Machine Learning and Statistics in Genetics and Genomics

VI: Introduction to Gaussian Processes

Christoph Lippert

Microsoft Research
eScience group

Los Angeles , USA

Current topics in computational biology
UCLA
Winter quarter 2014

Outline

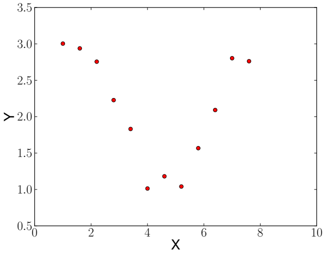
Outline

Why Gaussian processes?

- So far: linear models with a finite number of basis functions, e.g. $\phi(x) = (1, x, x^2, \ldots, x^K)$
- Open questions:
  - How to design a suitable basis?
  - How many basis functions to pick?
- Gaussian processes: accurate and flexible regression method yielding predictions alongside with error bars.

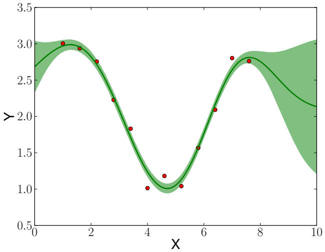
Why Gaussian processes?

- So far: linear models with a finite number of basis functions, e.g. $\phi(x) = (1, x, x^2, \ldots, x^K)$
- Open questions:
  - How to design a suitable basis?
  - How many basis functions to pick?
- Gaussian processes: accurate and flexible regression method yielding predictions alongside with error bars.



Why Gaussian processes?

- So far: linear models with a finite number of basis functions, e.g. $\phi(x) = (1, x, x^2, \ldots, x^K)$
- Open questions:
  - How to design a suitable basis?
  - How many basis functions to pick?
- Gaussian processes: accurate and flexible regression method yielding predictions alongside with error bars.



Making predictions with variance component models

- Linear model, accounting for a set of measured SNPs $\boldsymbol{X}$
$$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\theta}, \sigma^2) = \mathcal{N}\left(\boldsymbol{y} \mid \sum_{s=1}^{S} \boldsymbol{x}_s \theta_s, \sigma^2 \boldsymbol{I}\right)$$
- Prediction at unseen test input given max. likelihood weight:
$$p(y^\star \mid \boldsymbol{x}^\star, \hat{\boldsymbol{\theta}}) = \mathcal{N}\left(y^\star \mid \boldsymbol{x}^\star \hat{\boldsymbol{\theta}}, \sigma^2\right)$$
- Marginal likelihood
$$p(\boldsymbol{y} \mid \boldsymbol{X}, \sigma^2, \sigma_g^2) = \int_{\boldsymbol{\theta}} \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{\theta}, \sigma^2 \boldsymbol{I}\right) \mathcal{N}\left(\boldsymbol{\theta} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \boldsymbol{I}\right)$$
$$= \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{0}, \underbrace{\sigma_{\mathrm{g}}^2 \boldsymbol{X}\boldsymbol{X}^\top}_{\boldsymbol{K}} + \sigma^2 \boldsymbol{I}\right)$$
- Making predictions with variance component models?

Making predictions with variance component models

- Linear model, accounting for a set of measured SNPs $\boldsymbol{X}$
$$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\theta}, \sigma^2) = \mathcal{N}\left(\boldsymbol{y} \mid \sum_{s=1}^{S} \boldsymbol{x}_s \theta_s, \sigma^2 \boldsymbol{I}\right)$$
- Prediction at unseen test input given max. likelihood weight:
$$p(y^\star \mid \boldsymbol{x}^\star, \hat{\boldsymbol{\theta}}) = \mathcal{N}\left(y^\star \mid \boldsymbol{x}^\star \hat{\boldsymbol{\theta}}, \sigma^2\right)$$
- Marginal likelihood
$$p(\boldsymbol{y} \mid \boldsymbol{X}, \sigma^2, \sigma_g^2) = \int_{\boldsymbol{\theta}} \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{\theta}, \sigma^2 \boldsymbol{I}\right) \mathcal{N}\left(\boldsymbol{\theta} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \boldsymbol{I}\right)$$
$$= \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{0}, \underbrace{\sigma_{\mathrm{g}}^2 \boldsymbol{X}\boldsymbol{X}^\top}_{\boldsymbol{K}} + \sigma^2 \boldsymbol{I}\right)$$
- Making predictions with variance component models?

Making predictions with variance component models

- Linear model, accounting for a set of measured SNPs $\boldsymbol{X}$
  $$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\theta}, \sigma^2) = \mathcal{N}\left(\boldsymbol{y} \mid \sum_{s=1}^{S} \boldsymbol{x}_s \theta_s, \sigma^2 \boldsymbol{I}\right)$$
- Prediction at unseen test input given max. likelihood weight:
  $$p(y^\star \mid \boldsymbol{x}^\star, \hat{\boldsymbol{\theta}}) = \mathcal{N}\left(y^\star \mid \boldsymbol{x}^\star \hat{\boldsymbol{\theta}}, \sigma^2\right)$$
- Marginal likelihood
  $$p(\boldsymbol{y} \mid \boldsymbol{X}, \sigma^2, \sigma_g^2) = \int_{\boldsymbol{\theta}} \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{\theta}, \sigma^2 \boldsymbol{I}\right) \mathcal{N}\left(\boldsymbol{\theta} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \boldsymbol{I}\right)$$
  $$= \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{0}, \underbrace{\sigma_{\mathrm{g}}^2 \boldsymbol{X}\boldsymbol{X}^\top}_{\boldsymbol{K}} + \sigma^2 \boldsymbol{I}\right)$$
- Making predictions with variance component models?

Making predictions with variance component models

- Linear model, accounting for a set of measured SNPs $\boldsymbol{X}$

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\theta}, \sigma^2) = \mathcal{N}\left(\boldsymbol{y} \mid \sum_{s=1}^{S} \boldsymbol{x}_s \theta_s, \sigma^2 \boldsymbol{I}\right)$$

- Prediction at unseen test input given max. likelihood weight:

$$p(y^\star \,|\, \boldsymbol{x}^\star, \hat{\boldsymbol{\theta}}) = \mathcal{N}\left(y^\star \mid \boldsymbol{x}^\star \hat{\boldsymbol{\theta}}, \sigma^2\right)$$

- Marginal likelihood

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \sigma^2, \sigma_g^2) = \int_{\boldsymbol{\theta}} \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{\theta}, \sigma^2 \boldsymbol{I}\right) \mathcal{N}\left(\boldsymbol{\theta} \mid \boldsymbol{0}, \sigma_{\mathrm{g}}^2 \boldsymbol{I}\right)$$

$$= \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{0}, \underbrace{\sigma_{\mathrm{g}}^2 \boldsymbol{X}\boldsymbol{X}^\top}_{\boldsymbol{K}} + \sigma^2 \boldsymbol{I}\right)$$

- Making predictions with variance component models?

