Learning with Gaussian Processes

Nishanth Koganti

August 3, 2015

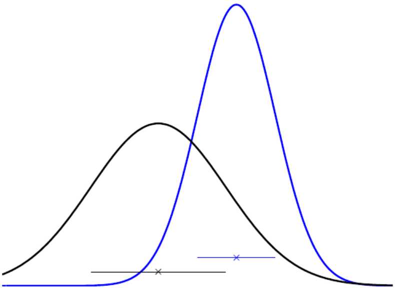
Supervised Learning: Ubiquitous questions

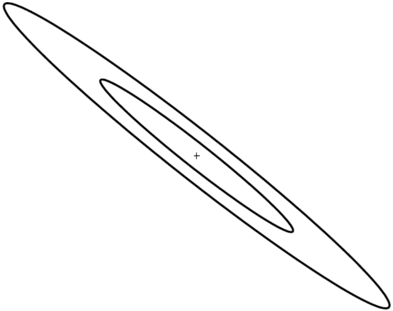
- Model fitting
  - How to fit parameters?
  - How to handle overfitting?
- Model selection
  - Which model best represents data?
  - How sure can I be?
- Interpretation
  - What is the accuracy of predictions?
  - Can I trust predictions under model uncertainity?

Gaussian Processes provides framework to address these issues.

Outline

Gaussian Distribution

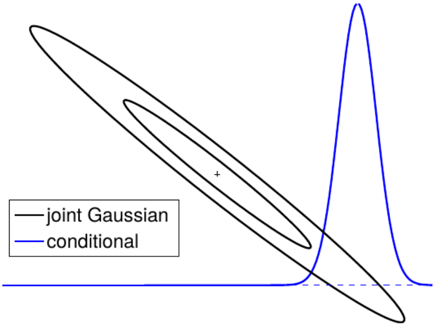


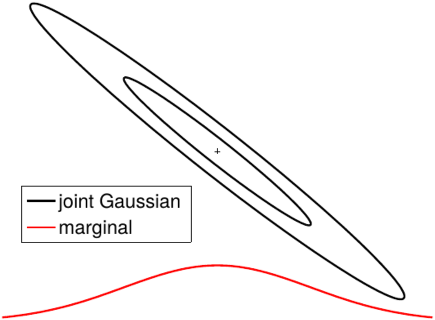


$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-D/2}|\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

$\boldsymbol{\mu}$: mean vector, $\boldsymbol{\Sigma}$: covariance matrix

Conditional and Marginal of a Gaussian





Conditional and Marginal of a joint Gaussian is also Gaussian.

What is a Gaussian Process?

Generalization of a multivariate Gaussian to infinitely many variables.

Definition: *Gaussian Process is a collection of random variables, any finite collection of which are Gaussian Distributed.*

What is a Gaussian Process?

Generalization of a multivariate Gaussian to infinitely many variables.

Definition: *Gaussian Process is a collection of random variables, any finite collection of which are Gaussian Distributed.*

Gaussian distribution: mean vector, $\boldsymbol{\mu}$, and covariance matrix $\boldsymbol{\Sigma}$:

$$\mathbf{f} = (f_1, \ldots, f_n)^T \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \text{indices } i = 1, \ldots, n$$

What is a Gaussian Process?

Generalization of a multivariate Gaussian to infinitely many variables.

Definition: *Gaussian Process is a collection of random variables, any finite collection of which are Gaussian Distributed.*

Gaussian distribution: mean vector, $\boldsymbol{\mu}$, and covariance matrix $\boldsymbol{\Sigma}$:

$$\mathbf{f} = (f_1, \ldots, f_n)^T \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad \text{indices } i = 1, \ldots, n$$

Gaussian process: mean function, $m(x)$, and covariance function $k(x, x')$:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')), \quad \text{indices: } x$$

Marginalization Property

How can we represent infinite mean vector and infinite covariance matrix?

...luckily saved by *marginalization property*:

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{y}) d\mathbf{y}$$

Marginalization Property

How can we represent infinite mean vector and infinite covariance matrix?

...luckily saved by *marginalization property*:

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{y}) d\mathbf{y}$$

For Gaussians:

$$p(\mathbf{x}, \mathbf{y}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \right)$$

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{a}, \mathbf{A})$$

Random sampling from Gaussian Process

Considering one dimensional Gaussian process:

$$p(f(x)) \sim \mathcal{GP}\left(m(x) = 0, k(x, x') = \exp\left(-\frac{1}{2}(x - x')^2\right)\right)$$

Random sampling from Gaussian Process

Considering one dimensional Gaussian process:

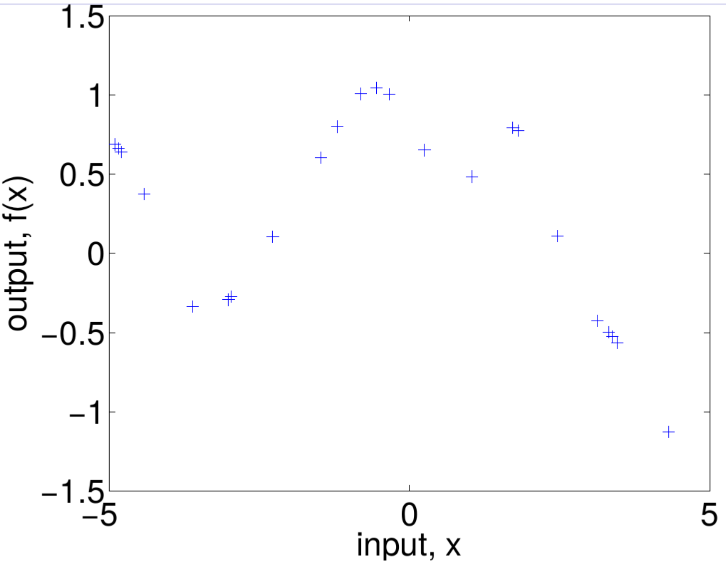
$$p(f(x)) \sim \mathcal{GP}\left(m(x) = 0, k(x, x') = \exp\left(-\frac{1}{2}(x - x')^2\right)\right)$$

