# Advanced Statistical Inference
# Gaussian Processes

Maurizio Filippone
Maurizio.Filippone@eurecom.fr

Department of Data Science
EURECOM

# Suggested readings

**Gaussian Processes for Machine Learning**

Carl E. Rasmussen and Christopher K. I. Williams

**Pattern Recognition and Machine Learning**

C. Bishop

# Gaussian Processes

- Linear models requires specifying a set of basis functions
  - Polynomials, Trigonometric, . . .??

# Gaussian Processes

- Linear models requires specifying a set of basis functions
  - Polynomials, Trigonometric, . . .??
- Can we use Bayesian inference to let data tell us?

# Gaussian Processes

- Linear models requires specifying a set of basis functions
  - Polynomials, Trigonometric, ...??
- Can we use Bayesian inference to let data tell us?
- Gaussian Processes work implicitly with an infinite set of basis functions and learn a probabilistic combination of these

# Gaussian Processes

Gaussian Processes can be explained in two ways

- Weight Space View
  - Bayesian linear regression with infinite basis functions
- Function Space View
  - Defined as priors over functions

# Gaussian Processes

Gaussian Processes can be explained in two ways

- **Weight Space View**
  - **Bayesian linear regression with infinite basis functions**
- Function Space View
  - Defined as priors over functions

# Bayesian Linear Regression - recap

- Modeling observations as noisy realizations of a linear combination of the features:

$$p(\mathbf{y}|\mathbf{w}, \mathbf{X}, \sigma^2) = \mathcal{N}(\mathbf{X}\mathbf{w}, \sigma^2\mathbf{I})$$

# Bayesian Linear Regression - recap

- Modeling observations as noisy realizations of a linear combination of the features:

$$p(\mathbf{y}|\mathbf{w}, \mathbf{X}, \sigma^2) = \mathcal{N}(\mathbf{X}\mathbf{w}, \sigma^2\mathbf{I})$$

- Gaussian prior over model parameters:

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{0}, \mathbf{S})$$

# Bayesian Linear Regression - recap

- Posterior **must be** Gaussian

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}, \sigma^2) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- Covariance:

$$\boldsymbol{\Sigma} = \left(\frac{1}{\sigma^2}\mathbf{X}^\top\mathbf{X} + \mathbf{S}^{-1}\right)^{-1}$$

- Mean:

$$\boldsymbol{\mu} = \frac{1}{\sigma^2}\boldsymbol{\Sigma}\,\mathbf{X}^\top\mathbf{y}$$

- Predictions

$$p(y_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*, \sigma^2) = \mathcal{N}(\mathbf{x}_*^\top\boldsymbol{\mu}, \sigma^2 + \mathbf{x}_*^\top\boldsymbol{\Sigma}\mathbf{x}_*)$$

# Introducing basis functions

- Imagine transforming the inputs using a set of $D$ functions

$$\mathbf{x} \rightarrow \boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_D(\mathbf{x}))^\top$$

- The functions $\phi_1(\mathbf{x})$ are also known as basis functions
- Define:

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \ldots & \phi_D(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_N) & \ldots & \phi_D(\mathbf{x}_N) \end{bmatrix}$$

# Introducing basis functions

- Applying Bayesian Linear Regression on the transformed features gives

$$p(\mathbf{w}|\mathbf{X}, \mathbf{y}, \sigma^2) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- Covariance:

$$\boldsymbol{\Sigma} = \left(\frac{1}{\sigma^2}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \mathbf{S}^{-1}\right)^{-1}$$

- Mean:

$$\boldsymbol{\mu} = \frac{1}{\sigma^2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^\top\mathbf{y}$$

- Predictions:

$$p(y_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*, \sigma^2) = \mathcal{N}(\phi_*^\top\boldsymbol{\mu}, \sigma^2 + \phi_*^\top\boldsymbol{\Sigma}\phi_*)$$

# Bayesian Linear Regression as a Kernel Machine

- We are going to show that predictions can be expressed exclusively in terms of scalar products as follows

$$k(\mathbf{x}, \mathbf{x}') = \psi(\mathbf{x})^\top \psi(\mathbf{x}')$$

- This allows us to work with either $k(\cdot, \cdot)$ or $\psi(\cdot)$
- Why is this useful??