Further reading

- C. E. Rasmussen, C. K. Williams
  Gaussian processes for machine learning
  - Comprehensive and freely available introduction (Appendix!).
- Christopher M. Bishop: Pattern Recognition and Machine learning

Outline

Outline

The Gaussian distribution

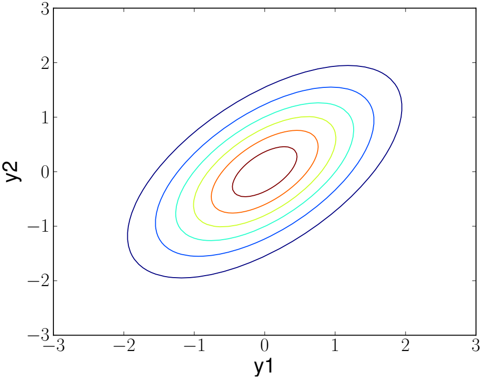
- Gaussian processes are merely based on the good old Gaussian

$$\mathcal{N}\left(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{K}\right) = \frac{1}{\sqrt{|2\pi \boldsymbol{K}|}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{K}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]$$

- Covariance matrix or kernel matrix

A 2D Gaussian

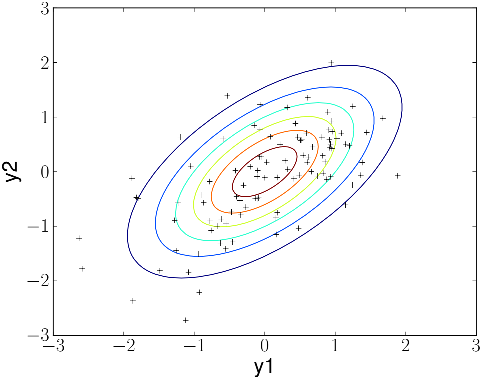
- Probability contour
- Samples



$$\boldsymbol{K} = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 1 \end{bmatrix}$$

A 2D Gaussian

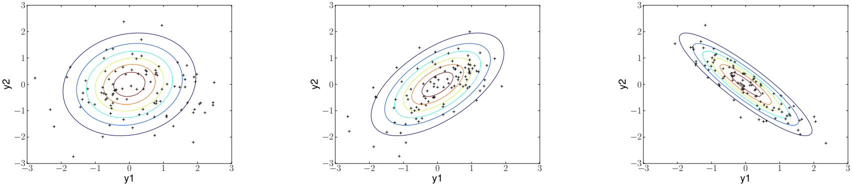
- Probability contour
- Samples



$$\boldsymbol{K} = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 1 \end{bmatrix}$$

A 2D Gaussian

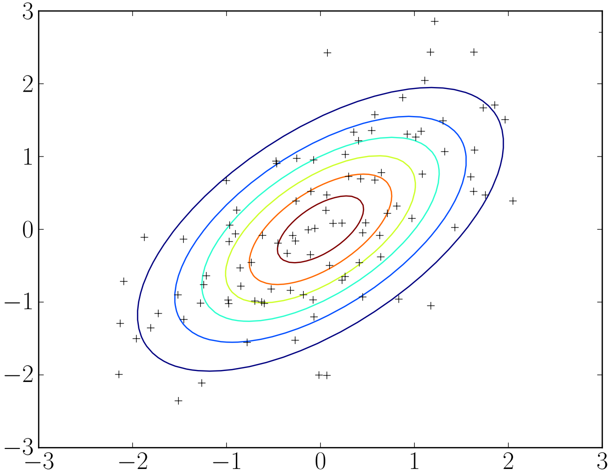
Varying the covariance matrix



$$K = \begin{bmatrix} 1 & 0.14 \\ 0.14 & 1 \end{bmatrix} \qquad K = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 1 \end{bmatrix} \qquad K = \begin{bmatrix} 1 & -0.9 \\ -0.9 & 1 \end{bmatrix}$$

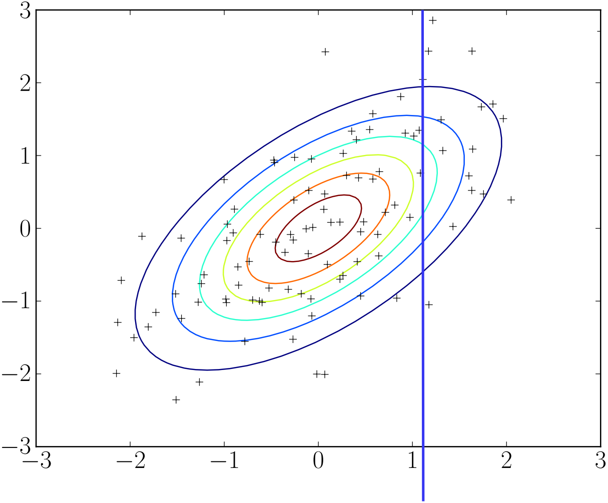
A 2D Gaussian

Inference



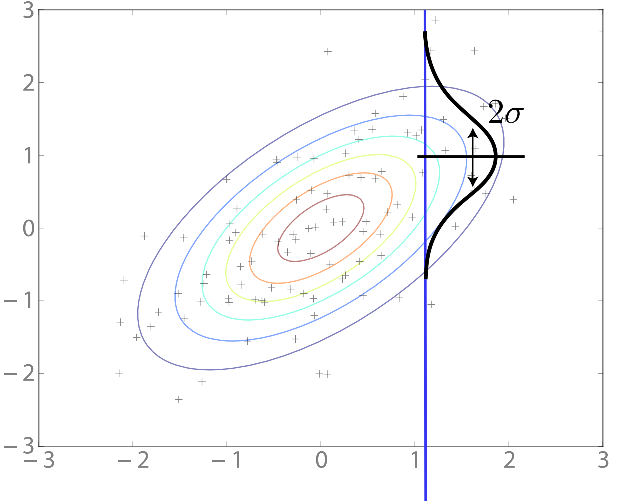
A 2D Gaussian

Inference



A 2D Gaussian

Inference



Inference

- Joint probability $p(y_1, y_2 \,|\, \boldsymbol{K}) = \mathcal{N}\left([y_1, y_2] \mid \mathbf{0}, \boldsymbol{K} \right)$
- Conditional probability

$$
\begin{aligned}
p(y_2 \,|\, y_1, \boldsymbol{K}) &= \frac{p(y_1, y_2 \,|\, \boldsymbol{K})}{p(y_1 \,|\, \boldsymbol{K})} \\
&\propto \exp\left\{ -\frac{1}{2} [y_1, y_2]\, \boldsymbol{K}^{-1} \left[\begin{array}{c} y_1 \\ y_2 \end{array} \right] \right\}
\end{aligned}
$$

- Completing the square yields a Gaussian with non-zero as posterior for y_2.

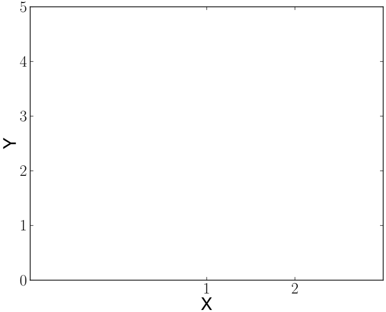
Inference

Gaussian conditioning in 2D

$$
\begin{aligned}
p(y_2 \mid y_1, \boldsymbol{K}) &= \frac{p(y_1, y_2 \mid \boldsymbol{K})}{p(y_1 \mid \boldsymbol{K})} \propto \exp\left\{-\frac{1}{2}[y_1, y_2]\, \boldsymbol{K}^{-1} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right\} \\
&= \exp\{-\frac{1}{2}\left[y_1^2 \boldsymbol{K}^{-1}_{1,1} + y_2^2 \boldsymbol{K}^{-1}_{2,2} + 2 y_1 \boldsymbol{K}^{-1}_{1,2} y_2\right]\} \\
&= \exp\{-\frac{1}{2}\left[y_2^2 \boldsymbol{K}^{-1}_{2,2} + 2 y_2 \boldsymbol{K}^{-1}_{1,2} y_1 + C\right]\} \\
&= Z \exp\{-\frac{1}{2}\boldsymbol{K}^{-1}_{2,2}\left[y_2^2 + 2 y_2 \frac{\boldsymbol{K}^{-1}_{1,2} y_1}{\boldsymbol{K}^{-1}_{2,2}}\right]\} \\
&= Z \exp\{-\frac{1}{2}\boldsymbol{K}^{-1}_{2,2}\left[y_2^2 + 2 y_2 \frac{\boldsymbol{K}^{-1}_{1,2} y_1}{\boldsymbol{K}^{-1}_{2,2}} + \frac{\boldsymbol{K}^{-1}_{1,2} y_1}{\boldsymbol{K}^{-1}_{2,2}}^2\right] + \frac{1}{2}\boldsymbol{K}^{-1}_{2,2} \frac{\boldsymbol{K}^{-1}_{1,2} y_1}{\boldsymbol{K}^{-1}_{2,2}}^2 \} \\
&= Z' \exp\{-\frac{1}{2}\underbrace{\boldsymbol{K}^{-1}_{2,2}}_{\sigma^2}\left[y_2 + \underbrace{\frac{\boldsymbol{K}^{-1}_{1,2} y_1}{\boldsymbol{K}^{-1}_{2,2}}}_{-\mu}\right]^2\} \propto \mathcal{N}\left(y_2 \mid \mu, \sigma^2\right)
\end{aligned}
$$

Extending the idea to higher dimensions

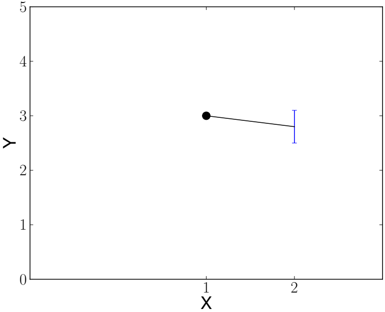
- Let us interpret y_1 and y_2 as outputs in a regression setting.
- We can introduce an additional 3rd point.



