Prop: $\underline{X} \sim N_m(0, \Sigma)$ $\quad i, j \in \{1, \dots, m\}$

$B = \{1, \dots, m\} \setminus \{i, j\}$ $\quad A = \{i, j\}$

TFAE

$\mathrm{Var}(X_A \mid X_B) =$
$= \Sigma_{AA} - \Sigma_{AB} \Sigma_{BB}^{-1} \Sigma_{BA}$

(i) $X_i \perp\!\!\!\perp X_j \mid X_B$

(ii) $\Sigma_{ij} = \underbrace{\Sigma_{iB}}_{\in \mathbb{R}^{m-2}} \Sigma_{BB}^{-1} \Sigma_{Bj}$

$\left(\mathrm{Cov}(X_i, X_j \mid X_B) = 0 \right)$

(iii) $(\Sigma^{-1})_{ij} = 0$

(iv) the j^{th} entry of

$\underbrace{\Sigma_{i, B \cup \{j\}}}_{\in \mathbb{R}^{m-1}} \Sigma_{B \cup \{j\}, B \cup \{j\}}^{-1}$

is zero.

DATA

$\underline{X}^{(1)}, \dots, \underline{X}^{(n)} \sim N_m(\mu, \Sigma)$

μ, Σ unknown

$$f(x) = \frac{1}{(2\pi)^{m/2}} (\det \Sigma)^{-1/2} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

$$\log f(x) = -\frac{m}{2}\log(2\pi) - \frac{1}{2}\log\det\Sigma - \frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)$$

log-likelihood

$$\ell(\mu, \Sigma) = \sum_{i=1}^{n} \log f(x^{(i)}) =$$

$$= -\frac{nm}{2}\log(2\pi) - \frac{n}{2}\log\det\Sigma - \frac{1}{2}\sum_{i=1}^{n}(x^{(i)}-\mu)^T\Sigma^{-1}(x^{(i)}-\mu)$$

→ optimize wrt $\mu \in \mathbb{R}^m$

$$(x^{(i)}-\mu)^T\Sigma^{-1}(x^{(i)}-\mu) =$$

$x^{(i)T}\Sigma^{-1}x^{(i)}$ ①

$- \underbrace{x^{(i)T}\Sigma^{-1}\mu}$ ② $- \underbrace{\mu^T\Sigma^{-1}x^{(i)}}$ ③ equal

$+ \mu^T\Sigma^{-1}\mu$ " $(\Sigma^{-1}x^{(i)})^T\mu$ " $\mu^T(\Sigma^{(i)}x^{(i)})$

④

Note ⊙ $w \in \mathbb{R}^m$ $y \in \mathbb{R}^m$

$$\nabla_y (w^T y) = \nabla_y (y^T w) = w$$

$$\frac{\partial}{\partial y_k}(w^T y) = \frac{\partial}{\partial y_k}(w_1 y_1 + \dots + w_m y_m) = w_k$$

⊙ A symmetric

$$\nabla_y (y^T A y) = 2 \cdot A \cdot y$$

$$\frac{\partial}{\partial y_k}(y^T A y) \stackrel{?}{=} (2Ay)_k$$

$$\| $$

$$\frac{\partial}{\partial y_k}\left(\sum_{i,j=1}^{m} A_{ij} y_i y_j\right)$$

$$\| \ \frac{\partial}{\partial y_k}\left(\sum_{i=1}^{m} A_{ii} y_i^2 + 2 \cdot \sum_{i<j} A_{ij} y_i y_j\right)$$

$$= 2A_{kk}y_k + 2\underbrace{\sum_{i\neq k} A_{ik}y_i}$$

$\hookrightarrow$ A symmetric

$$= 2\sum_{i=1}^{m} A_{ik}y_i = 2\sum_{i=1}^{m} A_{ki}y_i$$

$$= (2Ay)_k$$

$$\nabla\left((x^{(i)}-\mu)^T\Sigma^{-1}(x^{(i)}-\mu)\right) =$$

$$= \nabla(①) - \nabla(②) - \nabla(③) + \nabla(④)$$

$$= 0 - \Sigma^{-1}x^{(i)} - \Sigma^{-1}x^{(i)}$$

$$+ 2\cdot\Sigma^{-1}\mu$$

$$= 2\Sigma^{-1}(\mu - x^{(i)})$$

$$\nabla_\mu(\ell(\mu,\Sigma)) = -\frac{1}{2}\sum_{i=1}^{n}\left(2\Sigma^{-1}(\mu - x^{(i)})\right)$$

