第7回課題

33B21009　情報科学研究科 情報数理学専攻
羽田充宏

⑦

(a) $\Rightarrow$ (b)

$\exists B \in \mathbb{R}^{n \times n}$　$A = B^T B$ と表せるので

$$\forall x \in \mathbb{R}^n, \quad x^T A x = x^T B^T B x = \|Bx\|^2 \geq 0 \quad \square$$

(b) $\Rightarrow$ (c)

$\forall x \in \mathbb{R}^n$, $x^T A x \geq 0$ より

Aの固有ベクトル u として，対応する固有値 λ とすると，

$$u^T A u = \lambda u^T u = \lambda \|u\|^2 \geq 0$$

従って任意の固有値 $\lambda \geq 0$ である $\square$

(c) $\Rightarrow$ (a)

Aのすべての固有値が非負より

ある直交行列 $P \in \mathbb{R}^{n \times n}$ が存在して

$$P^T A P = \mathrm{diag}(\lambda_1, \dots, \lambda_n) \quad (\lambda_i \text{はAの固有値})$$

$$A = P\, \mathrm{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n})\, \mathrm{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n})\, P^T$$

$B^T = P\, \mathrm{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n})$ とすれば

$A = B^T B$ と表せる. $\square$

③

$A \in \mathbb{R}^{3\times 3}$ の固有値 $\lambda_1, \lambda_2, \lambda_3$ として

Aの固有多項式は

$$\Phi_A(\lambda) = \det(I\lambda - A) = (\lambda-\lambda_1)(\lambda-\lambda_2)(\lambda-\lambda_3)$$

と表せる

よって $\det(A) = (-1)^3 \Phi_A(0)$

$$= -(-\lambda_1)(-\lambda_2)(-\lambda_3)$$

$$= \lambda_1 \lambda_2 \lambda_3$$

また, $\det(A) < 0$ であれば $\lambda_1, \lambda_2, \lambda_3 \neq 0$ で,

少なくとも1つは負の固有値となるので A は非負定値とはならない

□

④

$A, B \succeq O \in \mathbb{R}^{n\times n}$

この時 $\circ$ を Hadamard 積として $A \circ B \succeq O$ を示す.

A, B を固有値分解して

$\lambda_1, \dots, \lambda_n \geq 0$ 　対応する固有ベクトル　$v_1, \dots, v_n \in \mathbb{R}^n$

$\nu_1, \dots, \nu_n \geq 0$ 　　　　　　　　　　　　$u_1, \dots, u_n \in \mathbb{R}^n$ 　として

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^T \qquad B = \sum_{j=1}^{n} \nu_j u_j u_j^T$$ と表せる.

ここで

$$v_i v_i^T \circ u_j u_j^T = \begin{pmatrix} v_{i1}v_{i1} & \cdots & v_{i1}v_{in} \\ \vdots & & \\ v_{in}v_{i1} & \cdots & v_{in}v_{in} \end{pmatrix} \circ \begin{pmatrix} u_{j1}u_{j1} & \cdots & u_{j1}u_{jn} \\ \vdots & & \vdots \\ u_{jn}u_{j1} & \cdots & u_{jn}u_{jn} \end{pmatrix}$$

$$= \begin{pmatrix} v_{i1}u_{j1}v_{i1}u_{j1} & \cdots & v_{i1}u_{j1}v_{in}u_{jn} \\ \vdots & & \vdots \\ v_{in}u_{jn}v_{i1}u_{j1} & \cdots & v_{in}u_{jn}v_{in}u_{jn} \end{pmatrix}$$

$$= (v_i \circ u_j)(v_i \circ u_j)^T$$

従って

$$A \circ B = \sum_i \sum_j \lambda_i \nu_j (v_i \circ u_j)(v_i \circ u_j)^T$$

よって $A \circ B \succeq O$ ◻

また正定値カーネル $k_1, k_2 : E \times E \to \mathbb{R}$ に対して $u, v \in E$

$k_1 k_2 (u, v) := k_1(u, v) k_2(u, v)$ で積を定義すると.