# Bayesian Linear Regression as a Kernel Machine

- Working with $\psi(\cdot)$ costs $O(D^2)$ storage, $O(D^3)$ time
- Working with $k(\cdot,\cdot)$ costs $O(N^2)$ storage, $O(N^3)$ time

# Bayesian Linear Regression as a Kernel Machine

- Working with $\psi(\cdot)$ costs $O(D^2)$ storage, $O(D^3)$ time
- Working with $k(\cdot, \cdot)$ costs $O(N^2)$ storage, $O(N^3)$ time
- Pick the one that makes computations faster . . . or

## Bayesian Linear Regression as a Kernel Machine

- Working with $\psi(\cdot)$ costs $O(D^2)$ storage, $O(D^3)$ time
- Working with $k(\cdot, \cdot)$ costs $O(N^2)$ storage, $O(N^3)$ time
- Pick the one that makes computations faster ... or
- What if we could pick $k(\cdot, \cdot)$ so that $\psi(\cdot)$ is infinite dimensional?

# Kernels

- It is possible to show that for

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2}\right)$$

  there exists a corresponding $\psi(\cdot)$ that is infinite dimensional!!!
- There are other kernels satisfying this property

# Kernels

## Proof that the Gaussian kernel induces an infinite dimensional $\psi(\cdot)$

- For simplicity consider one dimensional inputs $x$, $z$
- Expand the Gaussian kernel $k(x, z)$ as

$$\exp\left(-\frac{(x-z)^2}{2}\right) = \exp\left(-\frac{x^2}{2}\right)\exp\left(-\frac{z^2}{2}\right)\exp\left(xz\right)$$

- Focusing on the last term and applying the Taylor expansion of the $\exp(\cdot)$ function

$$\exp\left(xz\right) = 1 + (xz) + \frac{(xz)^2}{2!} + \frac{(xz)^3}{3!} + \frac{(xz)^4}{4!} + \dots$$

# Kernels

## Proof that the Gaussian kernel induces an infinite dimensional $\psi(\cdot)$

- Define the infinite dimensional mapping

$$\psi(x) = \exp\left(-\frac{x^2}{2}\right)\left(1, x, \frac{x^2}{\sqrt{2!}}, \frac{x^3}{\sqrt{3!}}, \frac{x^4}{\sqrt{4!}}, \ldots\right)^\top$$

- It is easy to verify that

$$k(x, z) = \exp\left(-\frac{(x-z)^2}{2}\right) = \psi(x)^\top \psi(z)$$

# Bayesian Linear Regression as a Kernel Machine

## Proof

- To show that Bayesian Linear Regression can be formulated through scalar products only, we need Woodbury identity:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$

- Do not memorize this!

# Bayesian Linear Regression as a Kernel Machine

## Proof

- Woodbury identity:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$

- We can rewrite:

$$\begin{aligned} \mathbf{\Sigma} &= \left(\frac{1}{\sigma^2}\mathbf{\Phi}^\top\mathbf{\Phi} + \mathbf{S}^{-1}\right)^{-1} \\ &= \mathbf{S} - \mathbf{S}\mathbf{\Phi}^\top\left(\sigma^2\mathbf{I} + \mathbf{\Phi}\mathbf{S}\mathbf{\Phi}^\top\right)^{-1}\mathbf{\Phi}\mathbf{S} \end{aligned}$$

- We set $A = \mathbf{S}$, $U = V^\top = \mathbf{\Phi}^\top$, and $C = \frac{1}{\sigma^2}\mathbf{I}$

# Bayesian Linear Regression as a Kernel Machine

## Proof

- Mean and variance of the predictions:

$$p(y_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*, \sigma^2) = \mathcal{N}(\phi_*^\top \boldsymbol{\mu}, \sigma^2 + \phi_*^\top \boldsymbol{\Sigma} \phi_*)$$

- Rewrite the variance:

$$\begin{aligned} \sigma^2 &+ \phi_*^\top \boldsymbol{\Sigma} \phi_* = \\ \sigma^2 &+ \phi_*^\top \mathbf{S} \phi_* - \phi_*^\top \mathbf{S} \boldsymbol{\Phi}^\top \left(\sigma^2 \mathbf{I} + \boldsymbol{\Phi} \mathbf{S} \boldsymbol{\Phi}^\top\right)^{-1} \boldsymbol{\Phi} \mathbf{S} \phi_* \end{aligned}$$

. . . continued

# Bayesian Linear Regression as a Kernel Machine

## Proof

- Mean and variance of the predictions:

$$p(y_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*, \sigma^2) = \mathcal{N}(\phi_*^\top \boldsymbol{\mu}, \sigma^2 + \phi_*^\top \mathbf{\Sigma} \phi_*)$$

