Gaussian Processes

Perry Groot

Radboud University Nijmegen

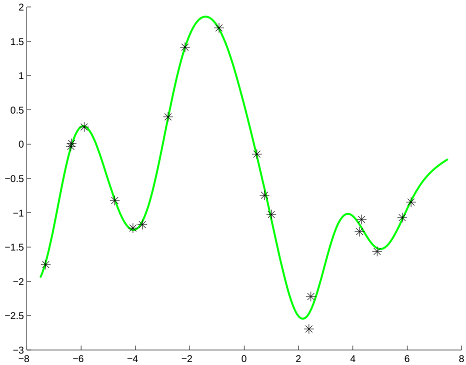
perry@cs.ru.nl

SIKS Course Computational Intelligence

11 & 12 March 2010

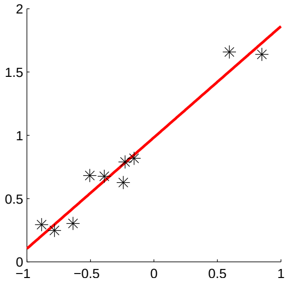
Supervised Learning: Regression

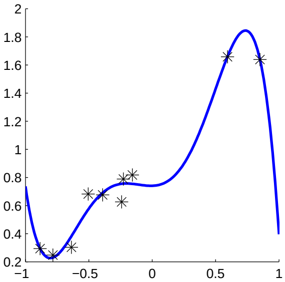
- Data $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) | i = 1, \ldots, n\}$;
- Input space $\mathcal{X} \subseteq \mathbb{R}^d$; Output space $\mathcal{Y} \subseteq \mathbb{R}$
- Goal predict functional relation $f : \mathcal{X} \to \mathcal{Y}$



Supervised Learning: Regression

- *parametric* regression: $f(\boldsymbol{x}; \boldsymbol{w})$
  - Linear model: $f(\boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{w}^T\boldsymbol{x} = \sum_{j=0}^{d} w_j x_j$
  - Polynomial model: $f(x; \boldsymbol{w}) = \sum_{j=0}^{M} w_j x^j$
- Loss function: $\mathcal{L}(\boldsymbol{w}) = \sum_{i=1}^{n}(y_i - f(\boldsymbol{x}_i; \boldsymbol{w}))^2$





Supervised Learning: Regression

There are a couple of disadvantages:

- Lack of error bars on predictions
- Problem of overfitting

Overfitting can be avoided by using simpler models, but its predictive performance may be poor.

Bayesian regression

Bayes' rule to obtain a *posterior* distribution:

$$p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X}, \sigma^2) = \frac{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}, \sigma^2)p(\boldsymbol{w})}{p(\boldsymbol{y}|\boldsymbol{X}, \sigma^2)}$$

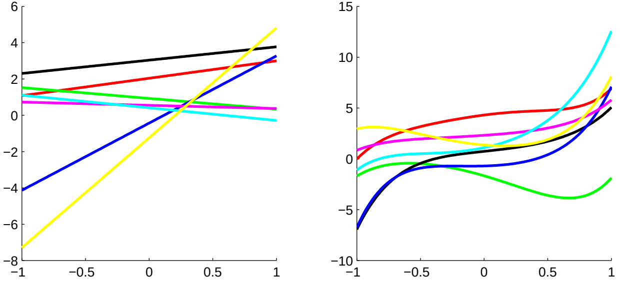
predictive distribution

$$p(y_*|\boldsymbol{x}_*, \boldsymbol{y}, \boldsymbol{X}, \sigma^2) = \int p(y_*|\boldsymbol{x}_*, \boldsymbol{w}, \sigma^2)p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X}, \sigma^2)\,\mathrm{d}\boldsymbol{w}$$

- *All* parameters contribute to a prediction
- Good generalization performance and robust to overfitting
- Allows for error bars on predictions

