Ridge正則化とStein推定量

黒木玄

2019-09-30

- ipynb版 (https://nbviewer.jupyter.org/github/genkuroki/Statistics/blob/master/Ridge%20regularization%20and%20Stein%20estimator.ipynb
- pdf版 (https://genkuroki.github.io/documents/Statistics/Ridge%20regularization%20and%20Stein%20estimator.pdf)

Ridge正則化とはパラメーター w の対数尤度函数の -1 倍 $L(w)$ そのものを最小化する最尤法を実行するのではなく, パラメーターの ℓ^2 ノルムの2乗に比例する罰則項 $\lambda\|w\|^2$ を加えた $L(w)+\lambda\|w\|^2$ の最小化によってパラメーターの推定値を決定する方法である.

Ridge正則化を非常にシンプルな場合に適用することによって, 最尤推定量よりも平均二乗誤差が小さいStein推定量が得られることを示す.

目次

1 設定

以下では, 平均 $a \in \mathbb{R}$, 分散 1 の正規分布の確率密度函数を

$$N(x|\theta) = \frac{1}{\sqrt{2\pi}} e^{-(x-\theta)^2/2}$$

と書くことにし, $\mu_{i0} \in \mathbb{R}$ $(i = 1, \dots, n)$ を任意に取って固定する. $X = (X_1, \dots, X_n)$ は確率密度函数

$$q(x) = \prod_{i=1}^{n} N(x|\mu_{i0})$$

で定義される確率分布に従うベクトル値確率変数であるとする. (これは, X_i は平均 μ_{i0}, 分散 1 の正規分布に従う確率変数であり, X_i 達は独立であると仮定したのと同じことである.)

以下 X をサンプルと呼ぶ.

このサンプルが従う分布の推定を, パラメーター $\mu = (\mu_1, \dots, \mu_n)$ を持つ $x = (x_1, \dots, x_n)$ に関する確率密度函数

$$p(x|\mu) = \prod_{i=1}^{n} N(x_i|\mu_i)$$

をモデルとして採用して行いたい. 以上が基本的な設定である.

1.1 平均汎化誤差と平均二乗誤差の関係

このとき $p(x|\mu)$ による $q(x)$ の予測の汎化誤差の2倍は

$$\int_{\mathbb{R}^n} q(x)(-2\log p(x|\mu))\,dx = \int_{\mathbb{R}^2} q(x)\left(n\log(2\pi) + \sum_{i=1}^{n}(x_i-\mu_i)^2\right)dx$$
$$= \int_{\mathbb{R}^2} q(x)\left(n\log(2\pi) + \sum_{i=1}^{n}((x_i-\mu_{i0})-(\mu_i-\mu_{i0}))^2\right)dx$$
$$= \int_{\mathbb{R}^2} q(x)\left(n\log(2\pi) + \sum_{i=1}^{n}((x_i-\mu_{i0})^2 - 2(\mu_i-\mu_{i0})(x_i-\mu_{i0}) + (\mu_i-\mu_{i0})^2)\right)dx$$
$$= n\log(2\pi) + n + \sum_{i=1}^{n}(\mu_i-\mu_{i0})^2.$$

ゆえに, もしも μ_i が $X=(X_1,\ldots,X_n)$ の函数 $\mu_i(X)$ ならば, サンプル X を動かす汎化誤差の平均の2倍は
$\mu(X)=(\mu_1(X),\ldots,\mu_n(X))$ の二乗誤差の平均と定数の和

$$n\log(2\pi) + n + E\left[\sum_{i=1}^{n}(\mu_i(X)-\mu_{i0})^2\right]$$

になる. ゆえに, 平均汎化誤差を小さくすること(平均予測誤差を小さくすること)と, 平均二乗誤差を小さくすることは同じことになる.

平均二乗誤差の小さな推定量の方が平均予測誤差も小さくなり, より優れた推定量だということになる.

