Advanced Bayesian Learning

Gaussian Process Regression and Classification - Lecture 2

Mattias Villani

Department of Statistics
Stockholm University

Department of Computer and Information Science
Linköping University

Stationary processes and smoothness

- A stochastic process (field) $\{f(\mathbf{x}), x \in \mathbb{R}^D\}$ is weakly stationary if $E(f(\mathbf{x})) = \mu$ and its covariance function $k(\mathbf{x}, \mathbf{x}')$ is a function of $\mathbf{t}$=$\mathbf{x} - \mathbf{x}'$

$$k(\mathbf{x}, \mathbf{x}') = Cov\left[f(\mathbf{x}), f(\mathbf{x}')\right] = k(\mathbf{t}).$$

- The covariance function is isotropic if it only depends on the distance $t = \|\mathbf{x} - \mathbf{x}\|$ (invariant to directions)

$$k(\mathbf{x}, \mathbf{x}') = Cov\left[f(\mathbf{x}), f(\mathbf{x}')\right] = k(t).$$

- A stationary process is continuous in quadratic mean

$$E\left(|f(x+t) - f(x)|^2\right) \rightarrow 0 \text{ as } t \rightarrow 0$$

iff $k(t)$ is continuous at $t = 0$.

- A stationary process is differentiable in quadratic mean

$$\frac{f(x+t) - f(x)}{t} \overset{q.m.}{\rightarrow} f'(x) \textbf{ as } t \rightarrow 0$$

iff $k(t)$ is twice continuously differentiable at $t = 0$.

Fourier analysis and orthogonal functions

- **Fourier series** for functions:

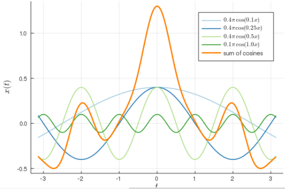
$$f(x) = \sum_k a_k \cos(2\pi s_k x) + \sum_k b_k \sin(2\pi s_k x)$$

$a_k = \int f(x) \cos(2\pi s_k x) dx$ and $b_k = \int f(x) \sin(2\pi s_k x) dx$.

- cos and sin are **orthogonal** at Fourier frequencies s_k and s_l:

$$\int \sin(2\pi s_k x) \cos(2\pi s_l x) dx = \delta_{kl}$$

- **Complex exponential**: $e^{it} \equiv \cos t + i \cdot \sin(t)$.
- Fourier: $f(x) = \sum_k c_k e^{i 2\pi s_k x}$ where $c_k = \int f(x) e^{i 2\pi s_k x} dx$.



Spectral density

- **Bochner's theorem**: A function $k(\cdot)$ is the covariance function of a stationary continuous process iff

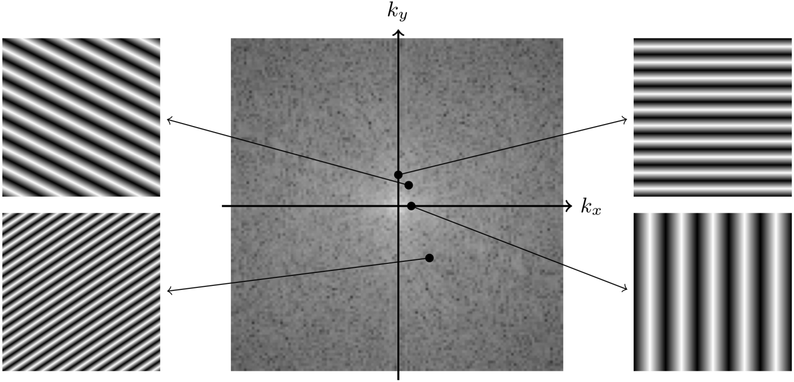
$$k(t) = \int_{\mathbb{R}^D} e^{2\pi i s t} S(s) ds$$

- $S(s)$ is the **spectral density**. $S(s)$ is the energy allocated to the basis function $e^{2\pi i s t}$ at frequency s.
- $S(s) \quad \Longleftrightarrow \quad k(t) \quad \Longleftrightarrow \quad \text{Smoothness of } f(x).$
- **Multivariate Bochner's**: A function $k(\cdot)$ on $\mathbb{R}^D$ is the covariance function of a stationary continuous process iff

$$k(\mathbf{t}) = \int_{\mathbb{R}^D} e^{2\pi i \mathbf{s}^T \mathbf{t}} S(\mathbf{s}) d\mathbf{s}$$

- $e^{2\pi i \mathbf{s}^T \mathbf{t}}$ is a D-dimensional sine wave with frequency $\mathbf{s}$ (with direction).

Fourier in 2D



Spectral density determines smoothness

- A stationary process $f(x)$ is continuous in q.m. if

$$\int S(s) ds < \infty$$

- The kth q.m. derivative process $f^{(k)}(x)$ has spectral density

$$S_{f^{(k)}}(s) = s^{2k} S_f(s)$$

- $f(x)$ is q.m. differentiable of order k iff $S(s)$ has moments order $2k$.

Spectral densities of common kernels

- Let $r = \|x - x'\|$. All kernels can be scaled by $\sigma_f > 0$.
- **Squared exponential (SE)** ($\ell > 0$)

$$K_{SE}(r) = \exp\left(-\frac{r^2}{2\ell^2}\right)$$

  - Spectral density $S(s) = (2\pi\ell^2)^{D/2} \exp\left(-2\pi^2\ell^2 s^2\right)$.
  - Higher freq tail of like a Gaussian with variance $1/(4\pi^2\ell^2)$.
  - Infinitely mean square differentiable. Very smooth.
- **Matérn** ($\ell > 0$, $\nu > 0$)

$$K_{Matern}(r) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu r}}{\ell}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu r}}{\ell}\right)$$

  - Spectral density: student-t density with 2ν degrees of freedom.
  - $\nu = 1/2$, $S(s)$ is Cauchy. Continuous in q.m., no derivatives.
  - As $\nu \to \infty$, Matérn approaches SE.