Sampling is done by focusing on subset $\mathbf{f} = (f(x_1), f(x_2), \ldots, f(x_n))^T$:

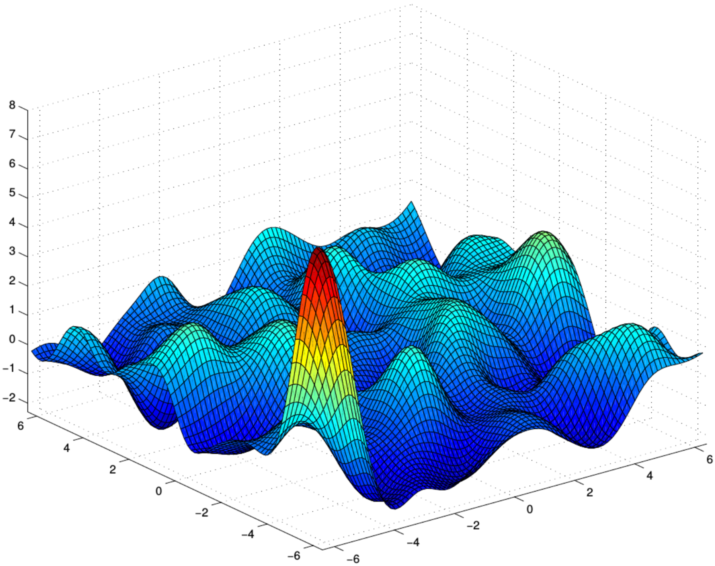
$$\mathbf{f} \sim \mathcal{N}(0, \mathbf{\Sigma}), \text{ where } \mathbf{\Sigma}_{ij} = k(x_i, x_j)$$

Coordinates of $\mathbf{f}$ are plot as a function of corresponding x

Random sample for single dimension



2 Dimensional Gaussian Process Sample



Sequential Generation of Samples

Factorize the joint distribution and generate function values sequentially:

$$p(f_1, \ldots, f_n | \mathbf{x}_1, \ldots, \mathbf{x}_n) = \prod_{i=1}^{n} p(f_i | f_{i-1}, \ldots, f_1, \mathbf{x}_1, \ldots, \mathbf{x}_n)$$

Sequential Generation of Samples

Factorize the joint distribution and generate function values sequentially:

$$p(f_1, \ldots, f_n | \mathbf{x}_1, \ldots, \mathbf{x}_n) = \prod_{i=1}^{n} p(f_i | f_{i-1}, \ldots, f_1, \mathbf{x}_1, \ldots, \mathbf{x}_n)$$

What do individual terms look like?

$$p(\mathbf{x}, \mathbf{y}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \right)$$

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{a} + \mathbf{B}\mathbf{C}^{-1}(\mathbf{y} - \mathbf{b}), \mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T)$$

Parametric Model and Maximum Likelihood

Parametric Model:

- data: $\mathbf{x}, \mathbf{y}$
- model: $\mathbf{y} = f_w(\mathbf{x}) + \epsilon$

Parametric Model and Maximum Likelihood

Parametric Model:

- data: $\mathbf{x}, \mathbf{y}$
- model: $\mathbf{y} = f_w(\mathbf{x}) + \epsilon$

Gaussian Likelihood:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M) \propto \prod_i \exp\left(-\frac{(y_i - f_{\mathbf{w}}(x_i))^2}{2\sigma_{noise}^2}\right)$$

Parametric Model and Maximum Likelihood

Parametric Model:

- data: $\mathbf{x}, \mathbf{y}$
- model: $\mathbf{y} = f_w(\mathbf{x}) + \epsilon$

Gaussian Likelihood:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M) \propto \prod_i \exp\left(-\frac{(y_i - f_{\mathbf{w}}(x_i))^2}{2\sigma^2_{noise}}\right)$$

Maximizing Likelihood:

$$\mathbf{w}_{ML} = \text{argmax}_{\mathbf{w}} p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M)$$

Parametric Model and Maximum Likelihood

Parametric Model:

- data: $\mathbf{x}, \mathbf{y}$
- model: $\mathbf{y} = f_w(\mathbf{x}) + \epsilon$

Gaussian Likelihood:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M) \propto \prod_i \exp\left(-\frac{(y_i - f_{\mathbf{w}}(x_i))^2}{2\sigma_{noise}^2}\right)$$

Maximizing Likelihood:

$$\mathbf{w}_{ML} = \text{argmax}_{\mathbf{w}} p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M)$$

Making predictions:

$$p(y^*|x^*, \mathbf{w}_{ML}, M)$$

Parametric Model and Bayesian Inference

Parametric Model:

- data: $\mathbf{x}, \mathbf{y}$
- model: $\mathbf{y} = f_w(\mathbf{x}) + \epsilon$