$$= \sum_{i=1}^{n} \Sigma^{-1}(x^{(i)} - \mu)$$

$$= \Sigma^{-1}\sum_{i=1}^{n}(x^{(i)} - \mu) \overset{?}{=} 0$$

irrespective of what Σ is the solution is the same as the solution to

$$\sum_{i=1}^{n}(x^{(i)} - \mu) = 0$$

equiv. $\frac{1}{n}\sum_{i=1}^{n} x^{(i)} - \mu = 0$

MLE $\boxed{\hat{\mu} = \bar{X}_n}$

Recall: $\bar{X}_n \sim N(\mu, \frac{1}{n}\Sigma)$

plug $\mu = \hat{\mu}$

$$\ell(\bar{x}_n, \Sigma) = -\frac{nm}{2}\log(2\pi)$$
$$-\frac{n}{2}\log\det\Sigma$$
$$-\frac{1}{2}\sum_{i=1}^{n}(x^{(i)}-\bar{x}_n)^T\Sigma^{-1}(x^{(i)}-\bar{x}_n)$$

$$\left\{\begin{array}{l} A\in\mathbb{R}^{n\times m} \quad B\in\mathbb{R}^{m\times n} \\ \mathrm{tr}(AB) = \mathrm{tr}(BA) \\ \\ \underline{x}, \underline{y}\in\mathbb{R}^{m} \\ \underline{x}^T\underline{y} = \mathrm{tr}(x^T y) = \mathrm{tr}(\underline{y}\,\underline{x}^T) \end{array}\right.$$

$$\ell(\bar{x}, \Sigma) = -\frac{nm}{2}\log(2\pi) - \frac{n}{2}\log\det\Sigma$$
$$-\frac{1}{2}\sum_{i=1}^{n}\mathrm{tr}\left(\Sigma^{-1}(x^{(i)}-\bar{x}_n)(x^{(i)}-\bar{x}_n)^T\right)$$

$$= -\frac{nm}{2}\log(2\pi) - \frac{n}{2}\log\det\Sigma - \frac{n}{2}\operatorname{tr}\left(\Sigma^{-1}\underbrace{\left(\frac{1}{n}\sum_{i=1}^{n}(x^{(i)}-\bar{x}_n)(x^{(i)}-\bar{x}_n)^T\right)}_{S_n}\right)$$

$$= -\frac{nm}{2}\log(2\pi) - \frac{n}{2}\log\det\Sigma - \frac{n}{2}\operatorname{tr}(\Sigma^{-1}S_n)$$

param. $K = \Sigma^{-1}$

$$\ell(\bar{x}; K) = -\frac{nm}{2}\log(2\pi) + \frac{n}{2}\log\det K - \frac{n}{2}\operatorname{tr}(KS_n)$$

(K ← inv. cov)

∇_K matrix with entries $\frac{\partial \ell}{\partial K_{ij}}$

$$\nabla_K \ell(\bar{x}; K) = \frac{n}{2}K^{-1} - \frac{n}{2}S_n$$

MLE $\hat{K}$: $\hat{K}^{-1} = S_n$

- is S_n not invertible, no solution
- if it is equiv. $\hat{\Sigma} = S_n$

Prop: $\underline{x}^{(1)}, \ldots, \underline{x}^{(n)} \sim N(\mu, \Sigma)$

$\bar{X}_n \perp\!\!\!\perp S_n$ $\qquad \underline{X} = \begin{pmatrix} - & x^{(1)} & - \\ & \vdots & \\ - & x^{(n)} & - \end{pmatrix}$