2 最尤推定量

まず, 最尤法を実行してみよう. 尤度函数の対数の -2 倍は

$$-2\log p(X|\mu) = -2\sum_{i=1}^{n}\log N(X_i|\mu_i) = n\log(2\pi) + \sum_{i=1}^{n}(X_i-\mu_i)^2$$

となるので, これの最小化は損失函数

$$L(\mu) = \sum_{i=1}^{n}(X_i-\mu_i)^2$$

の最小化と同じになる. これを最小化する μ_i 達は $\hat{\mu}_i = X_i$ となる. これが最尤法の解である. 最尤法の解の平均二乗誤差は, X_i が平均 μ_{i0}, 分散 1 の正規分布に従う確率変数なので,

$$E\left[\sum_{i=1}^{n}(\hat{\mu}_i-\mu_{i0})^2\right] = \sum_{i=1}^{n}E\left[(X_i-\mu_{i0})^2\right] = n$$

となる.

3 Ridge正則化とStein推定量

以下では $n \geqq 3$ であると仮定する.

Ridge正則化された損失函数 $R(\mu|\lambda)$ を

$$R(\mu|\lambda) = L(\mu) + \lambda\|\mu\|^2 = \sum_{i=1}^{n}(X_i-\mu_i)^2 + \lambda\sum_{i=1}^{n}\mu_i^2$$

と定める. $\lambda > 0$ には後でサンプル X_i から決まるある正の実数を代入することになる.

$R(\mu|\lambda)$ を最小化する $\mu_i = \tilde{\mu}_i$ は, 簡単な計算で

$$\tilde{\mu}_i = \frac{1}{1+\lambda}X_i = (1-\alpha)X_i, \quad \alpha = \frac{\lambda}{1+\lambda}$$

と書けることがわかる.

定数 c を用いて,

$$\alpha = \alpha(X) = \frac{c}{X^2}, \quad X^2 = \sum_{i=1}^{n} X_i^2$$

とおき, 推定量

$$\tilde{\mu}_i = (1 - \alpha(X))\, X_i = \left(1 - \frac{c}{X^2}\right) X_i$$

の平均二乗誤差

$$E\left[\sum_{i=1}^{n} (\tilde{\mu}_i - \mu_{i0})^2\right] = \sum_{i=1}^{n} E[(X_i - \mu_{i0})^2] - 2\sum_{i=1}^{n} E[\alpha(X) X_i (X_i - \mu_{i0})] + \sum_{i=1}^{n} E[\alpha(X)^2 X_i^2]$$

を最小化する c を求めよう.

3.1 $n \geqq 3$ という仮定からの帰結

$n \geqq 3$ と仮定したことより, $E[1/X^2]$ が有限の値になることを示そう.

$$E\left[\frac{1}{X^2}\right] = \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} e^{-(x-\mu_0)^2/2} \frac{1}{x^2}\, dx.$$

ここで $(x - \mu_0)^2 = \sum_{i=1}^{n} (x_i - \mu_{i0})^2$, $x^2 = \sum_{i=1}^{n} x_i^2$ である.

原点を中心とする $n-1$ 次元単位球面を S^{n-1} と書き, その面積を $A_n = 2\pi^{n/2}/\Gamma(n/2)$ と書き, S^{n-1} 上の一様分布を $d\omega$ と書くことにする. このとき, 極座標変換

$$x = r\omega, \quad (r > 0,\ \omega \in S^{n-1})$$

によって,

$$E\left[\frac{1}{X^2}\right] = \frac{1}{(2\pi)^{n/2}} \iint_{\mathbb{R}^n} e^{-(r\omega-\mu_0)^2/2} \frac{1}{r^2} A_n r^{n-1}\, dr\, d\omega = \frac{A_n}{(2\pi)^{n/2}} \iint_{\mathbb{R}^n} e^{-(r\omega-\mu_0)^2/2}\, r^{n-3}\, dr\, d\omega.$$

これは $n - 3 > -1$ すなわち $n > 2$ ならば絶対収束している.

