# DS-GA 3001.009
# Modeling Time Series Data
# Lab 9

Artie Shen | Center for Data Science

NYU

# Agenda

- Recap
  - Gaussian Process Regression
  - Cholesky Decomposition
  - Sampling from Multivariate Gaussian

- Programming
  - GP Sampling
  - GP Inference

# GP Basics

**Definition** A Gaussian Process (GP) is a collection of random variables, such that any subset with finite number of elements have Gaussian distributions which can be categorized by a mean function $m(x)$ and a covariance function $K(x, x')$.

- Functions can be viewed as infinitely long vectors $f(x) = [f(t_1), f(t_2), ..., f(t_\infty)]^T, t_i \in \mathbb{R}$.
- GP can be viewed as distribution over functions.
- For a function $f(x)$, in lots of cases, we only care about a subsets of $x \in \mathbb{X}$ (e.g. we have a test set).
- If $f(x) \sim GP(m(x), K(x, x'))$, we know that any finite subset of $f(x)$ have Gaussian distributions.

## Guassian Process Regression

- $y = f(x) + \epsilon\sigma_y, \epsilon \sim N(0, I)$
- $f(x) \sim GP(m(x), K(x, x'))$
- $y(x) \sim GP(m(x), K(x, x') + I\sigma_y^2)$
- $m(x) : \mathbb{R}^{d_x} \mapsto \mathbb{R}^{d_y}$, $K(x, x') : \mathbb{R}^{d_x} \times \mathbb{R}^{d_x} \mapsto \mathbb{R}$
- In the lab, we will assume $\sigma_y = 0$ and $m(x) = 0$.

# GP Inference

**Goal** Given training set $\mathbf{X}_2 \in \mathbb{R}^{n \times d_x}, \mathbf{y}_2 \in \mathbb{R}^{n \times d_y}$, test data $\mathbf{X}_1 \in \mathbb{R}^{m \times d_x}$, and a Gaussian Process Model $GP(m(x), K(x, x'))$, we would like to find $\mathbf{y}_1 \in \mathbb{R}^{m \times d_y}$ that maximize the posterior conditional distribution $p(\mathbf{y}_1|\mathbf{y}_2)$.

$$p(\mathbf{y}_1, \mathbf{y}_2) = \mathcal{N}\left( \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \begin{bmatrix} \mathrm{A} & \mathrm{B} \\ \mathrm{B}^\mathrm{T} & \mathrm{C} \end{bmatrix} \right)$$

$$p(\mathbf{y}_1|\mathbf{y}_2) = \frac{p(\mathbf{y}_1, \mathbf{y}_2)}{p(\mathbf{y}_2)}$$

$$p(\mathbf{y}_2) = \mathcal{N}(\mathbf{b}, \mathrm{C})$$

# GP Inference

$$p(\mathbf{y}_1, \mathbf{y}_2) = \mathcal{N}\left(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}\right)$$

$$p(\mathbf{y}_1 | \mathbf{y}_2) = \frac{p(\mathbf{y}_1, \mathbf{y}_2)}{p(\mathbf{y}_2)}$$

$$p(\mathbf{y}_2) = \mathcal{N}(\mathbf{b}, C)$$

- $\mathbf{a} \in \mathbb{R}^{m \times d_y}$, $\mathbf{b} \in \mathbb{R}^{n \times d_y}$, the prior mean for every single $y$ in $\mathbf{y}_1, \mathbf{y}_2$
- $A \in \mathbb{R}^{m \times m} = K(\mathbf{X_1}, \mathbf{X_1})$
- $B \in \mathbb{R}^{m \times n} = K(\mathbf{X_1}, \mathbf{X_2})$
- $C \in \mathbb{R}^{n \times n} = K(\mathbf{X_2}, \mathbf{X_2})$

# GP Inference

$$p(\mathbf{y}_1|\mathbf{y}_2) = N(\mu_{y_1|y_2}, \Sigma_{y_1|y_2})$$

- $\mu_{y_1|y_2} = a + BC^{-1}(y_2 - b)$
- $\Sigma_{y_1|y_2} = A - BC^{-1}B^T$
- If we further assume $m(x) = 0$, we will have $\mathbf{a}, \mathbf{b} = \mathbf{0}$. Our posterior becomes:

- $\mu_{y_1|y_2} = BC^{-1}y_2$
- $\Sigma_{y_1|y_2} = A - BC^{-1}B^T$

# Cholesky Decomposition

**Motivation** In GP inference, we need to compute $C^{-1}$. However, $C^{-1}$ is not guaranteed to be non-singular. Moreover, naive matrix inversion takes $O(n^3)$. We need a faster and more stable way to compute $\mu_{y_1|y_2}$ and $\Sigma_{y_1|y_2}$ without any naive matrix inversion.

**Algorithm** Cholesky Decomposition convert a Hermitian, positive-definite matrix $A$ into the product of a lower triangular matrix $L$ and its conjugate transpose $L^*$.

- $A = LL^*$
- In our case, $C$ is a covariance matrix, which is positive-definite. Moreover, $C$ is a real matrix that mirror itself along the diagonal $C_{i,j} = C_{j,i}$. Therefore, it's a Hermitian matrix.
- Using Cholesky Decomposition, we have $C = LL^* = L\bar{L}^T$. Since $L$ is a real-value matrix, its conjugate is itself. We will have $C = LL^T$.
- Cholesky Decomposition is usually implemented as a iterative algorithm. It takes $O(kn^2)$ where $k$ is the (small) number of iterations to reach the convergence.

## Use Cholesky Decomposition for GP Inference

- $\mu_{y_1|y_2} = BC^{-1}y_2 = B(LL^T)^{-1}y_2 = BL^{-T}L^{-1}y_2 = (L^{-1}B^T)^T(L^{-1}y2)$
- $\Sigma_{y_1|y_2} = A - BC^{-1}B^T = A - BL^{-T}L^{-1}B^T = A - (L^{-1}B^T)^T(L^{-1}B^T)$
- $A$, $B$, $C$, and $y_2$ are either given or can be computed using $K(x, x')$, $X_1$, and $X_2$.
- $L = cholesky(C)$
- $L^{-1}B^T$ can be obtained by solving a linear system $Lx = B^T$ (np.linalg.solve) which is rather fast.
- The same condition holds for $L^{-1}y_2$.

## Sampling from Multivariate Guassian

- $x \sim N(\mu, \Sigma)$, where $x, \mu \in \mathbb{R}^n$, $\Sigma \in \mathbb{R}^{n \times n}$
- sample $z \in N(0, I), I \in \mathbb{R}^n$
- $L = cholesky(\Sigma)$
- use property of multivatiate Guassian, we have $x = \mu + Lz$
- for GP, we set $\mu = \mu_{y_1|y_2}, \Sigma = \Sigma_{y_1|y_2}$

- **More about Gaussian Process**
  - Kernels
    - The Kernel Cookbook: Advice on Covariance functions by David Duvenaud.
  - Hyper-parameters
    - Cross Validation
    - Maximum Likelihood Estimation (sklearn)

# Programming

- Github:
  - https://github.com/charlieblue17/timeseries2018
- Due Date 04/12/2018 06:45 pm on NYU Classes
- Please rename your submission to **net_id.ipynb**