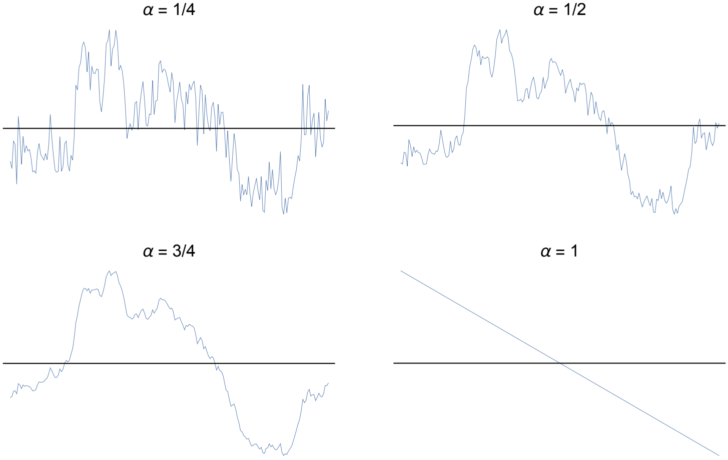
Sample paths 1-dim exponential kernel

Covariance kernel: $K(x, x') = e^{-\frac{\|x - x'\|^{2\alpha}}{2\sigma^2}}$



Sample paths FBM: 1-dim

Covariance kernel: $K(x, x') = \frac{1}{2}\left(\|x\|^{2\alpha} + \|x'\|^{2\alpha} - \|x - x'\|^{2\alpha}\right)$



Outline

Gaussian process (GP) regression

Assume

$$y_i = f(x_i) + \varepsilon_i \qquad\qquad i = 1, \ldots, n \tag{1}$$

where

$$\varepsilon_i \sim_{\text{iid}} N(0, \sigma_\varepsilon^2)$$
$$\pi(f) = \text{GP}(0, K)$$
$$f \text{ and } (\varepsilon_1, \ldots, \varepsilon_n) \text{ are independent}$$

Marginal distribution of y

The marginal distribution of y is multivariate normal with means

$$E(y_i) = E(f(x_i) + \varepsilon_i) = E(f(x_i)) + E(\varepsilon_i) = 0$$

and covariances

$$\begin{aligned} \text{cov}(y_i, y_j) &= \text{cov}(f(x_i) + \varepsilon_i, f(x_j) + \varepsilon_j) \\ &= \text{cov}(f(x_i), f(x_j)) + \text{cov}(\varepsilon_i, \varepsilon_j) \\ &= K(x_i, x_j) + \sigma_\varepsilon^2 I(i = j) \end{aligned}$$

In matrix notation, denoting with K_n the $n \times n$ matrix containing all the (x_i, x_j) pairs, we get

$$y \sim N(0,\ K_n + \sigma_\varepsilon^2 I_n)$$

Joint distribution of the y and f

For each $x_i \in \mathcal{X}$ and y_j

$$\begin{aligned}
\operatorname{cov}(f(x_i), y_j) &= Ef(x_i)y_j - Ef(x_i)Ey_j \\
&= Ef(x_i)(f(x_j) + \epsilon_j) - Ef(x_i) \times 0 \\
&= Ef(x_i)f(x_j) + Ef(x_i)E\epsilon_j = K(x_i, x_j)
\end{aligned}$$

Hence, the joint distribution of f and y is

$$\begin{pmatrix} f \\ y \end{pmatrix} \sim N\left[\begin{pmatrix} 0_n \\ 0_n \end{pmatrix}, \begin{pmatrix} K_n & K_n \\ K_n & K_n + \sigma_\varepsilon^2 I_n \end{pmatrix}\right]$$

where $f = [f(x_1), \ldots, f(x_n)]^\top$ and 0_n an $n \times 1$ vector of zeroes.

Conditional distribution of multivariate normals

A standard result is that if

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \sim N\left[0, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}\right]$$

then

$$\left(z_1 | z_2\right) \sim N\left[\Sigma_{12}\Sigma_{22}^{-1} z_2, \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\right]$$

Since the posterior of f is $f|y$, and since (f, y) have a joint Gaussian distribution, this formula can be used to obtain the posterior of f.

With $z_1 = f$ and $z_2 = y$, the above result gives the posterior distribution of f, given next.

Posterior distribution of f

Consider the model (1) with $\varepsilon_i \sim_{\text{iid}} N(0, \sigma_\varepsilon^2)$ and prior $\pi(f)$ a Gaussian process with mean 0 and covariance kernel K.

Under the above assumptions, the distribution of $f|y$ is a Gaussian process with mean M_f and covariance kernel V_f, which are given by

$$
\begin{aligned}
M_f &= K_n \left[K_n + \sigma_\varepsilon^2 I_n\right]^{-1} y \\
V_f &= K_n - K_n \left[K_n + \sigma_\varepsilon^2 I_n\right]^{-1} K_n
\end{aligned}
$$

Prediction a new point

Consider a new point y_{n+1} that we want to forecast based on x_{n+1}.