- Now $P([y_1, y_2, y_3] \mid \boldsymbol{K}_3) = \mathcal{N}\left([y_1, y_2, y_3] \mid \boldsymbol{0}, \boldsymbol{K}_3\right)$, where $\boldsymbol{K}_3$ is now a 3 x 3 covariance matrix!

Extending the idea to higher dimensions

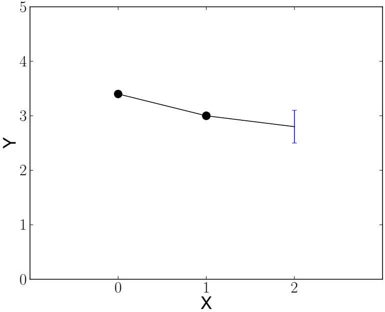
- Let us interpret y_1 and y_2 as outputs in a regression setting.
- We can introduce an additional 3rd point.



- Now $P([y_1, y_2, y_3] \mid \boldsymbol{K}_3) = \mathcal{N}\left([y_1, y_2, y_3] \mid \mathbf{0}, \boldsymbol{K}_3\right)$, where $\boldsymbol{K}_3$ is now a 3 x 3 covariance matrix!

Extending the idea to higher dimensions

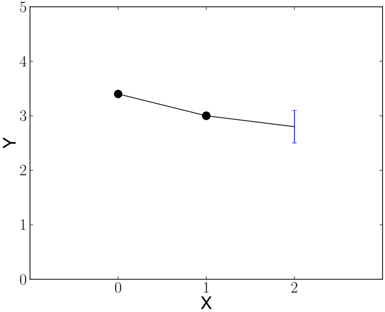
- Let us interpret y_1 and y_2 as outputs in a regression setting.
- We can introduce an additional 3rd point.



- Now $P([y_1, y_2, y_3] \,|\, \boldsymbol{K}_3) = \mathcal{N}\left([y_1, y_2, y_3] \mid \boldsymbol{0}, \boldsymbol{K}_3\right)$, where $\boldsymbol{K}_3$ is now a 3 x 3 covariance matrix!

Extending the idea to higher dimensions

- Let us interpret y_1 and y_2 as outputs in a regression setting.
- We can introduce an additional 3rd point.



- Now $P([y_1, y_2, y_3] \mid \boldsymbol{K}_3) = \mathcal{N}\left([y_1, y_2, y_3] \mid \mathbf{0}, \boldsymbol{K}_3\right)$, where $\boldsymbol{K}_3$ is now a 3 x 3 covariance matrix!

Constructing Covariance Matrices

- Analogously we can look at the joint probability for arbitrary many points and obtain predictions.
- Issue: how to construct a good covariance matrix?
- A simple heuristics

$$\boldsymbol{K}_2 = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 1 \end{bmatrix}$$

$$\boldsymbol{K}_3 = \begin{bmatrix} 1 & 0.6 & 0 \\ 0.6 & 1 & 0.6 \\ 0 & 0.6 & 1 \end{bmatrix}$$

- Note:
  - The ordering of the points y_1, y_2, y_3 matters.
  - Important to ensure that covariance matrices remain positive definite (matrix inversion).

Constructing Covariance Matrices

- Analogously we can look at the joint probability for arbitrary many points and obtain predictions.
- Issue: how to construct a good covariance matrix?
- A simple heuristics

$$\boldsymbol{K}_2 = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 1 \end{bmatrix}$$

$$\boldsymbol{K}_3 = \begin{bmatrix} 1 & 0.6 & 0 \\ 0.6 & 1 & 0.6 \\ 0 & 0.6 & 1 \end{bmatrix}$$

- Note:
  - The ordering of the points y_1, y_2, y_3 matters.
  - Important to ensure that covariance matrices remain positive definite (matrix inversion).

Constructing Covariance Matrices

- Analogously we can look at the joint probability for arbitrary many points and obtain predictions.
- Issue: how to construct a good covariance matrix?
- A simple heuristics

$$\boldsymbol{K}_2 = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 1 \end{bmatrix}$$

$$\boldsymbol{K}_3 = \begin{bmatrix} 1 & 0.6 & 0 \\ 0.6 & 1 & 0.6 \\ 0 & 0.6 & 1 \end{bmatrix}$$

- Note:
  - The ordering of the points y_1, y_2, y_3 matters.
  - Important to ensure that covariance matrices remain positive definite (matrix inversion).

Constructing Covariance Matrices

A general recipe

- Use a covariance function (kernel function) to construct $\boldsymbol{K}$:

$$\boldsymbol{K}_{i,j} = k(x_i, x_j; \boldsymbol{\Theta}_{\mathsf{K}})$$

- Example: The linear covariance function corresponds to a variance component model

$$k_{\mathsf{LIN}}(x_i, x_j, ; A) = A^2 x_i \cdot x_j$$

- Example: The squared exponential covariance function embodies the belief that points further apart are less correlated:

$$k_{\mathsf{SE}}(x_i, x_j, ; A, L) = A^2 \exp\left\{-0.5 \cdot \frac{(x_i - x_j)^2}{L^2}\right\}$$

- $\boldsymbol{\Theta}_{\mathsf{K}} = \{A, L\}$: hyperparameters.
  - A^2 Overall correlation, amplitude L^2 Scaling parameter, smoothness
- Denote the covariance matrix for a set of inputs $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ as: $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}})$

Constructing Covariance Matrices

A general recipe

- Use a covariance function (kernel function) to construct $\boldsymbol{K}$:

$$\boldsymbol{K}_{i,j} = k(x_i, x_j; \boldsymbol{\Theta}_{\mathsf{K}})$$

- Example: The linear covariance function corresponds to a variance component model

$$k_{\mathsf{LIN}}(x_i, x_j, ; A) = A^2 x_i \cdot x_j$$

- Example: The squared exponential covariance function embodies the belief that points further apart are less correlated:

$$k_{\mathsf{SE}}(x_i, x_j, ; A, L) = A^2 \exp\left\{-0.5 \cdot \frac{(x_i - x_j)^2}{L^2}\right\}$$

- $\boldsymbol{\Theta}_{\mathsf{K}} = \{A, L\}$: hyperparameters.
  - A^2 Overall correlation, amplitude L^2 Scaling parameter, smoothness
- Denote the covariance matrix for a set of inputs $X = \{x_1, \ldots, x_N\}$ as: $\boldsymbol{K}_{X,X}(\boldsymbol{\Theta}_{\mathsf{K}})$

Constructing Covariance Matrices

A general recipe

- Use a covariance function (kernel function) to construct $\boldsymbol{K}$:

$$\boldsymbol{K}_{i,j} = k(x_i, x_j; \boldsymbol{\Theta}_{\mathsf{K}})$$

- Example: The linear covariance function corresponds to a variance component model

$$k_{\mathsf{LIN}}(x_i, x_j, ; A) = A^2 x_i \cdot x_j$$

- Example: The squared exponential covariance function embodies the belief that points further apart are less correlated:

$$k_{\mathsf{SE}}(x_i, x_j, ; A, L) = A^2 \exp\left\{-0.5 \cdot \frac{(x_i - x_j)^2}{L^2}\right\}$$

