Gaussian Processes

Roy Stegeman

Definition

Gaussian Distribution

$\mathcal{N}(\mu, \Sigma)$

- Distribution over scalars or vectors
- Fully specified by mean and covariance

Gaussian Process

$\mathcal{GP}\left(m(\mathbf{x}), k\left(\mathbf{x}, \mathbf{x}'\right)\right)$

- Distribution over functions
- Fully specified by a mean function and covariance function

Supervised Machine Learning

Problem: Determine a continuous input to output mapping from discrete training data

Different output characteristics:

- Classification problem: Reading handwritten digits. Stars or galaxies?
- Regression problem (this talk): Distribution of gold based on boreholes

Supervised Machine Learning

Problem: Determine a continuous input to output mapping from discrete training data

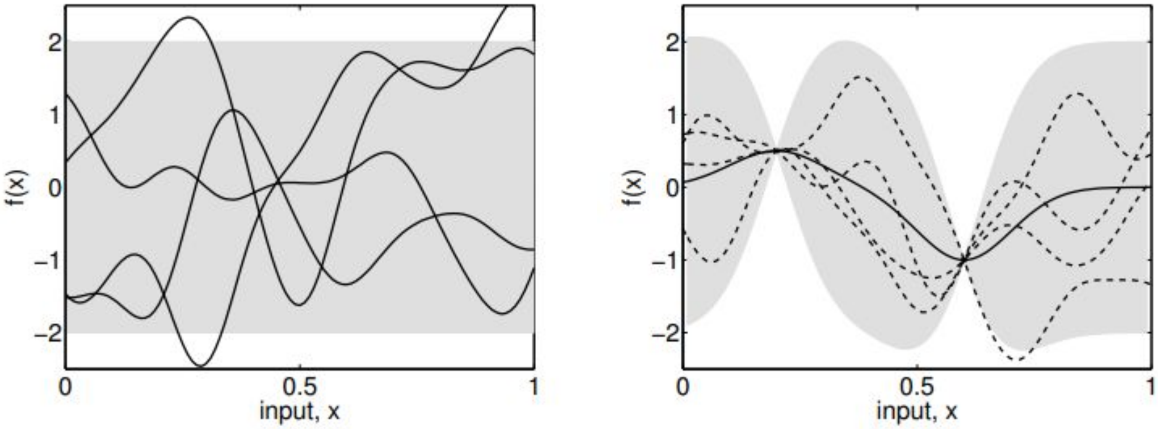
Common approaches:

- Parametric models: assume a functional form (e.g. linear function)
    - **Problem**: choosing the correct functional form
- Non-parametric models: give a prior probability to every possible function
    - **Problem**: how do we deal with an infinite set of functions?
    - **Solution**: Gaussian process

Bayesian modelling in pictures

- The covariance function defines the properties in function space
- Datapoints fix the function at certain locations

prior



posterior

Source: Rasmussen & Williams

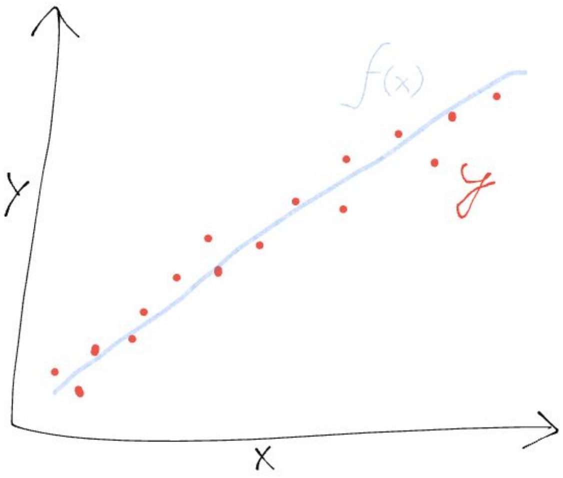
Left: samples drawn from prior distribution.
Right: samples drawn from the posterior after observing two datapoints

Regression: weight space view

Bayesian Linear Regression

A linear model:

- Easy implementation an interpretation
- Limited flexibility



Consider standard linear regression model

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{w},$$

with weights $\mathbf{w}$, and Gaussian noise

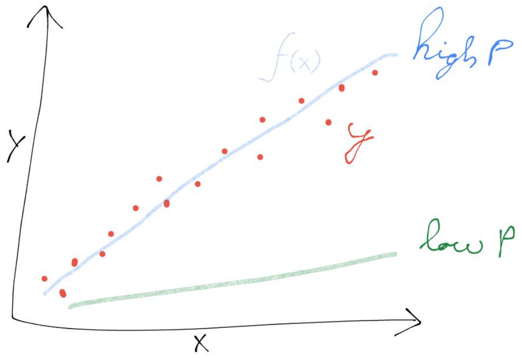
$$y = f(\mathbf{x}) + \epsilon$$

$$\epsilon \sim \mathcal{N}(0, \sigma^2)$$

Bayesian Linear Regression

The **likelihood**, probability density of observation given the parameters

$$
\begin{aligned}
p(\mathbf{y} \mid X, \mathbf{w}) &= \prod_{i=1}^{n} p\left(y_i \mid \mathbf{x}_i, \mathbf{w}\right) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{\left(y_i - \mathbf{x}_i^\top \mathbf{w}\right)^2}{2\sigma_n^2}\right) \\
&= \frac{1}{\left(2\pi\sigma_n^2\right)^{n/2}} \exp\left(-\frac{1}{2\sigma_n^2}\left|\mathbf{y} - X^\top \mathbf{w}\right|^2\right) \\
&= \mathcal{N}\left(X^\top \mathbf{w}, \sigma_n^2 I\right)
\end{aligned}
$$



Bayesian Linear Regression

In the Bayesian formalism the model includes a **prior** $p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, \Sigma_p)$

Posterior parameter distribution from Bayes' rule can be calculated

$$p(\mathbf{w} \mid \mathbf{y}, X) = \frac{p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|X)}, \quad \text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}$$

Where the marginal likelihood is

$$p(\mathbf{y} \mid X) = \int p(\mathbf{y} \mid X, \mathbf{w}) p(\mathbf{w}) d\mathbf{w}$$

Because all components are Gaussian we can find explicitly that

$$p(\mathbf{w} \mid X, \mathbf{y}) \sim \mathcal{N}\left(\frac{1}{\sigma_n^2} A^{-1} X\mathbf{y}, A^{-1}\right), \quad A = \sigma_n^{-2} X X^\top + \Sigma_p^{-1}$$

Bayesian Linear Regression

We can now make predictions by averaging over **all possible parameter values**:

$$p(f_* \mid \mathbf{x}_*, X, \mathbf{y}) = \int p(f_* \mid \mathbf{x}_*, \mathbf{w})\, p(\mathbf{w} \mid X, \mathbf{y}) d\mathbf{w}$$
$$= \mathcal{N}\left(\frac{1}{\sigma_n^2}\mathbf{x}_*^\top A^{-1} X\mathbf{y}, \mathbf{x}_*^\top A^{-1}\mathbf{x}_*\right).$$

which is again Gaussian.

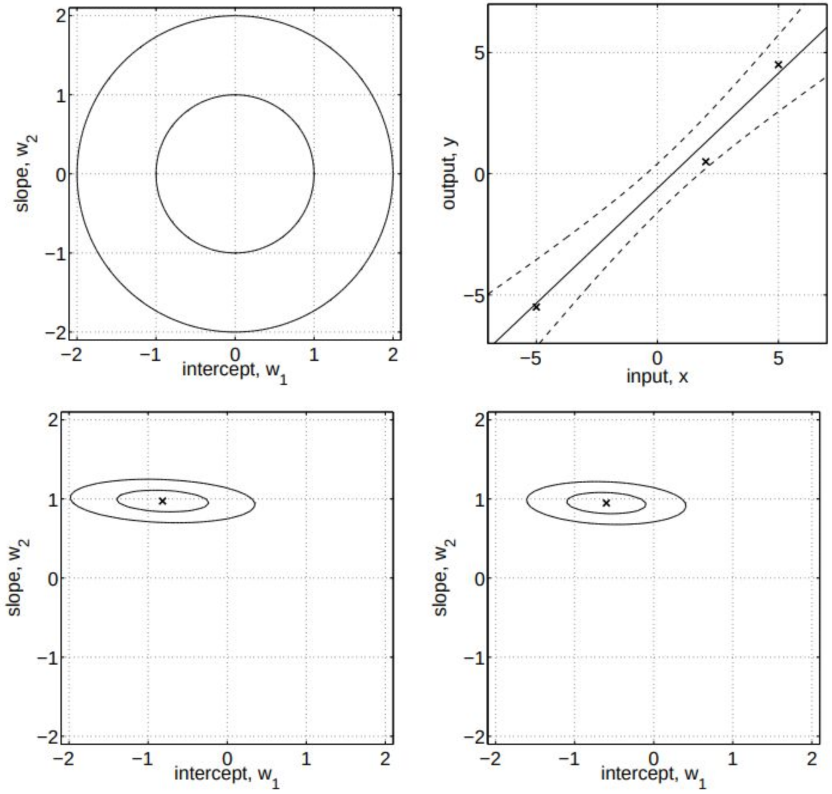
- Central value multiplied by test input
- Variance is quadratic in the test input

