Gaussian processes for Computational Biology

Peter Tonner

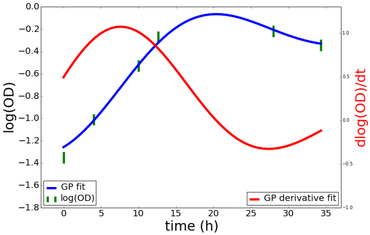
Computational Biology and Bioinformatics
Duke University

October 5, 2017

About me

- 6th year PhD student
- Computational Biology and Bioinformatics program
- Advisors: Amy Schmid and Scott Schmidler
- Defense: November 14th

Why Gaussian processes?



- Tonner *et al.*, "Detecting differential growth of microbial populations using Gaussian process regression", Genome Research 2016
- Darnell, Tonner *et al.* "Systematic Discovery of Archaeal Transcription Factor Functions in Regulatory Networks through Quantitative Phenotyping Analysis" MSystems, 2017
- Tonner *et al.* "Random effects modeling of microbial population growth", *in prep*

Why Gaussian processes?

- **Population genomics:** "Flexible Modelling of Genetic Effects on Function-Valued Traits", Fusi and Listgarten (2016)
- **Temporal modeling of microbial communities:** "Temporal probabilistic modeling of bacterial compositions derived from 16S rRNA sequencing", Aijo, Mueller, and Bonneau (2016)
- **Single cell RNA-seq time-series:** "Order Under Uncertainty: Robust Differential Expression Analysis Using Probabilistic Models for Pseudotime Inference", Campbell and Yau (2016)

About you

What is your background?

- statistics: multivariate normal, linear models, Bayesian vs. Frequentist statistics
- programming: python, jupyter notebooks
- linear algebra: matrix inversion, positive definiteness

Affiliations

- CBB
- Biology
- CMB
- MGM
- UPGG
- BME

About this workshop

Outline of sessions

1. Basic Theory and Applications
   - GP theory
   - GP regression & GPy
2. Bayesian Modeling
   - Simple example
   - Hierarchical models

Resources

- github: github.com/ptonner/gp-workshop

Major References

- *Gaussian Processes for Machine Learning*, Rasmussen and Williams (2006)
- *GPy*, a Gaussian process library in python. sheffieldml.github.io/GPy/
- Stan: mc-stan.org

Motivating example

Motivating example

- **Problem**: choosing $f(x)$ based on observed data, may need to change with new observations. no “best” way to choose f
- **Question**: what if we could represent $f(x)$ with a distribution?

Answer: Gaussian processes

def: A *Gaussian process* is an infinite collection of random variables, any finite number of which have a joint multivariate normal distribution.

$$f(x) \sim GP\Big(\mu(x), \kappa(x, x')\Big)$$

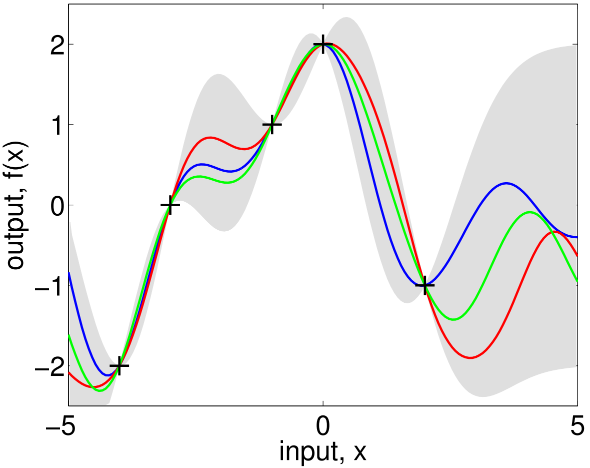
Most common uses for Gaussian processes

$$f(x) \sim GP\Big(\mu(x), \kappa(x, x')\Big)$$

regression	$y(x) = f(x) + \epsilon(x)$
classification	$p(y(x) = 1 \Big\vert f(x)) = \frac{1}{1+\exp(-f(x))}$
hierarchical	$p(y(x) \Big\vert f(x), \dots)$