- $\boldsymbol{\Theta}_{\mathsf{K}} = \{A, L\}$: hyperparameters.
  - A^2 Overall correlation, amplitude L^2 Scaling parameter, smoothness
- Denote the covariance matrix for a set of inputs $\boldsymbol{X} = \{x_1, \ldots, x_N\}$ as: $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}})$

Constructing Covariance Matrices

A general recipe

- Use a covariance function (kernel function) to construct $\boldsymbol{K}$:

$$\boldsymbol{K}_{i,j} = k(x_i, x_j; \boldsymbol{\Theta}_{\mathsf{K}})$$

- Example: The linear covariance function corresponds to a variance component model

$$k_{\mathsf{LIN}}(x_i, x_j, ; A) = A^2 x_i \cdot x_j$$

- Example: The squared exponential covariance function embodies the belief that points further apart are less correlated:

$$k_{\mathsf{SE}}(x_i, x_j, ; A, L) = A^2 \exp\left\{-0.5 \cdot \frac{(x_i - x_j)^2}{L^2}\right\}$$

- $\boldsymbol{\Theta}_{\mathsf{K}} = \{A, L\}$: hyperparameters.
  - A^2 Overall correlation, amplitude L^2 Scaling parameter, smoothness
- Denote the covariance matrix for a set of inputs $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ as: $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}})$

Constructing Covariance Matrices

A general recipe

- Use a covariance function (kernel function) to construct $\boldsymbol{K}$:

$$\boldsymbol{K}_{i,j} = k(x_i, x_j; \boldsymbol{\Theta}_{\mathsf{K}})$$

- Example: The linear covariance function corresponds to a variance component model

$$k_{\mathsf{LIN}}(x_i, x_j, ; A) = A^2 x_i \cdot x_j$$

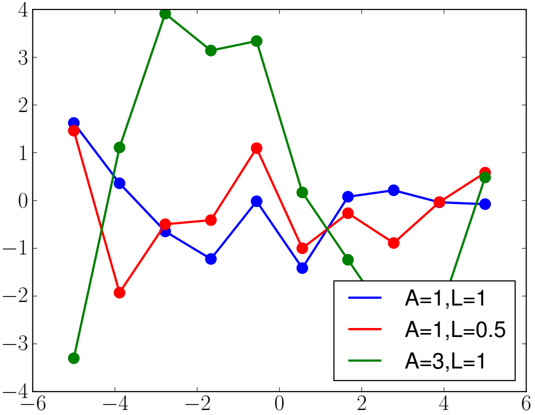
- Example: The squared exponential covariance function embodies the belief that points further apart are less correlated:

$$k_{\mathsf{SE}}(x_i, x_j, ; A, L) = A^2 \exp\left\{-0.5 \cdot \frac{(x_i - x_j)^2}{L^2}\right\}$$

- $\boldsymbol{\Theta}_{\mathsf{K}} = \{A, L\}$: hyperparameters.
  - A^2 Overall correlation, amplitude L^2 Scaling parameter, smoothness
- Denote the covariance matrix for a set of inputs $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ as: $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}})$

Constructing Covariance Matrices

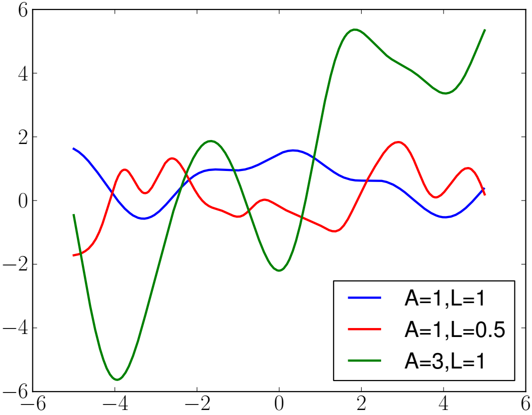
GP samples using the squared exponential covariance function



10D Gaussian

Constructing Covariance Matrices

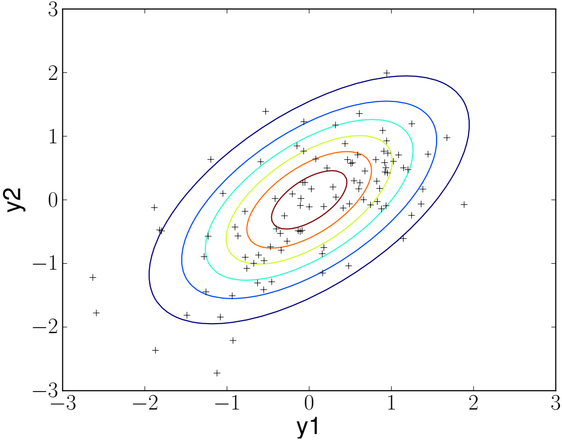
GP samples using the squared exponential covariance function



500D Gaussian

Constructing Covariance Matrices

GP samples using the squared exponential covariance function



Reminder: Every function line corresponds to a sample drawn from this 2D Gaussian!

Drawing samples from a Gaussian processes

For each sample do:

- Choose discretization of x axes $\boldsymbol{X} = \{x_0, x_1, \ldots, x_N\}$.
- Evaluate covariance $\boldsymbol{K} = \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}})$

Math

- Draw from

$$p(\boldsymbol{y} \mid \boldsymbol{K}) = \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{0}, \boldsymbol{K}\right)$$

"Matlab"

- Draw independent Gaussian variables

$$\tilde{\boldsymbol{y}} = \mathsf{randn}(N, 1)$$

- Rotate with $\sqrt{\boldsymbol{K}}$

$$\boldsymbol{y} = \mathsf{chol}(\boldsymbol{K}) \cdot \tilde{\boldsymbol{y}}$$

Drawing samples from a Gaussian processes

For each sample do:

- Choose discretization of x axes $\boldsymbol{X} = \{x_0, x_1, \ldots, x_N\}$.
- Evaluate covariance $\boldsymbol{K} = \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}})$

Math

- Draw from

$$p(\boldsymbol{y} \,|\, \boldsymbol{K}) = \mathcal{N}\left(\, \boldsymbol{y} \mid \boldsymbol{0}, \boldsymbol{K} \,\right)$$

"Matlab"

- Draw independent Gaussian variables

$$\tilde{\boldsymbol{y}} = \text{randn}(N, 1)$$

- Rotate with $\sqrt{\boldsymbol{K}}$

$$\boldsymbol{y} = \text{chol}(\boldsymbol{K}) \cdot \tilde{\boldsymbol{y}}$$

Drawing samples from a Gaussian processes

For each sample do:

- Choose discretization of x axes $\boldsymbol{X} = \{x_0, x_1, \ldots, x_N\}$.
- Evaluate covariance $\boldsymbol{K} = \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}})$

Math

- Draw from

$$p(\boldsymbol{y} \,|\, \boldsymbol{K}) = \mathcal{N}\left(\, \boldsymbol{y} \mid \mathbf{0}, \boldsymbol{K} \,\right)$$

"Matlab"

- Draw independent Gaussian variables

$$\tilde{\boldsymbol{y}} = \mathsf{randn}(N, 1)$$

- Rotate with $\sqrt{\boldsymbol{K}}$

$$\boldsymbol{y} = \mathsf{chol}(\boldsymbol{K}) \cdot \tilde{\boldsymbol{y}}$$

Why this all works

- Consistency of the 10D and 500D Gaussian.
- A small quiz:
  - Let y_1, y_2, y_3 have covariance matrix

$$\boldsymbol{K}_3 = \begin{bmatrix} 1 & 0.5 & 0 \\ 0.5 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix} \text{ and inverse } \boldsymbol{K}_3^{-1} = \begin{bmatrix} 1.5 & -1 & 0.5 \\ -1 & 2 & -1 \\ 0.5 & -1 & 1.5 \end{bmatrix}$$

i.e. $p(\{y_1, y_2, y_3\} \mid \boldsymbol{K}_3) = \mathcal{N}\left(\{y_1, y_2, y_3\} \mid \mathbf{0}, \boldsymbol{K}_3\right)$

