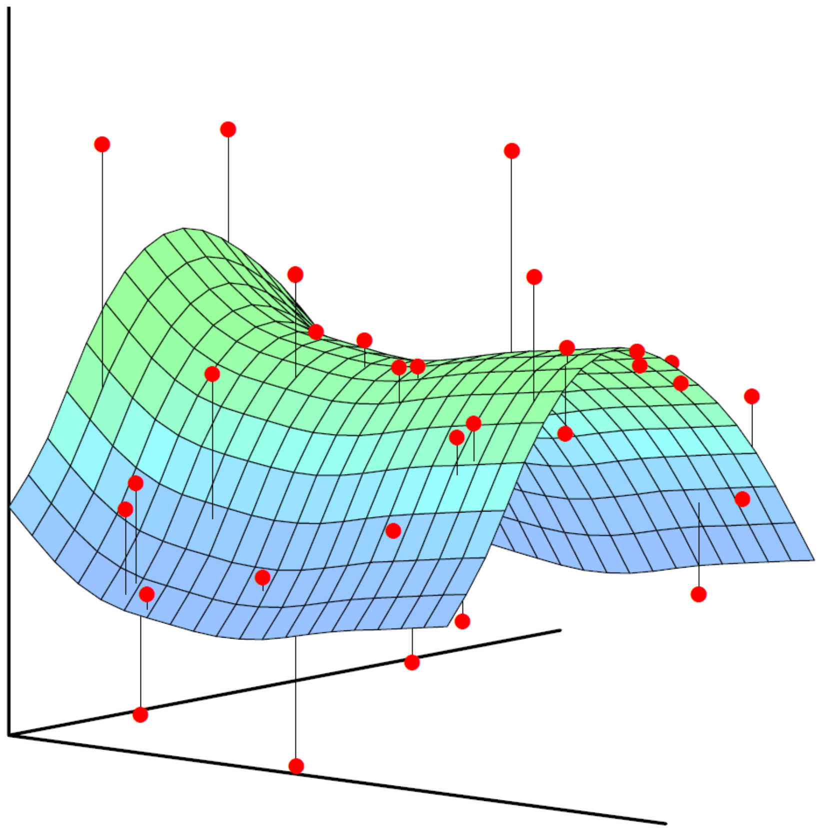
Machine Learning



第8讲 高斯过程
Gaussian Process

刘 峤

电子科技大学计算机科学与工程学院

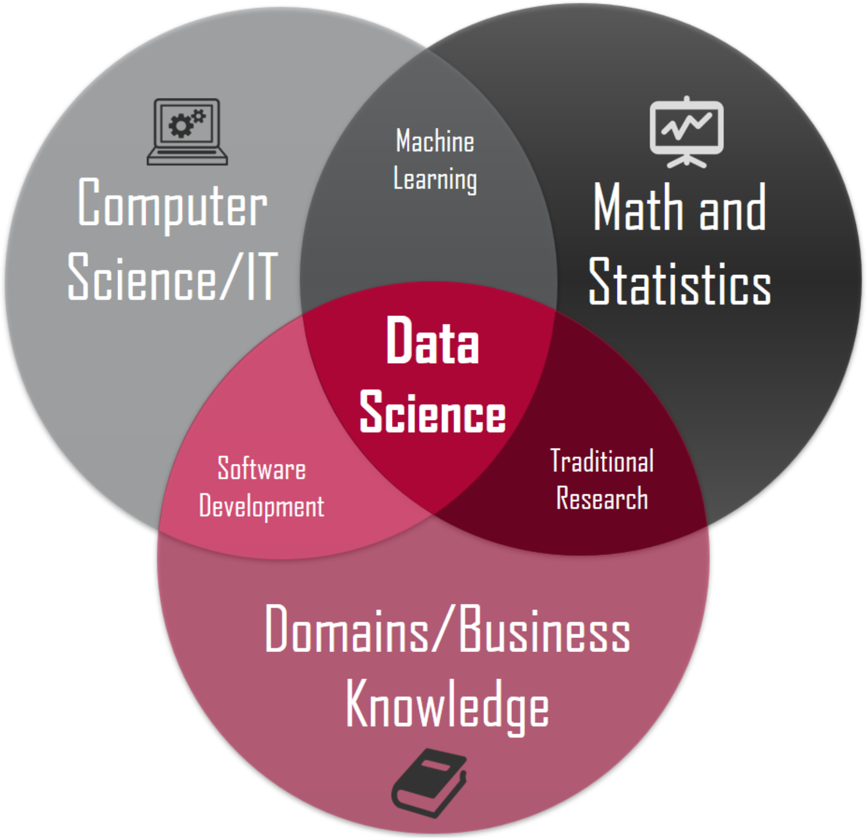
8.0 Preliminaries

What is Machine Learning?

A branch of artificial intelligence. As intelligence requires knowledge, it is necessary for the computers to acquire knowledge.

ML concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data.

Statistics + Algorithms



Rob Tibshirani: Stats 101: Data Science

Statistical learning refers to a set of tools for understanding data.

A Brief History of Statistical Learning

- **1900s：最小二乘法（Legendre and Gauss)**

  ※ 定量值预测问题（ predicting quantitative values)

- **1936年：线性判别分析法（ Fisher)**

  ※ 定性值预测问题（ predicting qualitative values)

- **1940s：逻辑斯蒂回归方法（ various authors)**
- **1970s：广义线性模型（ Nelder and Wedderburn)**
- **1980s：CART（ Breiman, Friedman, Olshen and Stone)**
- **1990s：支持向量机（ Vapnik-Chervonenkis)**

Types of Learning

supervised:

- **Given an observation x, what is the *best label y*?**
  - ※ **Predictive**: What will happen?

unsupervised:

- **Given a set of x's, *cluster* or *summarize* them?**
  - ※ **Descriptive**: What happened?

Reinforcement:

- ***Translate* state to action to *maximize reward*?**
  - ※ **Prescriptive**: What should we do?

Relation to probability

- **Machine learning is (often) modeling a probability distribution**
  - ※ Probabilistic reasoning is central to many machine learning tasks
  - ※ Probability is an useful way of quantifying our *beliefs* about the state of the world.
- **supervised learning**
  - ※ **optimization:** $argmin(\hat{f}(x) - y)$
  - ※ **conditional probability estimation:** $P(y|x)$
- **unsupervised learning**
  - ※ **generative model:** $P(x)$

Generative vs. discriminative models

- **Many of the methods model one of the following probability distributions**

  ※ **P(X)**， joint probability distribution **P(X, Y)**， conditional probability **P(Y|X)**.

- **Generative model**

  ※ **supervised :** *estimating* **P(X,Y)** **--> eg. Naive Bayes Model**

  ※ **unsupervised :** *estimating* **P(X)** **--> eg. Gaussian Mixture Model**

- **Discriminative model**

  ※ **supervised :** *estimating* **P(Y|X)** **--> eg. Logistic Regression**

8.1 Introduction to GP

The Data are Not Enough

- Four pillars:
  - ※ **Deterministic/Stochastic**
  - ※ **Mechanistic/Emipirical**
- Goal: model **complex** phenomena **over time**
- Problem:
  - ※ Mechanistic models are often **inaccurate**
  - ※ Data is often **not rich enough** for an empirical approach
- How do we combine inaccurate physical model with machine learning?

Need to Model $p(t)$

- **Gaussian process: a probabilistic model for functions.**

  ※ Formally known as a stochastic process.

- Multivariate Gaussian is normally defined by

  ※ a mean vector, $\boldsymbol{\mu}$, and a covariance matrix, $\boldsymbol{\Sigma}$.

$$\mathbf{y} \sim \mathcal{N}_k(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

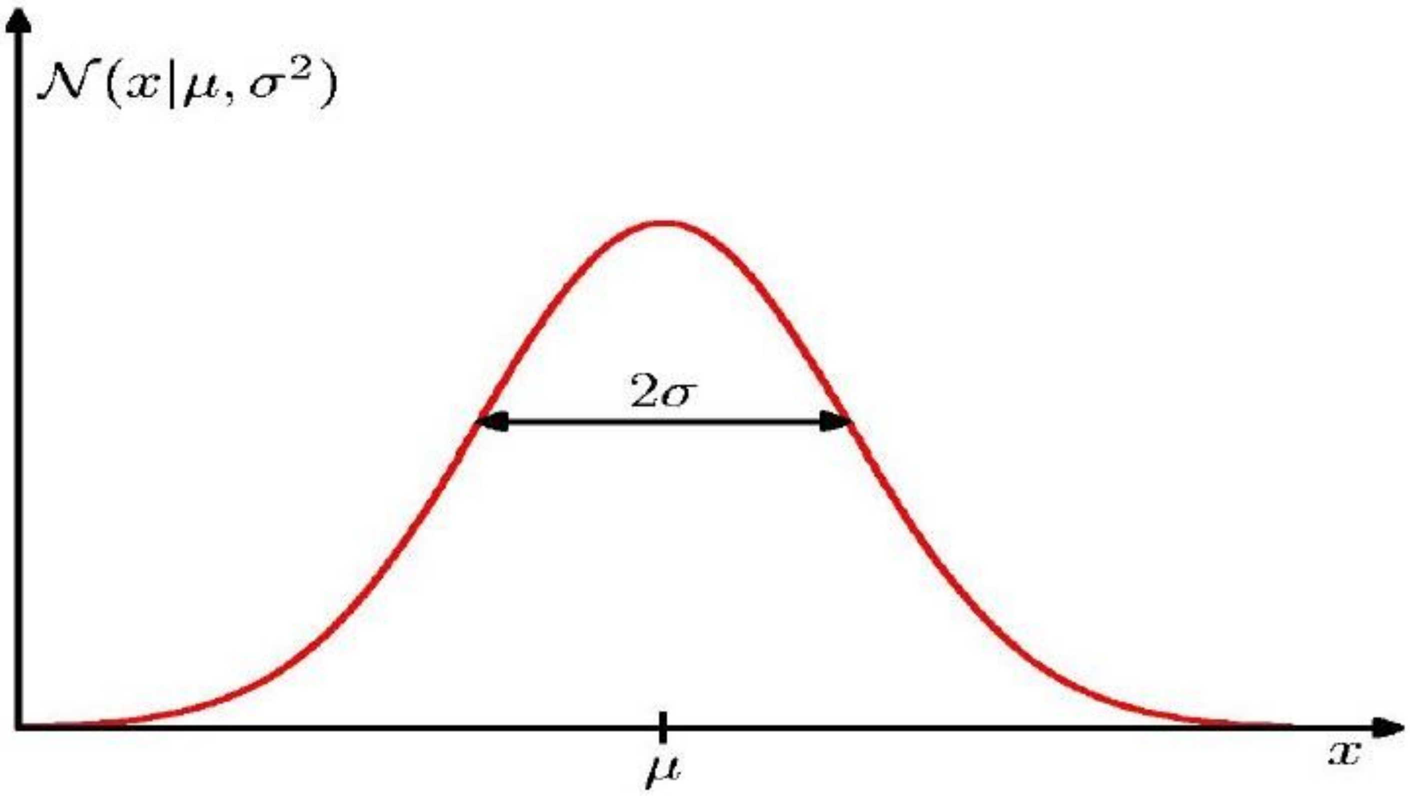
- Gaussian process defined by

  ※ a mean function, $\boldsymbol{\mu}(t)$, and a covariance function, $\boldsymbol{\Sigma}(t, t')$

$$\mathbf{y}(t) \sim \mathcal{N}_k(\boldsymbol{\mu}(t), \boldsymbol{\Sigma}(t, t'))$$