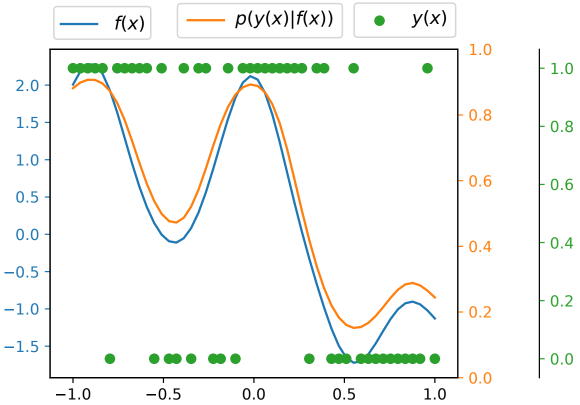
Gaussian processes regression

$$y(x) = f(x) + \epsilon(x)$$



Gaussian processes classification

$$p(y(x) = 1 \big| f(x)) = \frac{1}{1 + \exp(-f(x))}$$



Multivariate normal

$$X \sim N(\mu, \Sigma), \quad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix}, \quad \Sigma = \begin{pmatrix} \sigma^2_{1,1} & \cdots \\ \vdots & \ddots \end{pmatrix}$$

$$p(X = x|\mu, \Sigma) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \cdot \exp\left(\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

Multivariate normal

$$X \sim N(\mu, \Sigma), \quad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix}, \quad \Sigma = \begin{pmatrix} \sigma^2_{1,1} & \cdots \\ \vdots & \ddots \end{pmatrix}$$

All subsets of X are multivariate normal:

$$Y = \begin{bmatrix} X_i \\ X_j \\ X_k \end{bmatrix} = N\left(\begin{bmatrix} \mu_i \\ \mu_j \\ \mu_k \end{bmatrix}, \begin{pmatrix} \sigma^2_{i,i} & \cdots \\ \vdots & \ddots \end{pmatrix} \right)$$

Multivariate normal

$$X \sim N(\mu, \Sigma), \quad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix}, \quad \Sigma = \begin{pmatrix} \sigma^2_{1,1} & \cdots \\ \vdots & \ddots \end{pmatrix}$$

Any linear combination of X, $AX + b$ is

$$AX + b \sim N(A\mu + b, A\Sigma A^T)$$

Multivariate normal

$$X \sim N(\mu, \Sigma), \quad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix}, \quad \Sigma = \begin{pmatrix} \sigma^2_{1,1} & \cdots \\ \vdots & \ddots \end{pmatrix}$$

Conditional distribution:

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{1,1} & \Sigma_{1,2} \\ \Sigma_{2,1} & \Sigma_{2,2} \end{bmatrix} \right)$$

$$X_1 \mid X_2 = x_2 \sim N(\mu_1 + \Sigma_{1,2}\Sigma_{2,2}^{-1}(x_2 - \mu_2), \\ \Sigma_{1,1} - \Sigma_{1,2}\Sigma_{2,2}^{-1}\Sigma_{2,1})$$

Covariance matrix

definition

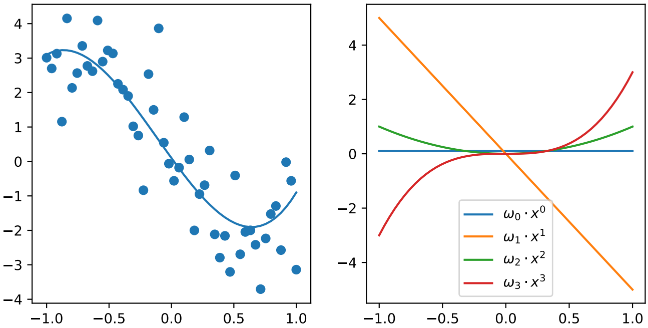
$$cov(X,Y) = E[(X-\mu_X)(Y-\mu_Y)] \quad (1)$$
$$= \frac{cor(X,Y)}{\sigma_X \sigma_Y} \quad (2)$$