Proof: Recall $S_n = \frac{1}{n} X^T H X$

$H = I_n - \frac{1}{n} \mathbb{1}_n \mathbb{1}_n^T$ centering matrix

$S_n = \frac{1}{n} (HX)^T (HX)$

equiv. show $\underline{\bar{X}_n} \perp\!\!\!\perp \underline{HX}$

k^{th} row of $HX = x^{(k)} - \bar{X}_n$

$$\mathrm{Cov}\left(\bar{X}_n, X^{(k)} - \bar{X}_n\right)$$

$$= \mathrm{Cov}\left(\underset{\frac{1}{n}\sum_{i=1}^{n} X^{(i)}}{\bar{X}_n}, X^{(k)}\right) - \underbrace{\mathrm{Var}(\bar{X}_n)}_{\frac{1}{n}\cdot\Sigma}$$

$$= \frac{1}{n}\cdot\underbrace{\mathrm{Cov}\left(X^{(k)}, X^{(k)}\right)}_{\Sigma} - \frac{1}{n}\Sigma = 0_{m\times m}$$

so $HX \perp\!\!\!\perp \bar{X}$

so $S_n \perp\!\!\!\perp \bar{X}$ $\Rightarrow$

STA 437/2005: Methods for Multivariate Data

Week 4: Gaussian Processes

Piotr Zwiernik

University of Toronto

Table of contents

Introduction to GPs

Marginal distribution of MVN

$X = (X_1, \ldots, X_m)$

Consider the following reformulation of the earlier result:

Suppose $X \sim N_m(\mu, \Sigma)$. Let $T := \{1, \ldots, m\}$ and define $= \mathbb{E} X_i$

- $m : T \to \mathbb{R}$ such that $m(i) := \mu_i$ (mean function)
- $k : T \times T \to \mathbb{R}$ such that $k(i, j) := \Sigma_{ij}$ (kernel function)

Then for every $A = \{t_1, \ldots, t_n\} \subseteq T$, the vector $X_A = (X_{t_1}, \ldots, X_{t_n})$ is Gaussian with

- The mean μ_A whose i-th entry is $m(t_i)$.
- The covariance matrix Σ_{AA} whose (i, j)-th entry is $k(t_i, t_j)$.

The set T indexes all random variables in the system.

For every $A = \{t_1, \ldots, t_n\} \subseteq T$, $(X_{t_1}, \ldots, X_{t_n})$ is Gaussian.

Gaussian Processes - an immediate generalization

A **Gaussian Process (GP)** is a generalization of the multivariate normal distribution to a collection of random variables indexed by an arbitrary set T.

Definition

A Gaussian Process is a collection of random variables $\{X_t\}_{t \in T}$ such that for any finite set of points $\{t_1, \ldots, t_n\} \subset T$, the corresponding vector $(X_{t_1}, \ldots, X_{t_n})$ follows a multivariate normal distribution.

In what follows we assume $T \subseteq \mathbb{R}^d$ with the Euclidean distance metric.

Often, the correlation between two variables X_s and X_t will depend on the distance $\|t - s\|$.

The mean and the kernel functions

A Gaussian Process is characterized by:

- A **mean function** $m : T \to \mathbb{R}$: $\quad m(t) = \mathbb{E}[X_t]$
- A **kernel function** $k : T \times T \to \mathbb{R}$: $\quad k(t, t') = \mathrm{Cov}(X_t, X_{t'})$

Note that m is pretty much arbitrary (often set to be zero) but k is highly constrained:

Positive semi-definitness:

For any finite set $\{t_1, \ldots, t_n\} \subset T$, the covariance matrix Σ with entries $\Sigma_{ij} = k(t_i, t_j)$ is positive semi-definite.

We can use feature maps $\psi : \mathbb{R}^d \to \mathbb{R}^p$ to define kernels:

$$k(s, t) = \psi(s)^\top \psi(t).$$

Feature maps define kernels but not all kernels are like that (this can be generalized to "infinite dimensional" feature maps).