To find the joint distribution of the y and y_{n+1} (given x and x_{n+1}) note that $\mathrm{cov}(y_i, y_{n+1}) = K(x_i, x_{n+1})$ for $i = 1, \ldots, n$.

Denoting $k_{n+1} = [K(x_1, x_{n+1}), \ldots, K(x_n, x_{n+1})]^\top$, we get as before

$$\begin{pmatrix} y_{n+1} \\ y \end{pmatrix} \sim N\left[\begin{pmatrix} 0_n \\ 0_n \end{pmatrix}, \begin{pmatrix} K(x_{n+1}, x_{n+1}) + \sigma_\varepsilon^2 & k_{n+1}^\top \\ k_{n+1} & K_n + \sigma_\varepsilon^2 I_n \end{pmatrix}\right]$$

Then $y_{n+1}|y$ is a Normal with mean M_{n+1} and variance V_{n+1}:

$$\begin{aligned} M_{n+1} &= k_{n+1}^\top \left[K_n + \sigma_\varepsilon^2 I_n\right]^{-1} y \\ V_{n+1} &= K(x_{n+1}, x_{n+1}) + \sigma_\varepsilon^2 - k_{n+1}^\top \left[K_n + \sigma_\varepsilon^2 I_n\right]^{-1} k_{n+1} \end{aligned}$$

Estimating σ_ε and the K hyper-parameters

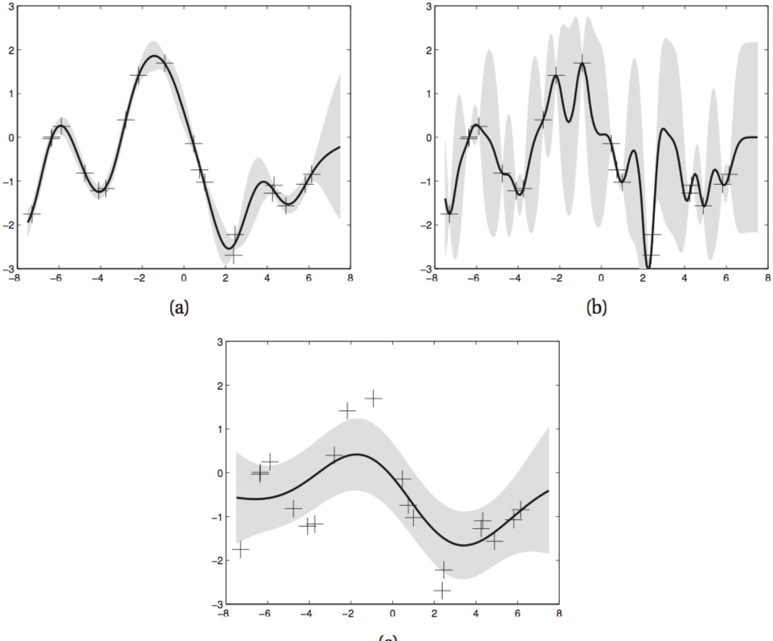
The posterior distribution of f still depends on unknown parameters θ consisting of σ_ε and any hyper-parameters of K, e.g. σ for squared exponential and α for FBM.

The marginal likelihood $\pi(y|\theta)$, multivariate normal density with mean 0_n and covariance matrix $V_y = K_n + \sigma_\varepsilon^2 I_n$, can be of help.

Two methods can be used to estimate θ:

- Maximum (marginal) likelihood, i.e., maximize $\pi(y|\theta)$. Aka empirical Bayes
- Put a prior on the hyper-parameters, and estimate them by their posterior means. Aka hierarchical Bayes

Fitting a Gaussian process - Regerssion



(a) (b)

Outline

Binary classification with Gaussian processes

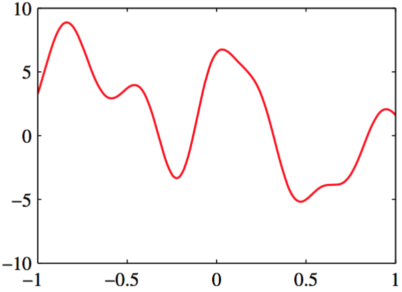
Let's consider the classification problem with a binary target variable y.
In llogistic regression we assume that