3.2 第1項

X_i は平均 μ_{i0}, 分散 1 の正規分布に従うので

$$\sum_{i=1}^{n} E[(X_i - \mu_{i0})^2] = n.$$

3.3 第2項

正規分布に関する部分積分の公式

$$E[\alpha(X) X_i (X_i - \mu_{i0})] = E\left[\frac{\partial}{\partial X_i}(\alpha(X) X_i)\right]$$

を使う.

$$\frac{\partial}{\partial X_i}(\alpha(X) X_i) = c \frac{\partial}{\partial X_i} \frac{X_i}{X^2} = c\left(\frac{1}{X^2} - \frac{2X_i^2}{X^4}\right)$$

より,

$$\sum_{i=1}^{n} E[\alpha(X) X_i (X_i - \mu_{i0})] = c\left(\sum_{i=1}^{n} E\left[\frac{1}{X^2}\right] - E\left[\frac{2X^2}{X^4}\right]\right) = c(n-2) E\left[\frac{1}{X^2}\right].$$

3.4 第3項

$\alpha(X)^2 = c^2/X^4$, $\sum_{i=1}^{n} X_i^2 = X^2$ なので

$$\sum_{i=1}^{n} E[\alpha(X)^2 X_i^2] = E[\alpha(X)^2 X^2] = E\left[\frac{c^2}{X^2}\right] = c^2 E\left[\frac{1}{X^2}\right].$$

3.5 Stein推定量

ゆえに, $\tilde{\mu}_i$ の平均二乗誤差は

$$\begin{aligned} E\left[\sum_{i=1}^{n}(\tilde{\mu}_i - \mu_{i0})^2\right] &= n - 2c(n-2)E\left[\frac{1}{X^2}\right] + c^2 E\left[\frac{1}{X^2}\right] \\ &= n + (c^2 - 2(n-2)c)E\left[\frac{1}{X^2}\right]. \end{aligned}$$

これを最小にする c は

$$c = n - 2$$

になる. このときの, 推定値

$$\tilde{\mu}_i = (1 - \alpha(X))X_i = \left(1 - \frac{n-2}{X^2}\right)X_i$$

を**Stein推定量** (https://www.google.com/search?q=Stein%E6%8E%A8%E5%AE%9A%E9%87%8F)と呼ぶことにする.

そのとき得られる $\tilde{\mu}_i$ の平均二乗誤差の最小値は

$$E\left[\sum_{i=1}^{n}(\tilde{\mu}_i - \mu_{i0})^2\right] = n - (n-2)^2 E\left[\frac{1}{X^2}\right] < n = E\left[\sum_{i=1}^{n}(\hat{\mu}_i - \mu_{i0})^2\right]$$

となる.

このようにRidge正則化によって得られたStein推定量 $\tilde{\mu}_i$ の平均二乗誤差は最尤推定量 $\hat{\mu}_i = X_i$ の平均二乗誤差より小さい.

3.6 すべての μ_{i0} が0の場合の平均二乗誤差

$\mu_0 = (\mu_{10}, \ldots, \mu_{n0}) = (0, \ldots, 0)$ のとき,

$$E\left[\frac{1}{X^2}\right] = \frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} e^{-x^2/2} \frac{1}{x^2}\, dx.$$

これは, x_i の分散を $1/t > 0$ にした場合より,

$$\frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} e^{-tx^2/2}\, dx = t^{-n/2}.$$

両辺を t について 1 から ∞ まで積分すると,

$$\frac{1}{(2\pi)^{n/2}} \int_{\mathbb{R}^n} e^{-x^2/2} \frac{2}{x^2}\, dx = \frac{1}{n/2 - 1}.$$

ゆえに

$$E\left[\frac{1}{X^2}\right] = \frac{1}{n-2}.$$

したがって, この場合には, Stein推定量の平均二乗誤差は

$$E\left[\sum_{i=1}^{n}(\tilde{\mu}_i - \mu_{i0})^2\right] = n - (n-2)^2 E\left[\frac{1}{X^2}\right] = 2$$

となる. これは $n \geqq 3$ が大きいとき, 最尤推定量の平均二乗誤差の n より相当に小さくなる.

3.7 正則化と事前分布の関係

最尤法では, 確率モデル $p(x|w)$ のサンプル X に関する対数尤度の -1 倍

$$L(w) = -\log p(X|w)$$

を最小化するパラメーター $w = \hat{w}$ を求め, $p(x|\hat{w})$ を予測分布として採用する.