Univariate Gaussian Distribution

- Parameters: **Mean** (μ) ; **Variance** (σ^2)



$$P(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$

Multivariate Gaussian Distribution

- aka： joint normal distribution
- a random vector is said to be k-variate normally distributed if every linear combination of its k components has a 1D normal distribution.

$$\mathbf{x} \ \sim \ \mathcal{N}_k(\boldsymbol{\mu}, \ \boldsymbol{\Sigma}) \qquad \text{where } \ \mathbf{x} = (x_1, \ldots, x_k)^{\mathrm{T}}$$

with k-dimensional mean vector

$$\boldsymbol{\mu} = \mathrm{E}[\mathbf{x}] = (\mathrm{E}[x_1], \mathrm{E}[x_2], \ldots, \mathrm{E}[x_k])^{\mathbf{T}}$$

and $k \times k$ covariance matrix

$$\Sigma_{i,j} = \mathrm{E}[(X_i - \mu_i)(X_j - \mu_j)] = \mathrm{Cov}[X_i, X_j]$$

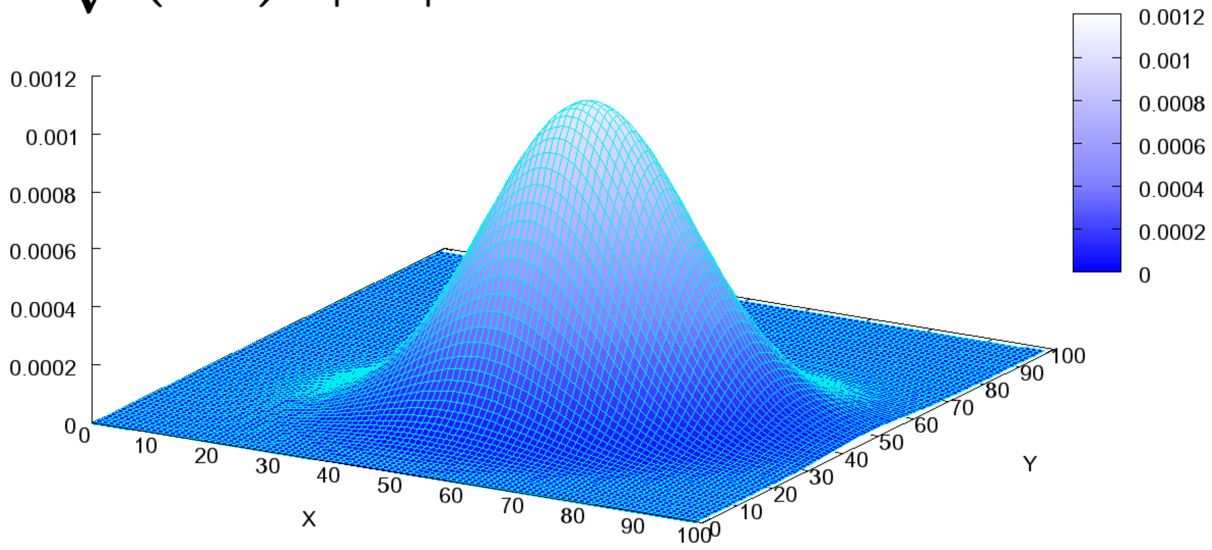
Multivariate Normal Distribution: Density function

- The multivariate normal distribution is said to be "non-degenerate" when the symmetric covariance matrix $\boldsymbol{\Sigma}$ is positive definite.
- In this case the distribution has density:

$$f_{\mathbf{X}}(x_1, \ldots, x_k) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^k|\boldsymbol{\Sigma}|}}$$

where $|\boldsymbol{\Sigma}| \equiv \det \boldsymbol{\Sigma}$

is the determinant of $\boldsymbol{\Sigma}$



Gaussian Process

- 随机过程：是依赖于时间参数的一组随机变量(的全体)。

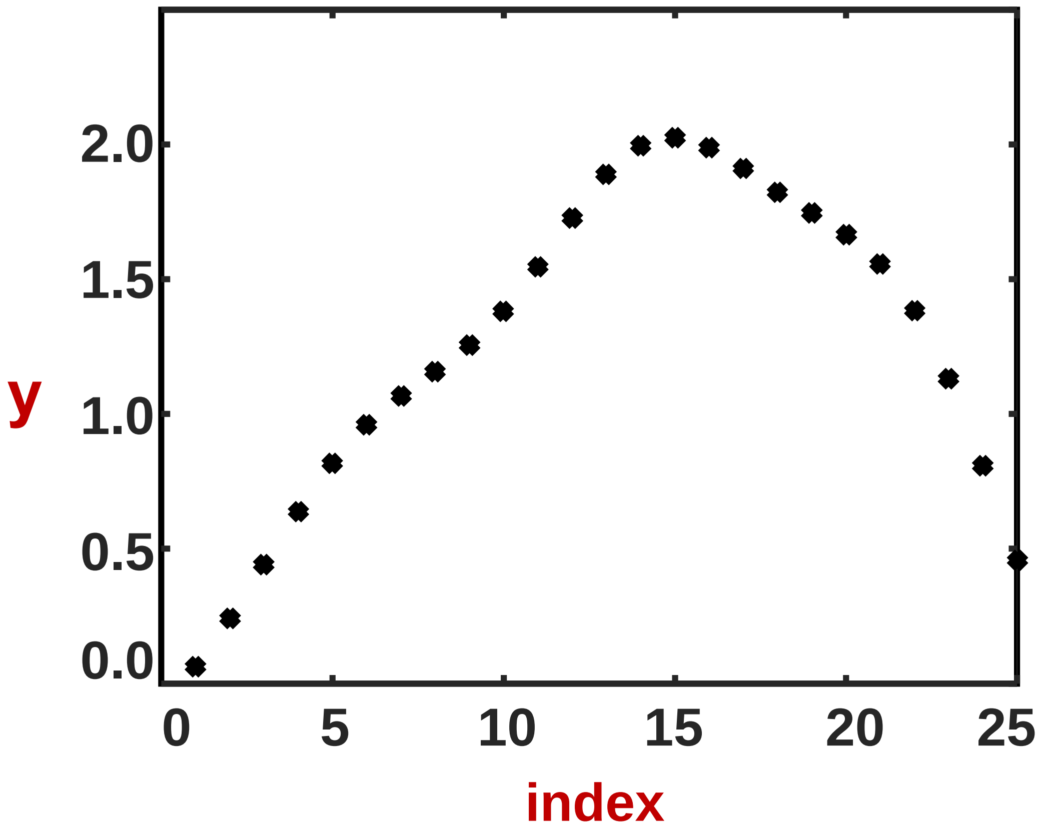
  ※ 布朗运动、泊松过程、马尔可夫过程、高斯过程等

- A **Gaussian process** is a collection of random variables，any finite number of which have (consistent) Gaussian distribution.

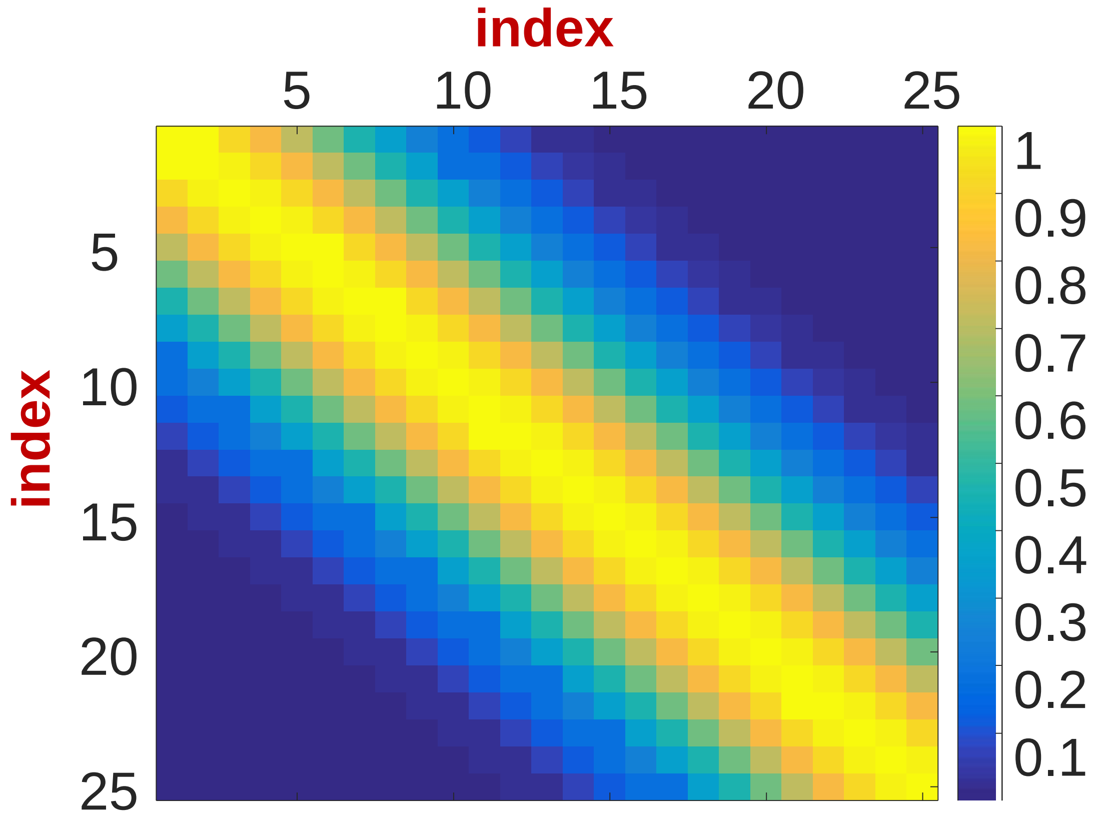
- Mathematically，for any set ***S***, a Gaussian process (GP) on ***S*** is a set of random variables ($f(x)$, $x \in S$) such that, for any $\{x_1, \ldots, x_n\} \subset S$, $P(f(x_1), \ldots, f(x_n))$ is multivariate Gaussian.