Common Kernels in GPs

- **Squared Exponential (RBF) Kernel:**

$$k_{\mathrm{E}}(t, t') = \sigma^2 \exp\left(-\frac{\|t - t'\|^2}{2\ell^2}\right).$$

  - Controls smoothness of the functions sampled from the GP.
  - Length scale ℓ: Correlation distance.
  - Signal variance σ^2: Scale of the output.

- **Matérn Kernel:**

$$k_{\mathrm{M}}(t, t') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{\|t - t'\|}{\ell}\right)^{\nu} K_{\nu}\left(\sqrt{2\nu}\frac{\|t - t'\|}{\ell}\right).$$

  - ν: Smoothness parameter.
  - More flexible than the RBF kernel for modeling rough functions.

Constructing kernels from kernels

Given valid kernels $k_1(\mathbf{x}, \mathbf{x}')$ and $k_2(\mathbf{x}, \mathbf{x}')$, the following kernels will also be valid:

$$
\begin{aligned}
k(\mathbf{x}, \mathbf{x}') &= c k_1(\mathbf{x}, \mathbf{x}') \quad \text{for } \; c > 0, \\
k(\mathbf{x}, \mathbf{x}') &= f(\mathbf{x}) k_1(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') \\
k(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}') \\
k(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') \cdot k_2(\mathbf{x}, \mathbf{x}') \\
k(\mathbf{x}, \mathbf{x}') &= \mathbf{x}^\top A \mathbf{x}' \qquad (A \text{ PSD}) \\
k(\mathbf{x}, \mathbf{x}') &= \exp(k_1(\mathbf{x}, \mathbf{x}')) \\
k(\mathbf{x}, \mathbf{x}') &= q(k_1(\mathbf{x}, \mathbf{x}'))
\end{aligned}
$$

where q polynomial with ≥ 0 coefficients.

Modelling with Gaussian processes

Working with Gaussian Processes we fix a kernel function.

Data: Suppose we observed $(X_{t_1}, \ldots, X_{t_n})$ for some $t_1, \ldots, t_n \in T$.

If the kernel function comes with some hyperparameters α, we can learn them maximizing the log-likelihood.

- By definition, $(X_{t_1}, \ldots, X_{t_n})$ is MVN with covariance that depends on α.
- This may be a complicated optimization procedure.

Suppose we want to predict the value of the process at some point t_{n+1}

- By definition $(X_{t_1}, \ldots, X_{t_n}, X_{t_{n+1}})$ is jointly Gaussian so simply compute the conditional distribution: $X_{t_{n+1}} | X_{t_1}, \ldots, X_{t_n}$.
- This gives both the point prodiction (the conditional mean) and uncertainty quantification (conditional variance).

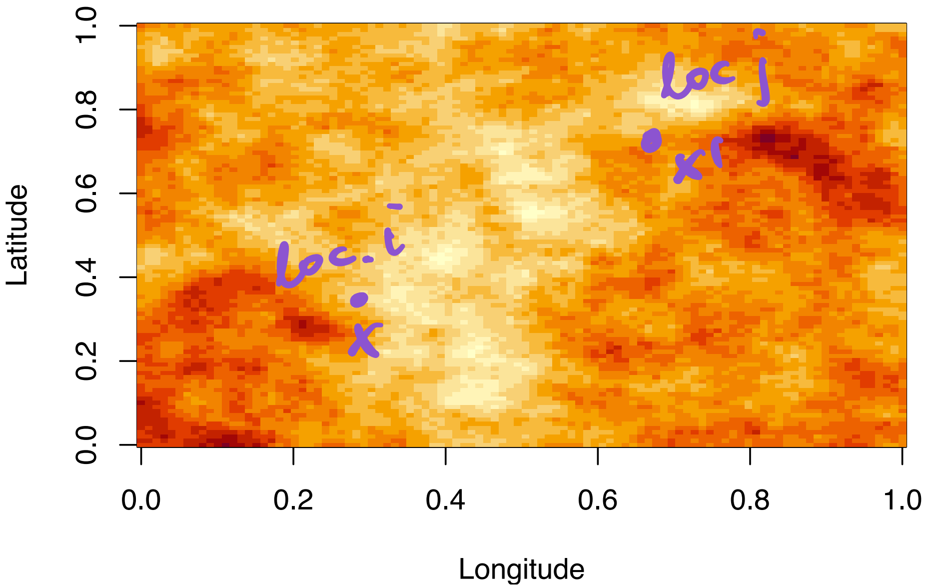
GPs for Spatial Data

Example: Modeling Spatial Data with GPs

$T = [0,1]^2$

GPs are widely used in spatial statistics, e.g. temperature across a grid of locations.