Bayesian Linear Regression

$$f = w_1 + w_2 x$$

prior

training data
predictive distribution



likelihood

posterior

A set of basis functions

(Bayesian) linear model is limited, instead we can generalize to **features** of $\mathbf{x}$

by changing the model to

$$f(\mathbf{x}) = \phi(\mathbf{x})^\top \mathbf{w}$$

where $\phi(\mathbf{x})$ projects the input onto a space of basis functions, e.g. polynomials:

$$\phi(\mathbf{x}) = (1, x, x^2, x^3, \ldots)^T$$

Kernel trick

The analysis is analogous to the linear model with $X \to \Phi(X)$

$$p(f_* \mid \mathbf{x}_*, X, \mathbf{y}) \sim \mathcal{N}\left(\frac{1}{\sigma_n^2}\phi(\mathbf{x}_*)^\top A^{-1}\Phi\mathbf{y}, \phi(\mathbf{x}_*)^\top A^{-1}\phi(\mathbf{x}_*)\right) \quad A = \sigma_n^{-2}\Phi\Phi^\top + \Sigma_p^{-1}$$

Feature space always enters in the form $\phi(\mathbf{x})^\top \Sigma_p \phi(\mathbf{x}')$

Let us define a covariance function or **kernel**

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \Sigma_p \phi(\mathbf{x}') = \phi(\mathbf{x})\Sigma_p^{1/2} \cdot \Sigma_p^{1/2}\phi(\mathbf{x}')$$

The covariance function must be

- Positive semi-definite
- Symmetric

Covariance function

Consider the commonly used **squared exponential** covariance function:

$$\mathrm{cov}\left(f\left(\mathbf{x}_p\right), f\left(\mathbf{x}_q\right)\right) = k\left(\mathbf{x}_p, \mathbf{x}_q\right) = \exp\left(-\frac{1}{2}\left|\mathbf{x}_p - \mathbf{x}_q\right|^2\right)$$

- Positive semi-definite
- Symmetric
- Nearby points are highly correlated
- Consistency: uncertainty cannot increase if data is added

Regression:
function space view

Function space view

We can derive identical results directly in **function space**

Definition: *A GP is a collection of random variables, any finite number of which have a joint Gaussian distribution*

A Gaussian process is completely specified by its mean function and covariance function:

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k\left(\mathbf{x}, \mathbf{x}'\right)\right)$$

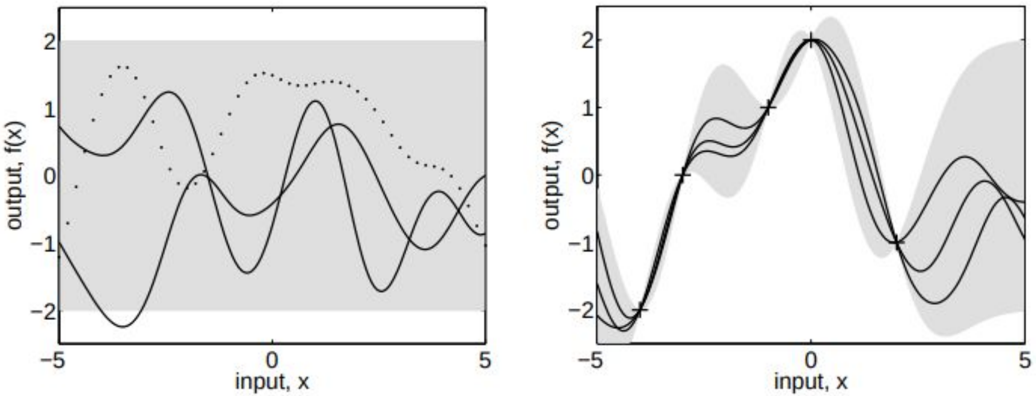
We can characterize a large number of functions with a GP:

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\ k(\mathbf{x}, \mathbf{x}) &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}) - m(\mathbf{x}))] \end{aligned}$$

Generating functions

The specification of the covariance function implies a distribution over functions. Consider random Gaussian vectors with a covariance matrix defined by X_*

$$\mathbf{f}_* \sim \mathcal{N}\left(\mathbf{0}, K\left(X_*, X_*\right)\right)$$



Predictions

We will mainly be interested in predictions instead of generating functions.