- Note: Although ***S*** can be any set, it usually is $\mathbb{R}^n$

Zero Mean Gaussian Sample

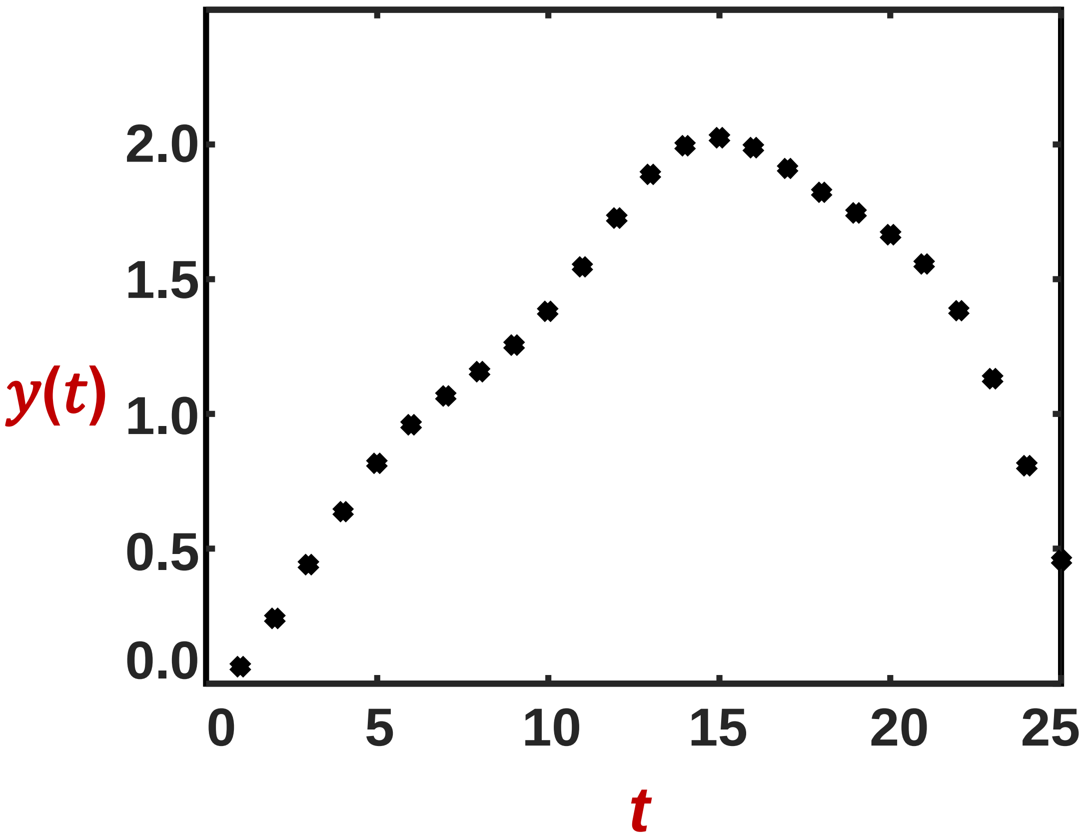


samples from Gaussian

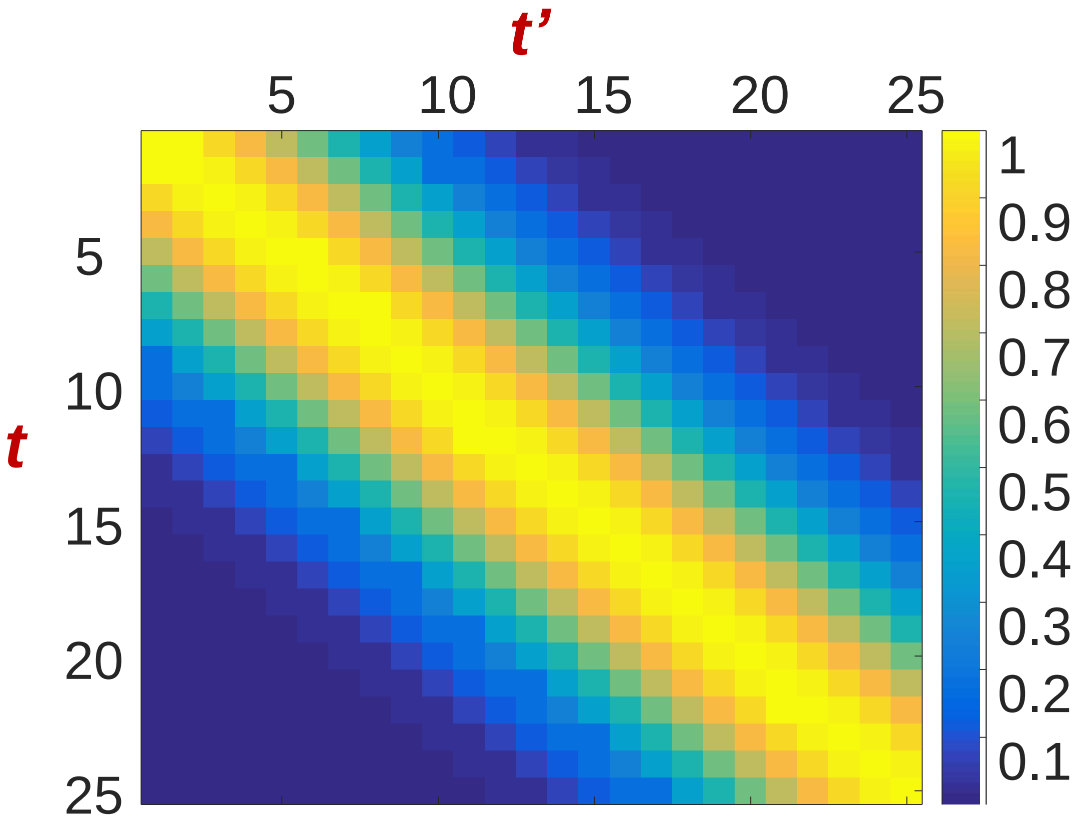


covariance Σ

Zero Mean Gaussian Process Sample



samples from Gaussian process



covariance function $\Sigma(t, t')$

高斯分布的优良性质

- **Marginalization and conditional distribution**

Let $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and partition $\boldsymbol{f}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ as

$$f = \begin{array}{cc} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} & \begin{array}{c} n_1 \\ n_2 \end{array} \end{array}, \quad \mu = \begin{array}{cc} \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} & \begin{array}{c} n_1 \\ n_2 \end{array} \end{array}, \quad \Sigma = \begin{array}{cc} \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} & \begin{array}{c} n_1 \\ n_2 \end{array} \\ \begin{array}{cc} n_1 & n_2 \end{array} & \end{array}$$

where: $\boldsymbol{f}, \boldsymbol{\mu} \in \mathbb{R}^n$, and $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$

Then: $P(\boldsymbol{f}_1) \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11})$

$$P(\boldsymbol{f}_2|\boldsymbol{f}_1) \sim \mathcal{N}(\boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}(\boldsymbol{f}_1 - \boldsymbol{\mu}_1),\ \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12})$$

高斯分布的优良性质

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad f = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \begin{matrix} n_1 \\ n_2 \end{matrix}, \quad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \begin{matrix} n_1 \\ n_2 \end{matrix}, \quad \Sigma = \begin{array}{cc} \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} & \begin{matrix} n_1 \\ n_2 \end{matrix} \\ \begin{matrix} n_1 & \quad n_2 \end{matrix} & \end{array}$$

- 若整体服从多元高斯分布，则部分也服从对应均值和协方差的高斯分布。

$$P(\boldsymbol{f}_1) \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_{11})$$

- 条件分布也服从高斯分布，且均值和协方差矩阵可以被唯一确定表示。

$$P(\boldsymbol{f}_2|\boldsymbol{f}_1) \sim \mathcal{N}(\boldsymbol{\mu}_2 + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}(\boldsymbol{f}_1 - \boldsymbol{\mu}_1),\ \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12})$$