$k_1 k_2$ のグラム行列 $K_{12} \in \mathbb{R}^{n \times n}$ は

k_1 のグラム行列 $K_1 \in \mathbb{R}^{n \times n}$

k_2 のグラム行列 $K_2 \in \mathbb{R}^{n \times n}$ を用いて

$$K_{12} = \begin{bmatrix} k_1(x_1, x_1) k_2(x_1, x_1) & \cdots & k_1(x_1, x_n) k_2(x_1, x_n) \\ \vdots & & \\ k_1(x_n, x_1) k_2(x_n, x_1) & \cdots & k_1(x_n, x_n) k_2(x_n, x_n) \end{bmatrix}$$

$$= K_1 \circ K_2$$

$K_1 \succeq O$, $K_2 \succeq O$ より $K_{12} \succeq O$ ◻

⑤ $\mathbb{R} \ni a > 0$ として $\begin{pmatrix} a & \cdots & a \\ \vdots & & \vdots \\ a & \cdots & a \end{pmatrix} \succeq O \in \mathbb{R}^{n \times n}$ を示す.

$$\begin{pmatrix} a & \cdots & a \\ \vdots & & \vdots \\ a & \cdots & a \end{pmatrix} = \begin{pmatrix} \sqrt{\frac{a}{n}} & \cdots & \sqrt{\frac{a}{n}} \\ \vdots & & \vdots \\ \sqrt{\frac{a}{n}} & \cdots & \sqrt{\frac{a}{n}} \end{pmatrix}^T \begin{pmatrix} \sqrt{\frac{a}{n}} & \cdots & \sqrt{\frac{a}{n}} \\ \vdots & & \vdots \\ \sqrt{\frac{a}{n}} & \cdots & \sqrt{\frac{a}{n}} \end{pmatrix}$$ と書ける.

よって命題が成り立つ

また, $\forall x, y \in E$, $k(x, y) \mapsto a > 0$ となるようなカーネル関数のグラム行列は上での定数行列と一致するため

このようなカーネル関数は正定値カーネルとなる ◻

⑥ 多項式カーネル $k_{3,2}(x,y) = (x^\top y+1)^3$, $x, y \in \mathbb{R}^2$

の特徴量写像

$$\Phi_{3,2}(x_1, x_2) = [\,1,\ \sqrt{3}\,x_1,\ \sqrt{3}\,x_2,\ \sqrt{6}\,x_1x_2,\ \sqrt{3}\,x_1^2,\ \sqrt{3}\,x_2^2,$$
$$x_1^3,\ \sqrt{3}\,x_1^2x_2,\ \sqrt{3}\,x_1x_2^2,\ x_2^3\,]$$

$\langle \Phi_{3,2}(x), \Phi_{3,2}(y) \rangle$

$$= 1 + 3x_1y_1 + 3x_2y_2 + 6x_1x_2y_1y_2$$
$$+ 3x_1^2y_1^2 + 3x_2^2y_2^2 + x_1^3y_1^3$$
$$+ 3x_1^2x_2y_1^2y_2 + 3x_1x_2^2y_1y_2^2$$
$$+ x_2^3y_2^3$$

m_0	$+\,m_1$	$+\,m_2 = 3$
3	0	0
2	1	0
2	0	1
1	1	1
1	2	0
1	0	2
0	3	0
0	2	1
0	1	2
0	0	3

一方

$$(1 + x_1y_1 + x_2y_2)^3$$
$$= (1 + x_1y_1 + x_2y_2)(1 + x_1^2y_1^2 + x_2^2y_2^2 + 2x_1y_1 + 2x_2y_2 + 2x_1y_1x_2y_2)$$
$$= 1 + x_1^2y_1^2 + x_2^2y_2^2 + 2x_1y_1 + 2x_2y_2 + 2x_1y_1x_2y_2$$
$$+ x_1y_1 + x_1^3y_1^3 + x_1y_1x_2^2y_2^2 + 2x_1^2y_1^2 + 2x_1y_1x_2y_2 + 2x_1^2y_1^2x_2y_2$$
$$+ x_2y_2 + x_1^2y_1^2x_2y_2 + x_2^3y_2^3 + 2x_1y_1x_2y_2 + 2x_2^2y_2^2 + 2x_1y_1x_2^2y_2^2$$

$$= 1 + 3x_1^2y_1^2 + 3x_2^2y_2^2 + 3x_1y_1 + 3x_2y_2 + 6x_1y_1x_2y_2$$
$$+ x_1^3y_1^3 + 3x_1y_1x_2^2y_2^2 + 3x_1^2y_1^2x_2y_2 + x_2^3y_2^3$$