Parametric Model and Bayesian Inference

Parametric Model:

- data: $\mathbf{x}, \mathbf{y}$
- model: $\mathbf{y} = f_w(\mathbf{x}) + \epsilon$

Gaussian Likelihood:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M) \propto \prod_i \exp\left(-\frac{(y_i - f_{\mathbf{w}}(x_i))^2}{2\sigma_{noise}^2}\right)$$

Parametric Model and Bayesian Inference

Parametric Model:

- data: $\mathbf{x}, \mathbf{y}$
- model: $\mathbf{y} = f_w(\mathbf{x}) + \epsilon$

Gaussian Likelihood:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M) \propto \prod_i \exp\left(-\frac{(y_i - f_{\mathbf{w}}(x_i))^2}{2\sigma_{noise}^2}\right)$$

Prior over parameters:

$$p(\mathbf{w}|M)$$

Parametric Model and Bayesian Inference

Parametric Model:

- data: $\mathbf{x}, \mathbf{y}$
- model: $\mathbf{y} = f_w(\mathbf{x}) + \epsilon$

Gaussian Likelihood:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M) \propto \prod_i \exp\left(-\frac{(y_i - f_{\mathbf{w}}(x_i))^2}{2\sigma_{noise}^2}\right)$$

Prior over parameters:

$$p(\mathbf{w}|M)$$

Posterior parameter distribution:

$$p(\mathbf{w}|\mathbf{x}, \mathbf{y}, M) = \frac{p(\mathbf{w}|M)p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M)}{p(\mathbf{y}|\mathbf{x}, M)}$$

Parametric Model and Bayesian Inference

Making predictions:

$$p(y^*|x^*, \mathbf{x}, \mathbf{y}, M) = \int p(y^*|\mathbf{w}, x^*, M) p(\mathbf{w}|\mathbf{x}, \mathbf{y}, M) d\mathbf{w}$$

Parametric Model and Bayesian Inference

Making predictions:

$$p(y^*|x^*, \mathbf{x}, \mathbf{y}, M) = \int p(y^*|\mathbf{w}, x^*, M) p(\mathbf{w}|\mathbf{x}, \mathbf{y}, M) d\mathbf{w}$$

Marginal Likelihood:

$$p(\mathbf{y}|\mathbf{x}, M) = \int p(\mathbf{w}|M) p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M) d\mathbf{w}$$

Parametric Model and Bayesian Inference

Making predictions:

$$p(y^*|x^*, \mathbf{x}, \mathbf{y}, M) = \int p(y^*|\mathbf{w}, x^*, M)p(\mathbf{w}|\mathbf{x}, \mathbf{y}, M)d\mathbf{w}$$

Marginal Likelihood:

$$p(\mathbf{y}|\mathbf{x}, M) = \int p(\mathbf{w}|M)p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M)d\mathbf{w}$$

Model probability:

$$p(M|\mathbf{x}, \mathbf{y}) = \frac{p(M)p(\mathbf{y}|\mathbf{x}, M)}{p(\mathbf{y}|\mathbf{x})}$$

Parametric Model and Bayesian Inference

Making predictions:

$$p(y^*|x^*, \mathbf{x}, \mathbf{y}, M) = \int p(y^*|\mathbf{w}, x^*, M)p(\mathbf{w}|\mathbf{x}, \mathbf{y}, M)d\mathbf{w}$$

Marginal Likelihood:

$$p(\mathbf{y}|\mathbf{x}, M) = \int p(\mathbf{w}|M)p(\mathbf{y}|\mathbf{x}, \mathbf{w}, M)d\mathbf{w}$$

Model probability:

$$p(M|\mathbf{x}, \mathbf{y}) = \frac{p(M)p(\mathbf{y}|\mathbf{x}, M)}{p(\mathbf{y}|\mathbf{x})}$$

Problem: integrals are intractable for most interesting models!

Non-parametric Gaussian Process Models

Parameters are replaced by “function” itself!

Non-parametric Gaussian Process Models

Parameters are replaced by “function” itself!
Gaussian Likelihood:

$$\mathbf{y}|\mathbf{x}, f(x), M \sim \mathcal{N}(\mathbf{f}, \sigma^2_{noise} I)$$

Non-parametric Gaussian Process Models

Parameters are replaced by "function" itself!
Gaussian Likelihood:

$$\mathbf{y}|\mathbf{x}, f(x), M \sim \mathcal{N}(\mathbf{f}, \sigma^2_{noise} I)$$

Gaussian Process Prior:

$$f(x)|M \sim \mathcal{GP}(m(x) = 0, k(x, x'))$$

Non-parametric Gaussian Process Models

Parameters are replaced by "function" itself!
Gaussian Likelihood:

$$\mathbf{y}|\mathbf{x}, f(x), M \sim \mathcal{N}(\mathbf{f}, \sigma^2_{noise} I)$$