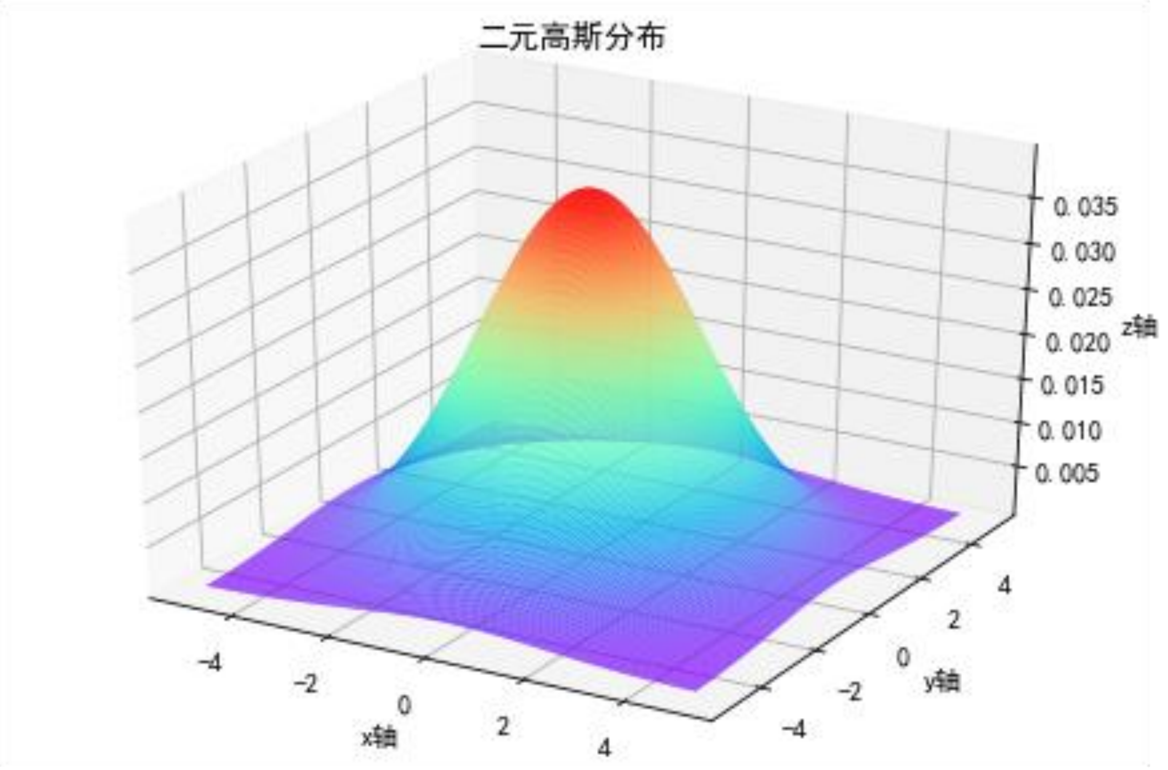
8.2 Kernels in Gaussian Process

Modeling Functions using Gaussians

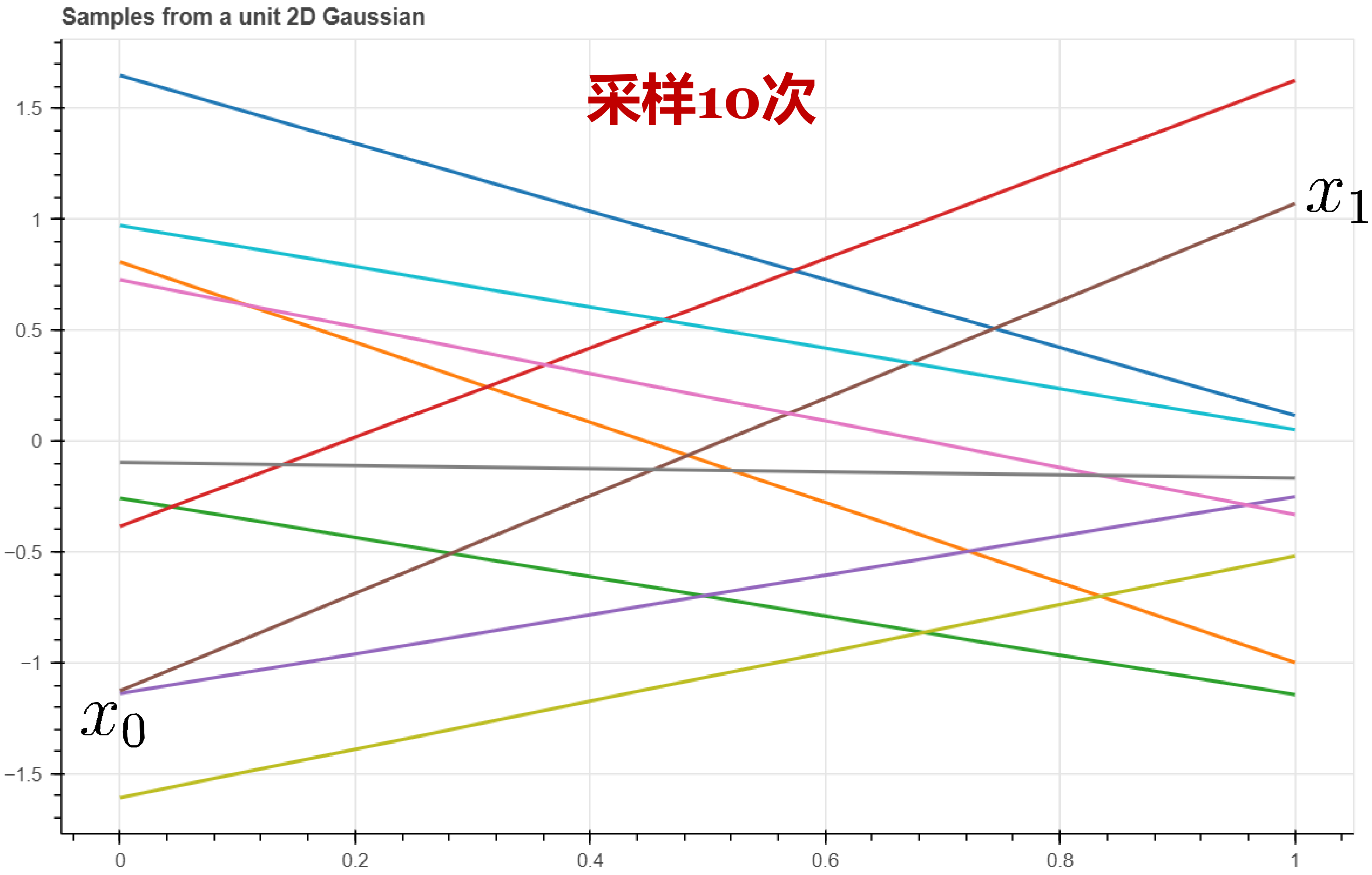
- 高斯过程的核心思想是可以用无限维多元高斯分布对函数进行建模
  - 即：将输入空间的每个点都关联到一个随机变量
  - 采用高斯分布对这组随机变量的联合分布进行建模
- 以二元高斯分布为例

$$\begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\right)$$

- 从中采样一个点得到：(x_0, x_1)

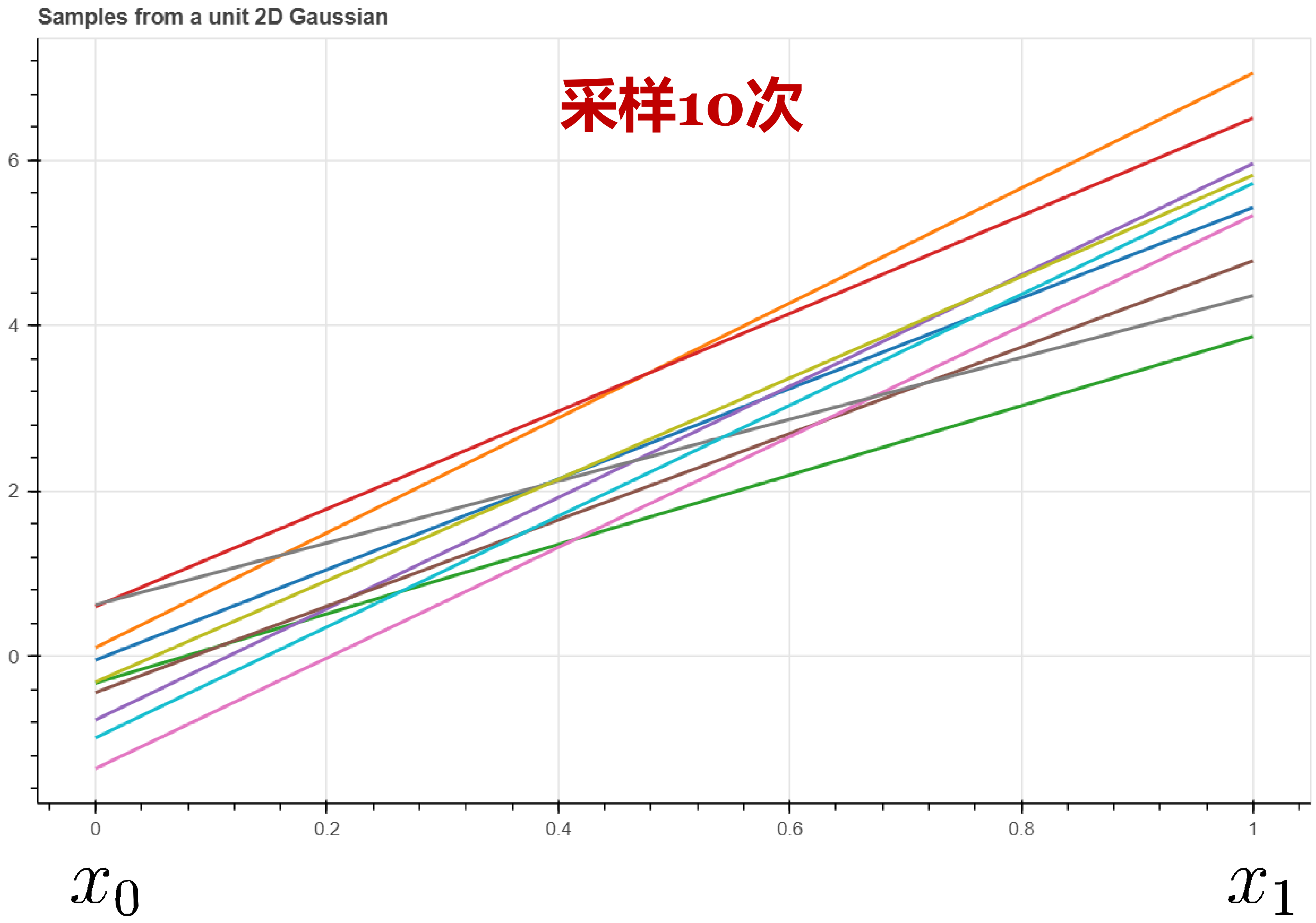


对二元高斯分布进行采样



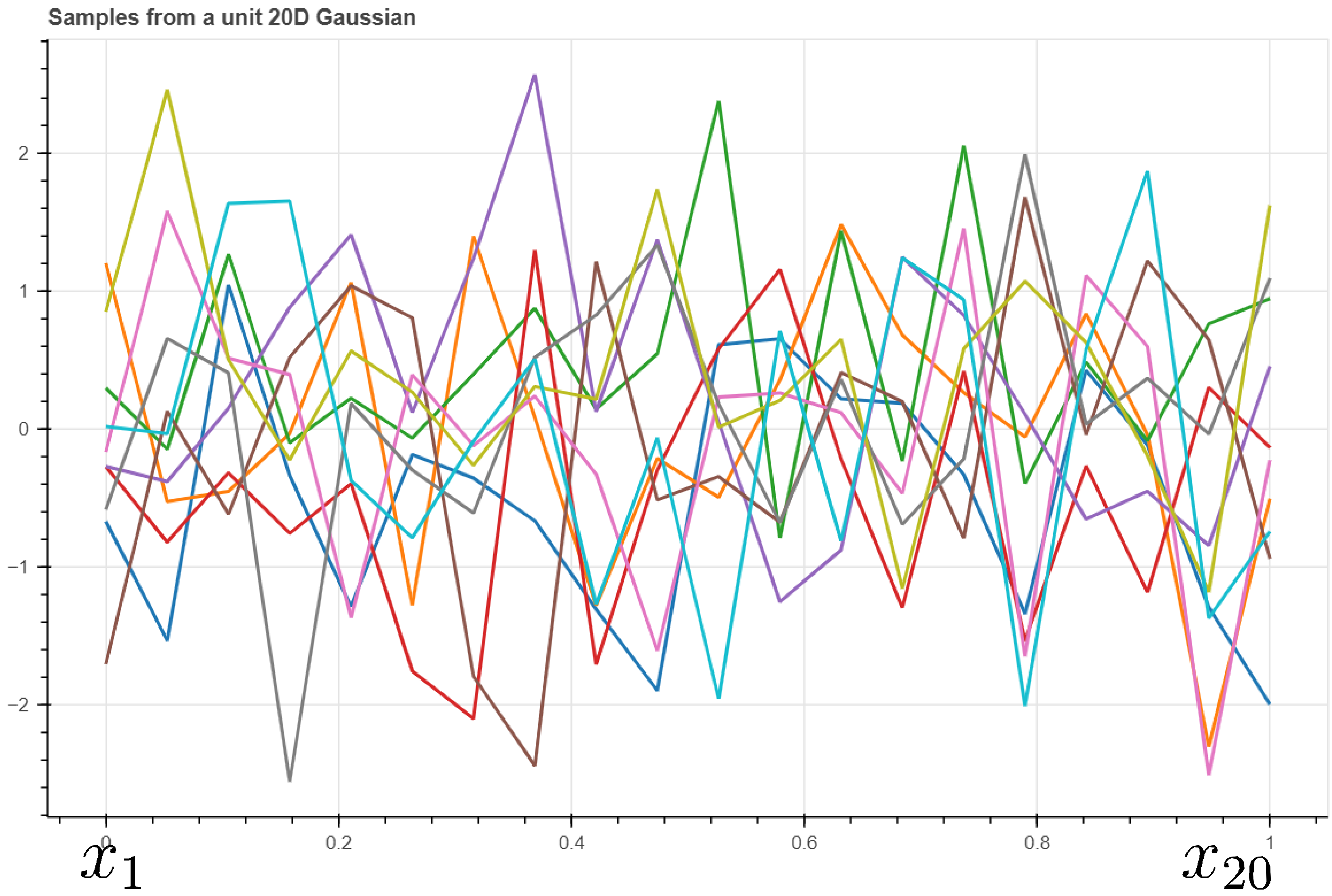
均值$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$