  - Now focus on the variables y_1, y_2, integrating out y_3.

$$p(\{y_1, y_2\}) = \int_{y_3} \mathcal{N}\left(\{y_1, y_2, y_3\} \mid \mathbf{0}, \boldsymbol{K}_3\right) = \mathcal{N}\left(\{y_1, y_2\} \mid \mathbf{0}, \boldsymbol{K}_2\right)$$

Which of the following statements is true

a) $\boldsymbol{K}_2 = \begin{bmatrix} 1 & 5 \\ 5 & 1 \end{bmatrix}$ b) $\boldsymbol{K}_2^{-1} = \begin{bmatrix} 1.5 & -1 \\ -1 & 2 \end{bmatrix}$

Why this all works

- Consistency of the 10D and 500D Gaussian.
- A small quiz:
  - Let y_1, y_2, y_3 have covariance matrix

$$\boldsymbol{K}_3 = \begin{bmatrix} 1 & 0.5 & 0 \\ 0.5 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix} \text{ and inverse } \boldsymbol{K}_3^{-1} = \begin{bmatrix} 1.5 & -1 & 0.5 \\ -1 & 2 & -1 \\ 0.5 & -1 & 1.5 \end{bmatrix}$$

    i.e. $p(\{y_1, y_2, y_3\} \mid \boldsymbol{K}_3) = \mathcal{N}\left(\{y_1, y_2, y_3\} \mid \mathbf{0}, \boldsymbol{K}_3\right)$
  - Now focus on the variables y_1, y_2, integrating out y_3.

$$\begin{aligned} p(\{y_1, y_2\}) &= \int_{y_3} \mathcal{N}\left(\{y_1, y_2, y_3\} \mid \mathbf{0}, \boldsymbol{K}_3\right) \\ &= \mathcal{N}\left(\{y_1, y_2\} \mid \mathbf{0}, \boldsymbol{K}_2\right) \end{aligned}$$

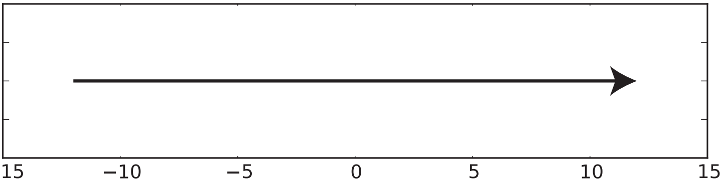
    Which of the following statements is true

$$\text{a) } \boldsymbol{K}_2 = \begin{bmatrix} 1 & 5 \\ 5 & 1 \end{bmatrix} \qquad \text{b) } \boldsymbol{K}_2^{-1} = \begin{bmatrix} 1.5 & -1 \\ -1 & 2 \end{bmatrix}$$

Why this all works

GP as infinite object (philosophical)

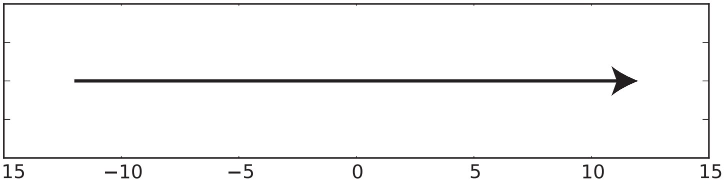
- A valid covariance function $k(x, x')$ defines recipe to calculate covariance for any choice of inputs.
- Prior on functions: all points on the real line are inputs; $K_{\mathcal{R},\mathcal{R}}$ is an infinite object!
- Numerical implementation: choose finite subset X and evaluate on a reduced, finite $K_{X,X}$, exploiting consistency rule.



Why this all works

GP as infinite object (philosophical)

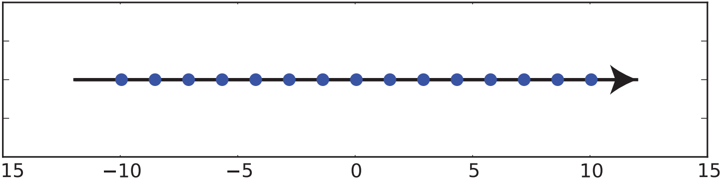
- A valid covariance function $k(x, x')$ defines recipe to calculate covariance for any choice of inputs.
- Prior on functions: all points on the real line are inputs; $\boldsymbol{K}_{\mathcal{R},\mathcal{R}}$ is an infinite object!
- Numerical implementation: choose finite subset $\boldsymbol{X}$ and evaluate on a reduced, finite $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}$, exploiting consistency rule.



Why this all works

GP as infinite object (philosophical)

- A valid covariance function $k(x, x')$ defines recipe to calculate covariance for any choice of inputs.
- Prior on functions: all points on the real line are inputs; $\boldsymbol{K}_{\mathcal{R},\mathcal{R}}$ is an infinite object!
- Numerical implementation: choose finite subset $\boldsymbol{X}$ and evaluate on a reduced, finite $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}$, exploiting consistency rule.



Outline

Function space view

So far

1. Joint Gaussian distribution over the set of all outputs $\boldsymbol{y}$.
2. Covariance function as a recipe to construct a suitable covariance matrices from the corresponding inputs $\boldsymbol{X}$.

Function space view

The Gaussian process as a prior on functions

- Covariance function and hyperparameters reflect the prior belief on function smoothness, lengthscales etc.
- The general recipe allows a joint Gaussian to be constructed for an arbitrary selection of input locations $\boldsymbol{X}$.

Prior on infinite function $f(x)$

$$p(f(x)) = \mathsf{GP}(f(x) \mid k)$$

Prior on function values $\boldsymbol{f} = (f_1, \ldots, f_N)$

$$p(\boldsymbol{f} \mid \boldsymbol{X}, \boldsymbol{\Theta}_{\mathsf{K}}) = \mathcal{N}\left(\boldsymbol{f} \mid \mathbf{0}, \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}}\right.$$

Noise-free observations

- Given noise-free training data $\mathcal{D} = \{\boldsymbol{x}_n, f_n\}_{n=1}^{N}$
- Want to make predictions $\boldsymbol{f}^{\star}$ at test points $\boldsymbol{X}^{\star}$
- Joint distribution of $\boldsymbol{f}$ and $\boldsymbol{f}^{\star}$ is

$$p([\boldsymbol{f}, \boldsymbol{f}^{\star}] \,|\, \boldsymbol{X}, \boldsymbol{X}^{\star}, \boldsymbol{\Theta}_{\mathsf{K}}) = \mathcal{N}\left([\boldsymbol{f}, \boldsymbol{f}^{\star}] \mid \boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} & \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}^{\star}} \\ \boldsymbol{K}_{\boldsymbol{X}^{\star},\boldsymbol{X}} & \boldsymbol{K}_{\boldsymbol{X}^{\star},\boldsymbol{X}^{\star}} \end{bmatrix} \right)$$

  (All kernel matrices $\boldsymbol{K}$ depend on hyperparameters $\boldsymbol{\Theta}_{\mathsf{K}}$ which are dropped for brevity.)
- Real data is rarely noise-free.

Noise-free observations

- Given noise-free training data $\mathcal{D} = \{\boldsymbol{x}_n, f_n\}_{n=1}^N$
- Want to make predictions $\boldsymbol{f}^\star$ at test points $\boldsymbol{X}^\star$
- Joint distribution of $\boldsymbol{f}$ and $\boldsymbol{f}^\star$ is

$$p([\boldsymbol{f}, \boldsymbol{f}^\star] \,|\, \boldsymbol{X}, \boldsymbol{X}^\star, \boldsymbol{\Theta}_\mathsf{K}) = \mathcal{N}\left([\boldsymbol{f}, \boldsymbol{f}^\star] \mid \mathbf{0}, \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} & \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}^\star} \\ \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}} & \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}^\star} \end{bmatrix} \right)$$

  (All kernel matrices $\boldsymbol{K}$ depend on hyperparameters $\boldsymbol{\Theta}_\mathsf{K}$ which are dropped for brevity.)
- Real data is rarely noise-free.