For a GP with zero mean and covariance $\mathrm{K}(X, X) + \sigma_n^2 I$ the joint distribution of training and test outputs is:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right)$$

The main predictive equations for GP regression is:

$$\mathbf{f}_* \mid X, \mathbf{y}, X_* \sim \mathcal{N}\left(\overline{\mathbf{f}}_*, \operatorname{cov}(\mathbf{f}_*)\right)$$

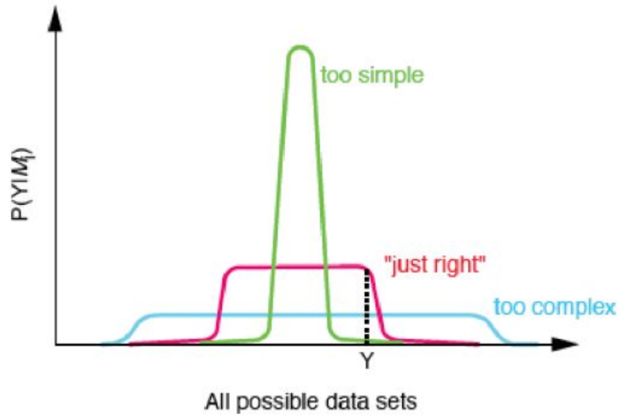
$$\overline{\mathbf{f}}_* = K(X_*, X)\left[K(X, X) + \sigma_n^2 I\right]^{-1} \mathbf{y}$$

$$\operatorname{cov}(\mathbf{f}_*) = K(X_*, X_*) - K(X_*, X)\left[K(X, X) + \sigma_n^2 I\right]^{-1} K(X, X_*)$$

Model selection

Model selection

Bayesian evidence is the probability of the data, given the model



While complex models can account for many datasets, the resulting evidence will be smaller.

Hyperparameters

A commonly used covariance function is

$$k\left(x_p, x_q\right) = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}\left(x_p - x_q\right)^2\right) + \sigma_n^2 \delta_{pq}$$

with hyperparameters

$l =$ characteristic length scale

$\sigma_f =$ signal variance

$\sigma_n =$ noise variance

Hyperparameter tuning

Maximize the *marginal* likelihood

$$p(\mathbf{y} \mid X) = \int p(\mathbf{y} \mid \mathbf{f}, X) p(\mathbf{f} \mid X) d\mathbf{f}$$

Or more precisely the log thereof:

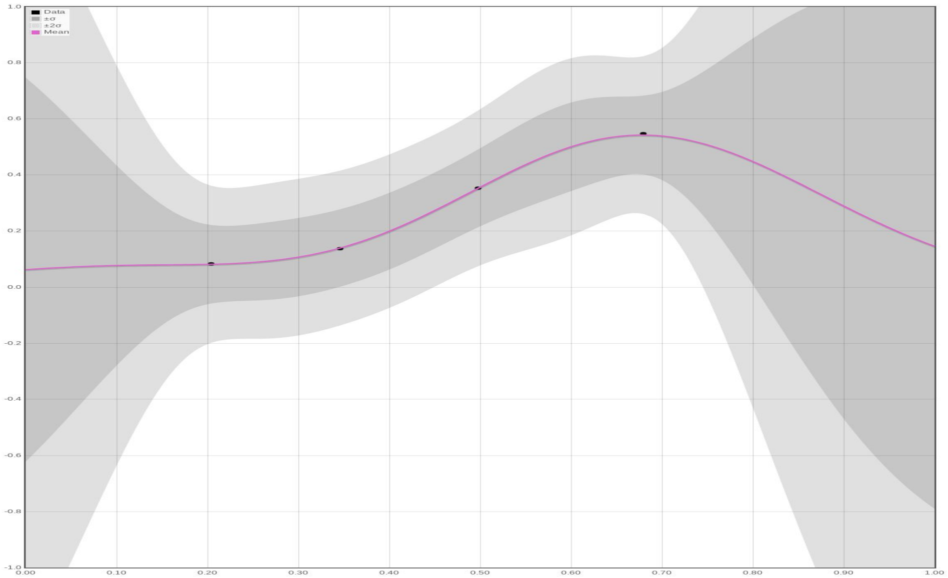
$$\log p(\mathbf{y} \mid X) = -\frac{1}{2}\mathbf{y}^\top \left(K + \sigma_n^2 I\right)^{-1}\mathbf{y} - \frac{1}{2}\log|K + \sigma_n^2 I| - \frac{n}{2}\log 2\pi$$