对二元高斯分布进行采样



均值 $\begin{pmatrix} 0 \\ 5 \end{pmatrix}$

对20元高斯分布进行采样



Smoothing with Kernels

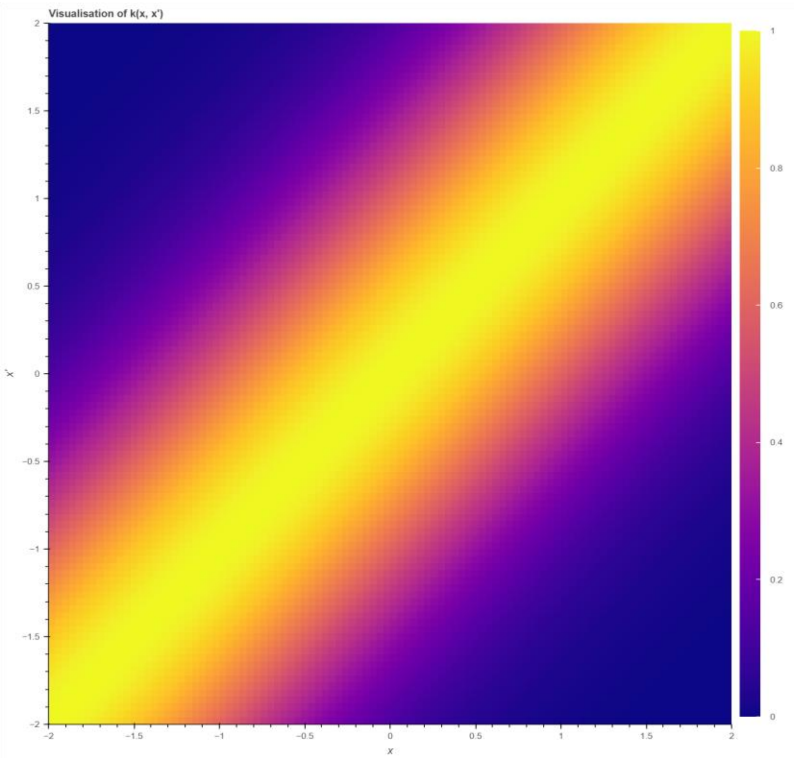
- 我们希望引入一些平滑性（smoothness） **目标:** $\mathrm{cov}(y_i, y_j) = \boldsymbol{K}(\mathbf{x}_i, \mathbf{x}_j)$
  - ※ 如果两个样本点彼此接近，则我们期望对应的随机变量值是相似的
  - ※ 即：相邻样本点对应的随机变量在联合分布中的协方差较高
- 如何定义协方差函数?
  - ※ 首先考察一下平方指数核函数

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

当 $\mathbf{x} = \mathbf{x}'$ 时: $\boldsymbol{K}(\mathbf{x}, \mathbf{x}') = 1$

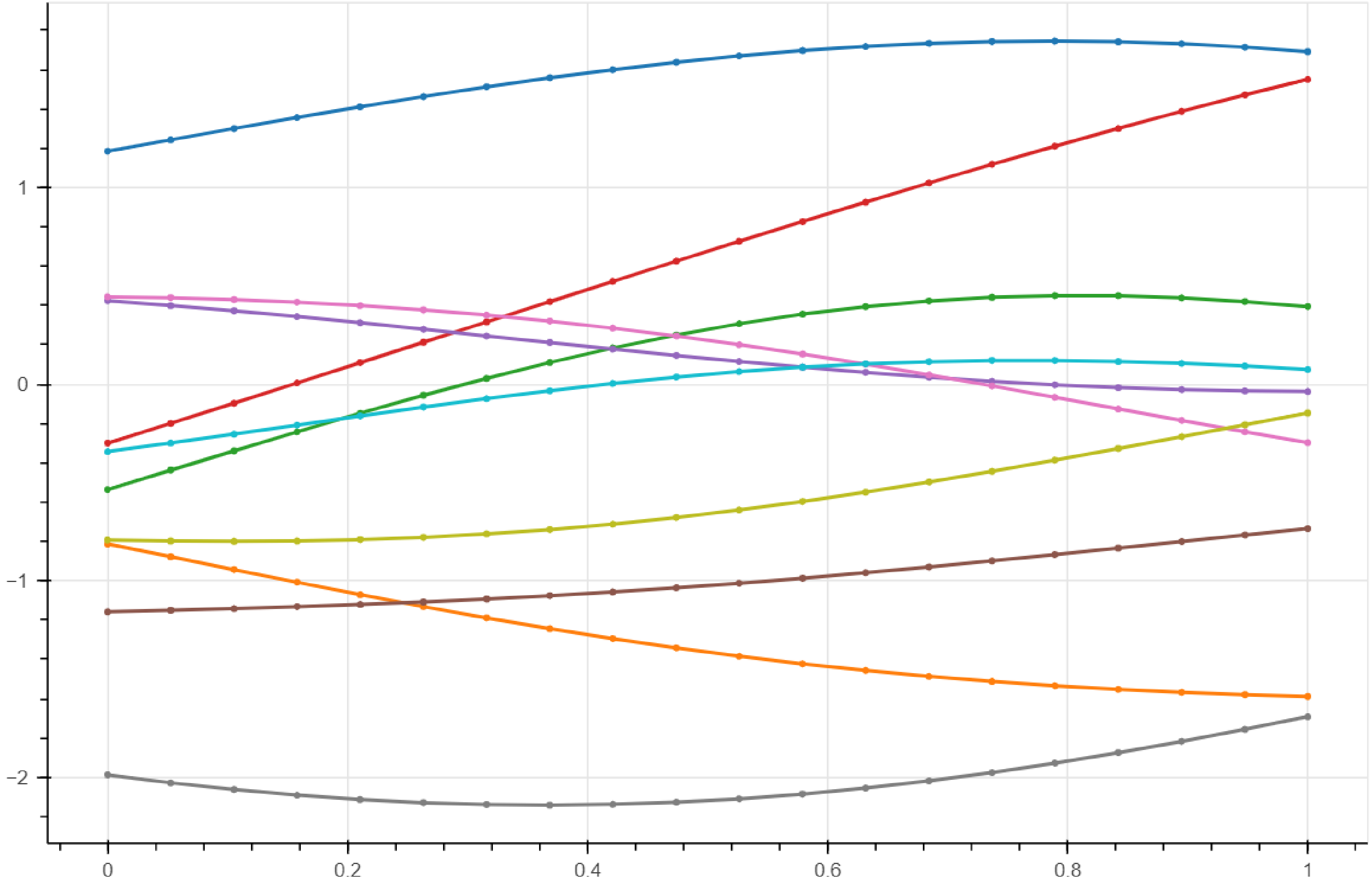
二者相差越大，核函数越趋于0

squared exponential kernel

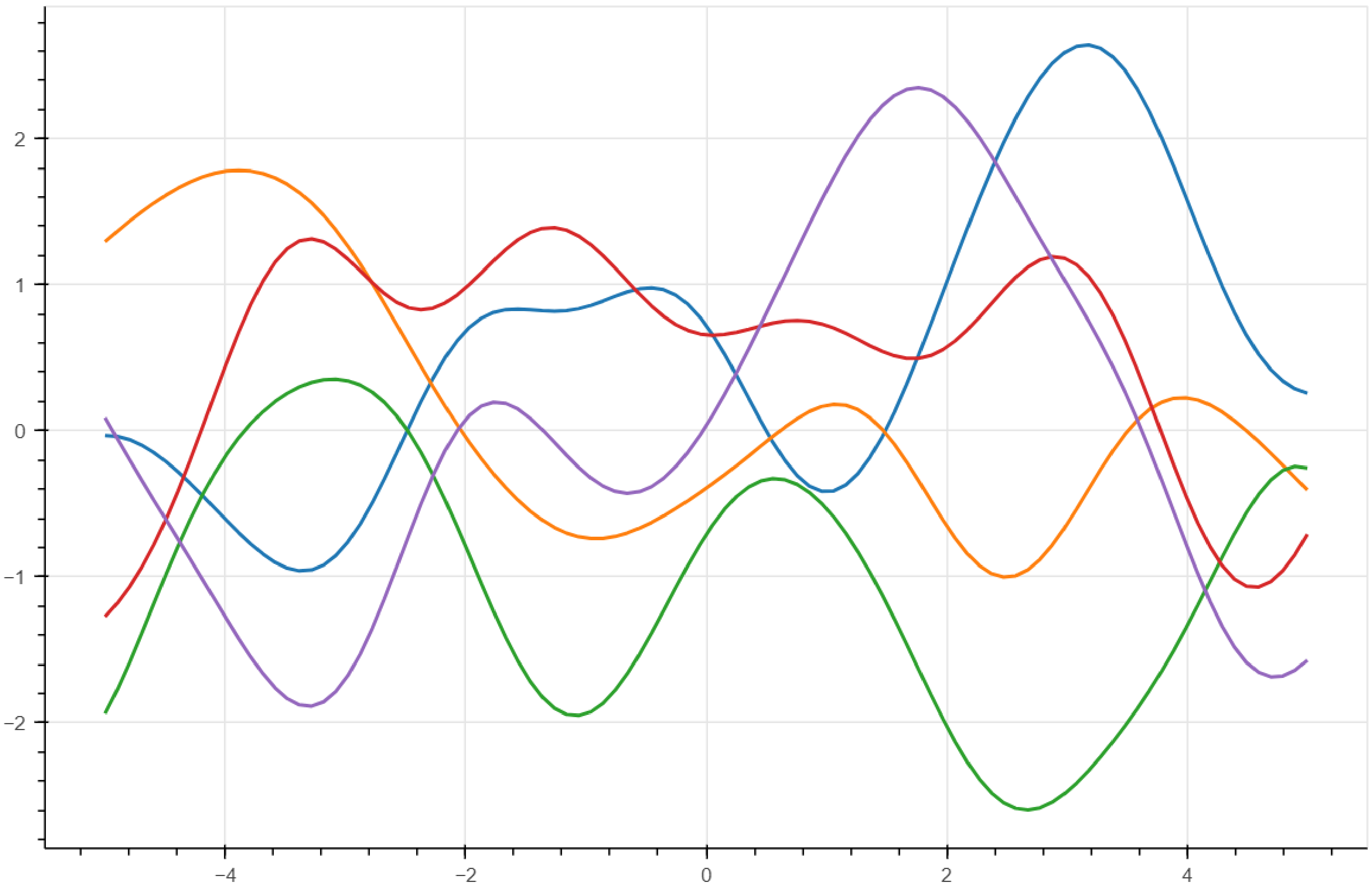


Smoothing with Kernels

x = np.linspace(0, 1, 20) $\text{cov}(y_i, y_j) = \boldsymbol{K}(\mathbf{x}_i, \mathbf{x}_j)$



对经过平滑处理后的100元高斯分布进行采样



x_1 x_{100}

Gaussian SVM revisit

- The RBF kernel is defined as

$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right)$$

Where γ is a parameter that sets the "spread" of the kernel.