Inference

- Given observed noisy data $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{y}\}$, the joint probability over latent function values $\boldsymbol{f}$ and $\boldsymbol{f}^\star$ given $\boldsymbol{y}$ is

$$p([\boldsymbol{f}, \boldsymbol{f}^\star] \,|\, \boldsymbol{X}, \boldsymbol{X}^\star, \boldsymbol{y}, \boldsymbol{\Theta}_\mathsf{K}, \sigma^2) \propto \overbrace{\mathcal{N}\left([\boldsymbol{f}, \boldsymbol{f}^\star] \mid \boldsymbol{0}, \boldsymbol{K} \right)}^{\text{Prior}} \\ \underbrace{\times \prod_{n=1}^{N} \mathcal{N}\left(y_n \mid f_n, \sigma^2 \right)}_{\text{Likelihood}},$$

Inference

- Given observed noisy data $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{y}\}$, the joint probability over latent function values $\boldsymbol{f}$ and $\boldsymbol{f}^\star$ given $\boldsymbol{y}$ is

$$p([\boldsymbol{f}, \boldsymbol{f}^\star] \,|\, \boldsymbol{X}, \boldsymbol{X}^\star, \boldsymbol{y}, \boldsymbol{\Theta}_{\mathsf{K}}, \sigma^2) \propto \overbrace{\mathcal{N}\left([\boldsymbol{f}, \boldsymbol{f}^\star] \mid \mathbf{0}, \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} & \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}^\star} \\ \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}} & \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}^\star} \end{bmatrix}\right)}^{\text{Prior}} \times \underbrace{\prod_{n=1}^{N} \mathcal{N}\left(y_n \mid f_n, \sigma^2\right)}_{\text{Likelihood}},$$

Inference

- Applying "Gaussian calculus", integrating out $\boldsymbol{f}$ yields

$$p([\boldsymbol{y}, \boldsymbol{f}^\star] \mid \boldsymbol{X}, \boldsymbol{X}^\star, \boldsymbol{y}, \boldsymbol{\Theta}_{\mathsf{K}}, \sigma^2) \propto \mathcal{N}\left([\boldsymbol{y}, \boldsymbol{f}^\star] \mid \mathbf{0}, \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2 \boldsymbol{I} & \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}^\star} \\ \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}} & \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}^\star} \end{bmatrix}\right.$$

- Note: Assuming noisy instead of perfect observation noise merely corresponds to adding a diagonal component to the self-covariance $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}$.

Inference

- Applying "Gaussian calculus", integrating out $\boldsymbol{f}$ yields

$$p([\boldsymbol{y}, \boldsymbol{f}^\star] \mid \boldsymbol{X}, \boldsymbol{X}^\star, \boldsymbol{y}, \boldsymbol{\Theta}_\mathsf{K}, \sigma^2) \propto \mathcal{N}\left([\boldsymbol{y}, \boldsymbol{f}^\star] \mid \boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2 \boldsymbol{I} & \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}^\star} \\ \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}} & \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}^\star} \end{bmatrix}\right.$$

- Note: Assuming noisy instead of perfect observation noise merely corresponds to adding a diagonal component to the self-covariance $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}$.

Making predictions

- The predictive distribution follows from the joint distribution by completing the square (conditioning)

$$p([\boldsymbol{y}, \boldsymbol{f}^\star] \mid \boldsymbol{X}, \boldsymbol{X}^\star, \boldsymbol{y}, \boldsymbol{\Theta}_\mathsf{K}, \sigma^2) \propto \mathcal{N}\left([\boldsymbol{y}, \boldsymbol{f}^\star] \mid \boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2 \boldsymbol{I} & \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}^\star} \\ \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}} & \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}^\star} \end{bmatrix}\right.$$

- Gaussian predictive distribution for $\boldsymbol{f}^\star$

$p(\boldsymbol{f}^\star \mid \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{X}^\star, \boldsymbol{\Theta}_\mathsf{K}, \sigma^2) = \mathcal{N}\left(\boldsymbol{f}^\star \mid \boldsymbol{\mu}^\star, \boldsymbol{\Sigma}^\star \right)$ with

$$\boldsymbol{\mu}^\star = \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}} \left[\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2 \boldsymbol{I} \right]^{-1} \boldsymbol{y}$$

$$\boldsymbol{\Sigma}^\star = \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}^\star} - \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}} \left[\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2 \boldsymbol{I} \right]^{-1} \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}^\star}$$

Making predictions

- The predictive distribution follows from the joint distribution by completing the square (conditioning)

$$p([\boldsymbol{y}, \boldsymbol{f}^\star] \mid \boldsymbol{X}, \boldsymbol{X}^\star, \boldsymbol{y}, \boldsymbol{\Theta}_\mathsf{K}, \sigma^2) \propto \mathcal{N}\left([\boldsymbol{y}, \boldsymbol{f}^\star] \mid \mathbf{0}, \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2 \boldsymbol{I} & \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}^\star} \\ \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}} & \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}^\star} \end{bmatrix}\right.$$

- Gaussian predictive distribution for $\boldsymbol{f}^\star$

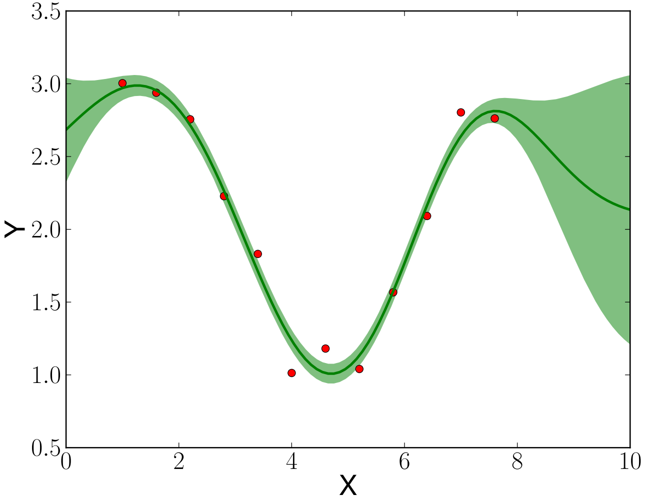
$$p(\boldsymbol{f}^\star \mid \boldsymbol{X}, \boldsymbol{y}, \boldsymbol{X}^\star, \boldsymbol{\Theta}_\mathsf{K}, \sigma^2) = \mathcal{N}\left(\boldsymbol{f}^\star \mid \boldsymbol{\mu}^\star, \boldsymbol{\Sigma}^\star \right) \text{ with}$$

$$\boldsymbol{\mu}^\star = \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}} \left[\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2 \boldsymbol{I}\right]^{-1} \boldsymbol{y}$$

$$\boldsymbol{\Sigma}^\star = \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}^\star} - \boldsymbol{K}_{\boldsymbol{X}^\star,\boldsymbol{X}} \left[\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}} + \sigma^2 \boldsymbol{I}\right]^{-1} \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}^\star}$$

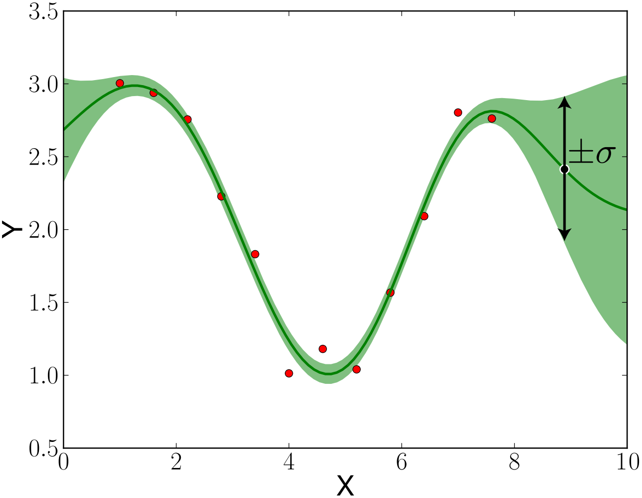
Making predictions

Example



Making predictions

Example



Learning hyperparameters

1. Fixed covariance matrix: $p(\boldsymbol{y} \mid \boldsymbol{K})$
2. Constructed covariance matrix: $\{\boldsymbol{K}\}_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j; \boldsymbol{\Theta}_{\mathsf{K}})$
3. Can we learn the hyperparameters $\boldsymbol{\Theta}_{\mathsf{K}}$?