従って $\langle \Phi_{3,2}(x), \Phi_{3,2}(y) \rangle = (1+x^T y)^3$ □

⑦ 指数カーネル

$$k_m(x,y) = 1 + \beta x^T y + \frac{\beta^2}{2}(x^T y)^2 + \cdots + \frac{\beta^m}{m!}(x^T y)^m \quad (m \geq 1)$$

を考える.

ここで $(x^T y)^k = \underbrace{(x^T y) \cdots (x^T y)}_{k\text{コ}}$ は,正定値カーネル(内積)の k 回の積で表せるので

これもまた正定値カーネル

正定値カーネルの非負係数の1次結合も正定値カーネルより

k_m は正定値カーネル よって命題4(3) より 指数カーネル $\lim_{m\to\infty} k_m(x,y)$ は正定値カーネル

Gaussカーネル $x, y \in \mathbb{R}^d$,

$$k(x,y) := \exp\left(-\frac{1}{2\sigma^2}\|x-y\|_2^2\right), \quad \sigma > 0$$

$$= \exp\left\{-\frac{1}{2\sigma^2}\left(\|x\|^2 - 2x^T y + \|y\|^2\right)\right\}$$

$$= \exp\left(-\frac{1}{2\sigma^2}\|x\|^2\right)\exp\left(\frac{1}{2\sigma^2}x^T y\right)\exp\left(-\frac{1}{2\sigma^2}\|y\|^2\right)$$

ここで

$$\exp\left(\frac{1}{2\sigma^2}x^T y\right) = \sum_{k=0}^{\infty}\frac{1}{k!}\left(\frac{x^T y}{2\sigma^2}\right)^k$$

よって 命題4 (3) より $k(x,y) = \exp\left(\frac{1}{2\sigma^2}x^T y\right)$ は正定値カーネルなので

命題4 (5) より Gaussカーネルは正定値カーネル.

線形カーネル

$$k_{m,d}(x,y) = (1+x^T y)^m = \underbrace{(1+x^T y)\cdots(1+x^T y)}_{m\text{コ}}$$

正定値カーネルの積より

線形カーネルも正定値カーネル □

正定値カーネル $k : E \times E \to \mathbb{R}$ に対して

$$\tilde{k}(x,y) = \frac{k(x,y)}{\sqrt{k(x,x)k(y,y)}}$$ で定める $\tilde{k}$ も

$$= \frac{1}{\sqrt{k(x,x)}} k(x,y) \frac{1}{\sqrt{k(y,y)}}$$

$$= f(x) \quad k(x,y) \quad f(y)$$ の形で表せるので

正定値カーネル

Gauss カーネル正規化

$$k(x,x) = \exp\left(-\frac{1}{2\sigma^2}\|x-x\|^2\right) = 1$$ より

正規化したカーネル $\tilde{k}$ として

$\tilde{k}(x,y) = k(x,y)$ で変化しない

指数カーネル正規化

$$\tilde{k}(x,y) = \frac{\exp(\beta x^T y)}{\sqrt{\exp(\beta x^T x)\exp(\beta y^T y)}}$$

$$= \exp\left\{-\frac{\beta}{2}(x^T x - 2x^T y + y^T y)\right\}$$

$$= \exp\left\{-\frac{\beta}{2}\|x-y\|^2\right\}$$

よって指数カーネルを正規化するとGauss カーネルとなる.

⑨

$(E, \mathcal{F}, \mu)$ 確率sp. $E = \{1,2,3,4,5,6\}$

$X : E \to \mathbb{R}$

$$X(e) = \begin{cases} 1 & (e = 1,3,5) \\ 0 & (\text{others}) \end{cases}$$

$\mathcal{F} = \{\{1,2,3\}, \{4,5,6\}, \phi, E\}$ の時.