Spectral mixture kernels

- Bochner's theorem for stationary processes:

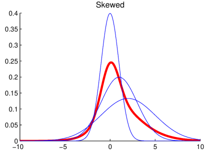
$$k(t) = \int e^{2\pi ist} S(s) ds.$$

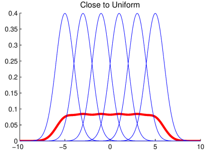
- Mixture of normals in frequency domain

$$S(s) = \sigma^2 \sum_{k=1}^{K} \pi_k \mathcal{N}(s|\mu_k, \psi_k^2)$$

- Bochner's theorem gives kernel in time domain

$$k(t) = \sigma^2 \sum_{k=1}^{K} \pi_k \cos(2\pi\mu_k t) \exp\left(-2\pi^2\psi_k^2 t^2\right)$$





SE as infinite basis expansion

- Regression with basis functions $\phi_1(x), \ldots, \phi_N(x)$

$$y = \sum_{c=1}^{N} w_c \phi_c(x) + \varepsilon$$

$$\phi_c(x) = \exp\left(-\frac{(x-c)^2}{2\ell^2}\right).$$

- Prior $\boldsymbol{w} \sim N\left(0, \frac{\sigma_p^2}{N} I\right)$.
- This is a GP with kernel

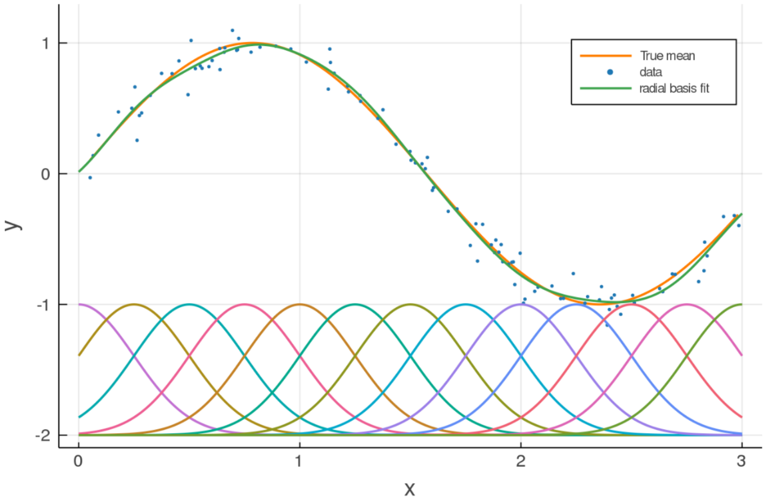
$$k(x_p, x_q) = \mathrm{cov}\left(\sum_{c=1}^{N} w_c \phi_c(x_p), \sum_{c=1}^{N} w_c \phi_c(x_q)\right) = \frac{\sigma_p^2}{N} \sum_{c=1}^{N} \phi_c(x_p)\phi_c(x_p) \rightarrow \sigma_p^2 \int_{c_{\min}}^{c_{\max}} \phi_c(x_p)\phi_c(x_p) dc$$

as the number of bases $N \rightarrow \infty$ over $[c_{\min}, c_{\max}]$.

- Letting $c_{\min} \rightarrow -\infty$ and $c_{\max} \rightarrow \infty$ we get

$$k(x_p, x_q) = \sigma_p^2 \int_{-\infty}^{\infty} \exp\left(-\frac{(x_p - c)^2}{2\ell^2}\right) \exp\left(-\frac{(x_c - c)^2}{2\ell^2}\right) dc = \sqrt{\pi}\ell\sigma_p^2 \exp\left(-\frac{(x_p - x_q)^2}{2(\sqrt{2}\ell)^2}\right)$$

Fitting basis expansion



Kernel composition

- **Periodic kernels**. When $f(x)$ is believed to be periodic with period d. Example:

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{2\sin^2\left(\pi\left|x - x'\right|/d\right)}{\ell^2}\right).$$

- **Product** of kernels is a kernel.
- Example: Locally periodic. Two nearby peaks are more dependent than two distant peaks.

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{2\sin^2\left(\pi\left|x - x'\right|^2/d\right)}{\ell^2}\right) \times \exp\left(-\frac{1}{2}\frac{\left|x - x'\right|^2}{\ell^2}\right.$$

- **Sum** of kernels is a kernel.

- Let $f_a \sim GP\left[m_a(\mathbf{x}), k_a(\mathbf{x}, \mathbf{x}')\right]$ independently of $f_b \sim GP\left[m_b(\mathbf{x}), k_b(\mathbf{x}, \mathbf{x}')\right]$ then

$$f_a + f_b \sim GP\left[m_a(\mathbf{x}) + m_b(\mathbf{x}), k_a(\mathbf{x}, \mathbf{x}') + k_b(\mathbf{x}, \mathbf{x}')\right]$$

Anisotropic kernels - ARD

- Anisotropic version of isotropic kernels by setting $r^2(\mathbf{x}, \mathbf{x}') = (\mathbf{x} - \mathbf{x}')^T \mathbf{M} (\mathbf{x} - \mathbf{x}')$ where $\mathbf{M}$ is positive definite.

- **Automatic Relevance Determination (ARD)**: $\mathbf{M} = Diag(\ell_1^{-2}, ..., \ell_D^{-2})$ is diagonal with different length scales.

- ARD does 'variable selection' since large ℓ_j means that the jth input essentially drops out of $f(\mathbf{x})$.