must be positive (semi)definite

$$x^T \Sigma x \geq 0 \quad \forall x$$

Bayesian linear regression

$$\omega_0 = 0.1,\ \omega_1 = -5.0,\ \omega_2 = 1.0,\ \omega_3 = 3.0$$



$$y(x) = f(x) + \epsilon(x)$$

$$f(x) = \phi_p(x)^T \omega$$

$$\phi_p(x) = \{1, x, x^2, \ldots, x^p\}$$

$$\omega \sim N(0, \Sigma_\omega)$$

$$\epsilon \sim N(0, \sigma_y^2)$$

Distribution of $f(x)$

$$f(x) = \phi_p(x)^T \omega$$

Mean

$$E[f(x)] = \phi(x)^T E[\omega] = 0$$

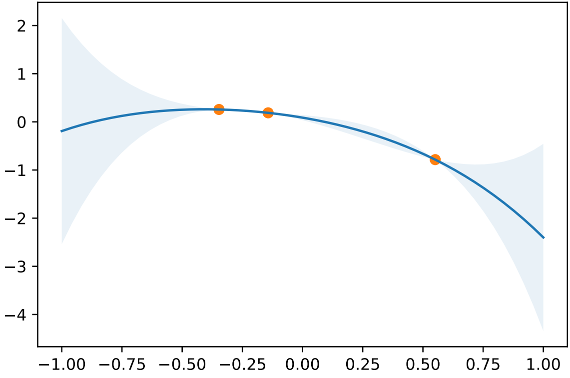
Covariance

$$\begin{aligned} Cov[f(x), f(x')] = E[f(x)f(x')] &= \phi(x)^T E[\omega\omega^T]\phi(x') \\ &= \phi(x)^T \Sigma_\omega \phi(x') \\ &= \kappa_f(x, x') \end{aligned}$$

Conditional distribution

$$\begin{bmatrix} f(X_1) \\ f(X_2) \end{bmatrix} \sim N\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \kappa_1 & \kappa_{1,2} \\ \kappa_{2,1} & \kappa_2 \end{bmatrix} \right)$$

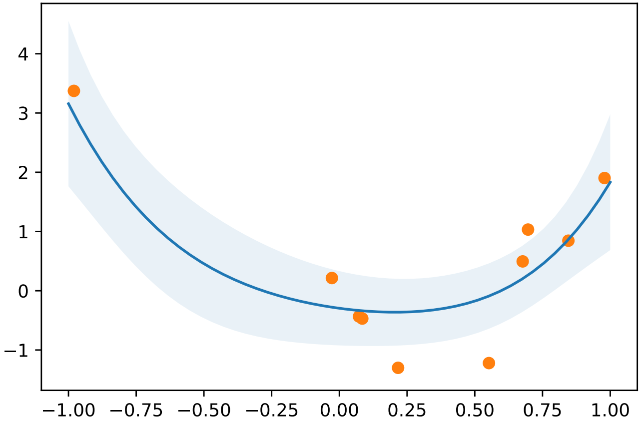
$$f(X_2) \mid f(X_1) \sim N(\kappa_{2,1}\kappa_1^{-1} f(X_1), \kappa_2 - \kappa_{2,1}\kappa_1^{-1}\kappa_{1,2})$$



adding observation noise is easy

$$\begin{bmatrix} y(X_1) \\ f(X_2) \end{bmatrix} \sim N\Big(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \kappa_1 + \sigma_y^2 I & \kappa_{1,2} \\ \kappa_{2,1} & \kappa_2 \end{bmatrix} \Big)$$

$$f(X_2) \mid y(X_1) \sim N(\kappa_{2,1}(\kappa_1 {\color{red}+\sigma_y^2 I})^{-1} y(X_1), \\ \kappa_2 - \kappa_{2,1}(\kappa_1 {\color{red}+\sigma_y^2 I})^{-1}\kappa_{1,2})$$



What did we learn?

- A procedure for building a covariance matrix (and mean) can describe a distribution on functions
- We can compute the conditional distribution of $f(x)$ at any x, given observations
- The (co)variance of these functions are dependent on the *input* (e.g. x), and not the output (y)

Formal derivation of GPs

Distribution of f

- f has a GP prior: $f(X) \sim GP\Big(\mu(X), \kappa(X)\Big)$
- which is MVN for finite X:

$$p(f(X)) = MVN\Big(\mu(X), \kappa(X)\Big)$$

Noisy observation $y(x)$

- conditional on f: $y(X)|f(X) \sim N\Big(f(x), \sigma_y^2 I\Big)$
- marginalized:

$$\begin{aligned} p(y) &= \int p(y|f)p(f)\partial f \\ &= N\big(\mu(X), \kappa(X) + \sigma_y^2 I\big) \end{aligned}$$

Idea: what if we changed $k_f(x, x')$ to a different function?