- Gaussian SVM:

※ find α_i to linearly combine Gaussians centered at SVs $\mathbf{x}_i$

$$g_{svm}(x) = \text{sign}\left(\sum_{SV_i} \alpha_i y_i \exp(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2) + b\right)$$

※ achieve large margin in infinite dimensional space

The RBF kernel

- Proof: Without loss of generality, let $\gamma = \frac{1}{2}$

$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left\{-\tfrac{1}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right\} = \exp\left\{-\tfrac{1}{2}(\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}')\right\}$$

$$= \exp\left\{-\tfrac{1}{2}\left(\mathbf{x}^T(\mathbf{x} - \mathbf{x}') - \mathbf{x}'^T(\mathbf{x} - \mathbf{x}')\right)\right\}$$

$$= \exp\left\{-\tfrac{1}{2}\left(\mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{x}' - \mathbf{x}'^T\mathbf{x} + \mathbf{x}'^T\mathbf{x}'\right)\right\}$$

$$= \exp\left\{-\tfrac{1}{2}\left(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - 2\mathbf{x}^T\mathbf{x}'\right)\right\}$$

$$= \exp\left\{-\tfrac{1}{2}\left(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2\right)\right\} \exp\left\{\mathbf{x}^T\mathbf{x}'\right\}$$

$$= Ce^{\mathbf{x}^T\mathbf{x}'} \Rightarrow C := \exp\left\{-\tfrac{1}{2}\left(\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2\right)\right\} \text{ is a constant}$$

$$= C\sum_{n=0}^{\infty} \frac{(\mathbf{x}^T\mathbf{x}')^n}{n!} \Rightarrow \text{Taylor expansion of } e^x \;\; = C\sum_{n=0}^{\infty} \frac{K_{poly(n)}(\mathbf{x}^T\mathbf{x}')}{n!}$$

8.3 Gaussian Process Regression

Parametric vs. Nonparametric Methods

- 在监督学习中，我们经常使用参数模型 $p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta})$ 来解释数据并通过极大似然法推断参数 $\boldsymbol{\theta}$ 的最优值。
- 使用具有固定数量参数的模型的方法称为：**Parametric methods**。
- 在非参数（**Nonparametric**）方法中：参数的数量取决于数据集的大小

※ 例如，在Nadaraya-Watson kernel regression中

※ 为每个观测目标 y_i 分配一个权重 w_i

※ 为了预测一个新的输入 $\mathbf{x}$ 对应的目标值，需要计算加权平均值：

$$f(\mathbf{x}) = \sum_{i=1}^{N} w_i(\mathbf{x}) y_i, \quad w_i(\mathbf{x}) = \frac{\kappa(\mathbf{x},\mathbf{x}_i)}{\sum_{i'=1}^{N} \kappa(\mathbf{x},\mathbf{x}_{i'})}$$

Gaussian Process

- 高斯过程可用于直接推断函数的分布

  ※ 而不是推断函数参数的分布，因此也属于非参数方法

- 高斯过程定义了一个函数的先验分布（prior distribution）

  ※ 可以利用贝叶斯定理将其转换为函数的后验（posterior）

$$P(f|D) = \frac{P(f)P(D|f)}{P(D)}$$

- 在这种情况下，对连续函数值的推断称为GP回归

  ※ Gaussian Process也可用于分类

Gaussian Process

- 高斯过程是一个随机过程，其中任意点 $\mathbf{x} \in \mathbb{R}^d$ 被分配一个随机变量 $f(\mathbf{x})$
- 有限个这样的随机变量 $(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))$ 的联合分布服从高斯分布：

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

$$\boldsymbol{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)), \quad \boldsymbol{\mu} = (\mu(\mathbf{x}_1), \ldots, \mu(\mathbf{x}_N)), \quad K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$$

- 通常令： $\boldsymbol{\mu}(\mathbf{x}) = 0$ —— 因为GP足够灵活，可以很好地建模任意均值
- 因此：高斯过程是函数的分布，其形状由协方差矩阵（核函数）定义

Gaussian Process Regression

- 对于训练集中的样本$\boldsymbol{X}$，假设观测值由一个不含噪声的函数$f(\boldsymbol{X})$生成
- 对于测试集中的样本$\boldsymbol{X}_*$，观测值$\boldsymbol{f}$和预测值$\boldsymbol{f}_*$的联合分布是高斯分布

$$P\begin{pmatrix}\boldsymbol{f}\\ \boldsymbol{f}_*\end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix}\mathbf{K} & \mathbf{K}_*\\ \mathbf{K}_*^T & \mathbf{K}_{**}\end{pmatrix}\right)$$

- 其中：$\mathbf{K}_* = \kappa(\mathbf{X}, \mathbf{X}_*) \quad \mathbf{K}_{**} = \kappa(\mathbf{X}_*, \mathbf{X}_*)$
- 利用高斯条件分布的性质，可以得到预测任务所需的GP后验分布

$$p(\boldsymbol{f}_*|\boldsymbol{X}_*, \boldsymbol{X}, \boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}_*|\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*) \qquad \begin{cases}\boldsymbol{\mu}_* = \mathbf{K}_*^T\mathbf{K}^{-1}\boldsymbol{f}\\ \boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^T\mathbf{K}^{-1}\mathbf{K}_*\end{cases}$$

Gaussian Process Regression

- 如果我们假设训练集是由含噪声的函数生成：$\boldsymbol{y} = f(\boldsymbol{X}) + \boldsymbol{\epsilon}$
- 其中：噪声 $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_y^2 \mathbf{I})$ 被独立地添加到每个观测值中
- 利用高斯条件分布的性质，可以得到预测任务所需的GP后验分布

$$p(\boldsymbol{f}_*|\boldsymbol{X}_*, \boldsymbol{X}, \boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}_*|\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$$

$$\begin{cases} \boldsymbol{\mu}_* = \mathbf{K}_*^T \mathbf{K}_y^{-1} \boldsymbol{y} \\ \boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}_y^{-1} \mathbf{K}_* \end{cases}$$

其中： $\mathbf{K}_y = \mathbf{K} + \sigma_y^2 \mathbf{I}$

$$p(\boldsymbol{f}_*|\boldsymbol{X}_*, \boldsymbol{X}, \boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}_*|\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$$

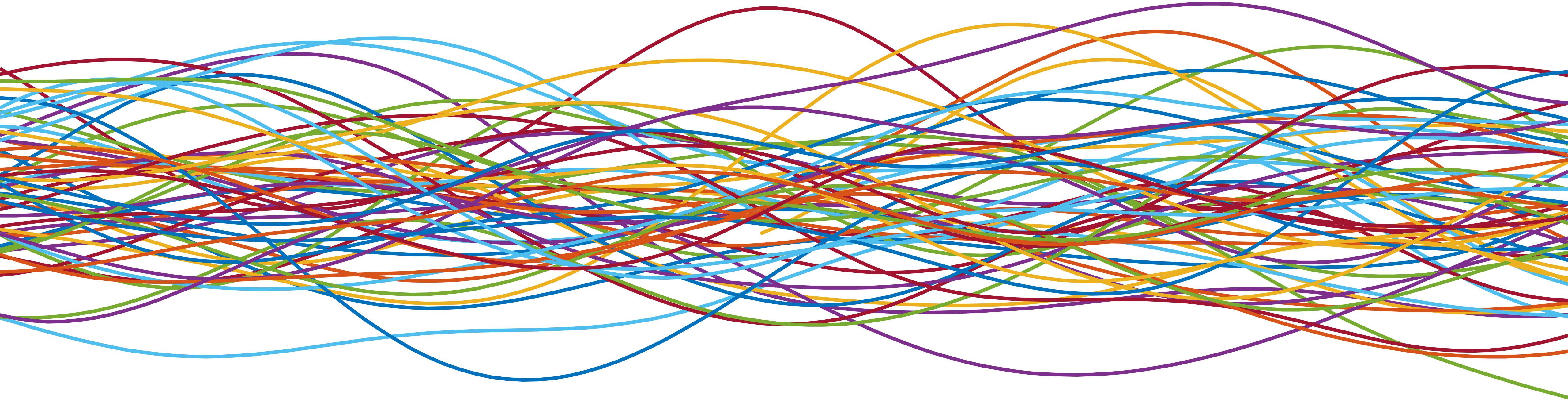
$$\begin{cases} \boldsymbol{\mu}_* = \mathbf{K}_*^T \mathbf{K}^{-1} \boldsymbol{f} \\ \boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{K}_* \end{cases}$$

Gaussian Process Regression

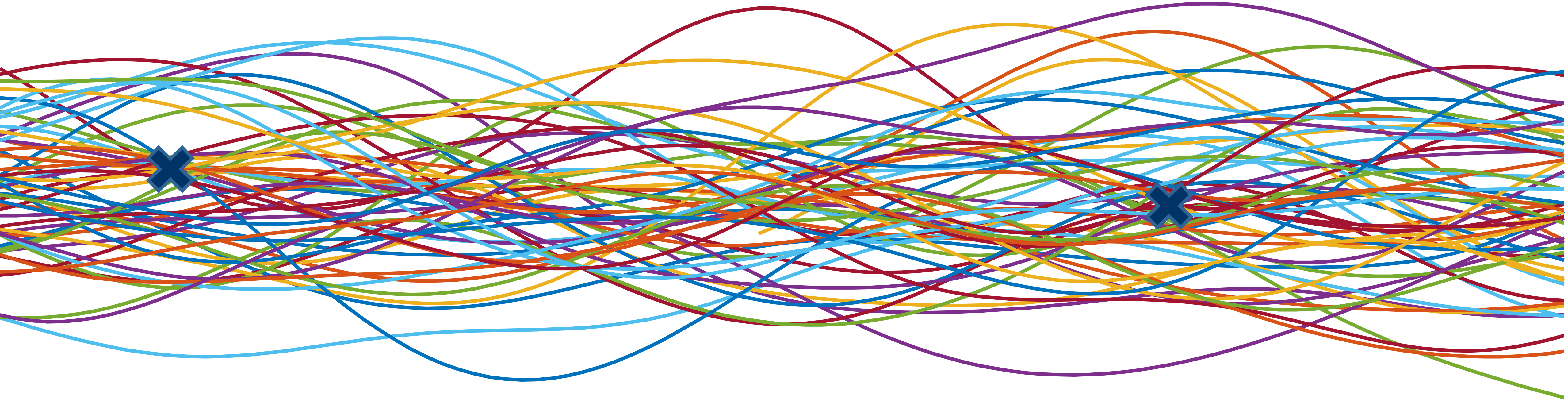
$$p(\boldsymbol{f}_*|\boldsymbol{X}_*, \boldsymbol{X}, \boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}_*|\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*) \quad \begin{cases} \boldsymbol{\mu}_* = \mathbf{K}_*^T \mathbf{K}_y^{-1} \boldsymbol{y} \\ \boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}_y^{-1} \mathbf{K}_* \end{cases} \qquad \mathbf{K}_y = \mathbf{K} + \sigma_y^2 \mathbf{I}$$