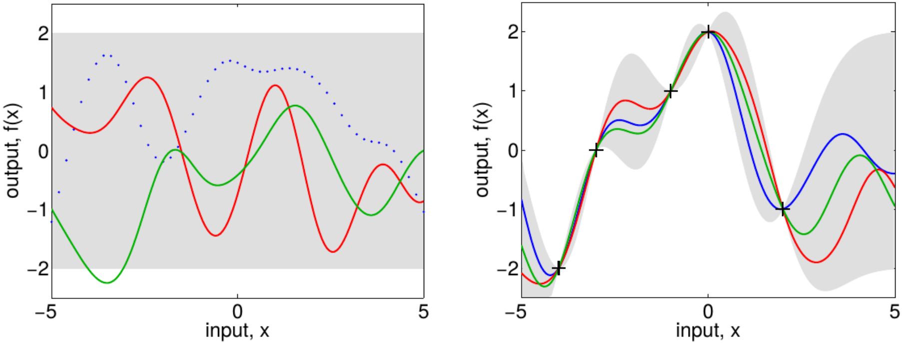
Gaussian Process Prior:

$$f(x)|M \sim \mathcal{GP}(m(x) = 0, k(x, x'))$$

Leading to Gaussian Process Posterior:

$$f(x)|\mathbf{x}, \mathbf{y}, M \sim \mathcal{GP}(m_{\text{post}}(x) = k(x, \mathbf{x})[K(\mathbf{x}, \mathbf{x}) + \sigma^2_{noise} I]^{-1}\mathbf{y},$$
$$k_{\text{post}}(x, x') = k(x, x') - k(x, \mathbf{x})[K(\mathbf{x}, \mathbf{x}) + \sigma^2_{noise} I]^{-1} k(\mathbf{x}, x'))$$

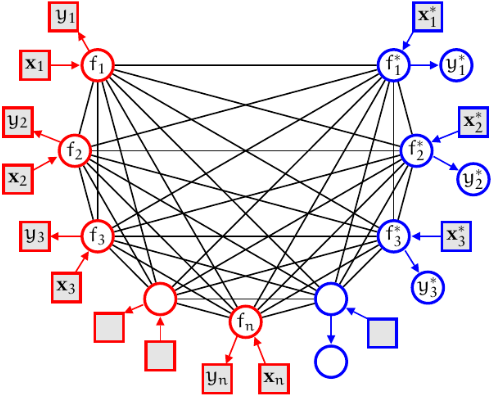
Prior and Posterior for $\mathcal{GP}$ Learning



Gaussian Process Predictive Distribution:

$$p(y^*|x^*, \mathbf{x}, \mathbf{y}) \sim \mathcal{N}(k(x^*, \mathbf{x})[K + \sigma^2_{noise}]^{-1}\mathbf{y},$$
$$k(x*, x*) - k(x^*, \mathbf{x})[K + \sigma^2_{noise} I]^{-1} k(\mathbf{x}, x^*))$$

Graphical Model for Gaussian Processes



- All pairs of latent variables are connected.
- Predictions y^* depend only on corresponding latent f^*.
- Adding x_m^*, y_m^*, f_m^* does not influence the distribution. Guaranteed by marginalization property.

Explains why inference uses finite amount of computation!

Interpretation of $\mathcal{GP}$ Inference

Recalling predictive distribution:

$$p(y^*|x^*, \mathbf{x}, \mathbf{y}) \sim \mathcal{N}(k(x^*, \mathbf{x})[K + \sigma^2_{noise}]^{-1}\mathbf{y},$$
$$k(x*, x*) - k(x^*, \mathbf{x})[K + \sigma^2_{noise} I]^{-1} k(\mathbf{x}, x^*))$$

Interpretation of $\mathcal{GP}$ Inference

Recalling predictive distribution:

$$p(y^*|x^*, \mathbf{x}, \mathbf{y}) \sim \mathcal{N}(k(x^*, \mathbf{x})[K + \sigma^2_{noise}]^{-1}\mathbf{y},$$
$$k(x*, x*) - k(x^*, \mathbf{x})[K + \sigma^2_{noise}I]^{-1}k(\mathbf{x}, x^*))$$

Mean can be linearly represented as:

$$\boldsymbol{\mu}(x^*) = k(x^*, \mathbf{x})[K + \sigma^2_{noise}]^{-1}\mathbf{y} = \sum_{i=1}^{n} \beta_i y_i = \sum_{i=1}^{n} \alpha_i k(x^*, x_i)$$

Interpretation of $\mathcal{GP}$ Inference

Recalling predictive distribution:

$$p(y^*|x^*, \mathbf{x}, \mathbf{y}) \sim \mathcal{N}(k(x^*, \mathbf{x})[K + \sigma^2_{noise}]^{-1}\mathbf{y},$$
$$k(x*, x*) - k(x^*, \mathbf{x})[K + \sigma^2_{noise}I]^{-1}k(\mathbf{x}, x^*))$$

Mean can be linearly represented as:

$$\boldsymbol{\mu}(x^*) = k(x^*, \mathbf{x})[K + \sigma^2_{noise}]^{-1}\mathbf{y} = \sum_{i=1}^{n} \beta_i y_i = \sum_{i=1}^{n} \alpha_i k(x^*, x_i)$$

Variance is composed of two terms:

$$\mathbf{\Sigma}x^* = \underbrace{k(x*, x*)}_{\textbf{prior variance}} - \underbrace{k(x^*, \mathbf{x})[K + \sigma^2_{noise}I]^{-1}k(\mathbf{x}, x^*)}_{\textbf{variance by data}}$$

Note that the variance is independent of observed outputs y.

Optimizing Marginal Likelihood

$$\log p(\mathbf{y}|\mathbf{x}, M) = -\frac{1}{2}\mathbf{y}^T K^{-1}\mathbf{y} - \frac{1}{2}\log|K| - \frac{n}{2}\log(2\pi)$$

is a combination of data fit and complexity penalty terms. Occam's razor is automatic!