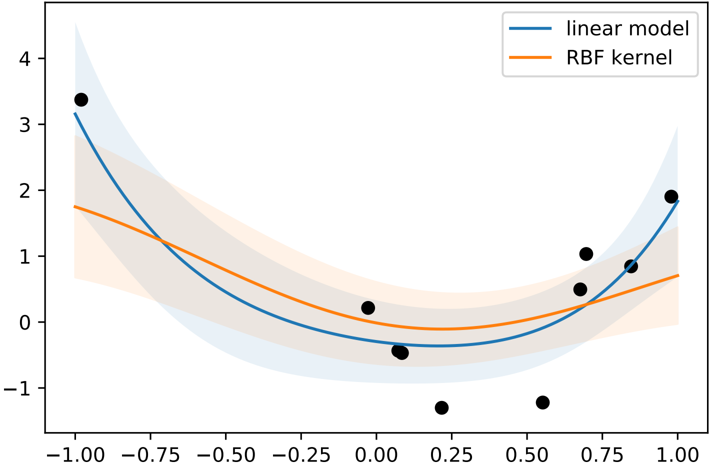
Radial basis function

$$cov(f(x_1), f(x_2)) = \kappa_{\text{RBF}}(x, x') = \sigma^2 \cdot \exp\left(\frac{-|x_1 - x_2|^2}{\ell}\right)$$

Define $f(x)$ as:

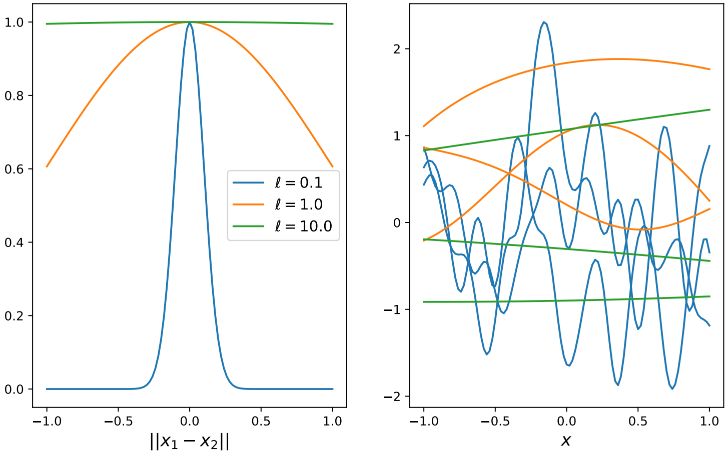
$$f(x) \sim N\Big(\mathbf{0}, \boldsymbol{\kappa}(x) = \{\kappa_{\text{RBF}}(x_i, x_j)\}\Big)$$

Replace κ_f with κ_{RBF}



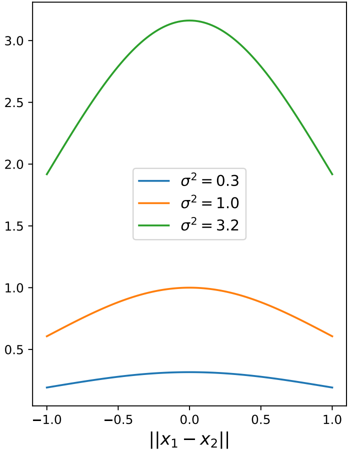
Lengthscale changes the smoothness

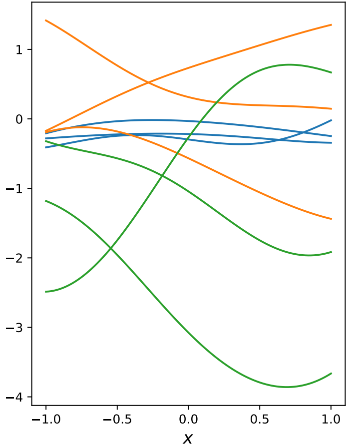
$$cov(f(x_1), f(x_2)) = \kappa_{\mathrm{RBF}}(x, x') = \sigma^2 \cdot \exp\left(\frac{-|x_1 - x_2|^2}{\ell}\right)$$