- ARD is a product of D one-dimensional kernels, one for each input variable

$$k_{ARD}(\mathbf{x}, \mathbf{x}') = \prod_{d=1}^{D} k_{SE,\ell_d}(x_d, x_d')$$

- **Factor kernels**: $M = \Lambda\Lambda^T + \Psi$, where Λ is $D \times k$ for low rank k.

Discrete covariates

- Suppose: x_1 is continuous (mg/week) and x_2 is binary (sex).
- Linear regression: just use x_2 coded as $x_2 = 0$ if male, $x_2 = 1$ if female.
- Implicit model:

$$y = \begin{cases} \beta_0 + \beta_1 x_1 & \text{if } x_2 = 0 \\ \beta_0 + \tilde{\beta}_0 + (\beta_1 + \tilde{\beta}_1) x_1 & \text{if } x_2 = 1 \end{cases}$$

- GP: add the 0-1 coded covariate and use ARD kernel:

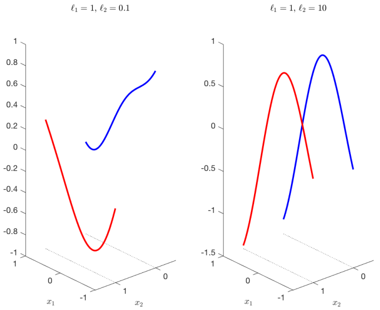
$$\exp\left(-\frac{1}{2}\left(\frac{x_1 - x_1'}{\ell_1}\right)^2\right)\exp\left(-\frac{1}{2}\left(\frac{x_2 - x_2'}{\ell_2}\right)^2\right)$$

So the covariance between $f(x_1, 0)$ and $f(x_1, 1)$ is

$$\exp\left(-\frac{1}{2}\left(\frac{1}{\ell_2}\right)^2\right)$$

Discrete covariates

- Large ℓ_2: men and female are believed to have similar profiles with respect to x_1.
- Small ℓ_2: men and female are believed to have potentially very different profiles with respect to x_1.



- Categorical covariates with K levels: create K *one-hot* variables.

Eigenfunction decomposition

- Eigenvalue decomposition of a $n \times n$ covariance matrix: $K = V\Lambda V^T$, or

$$K = \sum_{j=1}^{n} \lambda_j \boldsymbol{v}_j \boldsymbol{v}_j^T,$$

where $\Lambda = \mathrm{Diag}(\lambda_1, ..., \lambda_n)$, $Kv_j = \lambda_j v_j$, $\boldsymbol{v}_j^T \boldsymbol{v}_j = \delta_{ij}$.

- Simulation from $y \sim N(\mu, K)$: $y = \mu + V\Lambda^{1/2}\boldsymbol{z}$, and $\boldsymbol{z} \sim N(0, I)$. Principal components.
- Mercer's theorem for covariance kernels

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{\infty} \lambda_i \phi_i(\boldsymbol{x}) \phi_i^*(\boldsymbol{x}')$$

$$\int k(\boldsymbol{x}, \boldsymbol{x}') \phi(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} = \lambda \phi(\boldsymbol{x}')$$

the eigenfunctions $\phi(\boldsymbol{x})$ are orthogonal with respect $p(\boldsymbol{x})$.

- The eigenvalues determine the smoothness of the kernel.
- Bochner: $e^{2\pi i \mathbf{s} \cdot \boldsymbol{x}}$ are the eigenfunctions of stationary kernels.

Karhunen-Loève decomposition

- **Karhunen–Loève theorem**: a stochastic process X_t can be represented as

$$X_t = \sum_{k=1}^{\infty} Z_k e_k(t),$$

  where Z_k are uncorrelated variables and $e_k(t)$ orthogonal basis.
- $e_k(t)$ are determined by the covariance function of X_t.
- Karhunen-Loève adapts to X_t optimally.
- If X_t is a GP: the Z_k are Gaussian and independent.
- Can be use for simulation
- Truncate an infinite-dimensional process to finite dimension.

Large scale GPs

- GPs are **computationally challenging**.
- Need to invert $n \times n$ matrices such as $\left[K(\mathbf{x}, \mathbf{x}) + \sigma^2 I\right]^{-1}$.
- **Scales as** $O(n^3)$. Also with Cholesky.
- **Banded covariance functions**.
  - Special covariance functions that makes $K(\mathbf{x}, \mathbf{x})$ sparse.
  - Observations at a certain distance apart are uncorrelated.
  - Sparse matrix algebra.

Large scale GPs

- Introduce m latent inducing variables $\mathbf{u} = \{u_1, ..., u_m\}$ with inputs $\mathbf{X}_u = \{\mathbf{x}_{u_1}, \mathbf{x}_{u_2}, ..., \mathbf{x}_{u_m}\}$. Pseudo inputs.
- The Fully Independent Conditional (FIC) method *assumes* elements in $\mathbf{f}$ are independent given $\mathbf{u}$

$$p(\mathbf{f}|\mathbf{X}, \mathbf{X}_u, \mathbf{u}, \theta) = \prod_{i=1}^{n} p_i(f_i|\mathbf{X}, \mathbf{X}_u, \mathbf{u}, \theta)$$

- Computations are now $O(m^2 n)$, and often $m \ll n$. Fast!
- Partially Independent Conditional (PIC). Partition into blocks $\mathbf{f} = (\mathbf{f}_1, ..., \mathbf{f}_k)$, where each $\mathbf{f}_i$ has b elements. Assume indep. blocks given $\mathbf{u}$, but full dependence with blocks.
- $b = 1$ gives FIC. $b = n$ gives the original GP.
- The locations of $\mathbf{X}_u$ are learned by optimization.

Classification with logistic regression

- Classification: binary response $y \in \{-1, 1\}$.
- Example: linear logistic regression

$$Pr(y = 1|\mathbf{x}) = \lambda(\mathbf{x}^T\mathbf{w})$$

where $\lambda(z)$ is the logistic link function

$$\lambda(z) = \frac{1}{1 + \exp(-z)}$$

- $\lambda(z)$ 'squashes' the linear prediction $\mathbf{x}^T\mathbf{w} \in \mathbb{R}$ into $\in [0, 1]$.
- Logistic regression has linear decision boundaries.