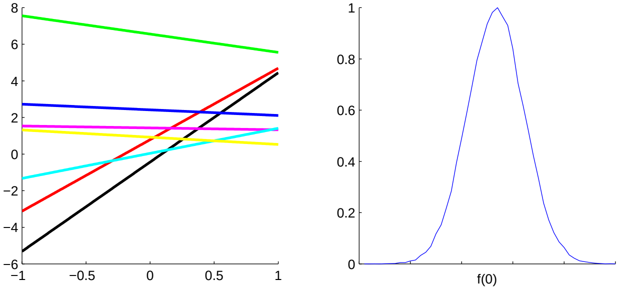
Weightspace view

Assuming a probability distribution over $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ leads to a probability distribution over functions $f(\cdot; \boldsymbol{w})$



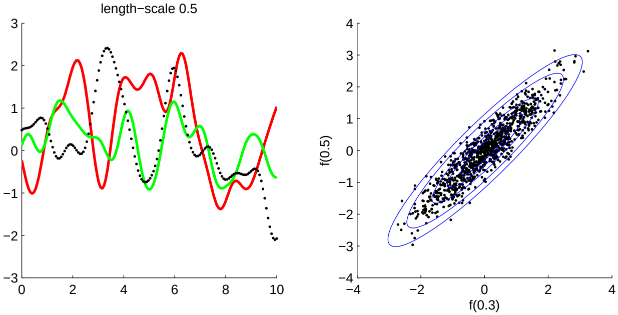
Weightspace view

Which leads to a distribution at each test point



Functionspace view

Instead of taking a distribution over weights, we can also directly consider distributions over functions. We will consider the following model $y_i = f_i + \epsilon_i$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$



Gaussian Processes

A Gaussian process (GP) is collection of random variables $\{f_i\}$ with the property that the joint distribution of any finite subset has a joint Gaussian distribution.

A GP specifies a probability distribution over functions $f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$ and is fully specified by its mean function $m(\boldsymbol{x})$ and covariance (or kernel) function $k(\boldsymbol{x}, \boldsymbol{x}')$.

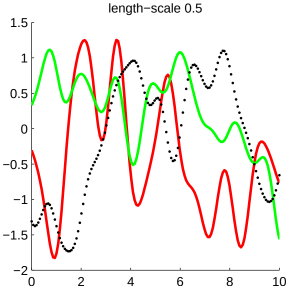
Typically $m(\boldsymbol{x}) = \mathbf{0}$, which gives

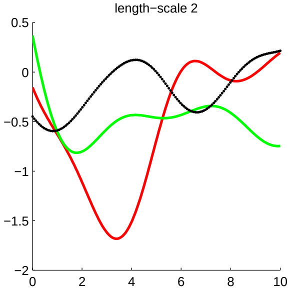
$$\{f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_l)\} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K}) \text{ with } K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Squared exponential (or Gaussian) covariance function:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2\ell^2}\sum_{n=1}^{N}(x_n - x'_n)^2\right)$$

where ℓ is a length-scale parameter denoting how quickly the functions are to vary.





A priori, given data $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{y}\}$ with $\boldsymbol{y} = f(\boldsymbol{X})$ and test points $\boldsymbol{X}_*$ we have

$$\begin{bmatrix} f(\boldsymbol{X}) \\ f(\boldsymbol{X}_*) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\boldsymbol{X},\boldsymbol{X}) & K(\boldsymbol{X},\boldsymbol{X}_*) \\ k(\boldsymbol{X}_*,\boldsymbol{X}) & K(\boldsymbol{X}_*,\boldsymbol{X}_*) \end{bmatrix}\right)$$