- 上述推导过程没有考虑到预测值中包含的噪声
- 将噪声 $\boldsymbol{\epsilon}$ 引入预测值 $\mathbf{y}_*$，可以得到预测任务所需的GP后验分布
- 方法是将 σ_y^2 添加到 $_*$的对角线上：$\boldsymbol{\Sigma}_* = \boldsymbol{\Sigma}_* + \sigma_y^2 \mathbf{I}$
- GP后验分布：$p(\boldsymbol{f}_*|\boldsymbol{X}_*, \boldsymbol{X}, \boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}_*|\boldsymbol{\mu}_*,\ \boldsymbol{\Sigma}_* + \sigma_y^2 \mathbf{I})$

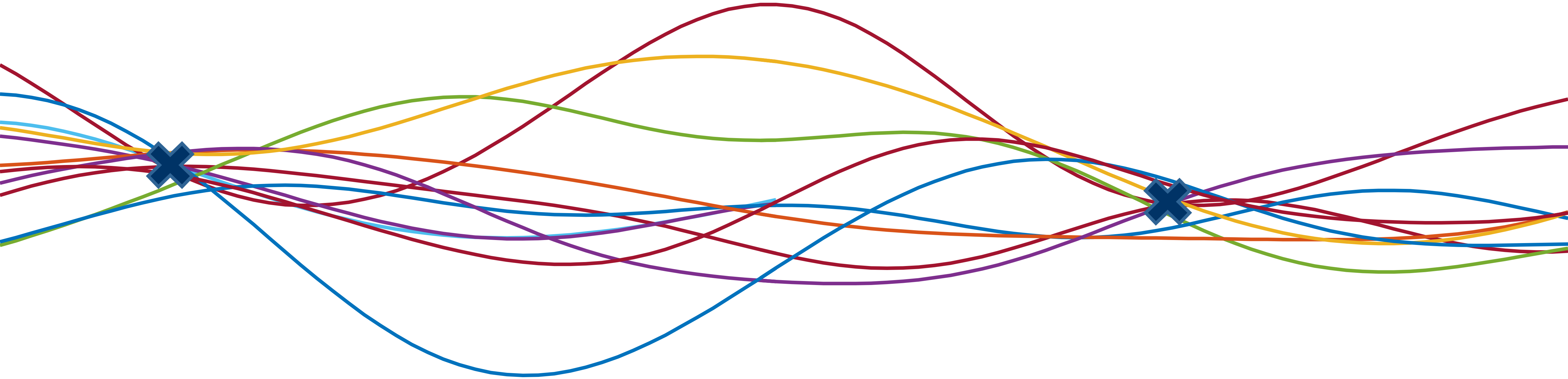
Gaussian Processes



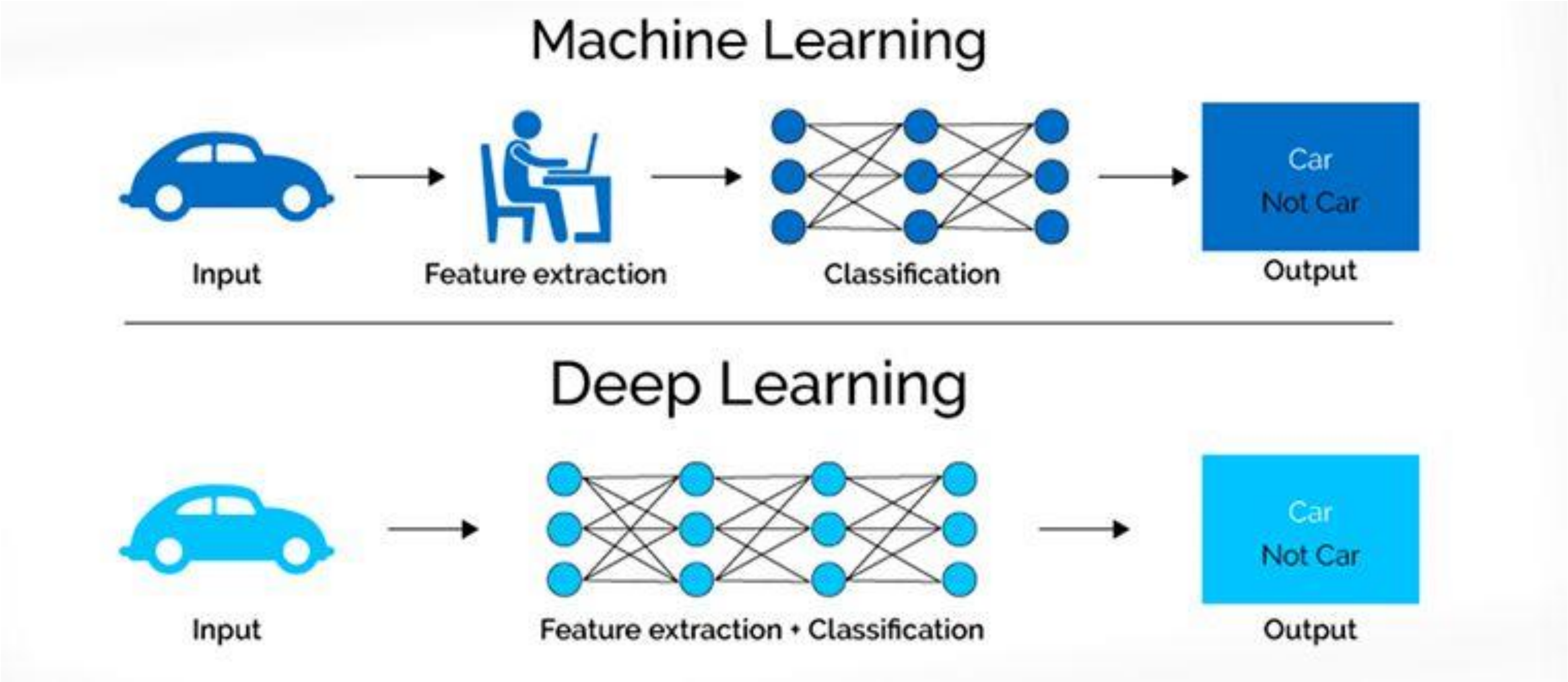
Gaussian Processes



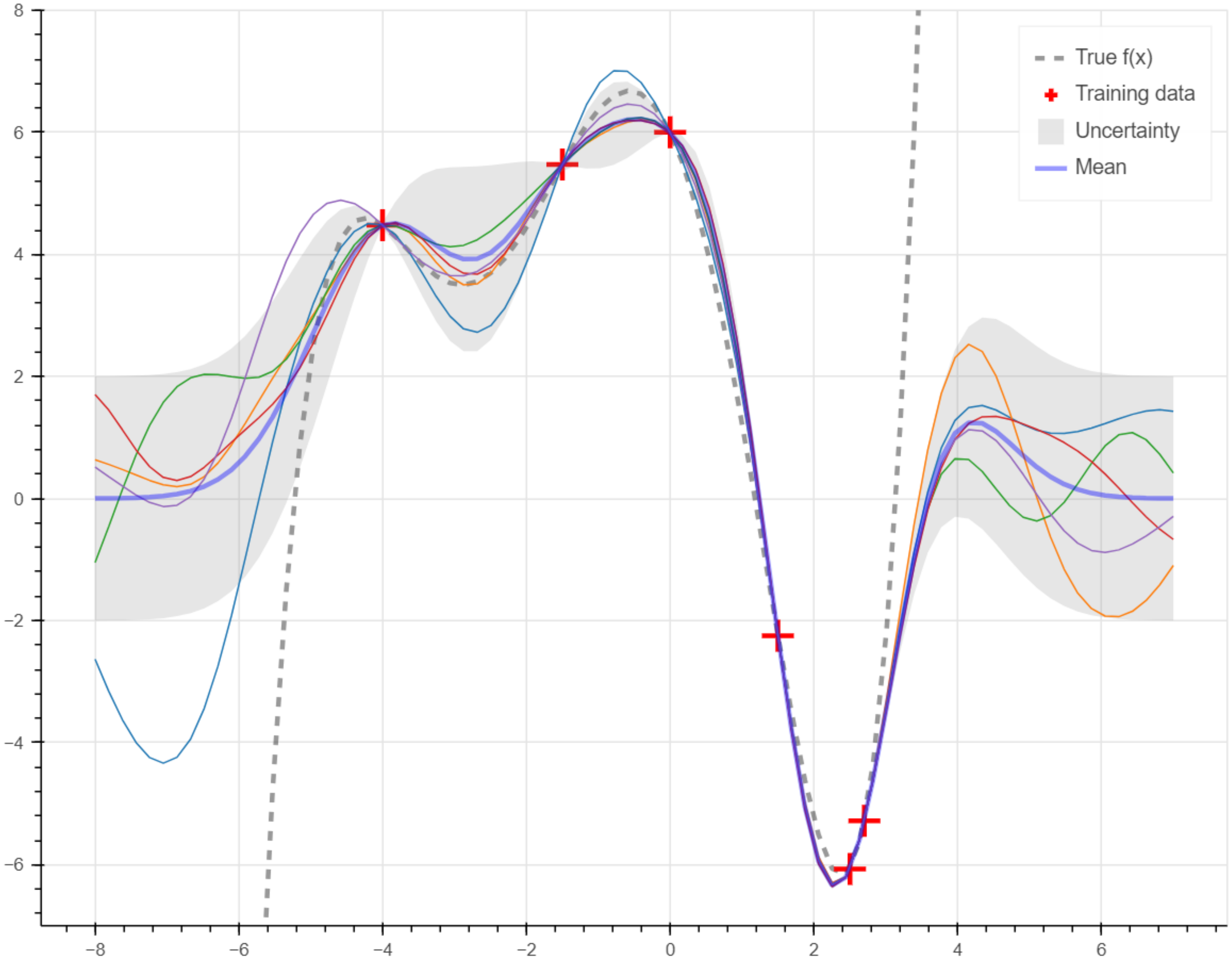
Gaussian Processes



Gaussian Processes



Making Predictions using the Prior & Observations



8.4 GPR in Action

定义高斯过程可视化函数

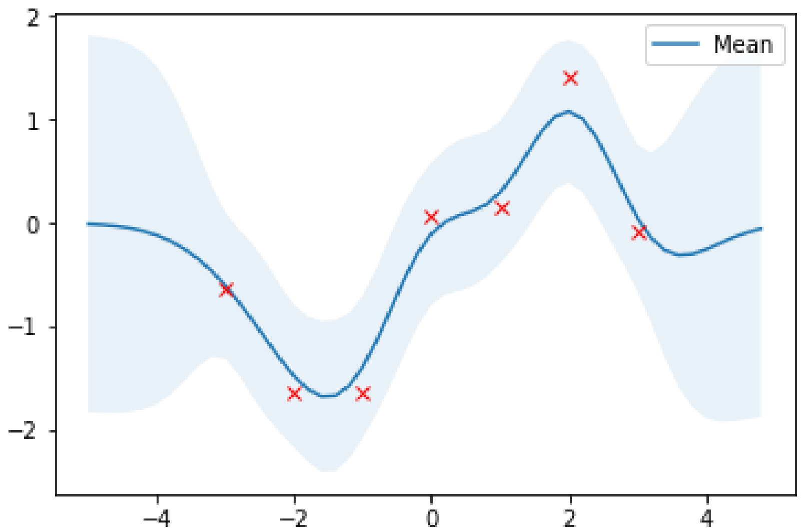
```
import numpy as np
import matplotlib.pyplot as plt

def plot_gp(mu, cov, X, X_train=None, Y_train=None, samples=[]):
    X = X.ravel()          # flatten
    mu = mu.ravel()     # flatten
    std = 1.96 * np.sqrt(np.diag(cov)) # 95%置信区间的标准差是1.96

    # X指定覆盖区域，后面两个参数指定下限和上限，alpha为透明度
    plt.fill_between(X, mu + std, mu - std, alpha=0.1)

    plt.plot(X, mu, label='Mean')  # 绘制均值

    if X_train is not None:
        plt.plot(X_train, Y_train, 'rx')  # 绘制训练样本
    plt.legend()
```



定义高斯核函数：Gaussian kernel

- 高斯核（RBF核）是一个平方指数核函数（squared exponential kernel）

$$\boldsymbol{K}(\mathbf{x}_i,\mathbf{x}_j)=\sigma_f^2\exp\left(-\frac{1}{2l^2}(\mathbf{x}_i-\mathbf{x}_j)^T(\mathbf{x}_i-\mathbf{x}_j)\right)$$

```python
def kernel(X1, X2, l=1.0, sigma=1.0):
    """
    Isotropic squared exponential kernel.
    X1: Array of m points (m x 1).
    X2: Array of n points (n x 1).
    Returns: (m x n) matrix.
    """
    sqdist = np.sum(X1**2, 1).reshape(-1, 1)
             + np.sum(X2**2, 1) - 2 * np.dot(X1, X2.T)
    return sigma**2 * np.exp(-0.5 / l**2 * sqdist)
```

generate synthetic data

```
X = np.arange(-5, 5, 0.2).reshape(-1, 1)

# Mean and covariance of the prior
mu = np.zeros(X.shape)   # (50, 1)
cov = kernel(X, X)

# Noisy training data:
noise = 0.4
X_train = np.arange(-3, 4, 1).reshape(-1, 1)
Y_train = np.sin(X_train) + noise * np.random.randn(*X_train.shape)