GP classification

- Obvious GP extension of logistic regression: replace $\mathbf{x}^T \mathbf{w}$ by

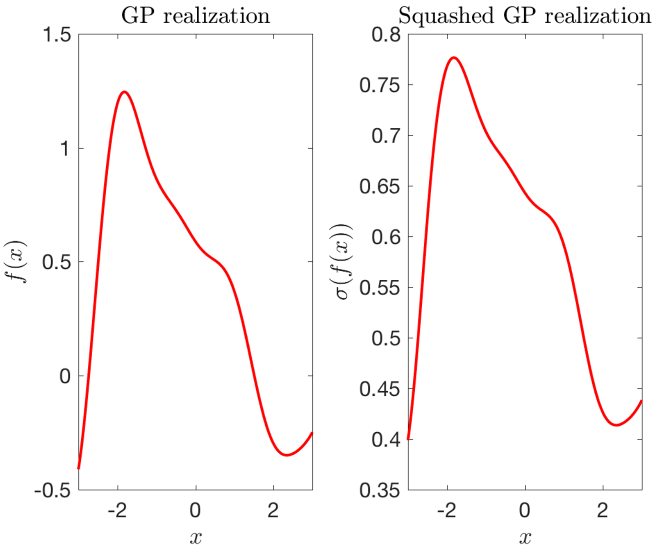
$$f(\mathbf{x}) \sim GP(0, k(\mathbf{x}, \mathbf{x}'))$$

  and squash

$$Pr(y = 1|\mathbf{x}) = \lambda(f(\mathbf{x}))$$

- Flexible decision boundaries (non-parametric, GP-style).

Squashing a GP function



GP classification - inference

- **Prediction** for a test case $\mathbf{x}_*$:

$$Pr(y_* = +1|\mathbf{X}, \mathbf{y}, \mathbf{x}_*) = \int \sigma(f_*) p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) df_*$$

  - $\sigma(f_*)$ is some sigmoidal function (logistic, normal CDF...)
  - f_* is the latent f at the test input $\mathbf{x}_*$.

- The posterior distribution of f_* is

$$p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \int p(f_*|\mathbf{f}, \mathbf{X}, \mathbf{x}_*) p(\mathbf{f}|\mathbf{X}, \mathbf{y}) d\mathbf{f}$$

  where $p(\mathbf{f}|\mathbf{X}, \mathbf{y})$ is the posterior of $\mathbf{f}$ from the training data

$$p(\mathbf{f}|\mathbf{X}, \mathbf{y}) \propto p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\mathbf{X})$$

- $p(\mathbf{y}|\mathbf{f})$ is no longer Gaussian. Posterior $p(\mathbf{f}|\mathbf{X}, \mathbf{y})$ intractable.

The Laplace approximation

- Approximates $p(\mathbf{f}|\mathbf{X},\mathbf{y})$ with $N(\hat{\mathbf{f}}, \mathbf{A}^{-1})$, where
  - $\hat{\mathbf{f}}$ is the posterior mode
  - $\mathbf{A} = -\nabla\nabla \log p(\mathbf{f}|\mathbf{y})$ is negative Hessian at $\mathbf{f} = \hat{\mathbf{f}}$.
- Log posterior

$$
\begin{aligned}
\Psi(\mathbf{f}) &= \log p(\mathbf{y}|\mathbf{f}) + \log p(\mathbf{f}|\mathbf{X}) \\
&= \log p(\mathbf{y}|\mathbf{f}) - \frac{1}{2}\mathbf{f}^T K^{-1}\mathbf{f} - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi
\end{aligned}
$$

- Differentiating wrt $\mathbf{f}$

$$
\begin{aligned}
\nabla\Psi(\mathbf{f}) &= \nabla \log p(\mathbf{y}|\mathbf{f}) - K^{-1}\mathbf{f} \\
\nabla\nabla\Psi(\mathbf{f}) &= \nabla\nabla \log p(\mathbf{y}|\mathbf{f}) - K^{-1} = -W - K^{-1}
\end{aligned}
$$

  where W is a diagonal matrix since y_i only depends on f_i.
- Use Newton's method to iterate to the mode.
- Approximate inference for f_* is possible.
- Predictions of y_* by one-dim numerical integration.

Saddlepoint approximation of marginal likelihood

- **Saddlepoint approximation** of **marginal likelihood**

$$p(\mathbf{y}|\theta) = \int p(\mathbf{y}, \mathbf{f}|\theta) d\mathbf{f} \approx \sqrt{2\pi} p(\mathbf{y}, \hat{\mathbf{f}}_\theta|\theta) \left(\frac{\partial^2 p(\mathbf{y}, \mathbf{f}|\theta)}{\partial \mathbf{f} \partial \mathbf{f}^T} |_{\mathbf{f}=\hat{\mathbf{f}}_\theta} \right)^{-1/2},$$

  where $\hat{\mathbf{f}}_\theta$ and $\frac{\partial^2 \log p(\mathbf{f}, \theta|\mathbf{y})}{\partial \mathbf{f} \partial \mathbf{f}^T}$ are the mode and Hessian for given θ.

- Joint posterior

$$\log p(\mathbf{y}, \mathbf{f}|\theta) = \log p(\mathbf{y}|\mathbf{f}, \theta) + \log p(\mathbf{f}|\theta)$$

- Saddlepoint approx = local Laplace approximation for given θ.

Hamiltonian Monte Carlo

- HMC/MCMC to **sample from training posterior**

$$\mathbf{f}|\mathbf{x}, \mathbf{y}, \theta$$

  Produces $\mathbf{f}^{(1)}, ..., \mathbf{f}^{(N)}$ draws.