Optimizing Marginal Likelihood

$$\log p(\mathbf{y}|\mathbf{x}, M) = -\frac{1}{2}\mathbf{y}^T K^{-1}\mathbf{y} - \frac{1}{2}\log|K| - \frac{n}{2}\log(2\pi)$$

is a combination of data fit and complexity penalty terms. Occam's razor is automatic!

Learning in Gaussian process models involves finding:

- Form of covariance matrix
- Unknown hyperparameter values θ

Optimizing Marginal Likelihood

$$\log p(\mathbf{y}|\mathbf{x}, M) = -\frac{1}{2}\mathbf{y}^T K^{-1}\mathbf{y} - \frac{1}{2}\log|K| - \frac{n}{2}\log(2\pi)$$

is a combination of data fit and complexity penalty terms. Occam's razor is automatic!

Learning in Gaussian process models involves finding:

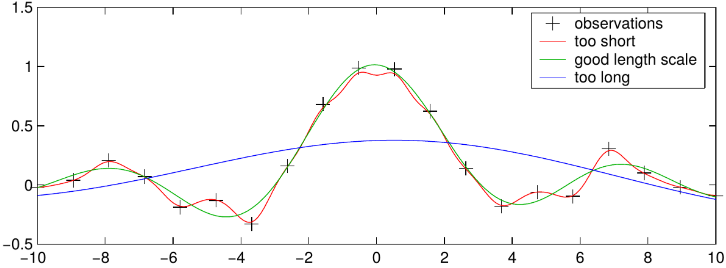
- Form of covariance matrix
- Unknown hyperparameter values θ

This can be done by optimizing the marginal likielihood:

$$\frac{\partial \log p(\mathbf{y}|\mathbf{x}, \theta, M)}{\partial \theta_j} = \frac{1}{2}\mathbf{y}^T K^{-1}\frac{\partial K}{\partial \theta_j}K^{-1}\mathbf{y} - \frac{1}{2}\text{trace}\left(K^{-1}\frac{\partial K}{\partial \theta_j}\right)$$

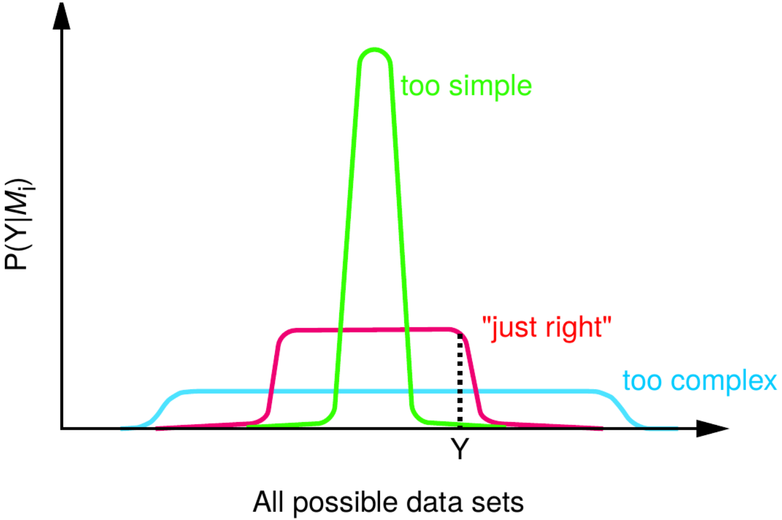
Example: Length Parameter Learning

Covariance function: $k(x,x') = \nu^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right) + \sigma^2_{noise}\delta_{xx'}$



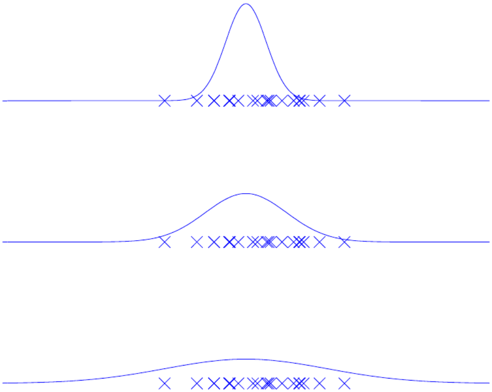
Posterior mean function is plotted for 3 different length scales, green curve maximizes marginal likelihood. Although exact fit for data can be found, marginal likelihood does not favour this!

Why does Bayesian Inference work?: Occam's Razor



Analogous Example

Task: Fitting variance, σ^2, of zero-mean Gaussian to n scalar observations.



Log likelihood is $\log p(y|\mu, \sigma^2) = -\frac{1}{2} \sum \frac{(y_i - \mu)}{\sigma^2} - \frac{n}{2} \log(\sigma^2) - \frac{n}{2} \log(2\pi)$

Covariance Function for Linear Models

Consider the class of linear functions:

$$f(x) = ax + b, \text{ where } a \sim \mathcal{N}(0, \alpha), \text{ and } b \sim \mathcal{N}(0, \beta)$$

Covariance Function for Linear Models

Consider the class of linear functions:

$$f(x) = ax + b, \text{ where } a \sim \mathcal{N}(0, \alpha), \text{ and } b \sim \mathcal{N}(0, \beta)$$

We can compute the mean as:

$$\mu(x) = E[f(x)] = \int \int f(x)p(a)p(b)dadb = \int axp(a)da + \int bp(b)db = 0$$

Covariance Function for Linear Models

Consider the class of linear functions:

$$f(x) = ax + b, \text{ where } a \sim \mathcal{N}(0, \alpha), \text{ and } b \sim \mathcal{N}(0, \beta)$$