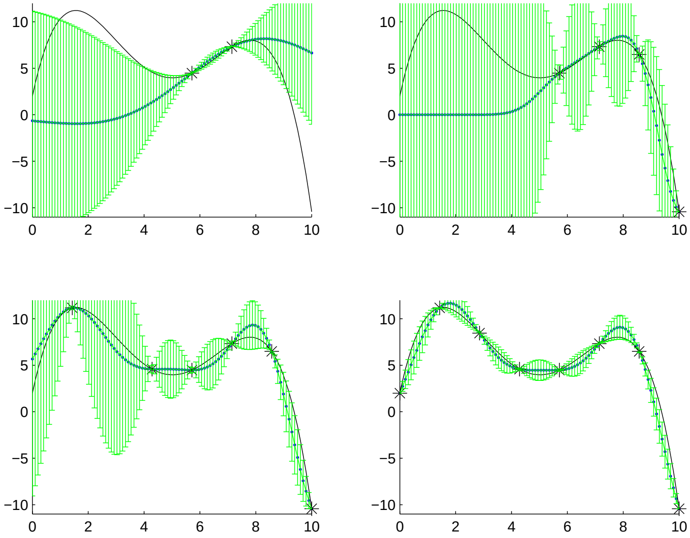
and after conditioning

$$f(\boldsymbol{X}_*)|\boldsymbol{X}_*, \boldsymbol{X}, \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$$

with

$$\begin{aligned} \boldsymbol{\mu} &= K(\boldsymbol{X}_*,\boldsymbol{X})K(\boldsymbol{X},\boldsymbol{X})^{-1}\boldsymbol{y} \\ \Sigma &= K(\boldsymbol{X}_*,\boldsymbol{X}_*) - K(\boldsymbol{X}_*,\boldsymbol{X})\underbrace{K(\boldsymbol{X},\boldsymbol{X})^{-1}}_{\mathcal{O}(n^3)}K(\boldsymbol{X},\boldsymbol{X}_*) \end{aligned}$$

Gaussian Processes - 1D demo



Gaussian Processes - Sampling

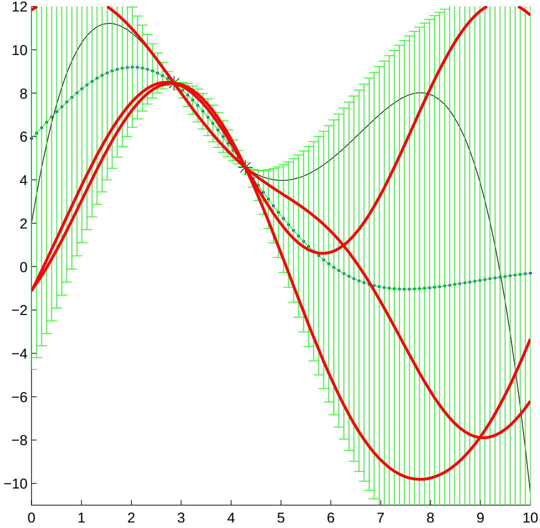
How to sample functions from a $\mathcal{GP}(\boldsymbol{m}, \boldsymbol{K})$?

This can done using the Cholesky decomposition, which is a lower triangular matrix $\boldsymbol{L}$ such that $\boldsymbol{L}\boldsymbol{L}^T = \boldsymbol{K}$

$$\boldsymbol{L} = \mathrm{chol}(\boldsymbol{K})^T;$$
$$\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I});$$
$$\boldsymbol{f} = \boldsymbol{m} + \boldsymbol{L}\boldsymbol{u}^T;$$

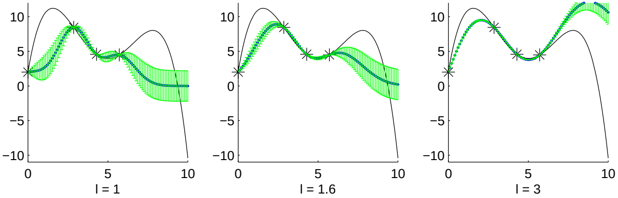
Then $\mathbb{E}[\boldsymbol{f}] = \boldsymbol{m} + \boldsymbol{L}\mathbb{E}[\boldsymbol{u}^T] = \boldsymbol{m}$ and
$var(\boldsymbol{f}) = var(\boldsymbol{L}\boldsymbol{u}^T) = \mathbb{E}[\boldsymbol{L}\boldsymbol{u}^T\boldsymbol{u}\boldsymbol{L}^T] = \boldsymbol{L}\mathbb{E}[\boldsymbol{u}\boldsymbol{u}^T]\boldsymbol{L}^T = \boldsymbol{L}\boldsymbol{I}\boldsymbol{L}^T = \boldsymbol{K}$

Gaussian Processes - Sampling



Model Selection: Hyperparameters

The kernel function and likelihood may depend on additional parameters (hyperparameters) that need to be set



How to choose the best hyperparameters θ?