- For each $\mathbf{f}^{(i)}$, **sample the test posterior** $\mathbf{f}_*$ from

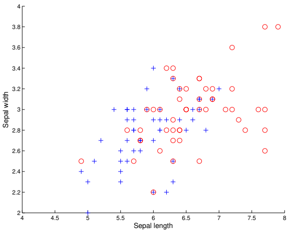
  $\mathbf{f}_*|\mathbf{f}^{(i)}, \mathbf{x}, \mathbf{x}_* \sim N\left(K(\mathbf{x}_*, \mathbf{x})K(\mathbf{x}, \mathbf{x})^{-1}\mathbf{f}^{(i)}, K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbf{x})K(\mathbf{x}, \mathbf{x})^{-1}K(\mathbf{x}, \mathbf{x}_*)\right)$

  Note that this does not depend on $\mathbf{y}$ since we condition on $\mathbf{f}$. Noise-free GP fit. Produces $\mathbf{f}_*^{(1)}, ..., \mathbf{f}_*^{(N)}$ draws.

- For each $\mathbf{f}_*^{(i)}$, **sample a prediction** from

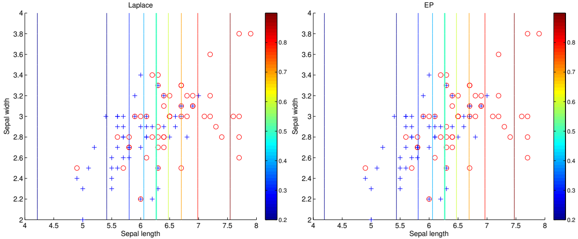
$$p(\mathbf{y}_*|\mathbf{f}_*^{(i)}, \theta).$$

Iris data - sepal - SE kernel with ARD

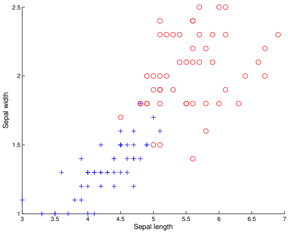


$$\text{Laplace: } \hat{\ell}_1 = 1.7214, \hat{\ell}_2 = 185.5040, \sigma_f = 1.4361$$

$$\text{EP: } \hat{\ell}_1 = 1.7189, \hat{\ell}_2 = 55.5003, \sigma_f = 1.4343$$

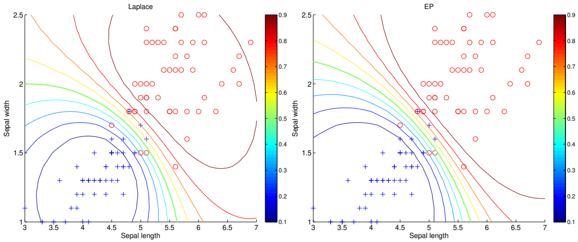


Iris data - petal - SE kernel with ARD

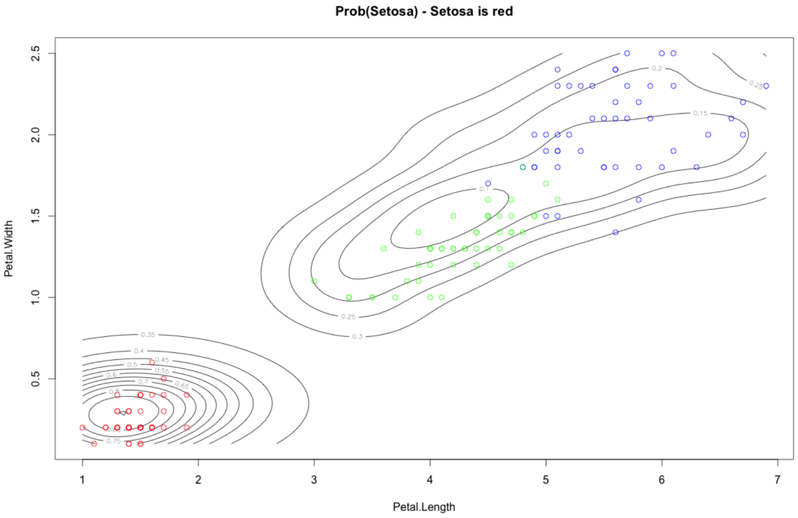


$$\text{Laplace: } \hat{\ell}_1 = 1.7606, \hat{\ell}_2 = 0.8804, \sigma_f = 4.9129$$

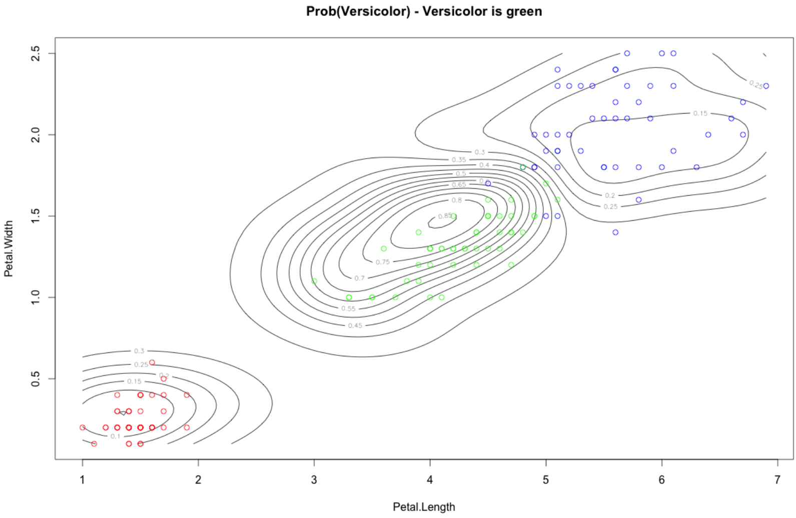
$$\text{EP: } \hat{\ell}_1 = 2.1139, \hat{\ell}_2 = 1.0720, \sigma_f = 5.3369$$