We can compute the mean as:

$$\mu(x) = E[f(x)] = \int \int f(x)p(a)p(b)dadb = \int axp(a)da + \int bp(b)db = 0$$

and covariance function as:

$$k(x, x') = E[(f(x) - 0)(f(x') - 0)] = \int \int (ax + b)(ax' + b)p(a)p(b)dadb$$
$$= \int a^2 xx' p(a)da + \int b^2 p(b)db + (x + x') \int abp(a)p(b)dadb = \alpha xx' + \beta$$

Regression with Basis Functions

Consider the class of linear functions:

$$
\begin{aligned}
f(x) &= \lim_{n \leftarrow \infty} \frac{1}{n} \sum_i \gamma_i \exp(-(x - i/n)^2), \text{ where } \gamma_i \sim \mathcal{N}(0,1), \forall i \\
&= \int_{-\infty}^{\infty} \gamma(u) \exp(-(x-u)^2) du, \text{ where } \gamma(u) \sim \mathcal{N}(0,1), \forall u
\end{aligned}
$$

Regression with Basis Functions

Consider the class of linear functions:

$$\begin{aligned} f(x) &= \lim_{n \leftarrow \infty} \frac{1}{n} \sum_i \gamma_i \exp(-(x - i/n)^2), \text{ where } \gamma_i \sim \mathcal{N}(0,1), \forall i \\ &= \int_{-\infty}^{\infty} \gamma(u) \exp(-(x-u)^2) du, \text{ where } \gamma(u) \sim \mathcal{N}(0,1), \forall u \end{aligned}$$

Mean function is:

$$\mu(x) = E[f(x)] = \int_{-\infty}^{\infty} \exp(-(x-u)^2) \int_{-\infty}^{\infty} \gamma p(\gamma) d\gamma du = 0$$

Regression with Basis Functions

Covariance function is:

$$E[f(x)f(x')] = \int \exp(-(x-u)^2 - (x'-u)^2)du$$
$$= \int \exp\left(-2\left(u - \frac{x+x'}{2}\right)^2 + \frac{(x-x')^2}{2} - x^2 - x'^2\right)du$$
$$\propto \exp\left(-\frac{(x-x')^2}{2}\right)$$

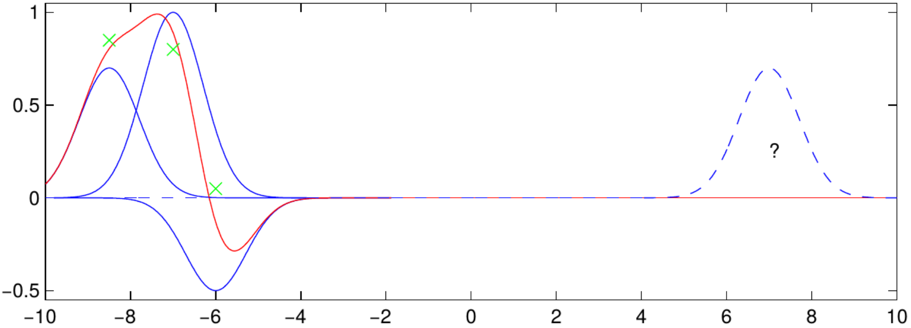
Regression with Basis Functions

Covariance function is:

$$E[f(x)f(x')] = \int \exp(-(x-u)^2 - (x'-u)^2)du$$
$$= \int \exp\left(-2\left(u - \frac{x+x'}{2}\right)^2 + \frac{(x-x')^2}{2} - x^2 - x'^2\right)du$$
$$\propto \exp\left(-\frac{(x-x')^2}{2}\right)$$

Using squared exponential covariance function is equivalent to regression using infinitely many bell-shaped basis functions!

Using finite basis functions can be dangerous!

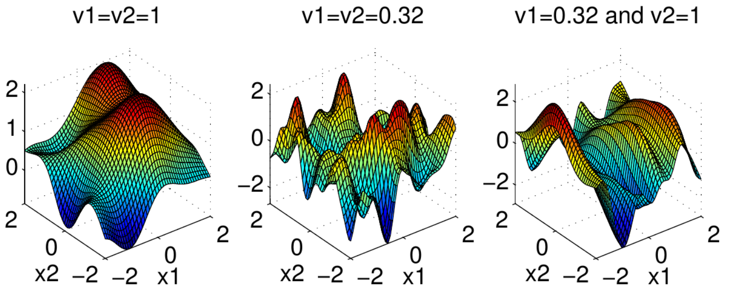


Model Selection in Practice

Two types of selection: *form* and *parameters* of covariance function.

Hyperparameters form a herarchical model. Eg, ARD Covariance Function:

$$k(x, x') = \nu_0^2 \exp\left(-\sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{2\nu_d^2}\right), \text{ hyperparameters } \theta = (\nu_0, \ldots, \sigma_{noise}^2)$$



Rational Quadratic (RQ) Covariance Function

$$k_{RQ}(r) = \left(1 + \frac{r^2}{2\alpha l^2}\right)^{-\alpha}$$

with $\alpha, l > 0$ can be seen as an infinite sum of squared exponential (SE) covariance functions with differen length-scales.