Learning hyperparameters

- Formally we are interested in the posterior

$$p(\boldsymbol{\Theta}_{\mathsf{K}} \,|\, \mathcal{D}) \propto p\,(\boldsymbol{y} \,|\, \boldsymbol{X},\, \boldsymbol{\Theta}_{\mathsf{K}})\, p(\boldsymbol{\Theta}_{\mathsf{K}})$$

- Inference is analytically intractable!
- MAP estimate instead of a full posterior. Set $\boldsymbol{\Theta}_{\mathsf{K}}$ to the most probable hyperparameter settings:

$$\begin{aligned}
\hat{\boldsymbol{\Theta}}_{\mathsf{K}} &= \underset{\boldsymbol{\Theta}_{\mathsf{K}}}{\operatorname{argmax}} \ln \left[p\,(\boldsymbol{y} \,|\, \boldsymbol{X},\, \boldsymbol{\Theta}_{\mathsf{K}})\, p(\boldsymbol{\Theta}_{\mathsf{K}})\right] \\
&= \underset{\boldsymbol{\Theta}_{\mathsf{K}}}{\operatorname{argmax}} \ln \mathcal{N}\left(\boldsymbol{y} \mid \mathbf{0}, \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}}) + \sigma^2 \boldsymbol{I}\right) + \ln p(\boldsymbol{\Theta}_{\mathsf{K}}) \\
&= \underset{\boldsymbol{\Theta}_{\mathsf{K}}}{\operatorname{argmax}} \Bigg[-\frac{1}{2} \log \det[\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}}) + \sigma^2 \boldsymbol{I}] \\
&\quad - \frac{1}{2} \boldsymbol{y}^\top [\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}}) + \sigma^2 \boldsymbol{I}]^{-1} \boldsymbol{y} - \frac{N}{2} \log 2\pi + \ln p(\boldsymbol{\Theta}_{\mathsf{K}}) \Bigg]
\end{aligned}$$

- Optimization can be carried out using standard optimization techniques.

Learning hyperparameters

- Formally we are interested in the posterior

$$p(\boldsymbol{\Theta}_{\mathsf{K}} \,|\, \mathcal{D}) \propto p\,(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\Theta}_{\mathsf{K}})\, p(\boldsymbol{\Theta}_{\mathsf{K}})$$

- Inference is analytically intractable!
- MAP estimate instead of a full posterior. Set $\boldsymbol{\Theta}_{\mathsf{K}}$ to the most probable hyperparameter settings:

$$\begin{aligned}
\hat{\boldsymbol{\Theta}}_{\mathsf{K}} &= \operatorname*{argmax}_{\boldsymbol{\Theta}_{\mathsf{K}}} \ln \left[p\,(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\Theta}_{\mathsf{K}})\, p(\boldsymbol{\Theta}_{\mathsf{K}})\right] \\
&= \operatorname*{argmax}_{\boldsymbol{\Theta}_{\mathsf{K}}} \ln \mathcal{N}\left(\boldsymbol{y} \mid \mathbf{0}, \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}}) + \sigma^2 \boldsymbol{I}\right) + \ln p(\boldsymbol{\Theta}_{\mathsf{K}}) \\
&= \operatorname*{argmax}_{\boldsymbol{\Theta}_{\mathsf{K}}} \left[-\frac{1}{2} \log \det[\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}}) + \sigma^2 \boldsymbol{I}] \right. \\
&\quad \left. - \frac{1}{2} \boldsymbol{y}^\top [\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}}) + \sigma^2 \boldsymbol{I}]^{-1} \boldsymbol{y} - \frac{N}{2} \log 2\pi + \ln p(\boldsymbol{\Theta}_{\mathsf{K}}) \right]
\end{aligned}$$

- Optimization can be carried out using standard optimization techniques.

Learning hyperparameters

- Formally we are interested in the posterior

$$p(\boldsymbol{\Theta}_{\mathsf{K}} \,|\, \mathcal{D}) \propto p\left(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\Theta}_{\mathsf{K}}\right) p(\boldsymbol{\Theta}_{\mathsf{K}})$$

- Inference is analytically intractable!
- MAP estimate instead of a full posterior. Set $\boldsymbol{\Theta}_{\mathsf{K}}$ to the most probable hyperparameter settings:

$$\begin{aligned}
\hat{\boldsymbol{\Theta}}_{\mathsf{K}} &= \underset{\boldsymbol{\Theta}_{\mathsf{K}}}{\operatorname{argmax}} \ln \left[p\left(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\Theta}_{\mathsf{K}}\right) p(\boldsymbol{\Theta}_{\mathsf{K}})\right] \\
&= \underset{\boldsymbol{\Theta}_{\mathsf{K}}}{\operatorname{argmax}} \ln \mathcal{N}\left(\boldsymbol{y} \mid \mathbf{0}, \boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}}) + \sigma^2 \boldsymbol{I}\right) + \ln p(\boldsymbol{\Theta}_{\mathsf{K}}) \\
&= \underset{\boldsymbol{\Theta}_{\mathsf{K}}}{\operatorname{argmax}} \left[-\frac{1}{2} \log \det[\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}}) + \sigma^2 \boldsymbol{I}] \right. \\
&\left. - \frac{1}{2} \boldsymbol{y}^{\top} [\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}}) + \sigma^2 \boldsymbol{I}]^{-1} \boldsymbol{y} - \frac{N}{2} \log 2\pi + \ln p(\boldsymbol{\Theta}_{\mathsf{K}}) \right]
\end{aligned}$$

- Optimization can be carried out using standard optimization techniques.

Choosing covariance functions

- The covariance function embodies the prior belief about functions.
- Example: linear regression

$$y_n = wx_n + c + \psi_n$$

- Covariance function denote covariation

$$\begin{aligned} k(x_n, x_n') &= \langle y_n y_n' \rangle \\ &= \langle (wx_n + c + \psi_n)(wx_n' + c + \psi_n') \rangle \\ &= \underbrace{w^2 \cdot x_n x_n' + c^2}_{\text{kernel: } k(x_n, x_n')} + \delta_{n,n'} \psi_n^2 \end{aligned}$$

Choosing covariance functions

Multidimensional input space

- Generalise squared exponential covariance function to multiple dimensions
  - 1 Dimension $k_{\mathsf{SE}}(x_i, x_j, ; A, L) = A^2 \exp\left\{-0.5 \cdot \frac{(x_i - x_j)^2}{L^2}\right\}$
  - D Dimensions dD

$$k_{\mathsf{SE}}(\boldsymbol{x}_i, \boldsymbol{x}_j, ; A, \boldsymbol{L}) = A^2 \exp\left\{-0.5 \sum_{d=1}^{D} \frac{(x_i^d - x_j^d)^2}{L_d^2}\right\}$$

- Lengthscale parameters L_d denote "relevance" of a particular data dimension.
  - Large L_d correspond to irrelevant dimensions.