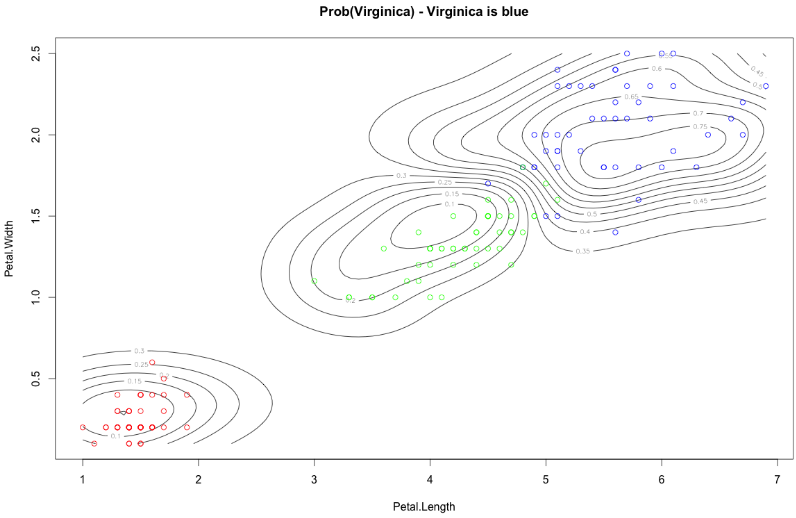
Iris data - petal - all three classes



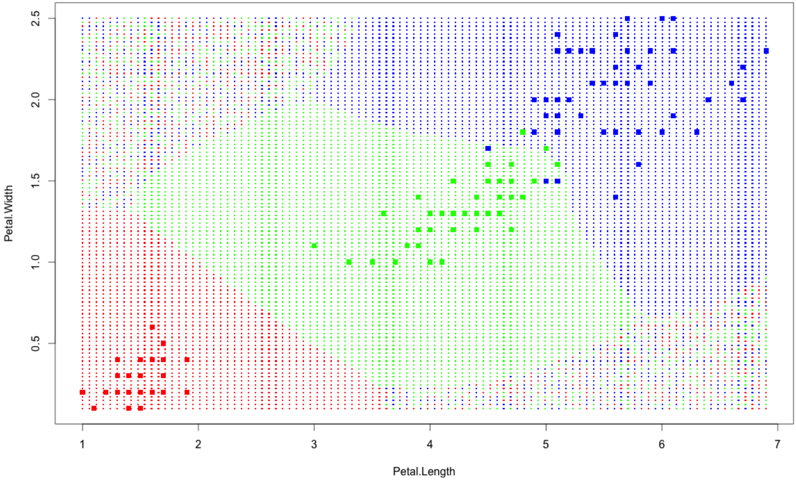
Iris data - petal - all three classes



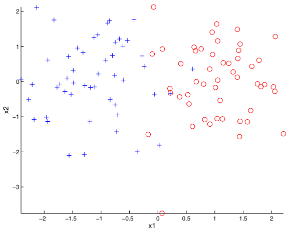
Iris data - petal - all three classes



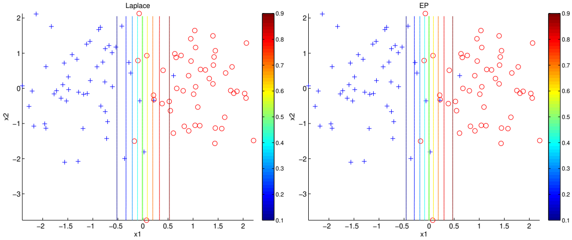
Iris data - petal - decision boundaries



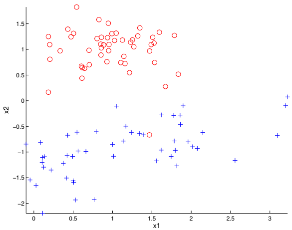
Toy data 1 - SE kernel with ARD



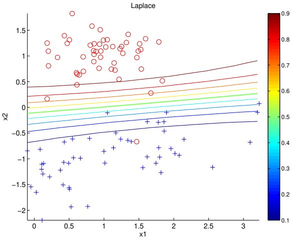
EP: $\hat{\ell}_1 = 2.4503, \hat{\ell}_2 = 721.7405, \sigma_f = 4.7540$

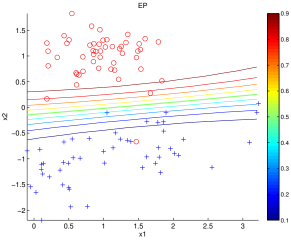


Toy data 2 - SE kernel with ARD

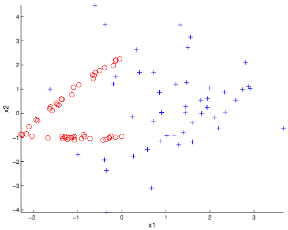


$$\text{EP: } \hat{\ell}_1 = 8.3831, \hat{\ell}_2 = 1.9587, \sigma_f = 4.5483$$



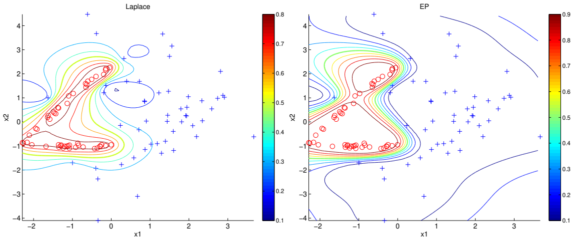


Toy data 3 - SE kernel with ARD



Laplace: $\hat{\ell}_1 = 0.7726, \hat{\ell}_2 = 0.6974, \sigma_f = 11.7854$

EP: $\hat{\ell}_1 = 1.2685, \hat{\ell}_2 = 1.0941, \sigma_f = 17.2774$



Bayesian Optimization (BO)