- Rewrite the variance:

$$\begin{aligned} \sigma^2 \quad &+ \quad \phi_*^\top \mathbf{S} \phi_* - \phi_*^\top \mathbf{S} \mathbf{\Phi}^\top \left(\sigma^2 \mathbf{I} + \mathbf{\Phi} \mathbf{S} \mathbf{\Phi}^\top\right)^{-1} \mathbf{\Phi} \mathbf{S} \phi_* = \\ \sigma^2 \quad &+ \quad k_{**} - \mathbf{k}_*^\top \left(\sigma^2 \mathbf{I} + \mathbf{K}\right)^{-1} \mathbf{k}_* \end{aligned}$$

- Where the mapping defining the kernel is

$$\psi(\mathbf{x}) = \mathbf{S}^{1/2} \phi(\mathbf{x}) \qquad \text{and}$$

$$\begin{aligned} k_{**} &= k(\mathbf{x}_*, \mathbf{x}_*) = \psi(\mathbf{x}_*)^\top \psi(\mathbf{x}_*) \\ (\mathbf{k}_*)_i &= k(\mathbf{x}_*, \mathbf{x}_i) = \psi(\mathbf{x}_*)^\top \psi(\mathbf{x}_i) \\ (\mathbf{K})_{ij} &= k(\mathbf{x}_i, \mathbf{x}_j) = \psi(\mathbf{x}_i)^\top \psi(\mathbf{x}_j) \end{aligned}$$

# Bayesian Linear Regression as a Kernel Machine

## Proof

- Mean and variance of the predictions:

$$p(y_*|\mathbf{X}, \mathbf{y}, \mathbf{x}_*, \sigma^2) = \mathcal{N}(\phi_*^\top \boldsymbol{\mu}, \sigma^2 + \phi_*^\top \boldsymbol{\Sigma} \phi_*)$$

- Rewrite the mean:

$$\begin{aligned}
\phi_*^\top \boldsymbol{\mu} &= \frac{1}{\sigma^2} \phi_*^\top \boldsymbol{\Sigma} \boldsymbol{\Phi}^\top \mathbf{y} \\
&= \frac{1}{\sigma^2} \phi_*^\top \left( \mathbf{S} - \mathbf{S} \boldsymbol{\Phi}^\top \left( \sigma^2 \mathbf{I} + \boldsymbol{\Phi} \mathbf{S} \boldsymbol{\Phi}^\top \right)^{-1} \boldsymbol{\Phi} \mathbf{S} \right) \boldsymbol{\Phi}^\top \mathbf{y} \\
&= \frac{1}{\sigma^2} \phi_*^\top \mathbf{S} \boldsymbol{\Phi}^\top \left( \mathbf{I} - \left( \sigma^2 \mathbf{I} + \boldsymbol{\Phi} \mathbf{S} \boldsymbol{\Phi}^\top \right)^{-1} \boldsymbol{\Phi} \mathbf{S} \boldsymbol{\Phi}^\top \right) \mathbf{y} \\
&= \frac{1}{\sigma^2} \phi_*^\top \mathbf{S} \boldsymbol{\Phi}^\top \left( \mathbf{I} - \left( \mathbf{I} + \frac{\boldsymbol{\Phi} \mathbf{S} \boldsymbol{\Phi}^\top}{\sigma^2} \right)^{-1} \frac{\boldsymbol{\Phi} \mathbf{S} \boldsymbol{\Phi}^\top}{\sigma^2} \right) \mathbf{y}
\end{aligned}$$

. . . continued

# Bayesian Linear Regression as a Kernel Machine

## Proof

- Define $\mathbf{H} = \frac{\mathbf{\Phi S \Phi}^\top}{\sigma^2}$
- The term in the parenthesis

$$\left(\mathbf{I} - \left(\mathbf{I} + \frac{\mathbf{\Phi S \Phi}^\top}{\sigma^2}\right)^{-1} \frac{\mathbf{\Phi S \Phi}^\top}{\sigma^2}\right)$$

  becomes

$$\left(\mathbf{I} - (\mathbf{I} + \mathbf{H})^{-1}\mathbf{H}\right) = \mathbf{I} - (\mathbf{H}^{-1} + \mathbf{I})^{-1}$$