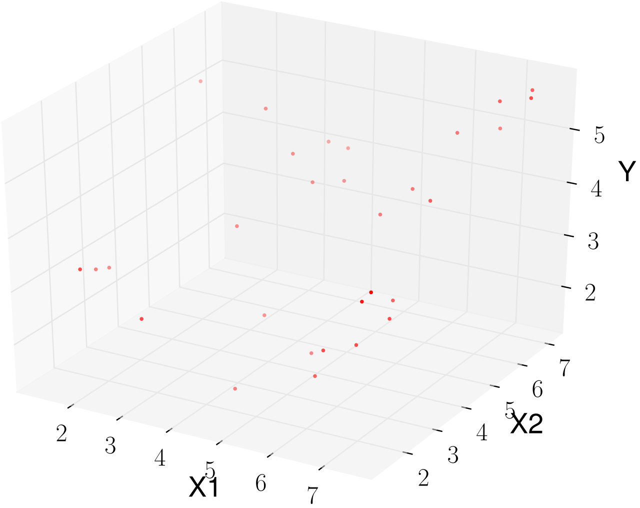
Choosing covariance functions

Multidimensional input space

- Generalise squared exponential covariance function to multiple dimensions
  - 1 Dimension $k_{\mathsf{SE}}(x_i, x_j, ; A, L) = A^2 \exp\left\{-0.5 \cdot \frac{(x_i - x_j)^2}{L^2}\right\}$
  - D Dimensions dD
    $$k_{\mathsf{SE}}(\boldsymbol{x}_i, \boldsymbol{x}_j, ; A, \boldsymbol{L}) = A^2 \exp\left\{-0.5 \sum_{d=1}^{D} \frac{(x_i^d - x_j^d)^2}{L_d^2}\right\}$$
- Lengthscale parameters L_d denote "relevance" of a particular data dimension.
  - Large L_d correspond to irrelevant dimensions.

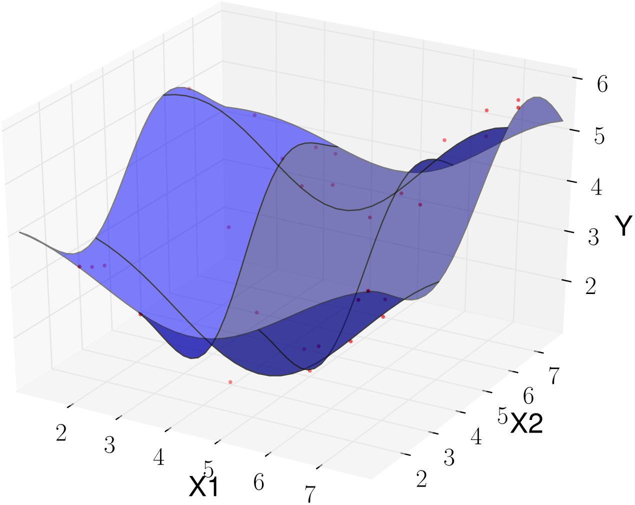
Choosing covariance functions

2D regression



Choosing covariance functions

2D regression



Choosing covariance functions

Any kernel will do

- Established kernels are all valid covariance functions, allowing for a wide range of possible input domains $\boldsymbol{X}$:
  - Graph kernels (molecules)
  - Kernels defined on strings (DNA sequences)

Choosing covariance functions

Combining existing covariance functions

- The sum of two covariances functions is itself a valid covariance function

$$k_S(x, x') = k_1(x, x') + k_2(x, x')$$

- The product of two covariance functions is itself a valid covariance function

$$k_P(x, x') = k_1(x, x') \cdot k_2(x, x')$$

GPs versus variance component models

Variance component

- Linear model

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\theta}, \sigma^2) = \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{\Phi}(\boldsymbol{X}) \cdot \boldsymbol{\theta}, \sigma^2 \boldsymbol{I}\right)$$

- Marginalize over $\boldsymbol{\theta}$

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \sigma_g^2, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \,|\, \boldsymbol{0}, \underbrace{\sigma_g^2 \boldsymbol{\Phi}(\boldsymbol{X}) \boldsymbol{\Phi}(\boldsymbol{X})^\top}_{\boldsymbol{K}} + \sigma^2 \boldsymbol{I}\big)$$

Gaussian process

- Define covariance through "recipe" $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}})$
- Implies marginal likelihood

$$p(\boldsymbol{y} \,|\, \boldsymbol{X}, \boldsymbol{\Theta}_{\mathsf{K}}, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \,|\, \boldsymbol{0}, \underbrace{\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}})}_{\boldsymbol{K}} + \sigma^2 \boldsymbol{I}\big)$$

- Any feature map $\boldsymbol{\Phi}$ implies a valid covariance function $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}})$.
- The inverse is not necessarily true!

GPs versus variance component models

Variance component

- Linear model

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\theta}, \sigma^2) = \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{\Phi}(\boldsymbol{X}) \cdot \boldsymbol{\theta}, \sigma^2 \boldsymbol{I}\right)$$

- Marginalize over $\boldsymbol{\theta}$

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \sigma_g^2, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \mid \boldsymbol{0}, \underbrace{\sigma_g^2 \boldsymbol{\Phi}(\boldsymbol{X}) \boldsymbol{\Phi}(\boldsymbol{X})^\top}_{\boldsymbol{K}} + \sigma^2 \boldsymbol{I}\big)$$

Gaussian process

- Define covariance through "recipe" $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}})$
- Implies marginal likelihood

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\Theta}_{\mathsf{K}}, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \mid \boldsymbol{0}, \underbrace{\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}})}_{\boldsymbol{K}} + \sigma^2 \boldsymbol{I}\big)$$

- Any feature map $\boldsymbol{\Phi}$ implies a valid covariance function $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}})$.
- The inverse is not necessarily true!

GPs versus variance component models

Variance component

- Linear model

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\theta}, \sigma^2) = \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{\Phi}(\boldsymbol{X}) \cdot \boldsymbol{\theta}, \sigma^2 \boldsymbol{I}\right)$$

- Marginalize over $\boldsymbol{\theta}$

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \sigma_g^2, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \mid \boldsymbol{0}, \underbrace{\sigma_g^2 \boldsymbol{\Phi}(\boldsymbol{X}) \boldsymbol{\Phi}(\boldsymbol{X})^\top}_{\boldsymbol{K}} + \sigma^2 \boldsymbol{I}\big)$$

Gaussian process

- Define covariance through "recipe" $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}})$
- Implies marginal likelihood

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\Theta}_{\mathsf{K}}, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \mid \boldsymbol{0}, \underbrace{\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}})}_{\boldsymbol{K}} + \sigma^2 \boldsymbol{I}\big)$$

- Any feature map $\boldsymbol{\Phi}$ implies a valid covariance function $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_{\mathsf{K}})$.
- The inverse is not necessarily true!

GPs versus variance component models

Variance component

- Linear model

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\theta}, \sigma^2) = \mathcal{N}\left(\boldsymbol{y} \mid \boldsymbol{\Phi}(\boldsymbol{X}) \cdot \boldsymbol{\theta}, \sigma^2 \boldsymbol{I}\right)$$

- Marginalize over $\boldsymbol{\theta}$

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \sigma_g^2, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \mid \boldsymbol{0}, \underbrace{\sigma_g^2 \boldsymbol{\Phi}(\boldsymbol{X}) \boldsymbol{\Phi}(\boldsymbol{X})^\top}_{\boldsymbol{K}} + \sigma^2 \boldsymbol{I}\big)$$

Gaussian process

- Define covariance through "recipe" $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_\mathsf{K})$
- Implies marginal likelihood

$$p(\boldsymbol{y} \mid \boldsymbol{X}, \boldsymbol{\Theta}_\mathsf{K}, \sigma^2) = \mathcal{N}\big(\boldsymbol{y} \mid \boldsymbol{0}, \underbrace{\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_\mathsf{K})}_{\boldsymbol{K}} + \sigma^2 \boldsymbol{I}\big)$$

- Any feature map $\boldsymbol{\Phi}$ implies a valid covariance function $\boldsymbol{K}_{\boldsymbol{X},\boldsymbol{X}}(\boldsymbol{\Theta}_\mathsf{K})$.
- The inverse is not necessarily true!