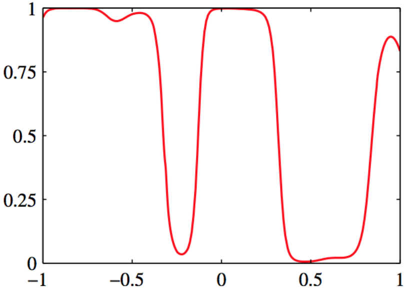
$$
\begin{aligned}
y &\sim \prod_{i=1}^{n} Bernoulli\big(\pi(x_i, \beta)\big), \\
\pi(x_i, \beta) &= \sigma(x_i \beta) = \frac{1}{1 + \exp(-x_i \beta)}
\end{aligned}
$$

In logistic regression with Gaussian processes we assume that

$$
\begin{aligned}
y &\sim \prod_{i=1}^{n} Bernoulli\big(\pi(f_i)\big), \; f = (f_1, \ldots, f_n)^\top \\
\pi(f_i) &= \sigma(f_i) = \frac{1}{1 + \exp(-f_i)} \\
f &= N(0_n, K_n)
\end{aligned}
$$

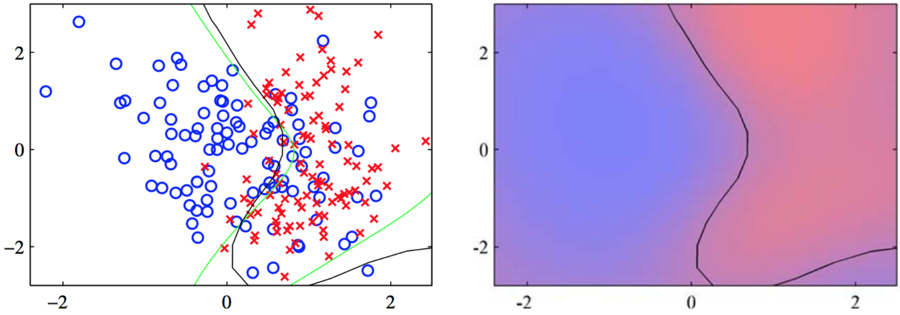
Gaussian process classification





Left: A simulated path $f(x)$ from a Gaussian Process
Right: The path $f(x)$ transformed to $[0, 1]$ scale reflecting $\pi(y_i = 1)$

Gaussian process classification in 2 dimensions



Left: Simulated points with green line being optimal boundary and black line being the GP boundary
Right: Prediction probabilities from Gaussian process classifier

Implementation

Let θ denote the Kernel hyper-parameters of f. The augmented likelihood can be written as

$$\pi(y, f|\theta) = \pi(f|\theta)\pi(y|f,\theta) = \pi(f|\theta)\prod_i Bernoulli\big((\pi(f_i))\big)$$

The posterior $\pi(f|y,\theta)$ is intractable.

The Laplace approximation can be used $N\big(f_M,\ H(f_m)^{-1}\big)$ where f_M is the mode of f and $H(\cdot)$ is the Hessian.

We can use Newton-Raphson as in the logistic regression with

$$\begin{aligned}\nabla_f \log\pi(y, f|\theta) &= \nabla_f \log\pi(y|f,\theta) + K_n^{-1} f \\ H(f) &= -\nabla_f\nabla_f \log\pi(y, f|\theta) + K_n^{-1}\end{aligned}$$

Implementation

MCMC on f provides a more accurate but also computationally expensive option.

Variational Bayes is also feasible.

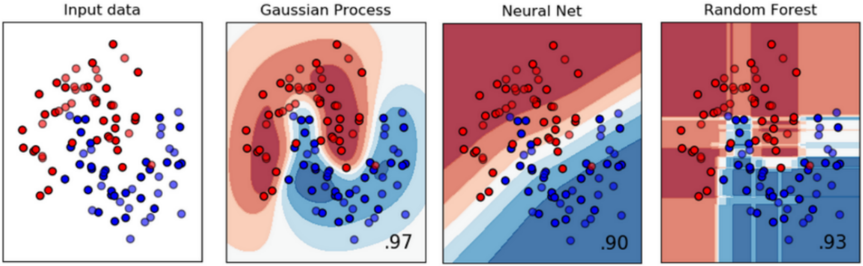
The predictive distribution for a future y_{n+1} can be written as

$$\pi(y_{n+1}|y) \approx \int \text{Bernoulli}\big(\sigma(f_{n+1})\big) N\left(f_{n+1}|f_M, H(f)^{-1}\right) df$$

One can sample from the above or just evaluate at f_M to classify y_{n+1}

θ can also be included in the Laplace/Variational approximation or MCMC together with f.

Gaussian process classification vs other classifiers



Summary

- Gaussian processes provide flexible tools in supervised learning settings.
- Fitting and prediction is carried out using Bayesian inference on that paths of the function f conditionally or jointly on hyper-parameters θ.
- The choice of kernel and θ is very important.
- Overall they perform very well but training is not always easy and also computationally expensive. Approximate and sparse versions are currently explored.

Today's lecture - Reading

Options reading:

Gelman et al, Chapter 21