- Using Woodbury ($A, U, V = \mathbf{I}$ and $C = \mathbf{H}^{-1}$)

$$\mathbf{I} - (\mathbf{H}^{-1} + \mathbf{I})^{-1} = (\mathbf{I} + \mathbf{H})^{-1}$$

# Bayesian Linear Regression as a Kernel Machine

## Proof

- Substituting into the expression of the predictive mean

$$
\begin{aligned}
\phi_*^\top \boldsymbol{\mu} &= \frac{1}{\sigma^2}\phi_*^\top \mathbf{S}\boldsymbol{\Phi}^\top \left(\mathbf{I} - \left(\mathbf{I} + \frac{\boldsymbol{\Phi}\mathbf{S}\boldsymbol{\Phi}^\top}{\sigma^2}\right)^{-1} \frac{\boldsymbol{\Phi}\mathbf{S}\boldsymbol{\Phi}^\top}{\sigma^2}\right)\mathbf{y} \\
&= \frac{1}{\sigma^2}\phi_*^\top \mathbf{S}\boldsymbol{\Phi}^\top \left(\mathbf{I} + \frac{\boldsymbol{\Phi}\mathbf{S}\boldsymbol{\Phi}^\top}{\sigma^2}\right)^{-1}\mathbf{y} \\
&= \phi_*^\top \mathbf{S}\boldsymbol{\Phi}^\top \left(\sigma^2\mathbf{I} + \boldsymbol{\Phi}\mathbf{S}\boldsymbol{\Phi}^\top\right)^{-1}\mathbf{y} \\
&= \mathbf{k}_*^\top \left(\sigma^2\mathbf{I} + \mathbf{K}\right)^{-1}\mathbf{y}
\end{aligned}
$$

- All definitions as in the case of the variance

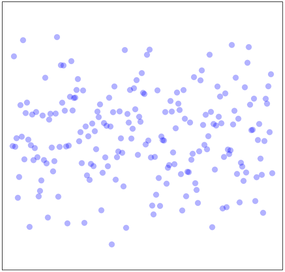
# Gaussian Processes

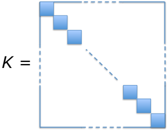
Gaussian Processes can be explained in two ways

- Weight Space View
  - Bayesian linear regression with infinite basis functions
- **Function Space View**
  - **Defined as priors over functions**

# Gaussian Processes - Prior over Functions

- Consider an infinite number of Gaussian random variables
- Think of them as indexed by the real line and as independent
- Denote them as $f(x)$





# Kernel

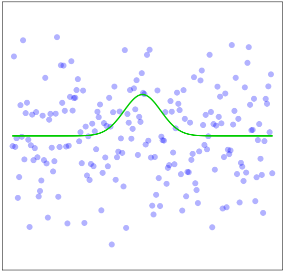
- Consider the Gaussian kernel again

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp(-\beta \|\mathbf{x} - \mathbf{x}'\|^2)$$

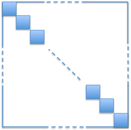
- We introduced some parameters for added flexibility

# Gaussian Processes - Prior over Functions

- Impose covariance using the kernel function

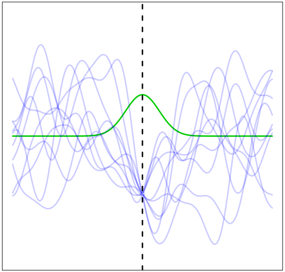


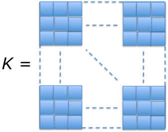
$K =$



# Gaussian Processes - Prior over Functions

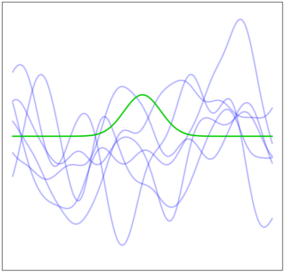
- Draw the infinite random variables again fixing one of them (the one at $x = 0$)

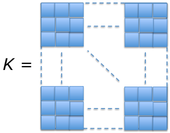




# Gaussian Processes - Prior over Functions

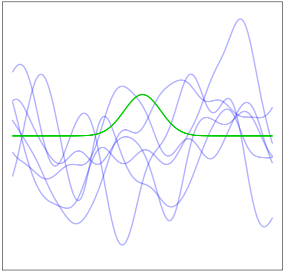
- Draw the infinite random variables again allowing the one at $x = 0$ to be random too