$\underline{x}, \underline{x}' \in T$



- Grid of 100^2 points.
- Fix the exponential kernel $\exp\{-\frac{1}{2}\|\mathbf{x} - \mathbf{x}'\|\}$
- Compute the $100^2 \times 100^2$ covariance matrix
- Get 1 sample from the corresponding distr.

Handling a 10000-dimensional Gaussian comes with its own computational challenges.

Spatial GP: Prediction

We explained how to make a prediction for $X_{t_{n+1}}$. This easily generalizes.

Suppose we observed the mean zero GP over some locations $\mathbf{x}_{\text{train}}$.

Our goal is to make predictions over some other points $\mathbf{x}_{\text{test}}$

1. Combine training and test locations.
2. Compute the covariance matrix using the kernel function.
3. Use Gaussian conditioning formulas:

$$\mathbb{E}[\mathbf{x}_{\text{test}}|\mathbf{x}_{\text{train}}] = \Sigma_{\text{test,train}}\Sigma^{-1}_{\text{train,train}}\mathbf{x}_{\text{train}},$$
$$\text{Cov}(\mathbf{x}_{\text{test}}|\mathbf{x}_{\text{train}}) = \Sigma_{\text{test,test}} - \Sigma_{\text{test,train}}\Sigma^{-1}_{\text{train,train}}\Sigma_{\text{test,train}}.$$

Nonparametric Regression with GPs

Nonparametric Regression

GPs can be used for nonparametric regression:

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), \quad i = 1, \ldots, n.$$

Prior over $f : \mathbb{R}^d \to \mathbb{R}$: GP defined by $m(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$.

- In this sense GP defines a distribution over (random) functions $f : \mathbb{R}^d \to \mathbb{R}$.

We have $(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)) \sim N_n(\mu, \Sigma)$

- $\mu_i = m(\mathbf{x}_i)$
- $\Sigma_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$

Say $d = 1$. Given $m(x)$ and $k(x, x')$, how would you plot random samples of the corresponding random functions on $\mathbb{R}$?