- Minimization of expensive function

$$\text{argmin}_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$$

- Hyperparameter estimation from marginal likelihood.
- BO idea[1]:
  - Assign GP prior to the unknown function f.
  - Evaluate the function at some values $x_1, x_2, ..., x_n$.
  - Update posterior $f|x_1, ..., x_n \sim GP$.
  - Use GP posterior of f to find new eval point x_{n+1}.
  - Repeat until convergence
- Explore vs Exploit.
- Bayesian Numerics[2]. Posterior of $\int f(\boldsymbol{x})d\boldsymbol{x}$ from $\{f(\boldsymbol{x}_i)\}$.

[1] Snoek et al (2012). Practical Bayesian Optimization of Machine Learning Algorithms.

[2] Hennig et al (2015). Probabilistic numerics and uncertainty in computations.

Acquisition functions

- **Probability of Improvement (PI)**

$$a_{PI}(\mathbf{x}; \mathcal{D}_n) \equiv \Pr\left(f(\mathbf{x}) < f(\mathbf{x}_{\text{best}}) | \mathcal{D}_n\right) = \Phi(\gamma(\mathbf{x}))$$

where $\mathcal{D}_n = \{y_i, \mathbf{x}_i\}_{i=1}^{n}$ are previous function evaluations and

$$\gamma(\mathbf{x}) = \frac{f(\mathbf{x}_{\text{best}}) - \mu(\mathbf{x}; \mathcal{D}_n)}{\sigma(\mathbf{x}; \mathcal{D}_n)}$$

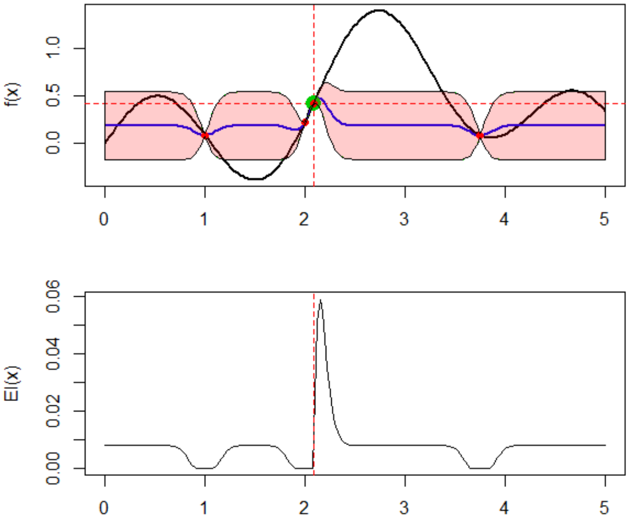
- **Expected Improvement (EI)**

$$a_{EI}(\mathbf{x}; \mathcal{D}_n) = \sigma(\mathbf{x}; \mathcal{D}_n)\left[\gamma(\mathbf{x})\Phi(\gamma(\mathbf{x})) + \phi(\gamma(\mathbf{x}))\right]$$

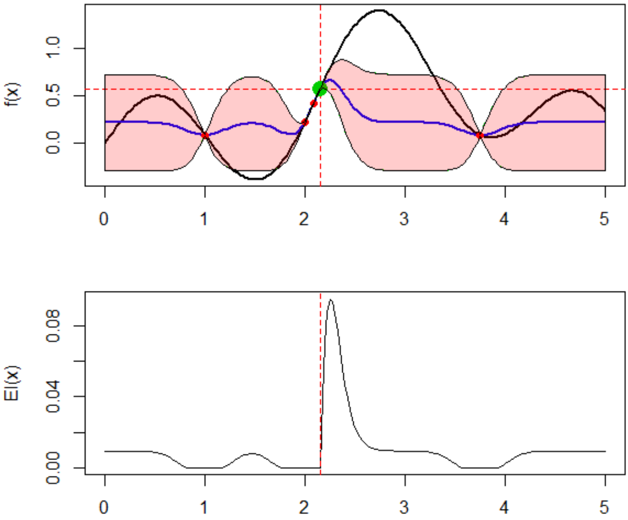
- Maximizing $a(\mathbf{x})$ to find $\mathbf{x}_{n+1}$ is simpler than minimizing $f(\mathbf{x})$.
- Noisy function evaluations $\hat{f}(\mathrm{x})$ (e.g. MCMC). Noisy GP.
- When precision of $\hat{f}(\mathrm{x})$ is controlled by user: BOOP. [3]

[3] Gustafsson et al (2020). Bayesian Optimization of Hyperparameters when the Marginal Likelihood is Estimated by MCMC. On arXiv next week ...