$X^{-1}(\{1\}) = \{1,3,5\} \notin \mathcal{F}$ より

X は $\mathcal{F}$-可測でないので確率変数でない □

(10) 正規分布の特性関数は

$$\varphi(t) = E[\exp(itX)]$$
$$= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2} - itx\right) dx$$

肩の部分について

$$-\frac{(x-\mu)^2}{2\sigma^2} - \frac{2\sigma^2 itx}{2\sigma^2} = -\frac{x^2 - 2\mu x + 2\sigma^2 itx + \mu^2}{2\sigma^2}$$
$$= -\frac{x^2 - 2(\mu+\sigma^2 it)x + \mu^2}{2\sigma^2}$$
$$= -\frac{(x-\mu+\sigma^2 it)^2 - (\mu+\sigma^2 it)^2 + \mu^2}{2\sigma^2}$$
$$= -\frac{(x-\mu+\sigma it)^2 - 2\mu\sigma^2 it + \sigma^4 t^2}{2\sigma^2}$$
$$= \frac{(x-\mu+\sigma it)^2}{2\sigma^2} + \mu it - \frac{\sigma^2 t^2}{2}$$

よって $\varphi(t) = \exp\left(-\frac{\sigma^2 t^2}{2} + \mu it\right)$

$\mu = 0$ の時 $\forall t \in \mathbb{R}$, $\varphi(t)$ は実数値をとる

ラプラス分布の特性関数は.

$$\varphi(t) = \int_{-\infty}^{\infty} \frac{\alpha}{2} \exp(-\alpha|x| + itx)\, dx$$
$$= \int_{-\infty}^{0} \frac{\alpha}{2} \exp((it+\alpha)x)\, dx + \int_{0}^{\infty} \frac{\alpha}{2} \exp((it-\alpha)x)\, dx$$
$$= \frac{\alpha}{2}\left\{\left[\frac{e^{(it+\alpha)x}}{it+\alpha}\right]_{-\infty}^{0} + \left[\frac{e^{(it-\alpha)x}}{it-\alpha}\right]_{0}^{\infty}\right\}$$
$$= \frac{\alpha}{2}\left(\frac{1}{it+\alpha} - \frac{1}{it-\alpha}\right) = \frac{\alpha}{2}\,\frac{it-\alpha-(it+\alpha)}{-t^2-\alpha^2} = \frac{\alpha^2}{\alpha^2+t^2}$$

$\forall \alpha > 0$, $\varphi(t)$ は実数値をとる.

□

⑬

文字列カーネル

E_1, E_3：有限長の文字列の集合

E_2：長さ p の文字列の集合　　とする.

$$R : E_1 \times E_2 \times E_3 \to E$$
$$(x_1, x_2, x_3) \longmapsto \underset{\sim}{x}$$

x_1, x_2, x_3 を連結した文字列

として写像 R を定義する.

この時　$k_1 : E_1 \times E_1 \to \mathbb{R}$

$k_2 : E_2 \times E_2 \to \mathbb{R}$

$k_3 : E_3 \times E_3 \to \mathbb{R}$　として

$$k_1(x_1, y_1) = 1$$
$$k_3(x_3, y_3) = 1$$
$$k_2(x_2, y_2) = \begin{cases} 1 & (\text{if } x_2 = y_2) \\ 0 & (\text{others}) \end{cases}$$

k_1, k_2, k_3 はそれぞれ正定値カーネルであり,

文字列カーネルは

$x, y \in E$

$$k(x, y) = \sum_{R(x_1, x_2, x_3) = x} \sum_{R(y_1, y_2, y_3) = y} 1 \cdot k_2(x_2, y_2) \cdot 1$$

以上より文字列カーネルはたたみ込みカーネルであり,

正定値カーネルの和で書けているので正定値カーネルである.