Prediction penalty: $-\frac{1}{2}\mathbf{y}^\top \left(K + \sigma_n^2 I\right)^{-1}\mathbf{y}$

Complexity penalty: $-\frac{1}{2}\log|K + \sigma_n^2 I|$

Hyperparameters - demonstration

[Gaussian process demo]



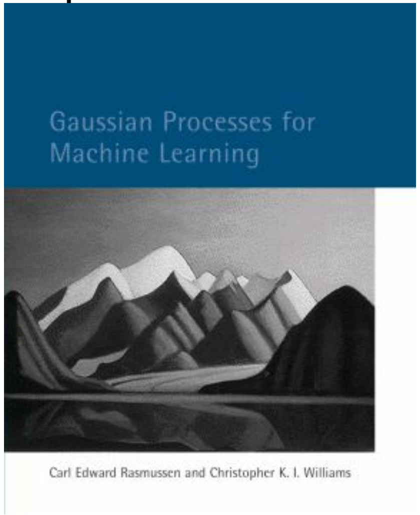
Resources

Literature:

- Gaussian Processes for Machine Learning, Rasmussen and Williams, Online version (this talk)
- Pattern Recognition and Machine Learning, Bishop

Software:

- GPML (Matlab, Rasmussen and Williams)
- Scikit-learn
- GPy (Python)
- GPyTorch (PyTorch implementation)
- GPflow (Tensorflow implementation)



Thank you!

Distribution over functions

One can generate a random sample $\boldsymbol{X}$ from a n-dimensional Gaussian distribution $N(\mu, \Sigma)$ as follows:

$$\boldsymbol{X} = \mu + AZ$$

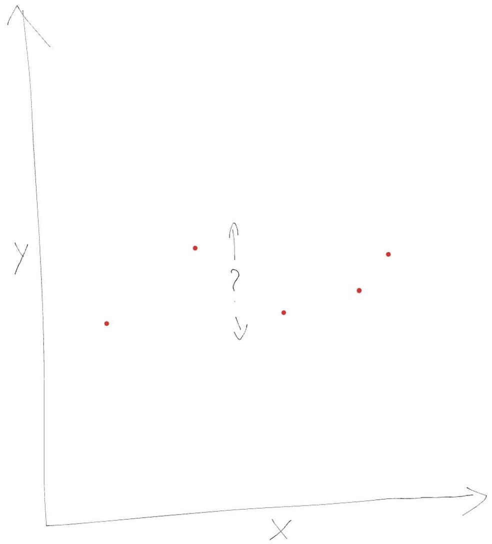
with a random n-dimensional vector Z and a matrix A that satisfies

$$\Sigma = AA^T$$

which can be found using **Cholesky decomposition**

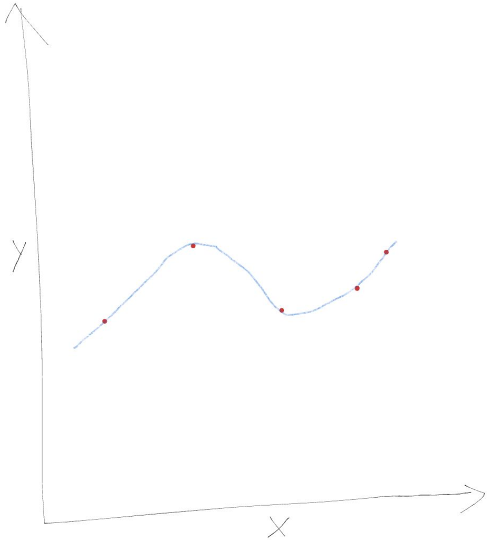
Non-linear regression

Consider noiseless data



Non-linear regression

What is the parametric form?



Non-linear regression

A Gaussian Process will provide the uncertainties without assuming a parametric form

How exactly does this work?