Model Selection: Marginal Likelihood

In Bayesian model selection one can think of a hierarchical specification. At the bottom level, the posterior over parameters is

$$p(\mathbf{w}|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}, \mathcal{H}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \boldsymbol{\theta}, \mathcal{H})p(\mathbf{w}|\boldsymbol{\theta}, \mathcal{H})}{p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}, \mathcal{H})}$$

where the evidence or marginal likelihood is

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}, \mathcal{H}) = \int p(\mathbf{y}|\mathbf{X}, \mathbf{w}, \mathcal{H})p(\mathbf{w}|\boldsymbol{\theta}, \mathcal{H})\,\mathrm{d}\mathbf{w}$$

Analogously, at the next level, the posterior over hyperparameters is

$$p(\boldsymbol{\theta}|\mathbf{y}, \mathbf{X}, \mathcal{H}) = \frac{p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}, \mathcal{H})p(\boldsymbol{\theta}|\mathcal{H})}{p(\mathbf{y}|\mathbf{X}, \mathcal{H})}$$

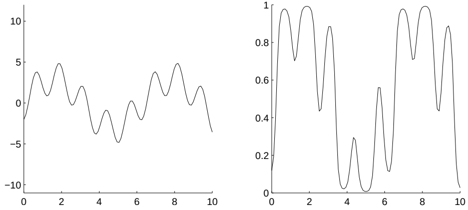
Model Selection: Marginal Likelihood

- This hierarchical specification can be continued analogously to obtain a posterior over models $\mathcal{H}$.
- Depending on the model, however, some integrals may be intractable and approximations are needed.
- In a type II maximum likelihood approximation we *maximize the marginal likelihood*.

$$\log p(\boldsymbol{y}|\boldsymbol{X}, \theta) = -\frac{1}{2}\log|\boldsymbol{K}| - \frac{1}{2}\boldsymbol{y}^T\boldsymbol{K}\boldsymbol{y} - \frac{n}{2}\log 2\pi$$

Classification

- GPs can alo be used for classification, but computations are intractable (needs approximations).
- The idea is to squash a regression function in the domain $(-\infty, \infty)$ to the domain $[0, 1]$
  - Logistic regression: $\lambda(\boldsymbol{x}^T \boldsymbol{w})$ with $\lambda(z) = \frac{1}{1+\exp(-z)}$
  - Probit regression: $\Phi(z) = \int_{-\infty}^{z} \mathcal{N}(x|0,1)\, \mathrm{d}x$



Gaussian Process Applications

- Regression / Classification
- Clustering
- Ordinal Regression
- Preference Learning
- Ranking
- Surrogate modeling / Global Optimization
- Relational Learning
- Reinforcement Learning
- Visualization of high dimensional data
- Nonrigid Shape Recovery
- Evaluating Integrals
- . . .

Preference Learning

Problem: Given a data set of M pairwise preferences (i.e., a set of pairs $(\boldsymbol{x}_1, \boldsymbol{x}_2)$ and whether $\boldsymbol{x}_1 \succ \boldsymbol{x}_2$ or $\boldsymbol{x}_1 \prec \boldsymbol{x}_2$ holds)

$$\mathcal{D} = \{(\boldsymbol{x}_{m_1}, \boldsymbol{x}_{m_2}, d_m) | 1 \leq m \leq M, d_m \in \{-1, 1\}\}$$

predict for new instances $\boldsymbol{x}, \boldsymbol{y}$ which one is preferred.