木カーネル

木 x, y に対して

$$k(x,y) := \sum_t C_t(x) C_t(y)$$

ここで $C_t(x)$：木 x における部分木 t の頻度

t をとり出す集合としては x と y の共通の木を考えれば十分である。

ここで

$$I_t(a) = \begin{cases} 1 & t \text{ が } a \text{ をルートにもつ部分木} \\ 0 & \text{others} \end{cases}$$

によって $I_t(\cdot)$ を定義すると、

$$C_t(x) = \sum_{a \in V_x} I_t(a)$$ と表せるので

$$k(x,y) = \sum_t \sum_{a \in V_x} \sum_{b \in V_y} I_t(a) I_t(b)$$

$$\sum_i \sum_j z_i z_j k(x_i, x_j) = \sum_t \sum_i \sum_j \sum_{a \in x_i} \sum_{b \in x_j} z_i z_j I_t(a) I_t(b)$$

$$= \sum_t \sum_i \sum_j z_i z_j C_t(x_i) C_t(x_j)$$

$$= \sum_t \left\{ \sum_i z_i C_t(x_i) \right\}^2 \geq 0$$

よって木カーネルは正定値カーネル。

周辺化カーネル

確率変数 X, Y についてそのとりうる値の集合 E_X, E_Y で、
$P(y|x)$ $(x \in E_X, y \in E_Y)$ で条件付き確率が与えられ、
$k_{XY} : (E_X \times E_Y) \times (E_X \times E_Y) \to \mathbb{R}$ が与えられている時、

$$k(x, x') := \sum_{y \in E_Y} \sum_{y' \in E_Y} k_{XY}((x,y),(x',y')) P(y|x) P(y'|x')$$

$x, x' \in E_X$

この時　　　　　　　　　　　　　　　　を周辺化カーネルとよぶ

k_{XY} が正定値カーネルとすれば

$\exists \Phi : E_X \times E_Y \to \mathcal{H}$,

$k_{XY}((x,y),(x',y')) = \langle \Phi(x,y), \Phi(x',y') \rangle$ と表せる.

$$k(x,x') = \sum_{y \in E_Y} \sum_{y' \in E_Y} P(y|x) P(y'|x') \langle \Phi(x,y), \Phi(x',y') \rangle$$

$$= \left\langle \sum_{y \in E_Y} P(y|x) \Phi(x,y), \sum_{y' \in E_Y} P(y'|x') \Phi(x',y') \right\rangle$$

新たに $\Phi' : E_X \to \mathcal{H}$

$x \mapsto \Phi'(x) = \sum_{y \in E_Y} P(y|x) \Phi(x,y)$ と定めれば

$k(x,x') = \langle \Phi'(x), \Phi'(x') \rangle_{\mathcal{H}}$ と書くことができ,

$$\sum_i \sum_j z_i z_j \langle \Phi'(x_i), \Phi'(x_j) \rangle_{\mathcal{H}} = \left\langle \sum_i z_i \Phi'(x_i), \sum_j z_j \Phi'(x_j) \right\rangle_{\mathcal{H}}$$

$$= \left\| \sum_i z_i \Phi'(x_i) \right\|_{\mathcal{H}}^2 \geq 0$$

で周辺化カーネルは正定値カーネル　□

グラフカーネル

グラフ G_1, G_2 に対して以下でカーネルを定める.

$$k(G_1, G_2) := \sum_{\pi_1} \sum_{\pi_2} P(\pi_1 | G_1) P(\pi_2 | G_2) I(L(\pi_1) = L(\pi_2))$$

上の周辺化カーネルの定ぎで

$$k_{XY}((G_1,\pi_1),(G_2,\pi_2)) = I[L(\pi_1) = L(\pi_2)]$$

とすれば

グラフカーネルは周辺化カーネル

□

(14)

(a) $3 \to 1 \to 4 \to 3 \to 5$

$$\frac{1}{5}\left(\frac{\cancel{2}}{3}\cdot\frac{1}{\cancel{2}}\right)\left(\frac{\cancel{2}}{3}\cdot\frac{1}{\cancel{2}}\right)\left(\frac{2}{3}\cdot 1\right)\left(\frac{\cancel{2}}{3}\cdot\frac{1}{\cancel{2}}\right)\cdot\frac{1}{3} = \frac{1}{5}\cdot\frac{1}{3}\cdot\frac{1}{3}\cdot\frac{2}{3}\cdot\frac{1}{3}\cdot\frac{1}{3}$$

$$= \frac{2}{5\cdot 3^5}$$

(b) $1 \to 2 \to 4 \to 1 \to 2$

$$\frac{1}{5}\left(\frac{2}{3}\cdot\frac{1}{2}\right)\left(\frac{2}{3}\cdot 1\right)\left(\frac{2}{3}\cdot 0\right)\left(\frac{2}{3}\cdot\frac{1}{2}\right)\cdot\frac{1}{3} = 0$$