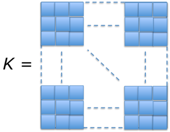




# Gaussian Processes - Prior over Functions

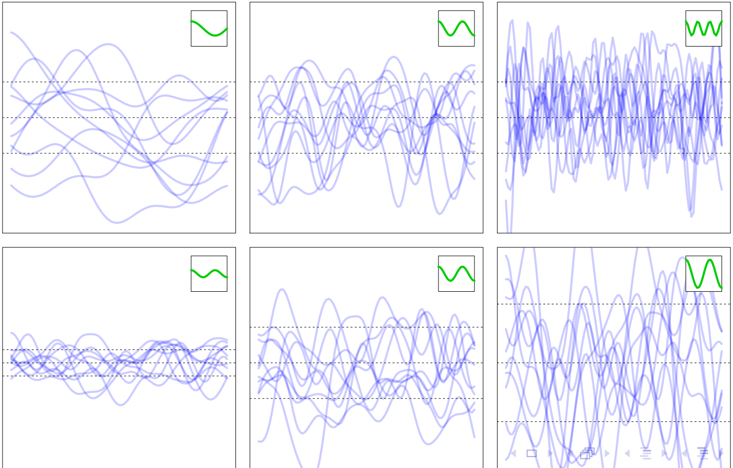
- This can be used as a prior over functions!





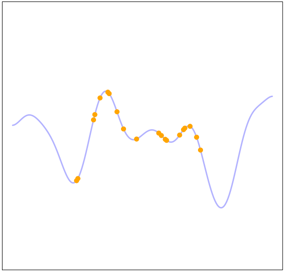
# Gaussian Processes - Priors over Functions

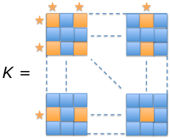
- Infinite Gaussian random variables with parameterized and input-dependent covariance



# Gaussian Processes - Prior over Functions

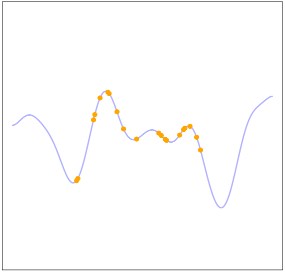
- The distribution of $N$ random variables $f(x_1), \ldots, f(x_N)$ depends exclusively on the corresponding rows and columns of the infinite by infinite kernel matrix $K$

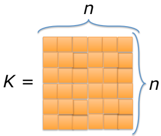




# Gaussian Processes - Prior over Functions

- The distribution of $N$ random variables $f(x_1), \ldots, f(x_N)$ depends exclusively on the corresponding rows and columns of the infinite by infinite kernel matrix $K$





# Gaussian Processes - Prior over Functions

- The marginal distribution of $\mathbf{f} = (f(x_1), \dots, f(x_N))^\top$ is

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{0}, \mathbf{K})$$

- The conditional distribution of $f_*$ given $\mathbf{f}$

$$p(f_*|\mathbf{f}, \mathbf{x}_*, \mathbf{X}) = \mathcal{N}(\bar{m}, \bar{s}^2)$$

with

$$\bar{m} = \mathbf{k}_*^\top \mathbf{K}^{-1} \mathbf{f}$$

$$\bar{s}^2 = k_{**} - \mathbf{k}_*^\top \mathbf{K}^{-1} \mathbf{k}_*$$

# Gaussian Processes - Prior over Functions

- Remember that when we modeled labels $\mathbf{y}$ in the linear model we assumed noise with variance $\sigma$ around $\mathbf{w}^\top \mathbf{x}$
- We can do the same in Gaussian processes

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{N} p(y_i|f_i)$$

with

$$p(y_i|f_i) = \mathcal{N}(y_i|f_i, \sigma^2)$$

- Likelihood and prior are both Gaussian - conjugate!

# Gaussian Processes - Prior over Functions

- Remember that when we modeled labels $\mathbf{y}$ in the linear model we assumed noise with variance $\sigma$ around $\mathbf{w}^\top \mathbf{x}$
- We can do the same in Gaussian processes

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{N} p(y_i|f_i)$$

with

$$p(y_i|f_i) = \mathcal{N}(y_i|f_i, \sigma^2)$$

- Likelihood and prior are both Gaussian - conjugate!
- We can integrate out the Gaussian process prior over $\mathbf{f}$

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

- This gives

$$p(\mathbf{y}|\mathbf{X}) = \mathcal{N}(\mathbf{0}, \mathbf{K} + \sigma^2\mathbf{I})$$