Idea: Assume a latent (utility) function f over instances that preserves user preferences, i.e., basically $f(\boldsymbol{x}_1) > f(\boldsymbol{x}_2)$ when $\boldsymbol{x}_1 \succ \boldsymbol{x}_2$.

Bayesian framework

$$p(\mathbf{f}|\mathcal{D},\mathcal{H}) = \frac{p(\mathbf{f}|\mathcal{H})p(\mathcal{D}|\mathbf{f},\mathcal{H})}{p(\mathcal{D}|\mathcal{H})}$$

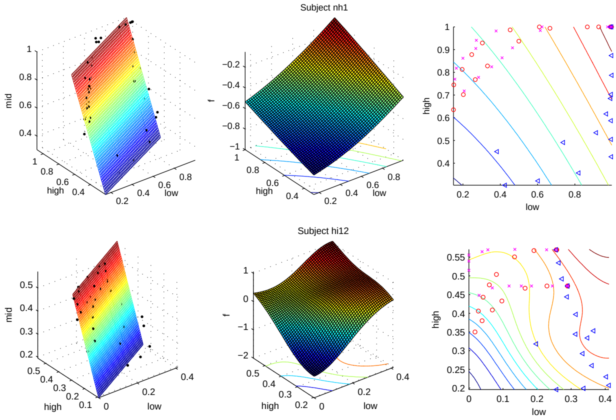
with a likelihood function, for $b \in \mathbb{R}$, $\delta_1, \delta_2 \sim \mathcal{N}(0, \sigma^2)$,

$$\begin{aligned} p(\mathbf{x}_1 \succ \mathbf{x}_2 | f(\mathbf{x}_1), f(\mathbf{x}_2)) &= p(f(\mathbf{x}_1) + \delta_1 > f(\mathbf{x}_2) + b + \delta_2) \\ &= \Phi(z) \end{aligned}$$

with

$$z = \frac{d(f(\mathbf{x}_1) - f(\mathbf{x}_2) - b)}{\sqrt{2}\sigma}$$

Applied to 14 normal-hearing and 18 hearing-impaired subjects. Obtained significant improvement for predicting preferences of hearing-impaired subjects.



Surrogate Modelling with GPs

Complex (physical) systems can be studied nowadays by computer simulations, but often need long running times.

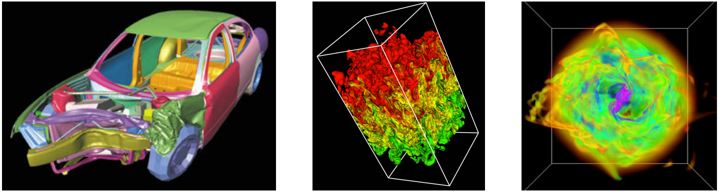


Figure: 1. Car collision; 2. Turbulent-mixing dynamics of a supernova; 3. Gas cloud collapsing inwards to form a star.

Surrogate Modelling with GPs

Idea: replace costly simulation model by a fast Gaussian process surrogate model.

Choose function evaluations in a "smart way" (e.g., reducing overall variance) to obtain a good model fit.

Sometimes, however, we are not interested in a good global model, but only in one specific point (e.g., the best parameter setting).

$$\tilde{\boldsymbol{x}} = \operatorname{argmax}_{\boldsymbol{x}} f(\boldsymbol{x})$$

Function Optimization

Let $f_{\max} = \max\{f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)\}$ be the best value so far. The improvement at a new point $y = f(\boldsymbol{x})$ is defined as