(c) $3 \to 5 \to 3 \to 5$

$$\frac{1}{5}\left(\frac{\cancel{2}}{3}\cdot\frac{1}{\cancel{2}}\right)\left(\frac{2}{3}\cdot 1\right)\left(\frac{\cancel{2}}{3}\cdot\frac{1}{\cancel{2}}\right)\frac{1}{3} = \frac{1}{5}\cdot\frac{1}{3}\cdot\frac{2}{3}\cdot\frac{1}{3}\cdot\frac{1}{3}$$

$$= \frac{2}{5\cdot 3^4}$$

Ch1

In [1]:
```
# 第 1 章のプログラムは,事前に下記が実行されていることを仮定する。
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import style
style.use("seaborn-ticks")
```

1

In [2]:
```
B = np.random.randn(5,5)
A = B.T@B
```

In [3]:
```
for i in range(5):
    x = np.random.randn(5).reshape(-1,)
    print(x.T@A@x)
```

```
6.44550819001953
2.8611258593266715
26.50609897398167
39.920400454824595
10.068269950769912
```

2次形式の全ての値は非負となっている

2

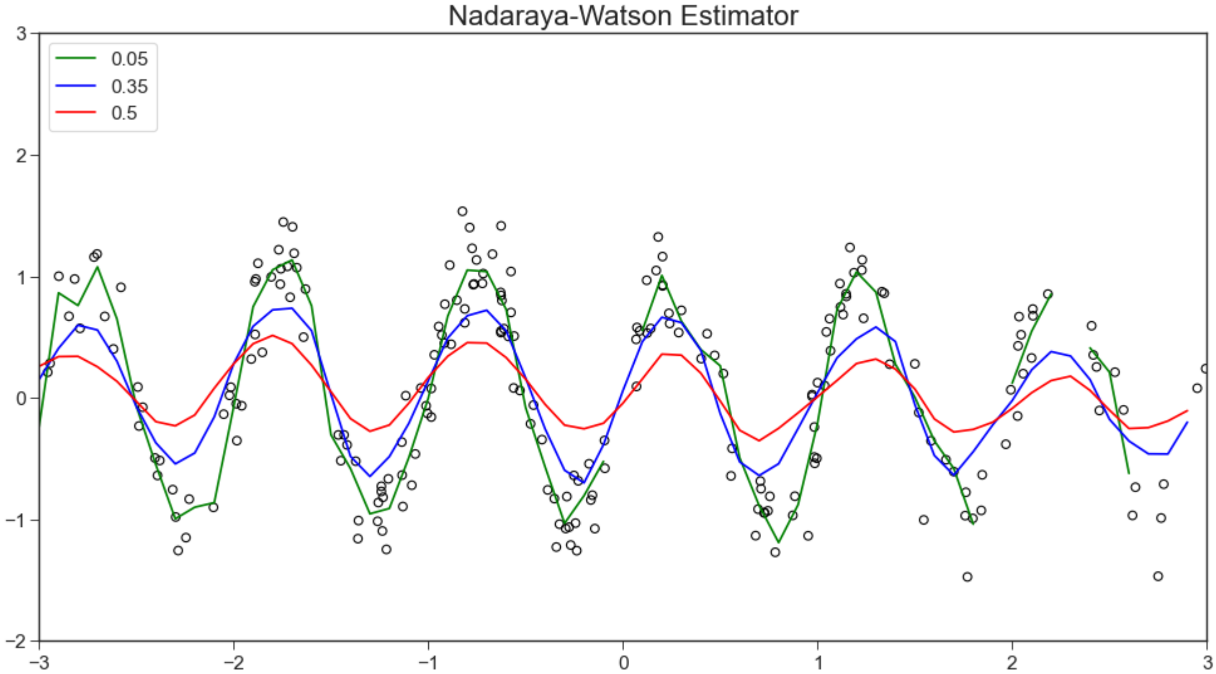
In [4]:
```
def k(x, y, lam):
    return D(np.abs((x - y) / lam))
```