Rational Quadratic (RQ) Covariance Function

$$k_{RQ}(r) = \left(1 + \frac{r^2}{2\alpha l^2}\right)^{-\alpha}$$

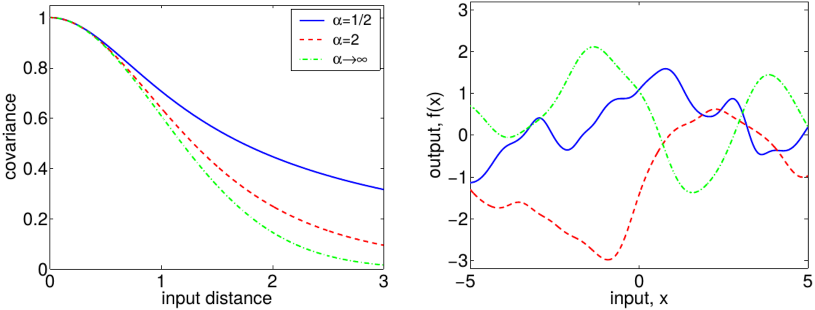
with $\alpha, l > 0$ can be seen as an infinite sum of squared exponential (SE) covariance functions with differen length-scales.

Using $\tau = l^2$ and $p(\tau|\alpha, \beta) \propto \tau^{\alpha-1}\exp(-\alpha\tau/\beta)$:

$$k_{RQ}(r) = \int p(\tau|\alpha, \beta) k_{SE}(r|\tau) d\tau$$

$$\propto \int \tau^{\alpha-1}\exp\left(-\frac{\alpha\tau}{\beta}\right)\exp\left(-\frac{\tau r^2}{2}\right) d\tau \propto \left(1 + \frac{r^2}{2\alpha l^2}\right)^{-\alpha}$$

Rational Quadratic Covariance Function



Limit $\alpha \leftarrow \infty$ of the RQ covariance function is SE.

Matern Covariance Function

$$k(x, x') = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left[\frac{\sqrt{2\nu}}{l} |x - x'| \right]^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu} l}{|} x - x'| \right)$$

where K_ν is a Bessel function of order ν, and l is the length scale.

Matern Covariance Function

$$k(x, x') = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left[\frac{\sqrt{2\nu}}{l} |x - x'| \right]^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu} l}{|} x - x'| \right)$$

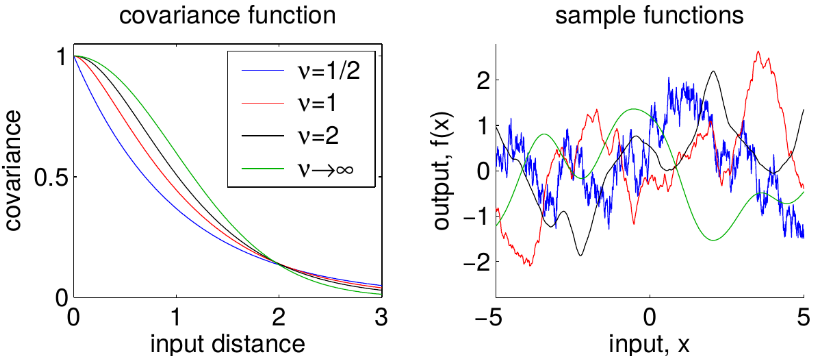
where K_ν is a Bessel function of order ν, and l is the length scale.

Samples of Matern forms are $\lfloor \nu - 1 \rfloor$ times differentiable.

- $k_{\nu=5/2}(r) = \left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}\right) \exp\left(-\frac{\sqrt{5}r}{l}\right)$: Twice differentiable
- $k_{\nu\leftarrow\infty}(r) = \exp\left(-\frac{r^2}{2l^2}\right)$: Smooth (Infinite differentiable)

Matern Covariance Function

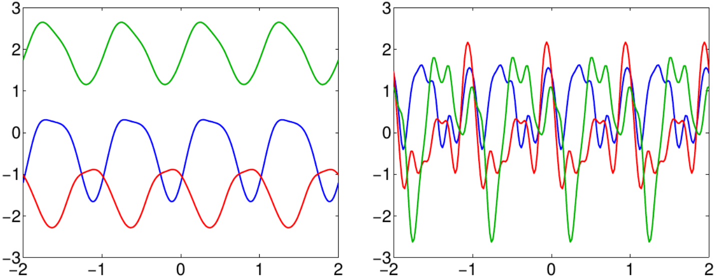
Univariate Matern covariance functions with unit length scale and unit variance:



Periodic Covariance Function

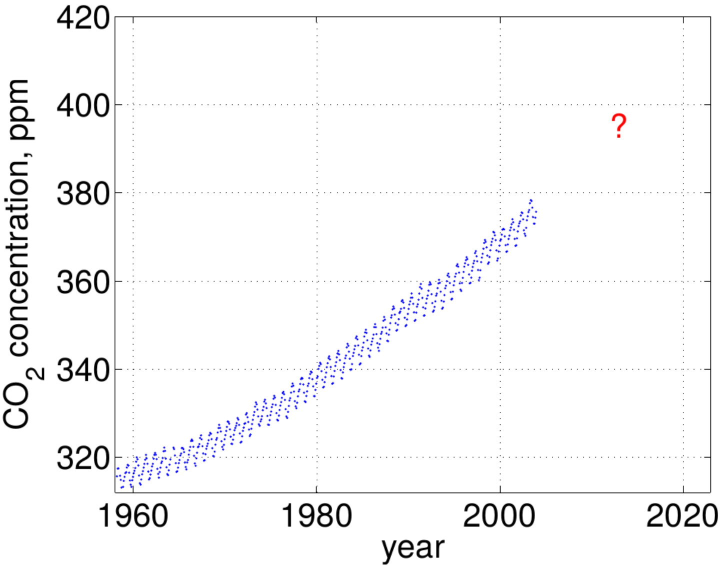
Periodic covariance functions can be obtained by mapping x to $u = (\sin(x), \cos(x))^T$ and combine with SE covariance function:

$$k_{periodic}(x, x') = \exp\left(-\frac{2\sin^2(\pi(x - x'))}{l^2}\right)$$



3 random samples with: left $l > 1$ and right $l < 1$

Prediction Problem



Covariance Functions

- long term smooth trend (squared exponential)

$$k_1(x, x') = \theta_1^2 \exp\left(\frac{(x - x')^2}{\theta_2^2}\right)$$

- seasonal trend (quasi-periodic smooth)