Variance changes the magnitude

$$cov(f(x_1), f(x_2)) = \kappa_{\text{RBF}}(x, x') = \sigma^2 \cdot \exp\left(\frac{-|x_1 - x_2|^2}{\ell}\right)$$



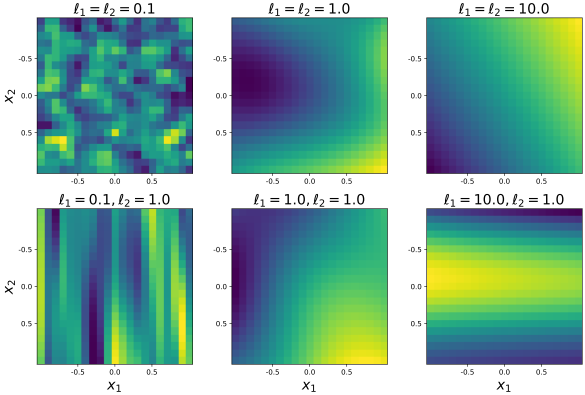


Support for multi-dim x

$x = \{x_1, x_2, \ldots, x_p\}$,

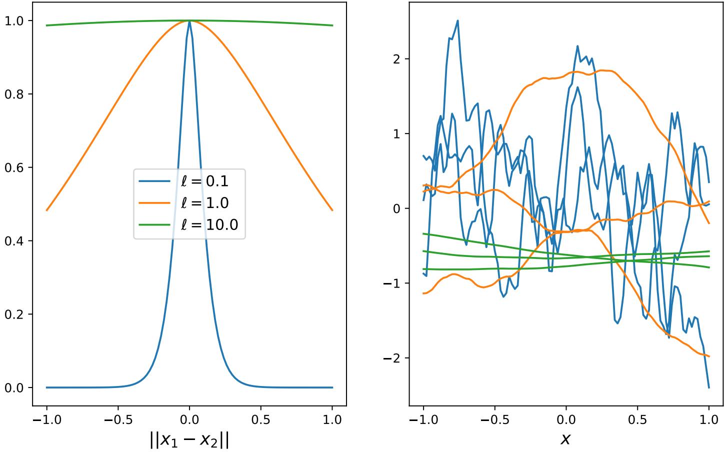
$$\kappa(x_1, x_2) = \sigma^2 \cdot \exp\left(\sum_{i=1}^{p} \frac{-|x_{1,i} - x_{2,i}|^2}{\ell_p}\right)$$

Referred to as auto-relevance detection (ARD)



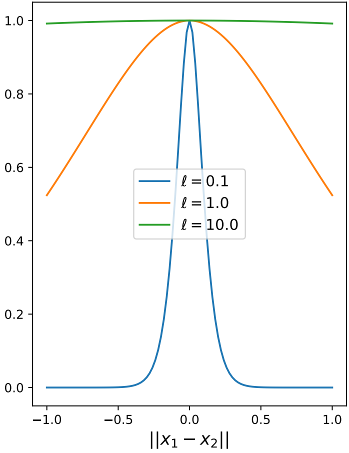
Matern kernel

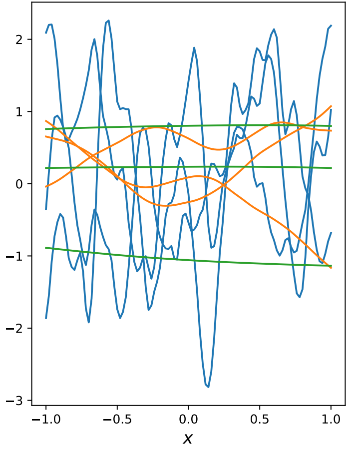
$$\kappa_{\frac{3}{2}} = \left(1 + \frac{\sqrt{3}r}{\ell}\right) \cdot \exp\left(\frac{-\sqrt{3}r}{\ell}\right)$$