In [5]:
```
n = 250
x = 2 * np.random.normal(size=n)
y = np.sin(2 * np.pi * x) + np.random.normal(size=n) / 4  # データ生成


def D(t):         # 関数定義 D
    return np.maximum(0.75 * (1 - t**2), 0)


def k(x, y, lam):  # 関数定義 K
    return D(np.abs((x - y) / lam))


def f(z, lam):     # 関数定義 f
    S = 0
    T = 0
    for i in range(n):
        S = S + k(x[i], z, lam) * y[i]
        T = T + k(x[i], z, lam)
    return S / T

plt.figure(num=1, figsize=(15, 8), dpi=80)
plt.xlim(-3, 3)
plt.ylim(-2, 3)
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)
plt.scatter(x, y, facecolors="none", edgecolors="k", marker="o")
```

```
xx = np.arange(-3, 3, 0.1)
yy = [[] for _ in range(3)]
lam = [0.05, 0.35, 0.50]
color = ["g", "b", "r"]
for i in range(3):
    for zz in xx:
        yy[i].append(f(zz, lam[i]))
    plt.plot(xx, yy[i], c=color[i], label=lam[i])

plt.legend(loc="upper left", frameon=True, prop={"size": 14})
plt.title("Nadaraya-Watson Estimator", fontsize=20)
```

```
<ipython-input-5-07b6850729b4>:20: RuntimeWarning: invalid value encountered in double_sca
lars
  return S / T
```

Out[5]: Text(0.5, 1.0, 'Nadaraya-Watson Estimator')



8

In [6]:

```
def K(x, y, sigma2):
    return np.exp(-np.linalg.norm(x - y)**2 / 2 / sigma2)


def F(z, sigma2):  # 関数定義 f
    S = 0
    T = 0
    for i in range(n):
        S = S + K(x[i], z, sigma2) * y[i]
        T = T + K(x[i], z, sigma2)
    return S / T
```

In [7]:

```
n = 100
x = 2 * np.random.normal(size=n)
y = np.sin(2 * np.pi * x) + np.random.normal(size=n) / 4  # データ生成

# 最適な lambda の値の計算
m = int(n / 10)
sigma2_seq = np.arange(0.001, 0.01, 0.001)
SS_min = np.inf
for sigma2 in sigma2_seq:
```

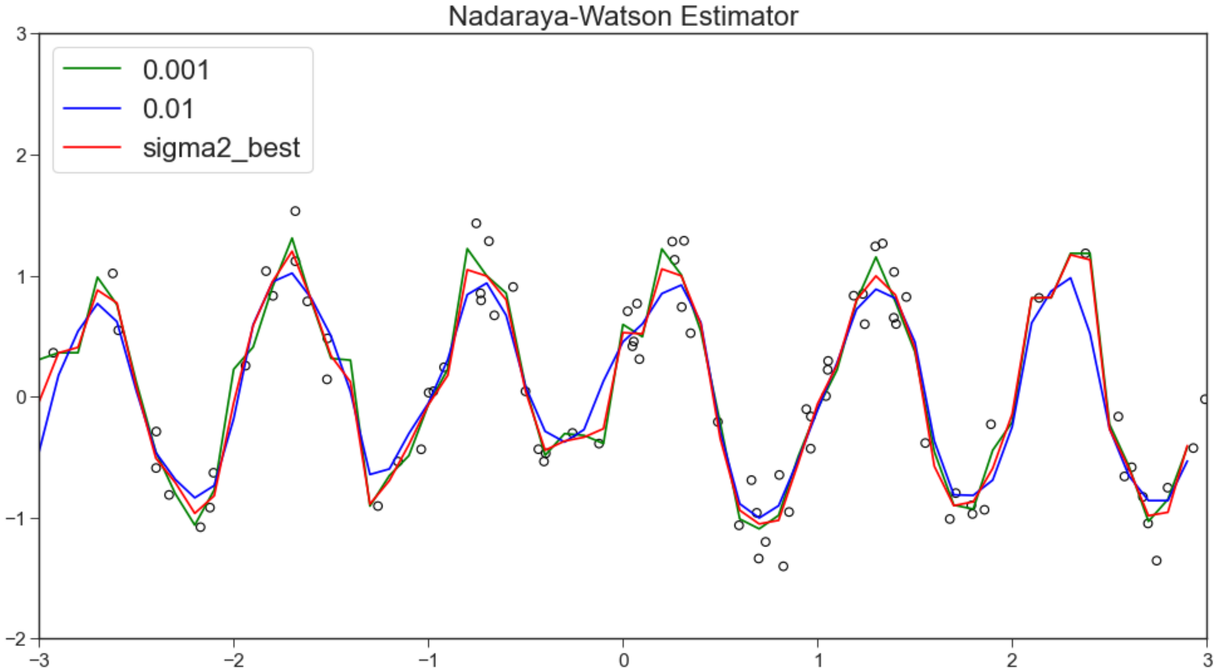
```
    SS = 0
    for k in range(10):
        test = range(k*m, (k+1)*m)
        train = [x for x in range(n) if x not in test]
        for j in test:
            u, v = 0, 0
            for i in train:
                kk = K(x[i], x[j], sigma2)
                u = u + kk * y[i]
                v = v + kk
            if v != 0:
                z = u / v
                SS = SS + (y[j] - z)**2
    if SS < SS_min:
        SS_min = SS
        sigma2_best = sigma2
print("Best sigma2 =", sigma2_best)
```