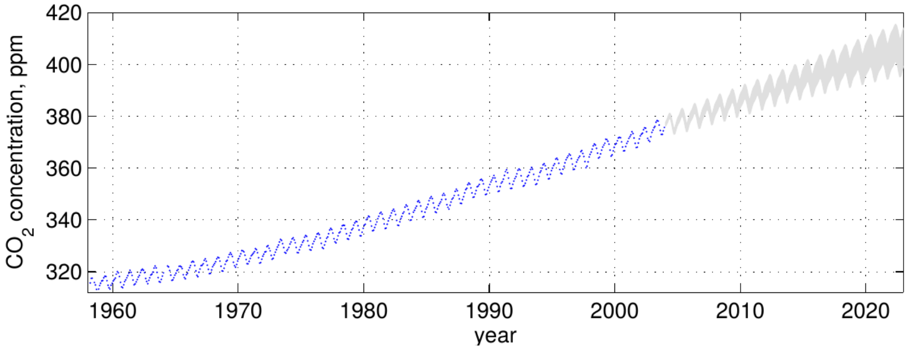
$$k_2(x, x') = \theta_3^2 \exp\left(-\frac{2\sin^2(\pi(x - x'))}{\theta_5^2}\right) \times \exp\left(\frac{(x - x')^2}{2\theta_4^2}\right)$$

- short and medium term anomaly (rational quadratic)

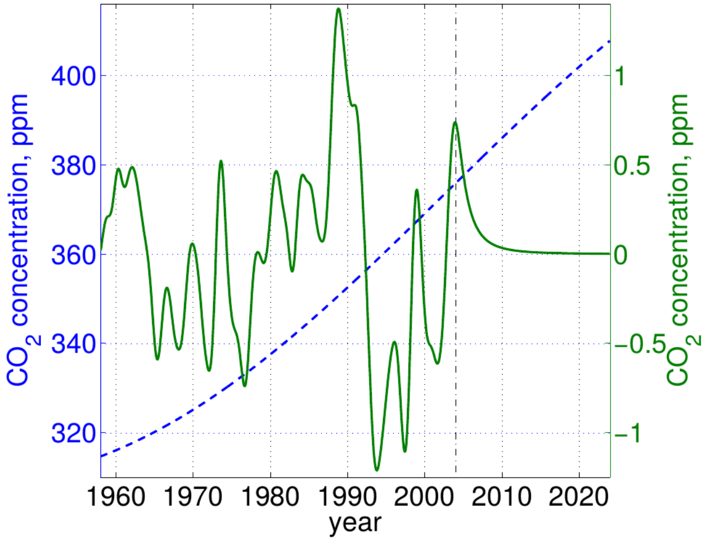
$$k_3(x, x') = \theta_6^2 \left(1 + \frac{(x - x')^2}{2\theta_8 \theta_7^2}\right)^{-\theta_8}$$

$$k(x, x') = k_1(x, x') + k_2(x, x') + k_3(x, x') + \text{noise kernel}$$

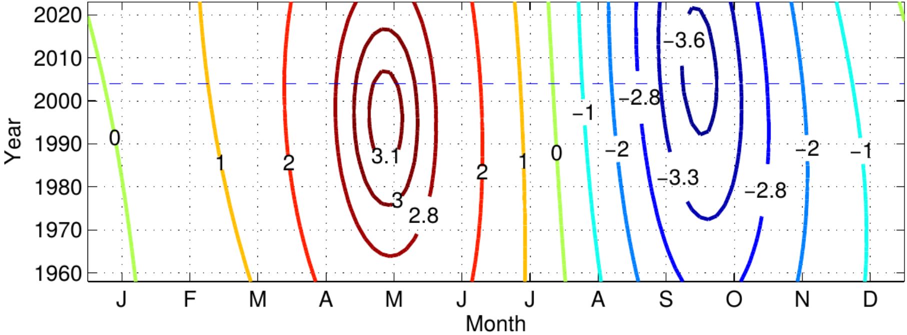
Carbon Dioxide Predictions



Long and Medium-term Predictions



Mean Seasonal Component



Conclusions

Complex non-linear inference problems can be solved by manipulating plain old Gaussian Distributions

- Bayesian inference is tractable for GP Regression
- Predictions are probabilistic
- Comparison of different models possible via Marginal Likelihood

Conclusions

Complex non-linear inference problems can be solved by manipulating plain old Gaussian Distributions

- Bayesian inference is tractable for GP Regression
- Predictions are probabilistic
- Comparison of different models possible via Marginal Likelihood

Scope for research:

- Interesting covariance functions
- Search for efficient approximations and sparse methods
- Application to high-dimensional data (Deep Learning)