```

用scikit-learn实现GP回归模型

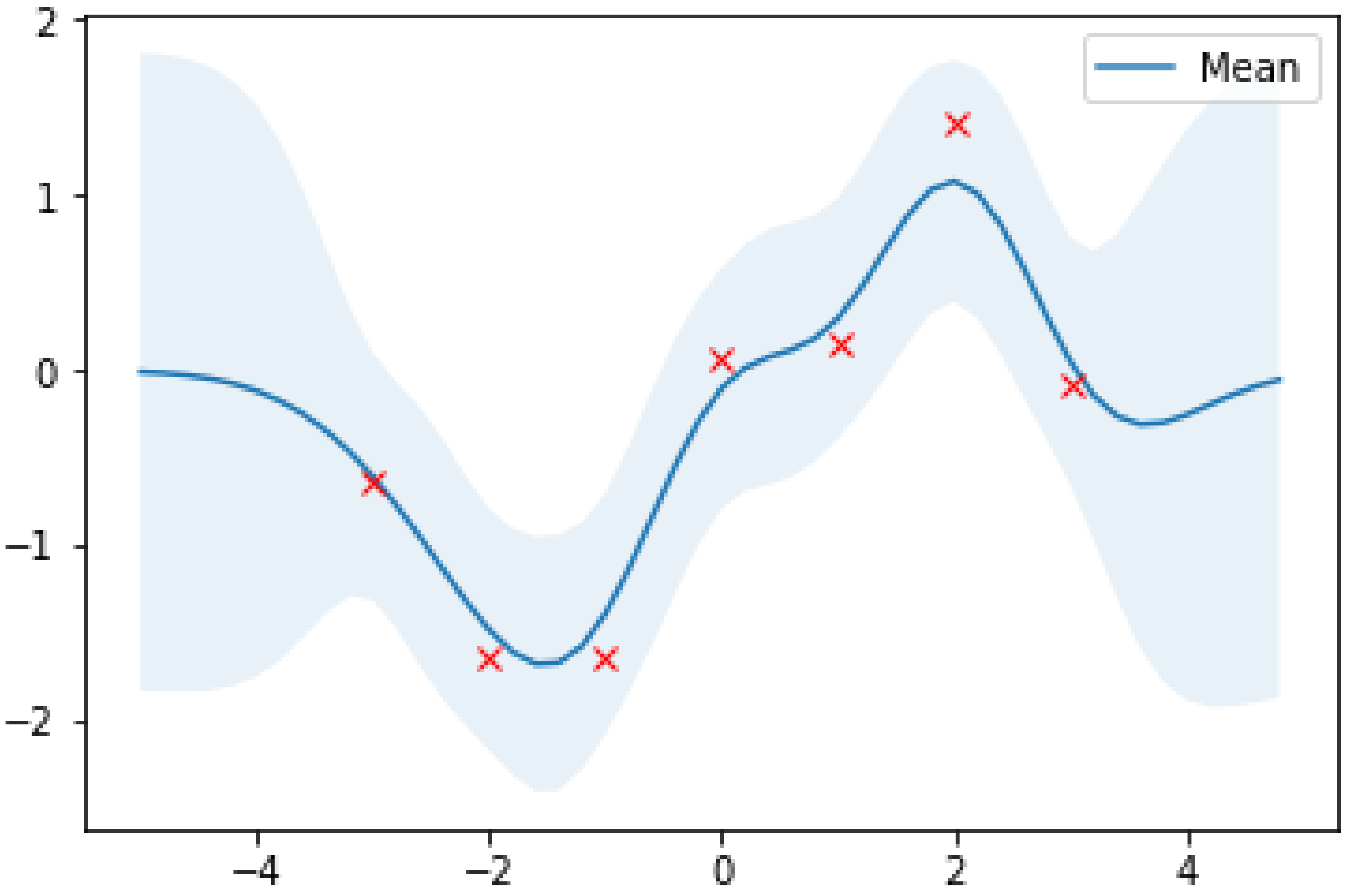
```
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import ConstantKernel, RBF

rbf = ConstantKernel(1.0) * RBF(length_scale=1.0)
gpr = GaussianProcessRegressor(kernel=rbf, alpha=noise**2)
gpr.fit(X_train, Y_train)

# Compute posterior mean and covariance
mu_s, cov_s = gpr.predict(X, return_cov=True)

# Obtain optimized kernel parameters
l = gpr.kernel_.k2.get_params()['length_scale']
sigma_f = np.sqrt(gpr.kernel_.k1.get_params()['constant_value'])
# Plot the results
plot_gp(mu_s, cov_s, X, X_train=X_train, Y_train=Y_train)
```

用scikit-learn实现GP回归模型



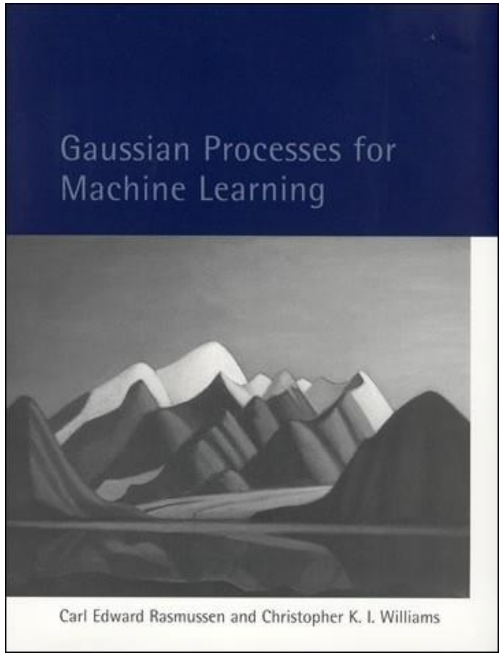
8.5 Brief Summary

Gaussian Process Regression

- Advantages
  - ※ a basic framework for statistical machine learning.
  - ※ it combines kernel machine learning with Bayesian inference learning
- Disadvantages
  - ※ GP model does not work well on sparse samples.
  - ※ The hyperparameters of the GP model, such as the covariance function and the pending parameters in the prior distribution, have a large impact on the learning and prediction results. But there is no clear explanation of how to determine the appropriate initial value.

Gaussian Process Regression

- 使用GPR模型需要考虑如下两个基本问题
  - ※ 选择合适的kernel function
  - ※ 估计kernel中的hyperparameters



- Gaussian Process学习资料
  - ※ 《Gaussian Processes for Machine Learning》
    - ➢ By Carl Edward Rasmussen and Christopher K. I. Williams
  - ※ Gaussian Process Summer School -- held in Sheffield, UK
  - ※ Gaussian Processes - A List of References

课程设计说明

- **推荐选题**
  1. 基于华为昇腾MindSpore框架的广告推荐
  2. 基于华为昇腾CANN的车辆图片分类
     - 前两个实验会发放硬件设备并提供组长培训
     - 背景：华为教育部产学合作协同育人项目（可写入个人简历）
  3. 基于RNN的古诗词（或其他类型文本）写作软件
     - 参考资料：https://github.com/norybaby/poet
  4. 基于CNN的人脸打分软件（加分点：哪里好看）
     - 参考资料：https://github.com/fendouai/FaceRank
- **自拟题目：要求是有创意，有一定挑战性**
- **自由分组：组长负责制，分工明确（答辩要问），共29组**
- **答辩安排：第8周随堂，每组答辩5分钟，回答问题3分钟，组长参与打分**

请选择华为昇腾课程设计题目的同学关注

说明

请选择华为昇腾课程设计题目的同学扫描左侧二维码注册华为**账号**，申请华为针对合作课程发放的**代金券**。

代金券发放额度为500元/人，有效期3个月，包含合作课程的老师、助教、学生

Next chapter: Ensemble Learning