Nonparametric Regression

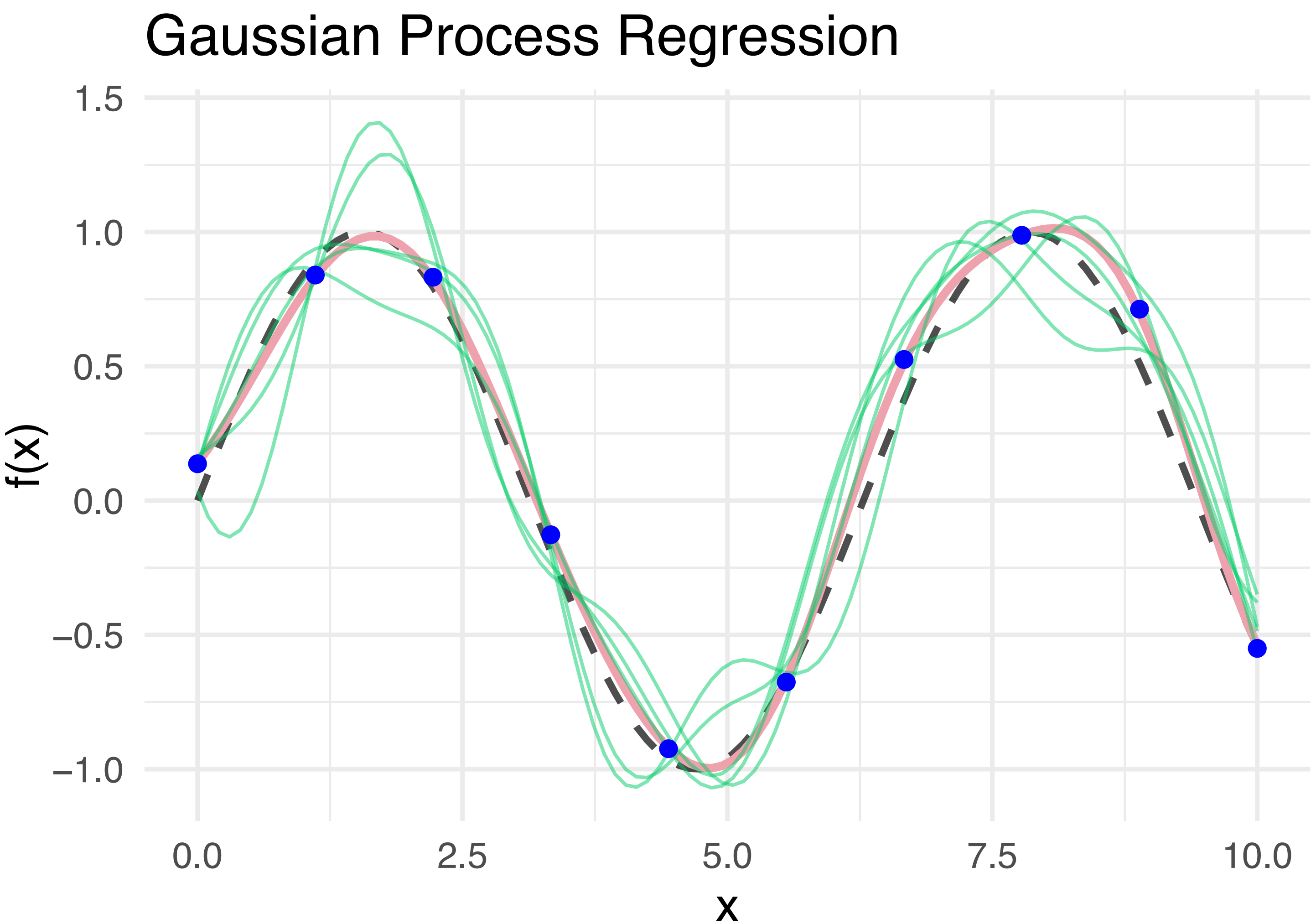
Note that $\mathbf{y} = (y_1, \ldots, y_n) = (f(\mathbf{x}_1) + \varepsilon_1, \ldots, f(\mathbf{x}_n) + \varepsilon_n)$.

Consider the underlying Gaussian Process $y(\mathbf{x})$:

- The mean is $m(\mathbf{x})$.
  - $\mathbb{E}[y(\mathbf{x}_i)] = \mathbb{E}[f(\mathbf{x}_i) + \varepsilon_i] = m(\mathbf{x}_i)$.
- The kernel is $k(\mathbf{x}, \mathbf{x}') + \sigma^2 \mathbf{1}\{\mathbf{x} = \mathbf{x}'\}$.
  - $\text{cov}[y(\mathbf{x}_i), y(\mathbf{x}_j)] = \text{cov}(f(\mathbf{x}_i) + \varepsilon_i, f(\mathbf{x}_j) + \varepsilon_j) = k(\mathbf{x}_i, \mathbf{x}_j) + \sigma^2 \mathbf{1}\{\mathbf{x}_i = \mathbf{x}_j\}$.

Given data $(y_1, \mathbf{x}_1)$, ..., $(y_n, \mathbf{x}_n)$ we can now easily predict y at any other point $\mathbf{x}$.

Illustration



Summary

- Gaussian Processes are a versatile tool for regression and spatial modeling.
- Key components:
  - Mean function.
  - Kernel function.
- Takeaway: Conceptually it is not harder than MVNs and the same formulas apply.
- Computational issues can be significant.