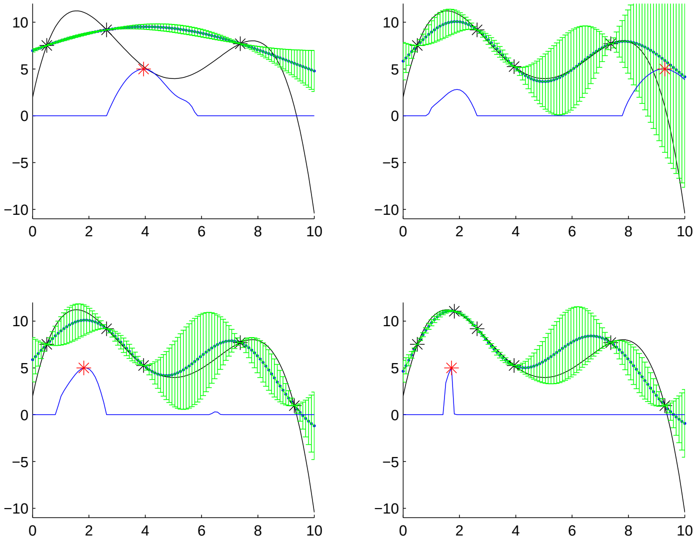
$$I(\boldsymbol{x}) = max\{0, f(\boldsymbol{x}) - f_{\max}\}$$

Using the GP prediction $y = f(\boldsymbol{x}) \sim \mathcal{N}(m, s^2)$ we obtain the Expected Improvement (EI):

$$E(I) = \begin{cases} (m - f_{\max})(1 - \Phi(d)) + s\phi(d) & s > 0 \\ 0 & s = 0 \end{cases}$$

with $d = (f_{\max} - m)/s$ and where $\Phi()$ and $\phi()$ denote the cdf and pdf of the standard normal distribution.

EI - 1D example



Generalized Expected Improvement

Advantages and disadvantages EI:

- EI allows for exploration and exploitation
- Expected Improvement can be sampled fast
- Expected Improvement often converges to a local optimum

Generalized Improvement (g positive integer):

$$I^g(\boldsymbol{x}) = \begin{cases} (f(\boldsymbol{x}) - f_{\text{max}})^g & f(\boldsymbol{x}) > f_{\text{max}} \\ 0 & \text{otherwise} \end{cases}$$

Larger g results in more exploration.

Bayesian Monte Carlo

Consider the following integral

$$F = \int_{\boldsymbol{x}} f(\boldsymbol{x}) p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$$

where p is a known distribution over the inputs X.

For example,

- f could be a computer simulation needing several hours of computation to evaluate f in a single point
- $p(\boldsymbol{x})$ is the posterior distribution and $f(\boldsymbol{x})$ predictions made by the model with parameters $\boldsymbol{x}$,
- $p(\boldsymbol{x})$ is the parameter prior and $f(\boldsymbol{x}) = p(y|\boldsymbol{x})$ the likelihood (i.e., integral computes the evidence)

Bayesian Monte Carlo

Monte Carlo makes the approximation

$$F \simeq \frac{1}{T} \sum_{t=1}^{T} f(\boldsymbol{x}^{(t)})$$

with $\boldsymbol{x}^{(t)}$ random draws from $p(\boldsymbol{x})$. Disadvantages are (c.f. [O'Hagan, 1987] 'Monte Carlo is Fundamentally Unsound'):

- MC is a frequentist approach
- MC can use an irrelevant *importance sampling distribution* $q(\boldsymbol{x})$ when sampling is hard from $p(\boldsymbol{x})$
- MC ignores the values $\boldsymbol{x}^{(t)}$

Bayesian Monte Carlo

We can think of F as being random as we are uncertain about $f(\boldsymbol{x})$ because we cannot afford to compute $f(\boldsymbol{x})$ at every location.