Best sigma2 = 0.003

In [8]:

```
plt.figure(num=1, figsize=(15, 8), dpi=80)
plt.scatter(x, y, facecolors="none", edgecolors="k", marker="o")
plt.xlim(-3, 3)
plt.ylim(-2, 3)
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)

xx = np.arange(-3, 3, 0.1)
yy = [[] for _ in range(3)]
sigma2 = [0.001, 0.01, sigma2_best]
labels = [0.001, 0.01, "sigma2_best"]
color = ["g", "b", "r"]

for i in range(3):
    for zz in xx:
        yy[i].append(F(zz, sigma2[i]))
    plt.plot(xx, yy[i], c=color[i], label=labels[i])
plt.legend(loc="upper left", frameon=True, prop={"size": 20})
plt.title("Nadaraya-Watson Estimator", fontsize=20)
```

Out[8]: Text(0.5, 1.0, 'Nadaraya-Watson Estimator')



12

In [9]:
```
def string_kernel(x, y):
    m, n = len(x), len(y)
    S = 0
    for i in range(m):
        for j in range(i, m):
            for k in range(n):
                if x[(i-1):j] == y[(k-1):(k+j-i)]:
                    S = S + 1
    return S
```

In [10]:
```
C = ["a", "b", "c"]
m = 10
w = np.random.choice(C, m, replace=True)
x = ""
for i in range(m):
    x = x + w[i]
n = 12
w = np.random.choice(C, n, replace=True)
y = ""
for i in range(n):
    y = y + w[i]
```

In [11]:
```
x,y
```

Out[11]: ('baacccbbac', 'babbaccacbba')

In [12]:
```
string_kernel(x,y)
```

Out[12]: 61

15

In [13]:
```
def k(s, p):
    return prob(s, p) / len(node)


def prob(s, p):
    if len(node[s[0]]) == 0:
        return 0
    if len(s) == 1:
        return p
    m = len(s)
    S = (1 - p) / len(node[s[0]]) * prob(s[1:m], p)
    return S
```

In [14]:
```
node = [[] for _ in range(5)]
node[0] = [1, 3]
node[1] = [3]
node[2] = [0, 4]
node[3] = [2]
node[4] = [2]
k([0, 3, 2, 4, 2], 1 / 3)
```

Out[14]: 0.0032921810699588485

In [15]:
```
2**2 / (5*3**5)
```

Out[15]: 0.0032921810699588477

この時，パスとして存在し得ないものを関数kの引数としても値を返してしまう
例として以下のようなあり得ないウォークを指定しても
1 -> 3 -> 5

In [16]:
```
k([0,2,4], 1/3)
```

Out[16]:
```
0.007407407407407408 6
```

非ゼロの値を返してしまう.
解決策としては隣接リストnodeを参照して次のノードへの遷移がありうるものかどうかの判定を行う必要がある.

In [ ]:
```
```