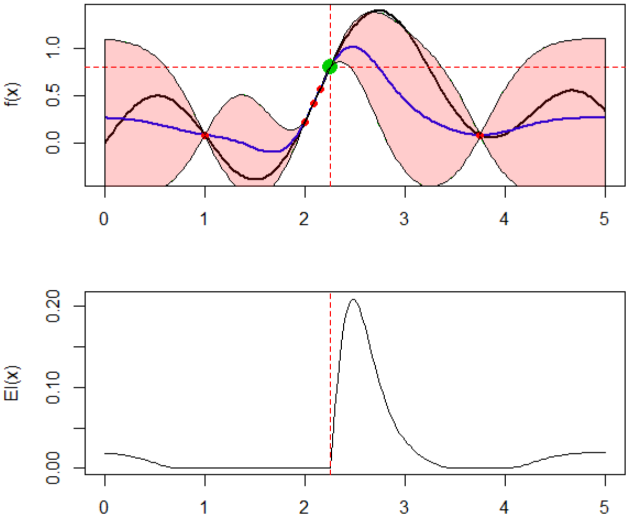
BO illustration - EI



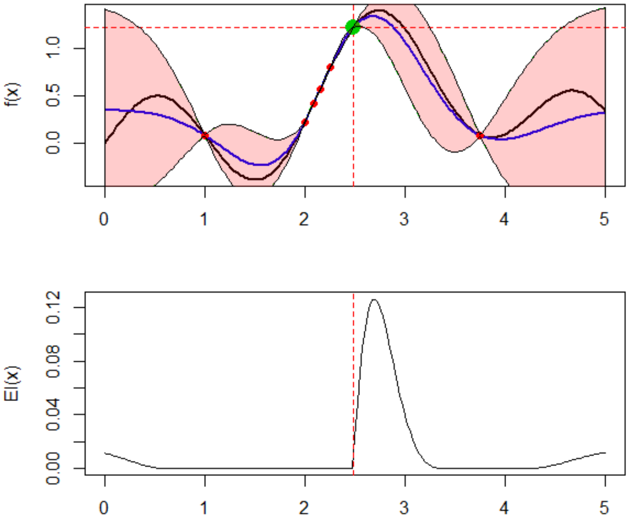
BO illustration - EI



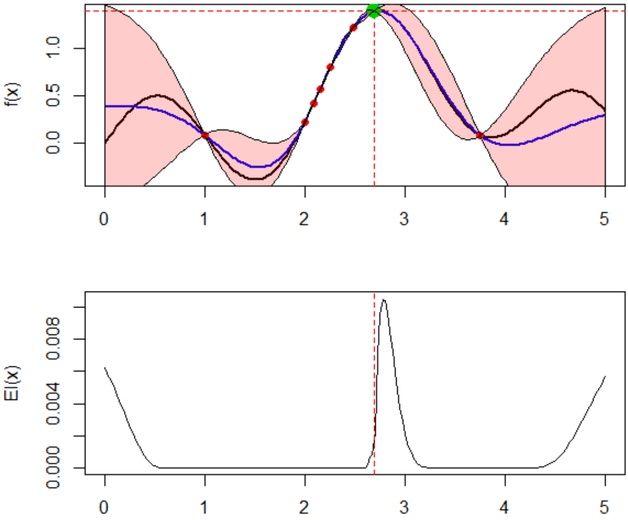
BO illustration - EI



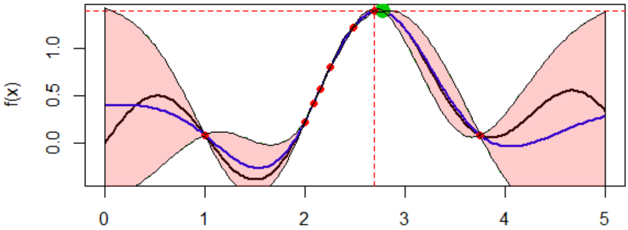
BO illustration - EI

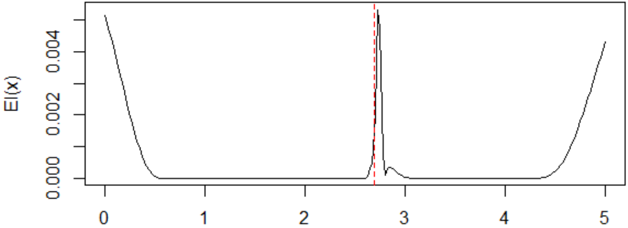


BO illustration - EI

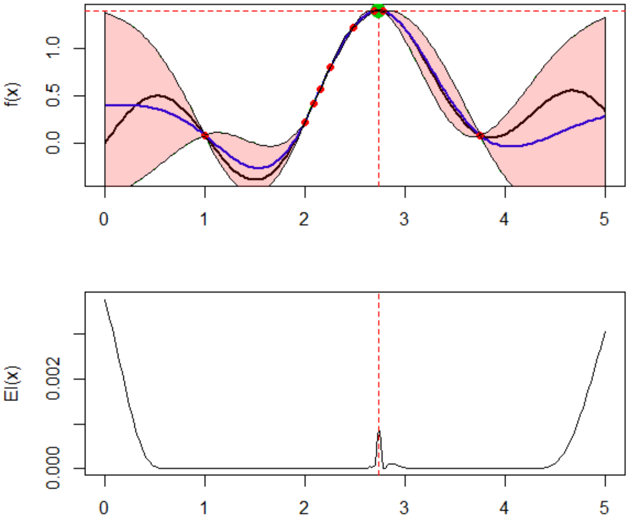


BO illustration - EI

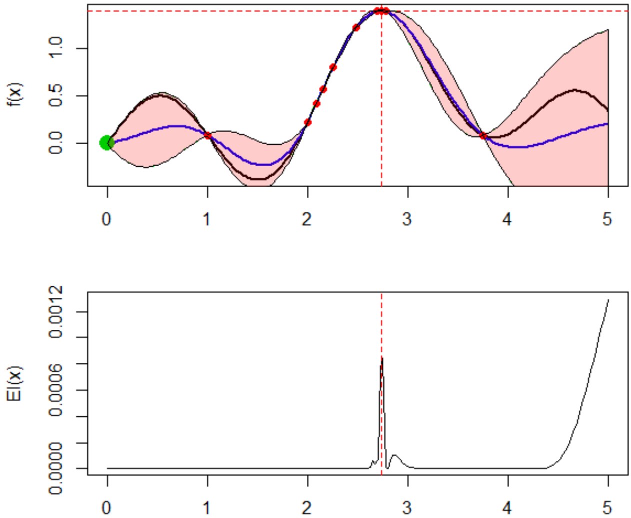




BO illustration - EI



BO illustration - EI



BO illustration - EI