The integral is then a Bayesian inference problem:

- put a prior on f,
- for observations, evaluate f in a number of points
- combine the prior and observations into a posterior distribution over f (which implies a distribution over F)

Bayesian Monte Carlo

When the prior f and posterior $f|\mathcal{D}$ are GPs, the distribution of F is Gaussian, $F \sim \mathcal{N}(\overline{F}, \mathrm{cov}(F))$, and is fully characterized by its mean and variance

$$\begin{aligned} \overline{F} &= \int_{\boldsymbol{x}} \bar{f}_{\mathcal{D}}(\boldsymbol{x}) p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \\ \mathrm{cov}(F) &= \int_{\boldsymbol{x}} \int_{\boldsymbol{x}'} \mathrm{cov}(f_{\mathcal{D}}(\boldsymbol{x}), f_{\mathcal{D}}(\boldsymbol{x}')) p(\boldsymbol{x}) p(\boldsymbol{x}') \, \mathrm{d}\boldsymbol{x} \, \mathrm{d}\boldsymbol{x}' \end{aligned} \tag{1}$$

with $\bar{f}_{\mathcal{D}}$ and $\mathrm{cov}(f_{\mathcal{D}}(\boldsymbol{x}), f_{\mathcal{D}}(\boldsymbol{x}'))$ the posterior mean and posterior variance, respectively.

Bayesian Monte Carlo - Special case

Sometimes the problem can be reduced to products of one dimensional integrals and/or some analytic expression, e.g.,

$$\begin{aligned} p(\boldsymbol{x}) &\sim \mathcal{N}(\boldsymbol{b}, \boldsymbol{B}) \\ k(\boldsymbol{x}, \boldsymbol{x}') &= w_0 \exp\left(-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{x}')^T \boldsymbol{A}^{-1}(\boldsymbol{x}-\boldsymbol{x}')\right) \end{aligned}$$

with $\boldsymbol{A} = \mathrm{diag}(w_1^2, \ldots, w_N^2)$. Then

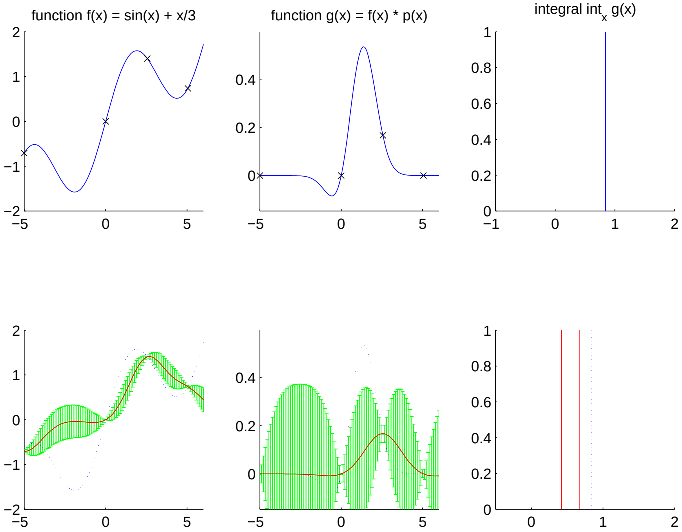
$$\overline{F} = \boldsymbol{z}\boldsymbol{K}^{-1} Y, \qquad \mathrm{cov}(F) = k_c - \boldsymbol{z}\boldsymbol{K}^{-1}\boldsymbol{z}^T$$

with

$$k_c = w_0 |2\boldsymbol{A}^{-1}\boldsymbol{B} + \boldsymbol{I}|^{-1/2}$$

$$z_l = w_0 |\boldsymbol{A}^{-1}\boldsymbol{B} + \boldsymbol{I}|^{-1/2} \exp\left(-\frac{1}{2}(\boldsymbol{x}_l - \boldsymbol{b})^T (\boldsymbol{A}+\boldsymbol{B})^{-1}(\boldsymbol{x}_l - \boldsymbol{b})\right)$$

Bayesian Monte Carlo - 1D demo



Suppose we want to introduce a new cake mix into the consumer market which is robust to an inaccurate setting of oven temperature and baking time.

3 Control variables: The amount of flour (F), the amount of sugar (S), and the amount of egg powder (E).

2 Noise variables: Oven temparature (T) and baking time (t).

References

- Gaussian Processes for Machine Learning, C.E. Rasmussen and K.I. Williams
- The Gaussian Process Web Site `http://www.gaussianprocess.org/`