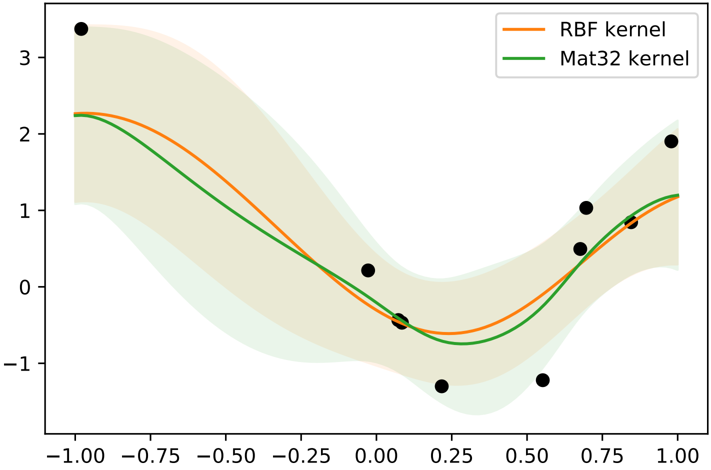
Matern kernel

$$\kappa_{\frac{5}{2}} = \left(1 + \frac{\sqrt{5}r}{\ell} + \frac{5r^2}{3\ell^2}\right) \cdot \exp\left(\frac{-\sqrt{3}r}{\ell}\right)$$





κ_{RBF} vs κ_{Mat32}



Kernel operations

if κ_A and κ_B are both valid kernels, then so are:

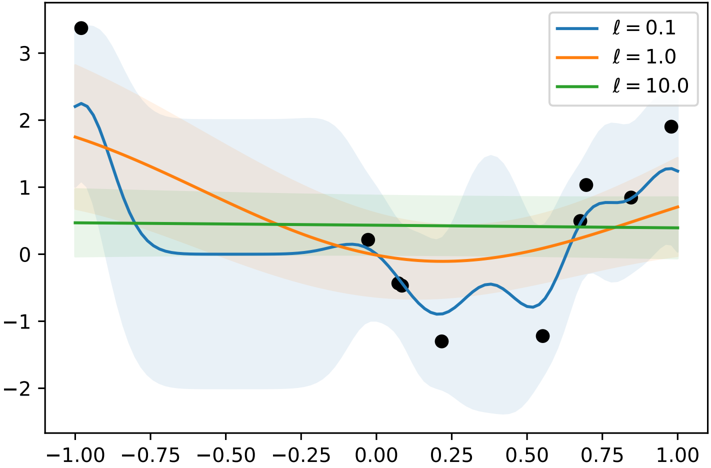
- $\kappa_A + \kappa_B$, interpret as

$$y(x) = f_a(x) + f_b(x)$$

- $\kappa_A * \kappa_B$, interpret as

$$y(x) = f_a(x) \times f_b(x)$$

Hyperparameters change model interpretation



How do we determine hyperparameters?

1. Maximize likelihood

- gradient optimization, point estimate
- **pros:** many existing implementations (e.g. *GPy*), often gets answer the fastest
- **cons:** requires "conjugacy" b/w f and y for analytical sol'n

2. Model hyperparameter uncertainty

- Posterior sampling (MCMC)
- **pros:** Doesn't care about form of $p(y)$, can be placed in larger model easily
- **cons:** sampling time (can be long)

1. Maximal likelihood

$$\mathcal{L}(\theta) = \log p(y|\theta) = \frac{1}{2}\log\det\Sigma(\theta) + \frac{1}{2}y^t\Sigma(\theta)^{-1}y + \frac{N}{2}\log(2\pi)$$