# Gaussian Processes - Prior over Functions

- We can derive the predictive distribution as follows:

$$p(f_*|\mathbf{y}, \mathbf{x}_* \mathbf{X}) = \int p(f_*|\mathbf{f}, \mathbf{x}_*, \mathbf{X}) p(\mathbf{f}|\mathbf{y}, \mathbf{X}) d\mathbf{f} df_* = \mathcal{N}(m, s^2)$$

  with

$$m = \mathbf{k}_*^\top \left(\mathbf{K} + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{y}$$

$$s^2 = k_{**} - \mathbf{k}_*^\top \left(\mathbf{K} + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{k}_*$$

- Same expression as in the "Weight-Space View" section

# Gaussian Processes - Prior over Functions

- We can also make predictions as follows:

$$\begin{aligned} p(y_*|\mathbf{y}, \mathbf{x}_* \mathbf{X}) &= \int p(y_*|f_*)p(f_*|\mathbf{f}, \mathbf{x}_*, \mathbf{X})p(\mathbf{f}|\mathbf{y}, \mathbf{X})d\mathbf{f}df_* \\ &= \mathcal{N}(m_y, s_y^2) \end{aligned}$$

with

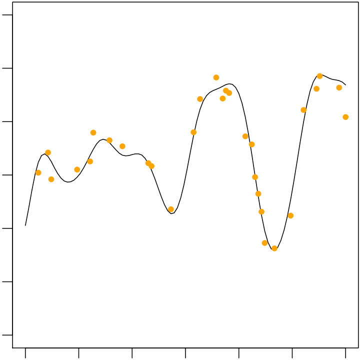
$$m_y = \mathbf{k}_*^\top \left(\mathbf{K} + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{y}$$

$$s_y^2 = \sigma^2 + k_{**} - \mathbf{k}_*^\top \left(\mathbf{K} + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{k}_*$$

- Same expression as in the “Weight-Space View” section

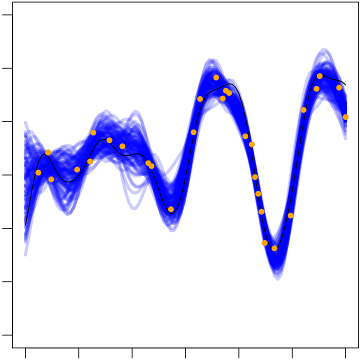
# Gaussian Processes - Regression example

- Some data generated as a noisy version of some function



# Gaussian Processes - Regression example

- Draws from the posterior distribution over $f_*$ on the real line



# Optimization of Gaussian Process parameters

- The kernel has parameters that have to be tuned

$$k(\mathbf{x}, \mathbf{x}') = \alpha \exp(-\beta \|\mathbf{x} - \mathbf{x}'\|^2)$$

  ... and there is also the noise parameter $\sigma^2$.
- Define $\boldsymbol{\theta} = (\alpha, \beta, \sigma^2)$
- How should we tune them?

# Optimization of Gaussian Process parameters

- Define $\mathbf{C} = \mathbf{K} + \sigma^2 \mathbf{I}$
- Maximize the logarithm of the likelihood

$$p(\mathbf{y}|\mathbf{X}, \theta) = \mathcal{N}(\mathbf{0}, \mathbf{C})$$

that is

$$-\frac{1}{2} \log |\mathbf{C}| - \frac{1}{2} \mathbf{y}^\top \mathbf{C}^{-1} \mathbf{y} + \text{const.}$$

- Derivatives can be useful for gradient-based optimization

$$\frac{\partial \log[p(\mathbf{y}|\mathbf{X}, \theta)]}{\partial \boldsymbol{\theta}_i}$$

# Optimization of Gaussian Process parameters

- Log-likelihood

$$-\frac{1}{2}\log|\mathbf{C}| - \frac{1}{2}\mathbf{y}^\top \mathbf{C}^{-1}\mathbf{y} + \text{const.}$$

- Derivatives can be useful for gradient-based optimization:

$$\frac{\partial \log[p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta})]}{\partial \boldsymbol{\theta}_i} = -\frac{1}{2}\mathrm{Tr}\left(\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \boldsymbol{\theta}_i}\right) + \frac{1}{2}\mathbf{y}^\top \mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \boldsymbol{\theta}_i}\mathbf{C}^{-1}\mathbf{y}$$

# Summary

- Introduced Gaussian Processes
  - Weight space view
  - Function space view
- Gaussian processes for regression
- Optimization of kernel parameters
- To think about:
  - Gaussian processes for classification?
  - Scalability?