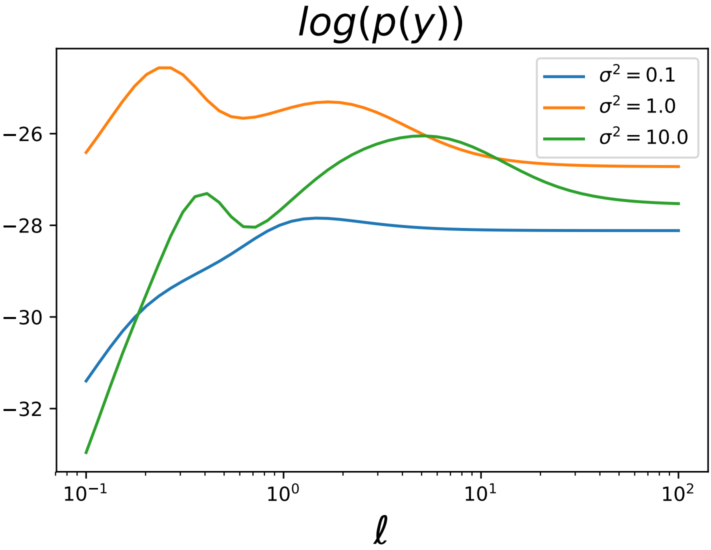
where: $\Sigma(\theta) = \kappa(\theta) + \sigma^2 I$, $\theta = \{\ell, \sigma_f^2, \sigma_y^2\}$

- $\mathcal{L}(\theta)$ is non-convex
- Use gradient based optimization:

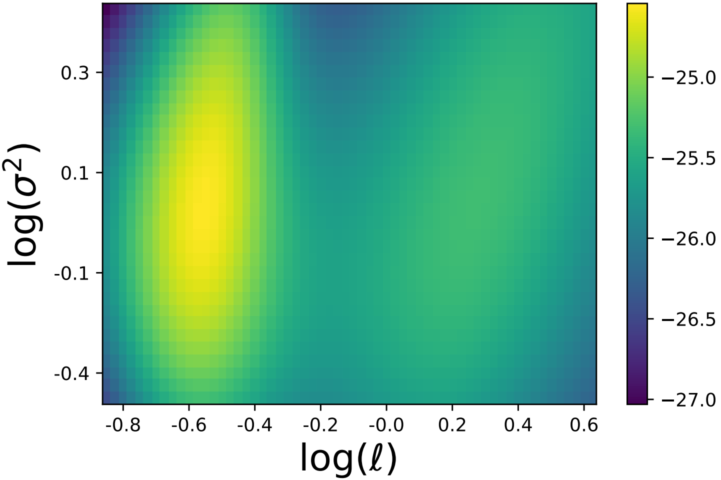
$$\frac{\partial\mathcal{L}}{\partial\theta_i} = \frac{1}{2}\mathrm{tr}\Sigma(\theta)\frac{\partial\Sigma(\theta)}{\partial\theta_i} + \frac{1}{2}y^t\Sigma(\theta)^{-1}\frac{\partial\Sigma(\theta)}{\partial\theta_i}\Sigma(\theta)^{-1}y$$

- $\frac{\partial\Sigma(\theta)}{\partial\theta_i}$: need to compute derivative of kernel function *wrt* each hyperparameter
- $\Sigma(\theta)^{-1}$: $O(n^3)$ operation

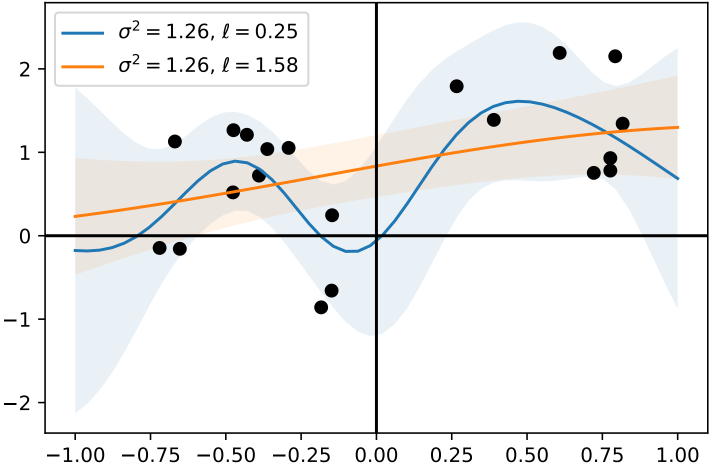
Data likelihood as a function of hyperparameters



Data likelihood as a function of hyperparameters



Data likelihood as a function of hyperparameters



What next?

Further reading

- Rasmussen and Williams *Gaussian Processes for Machine Learning*, 2006: Chs. 2,4,5

Remaining sessions

- Applications (*GPy* and *Stan*)
- Case studies