事前分布 $\varphi(w)$ と尤度函数の積の対数の -1 倍

$$R(w) = -\log p(X|w) - \log \varphi(w)$$

を最小化するパラメーター $w = \tilde{w}$ を求めて $p(x|\tilde{w})$ を予測分布として採用する推定法を**MAP法 (最大事後確率法)** と呼ぶ.
$-\log \varphi(w)$ の項を罰則項とみなすとき, この方法は**正則化 (regularization)** とも呼ばれる.

事前分布 $\varphi(w)$ が正規分布の積の場合の正則化は**Ridge正則化**と呼ばれている. 事前分布がLaplace分布の積の場合の正則化は**LASSO正則化**と呼ばれている.

以上で示したStein推定量の例は, 最尤法が必ずしも最良の推定量を与えるとは限らず, 正則化によって平均二乗誤差がより小さな得られる場合があることを示している.

すなわち, 最尤法の代わりに事前分布を与えたMAP法を使った方が平均予測誤差が小さくなることがありえる.

このような事実は「事前分布は主観や確信や信念を表す」と信じ込んで疑わない人達には思いもよらないことだと思われる. 事前分布は予測誤差を小さくするためにも有効に利用可能である.

3.8 平均に向けて縮小するタイプのStein推定量

以上においては, 原点 0 に寄せて縮小するタイプのStein推定量

$$\tilde{\mu}_i = \left(1 - \frac{n-2}{X^2}\right) X_i, \quad X^2 = \sum_{i=1}^{n} X_i^2$$

を扱った. これは 0 に向けて縮小するタイプの推定量である. $X_1, \ldots, X_n$ の平均と平均との差の二乗和を

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i, \quad (X - \overline{X})^2 = \sum_{i=1}^{n} (X_i - \overline{X})^2.$$

と書くことにする. 0 ではなく, 平均 $\overline{X}$ に向けて縮小するタイプのStein推定量が

$$\overline{\tilde{\mu}}_i = \overline{X} + \left(1 - \frac{n-3}{(X - \overline{X})^2}\right)(X_i - \overline{X})$$

と定義される. 以下ではこれが上の 0 に向けて寄せるタイプのStein推定量から導かれることを説明しよう. ポイントは二乗和分母に対する分子が $n-2$ から $n-3$ に下がる理由が「自由度が1減ったこと」によって説明できることである.

$A = [a_{ij}]$ は任意の $n \times n$ の直交行列であるとし, $Y_j = \sum_{i=1}^{n} X_i a_{ij}$ とおく. このとき, 直交行列の定義とサンプルの分散と二乗和の関係より,

$$\sum_{i=1}^{n} Y_i^2 = \sum_{i=1}^{n} X_i^2 = \sum_{i=1}^{n} (X_i - \overline{X})^2 + n\overline{X}^2 = (X - \overline{X})^2 + n\overline{X}^2.$$

さらに, Y_j 達は独立な確率変数になり, 各 Y_j は平均 $\sum_{i=1}^{n} \mu_{i0} a_{ij}$, 分散 1 の正規分布に従う.

直交行列 A の第 n 列のすべての成分を $1/\sqrt{n}$ にする. このとき, $Y_n = \sqrt{n}\,\overline{X}$ となるので, 上の計算結果より,

$$(X - \overline{X})^2 = \sum_{i=1}^{n-1} Y_i^2.$$

$Y_1, \ldots, Y_{n-1}$ 達の平均に関する 0 に向けて縮小するタイプのStein推定量は

$$\left(1 - \frac{n-3}{\sum_{i=1}^{n-2} Y_i^2}\right) Y_i = \left(1 - \frac{n-3}{(X - \overline{X})^2}\right) Y_i \quad (i = 1, \ldots, n-1).$$

分子が $n-3$ になる理由は $Y_1, \ldots, Y_{n-1}$ が $n-1$ 個しかないからである. Y_n の平均の推定量としては Y_n 自身を採用しよう. このとき, $\alpha = 1 - (n-3)/(X - \overline{X})^2$ とおくと, $j = 1, \ldots, n-1$ のとき, $\sum_{i=1}^{n} a_{ij} = 0$ より,

$$\sum_{i=1}^{n} \left(\overline{X} + (1 - \alpha(X_i - \overline{X})\right) a_{ij} = \sum_{i=1}^{n} \left((1-\alpha)X_i + \alpha\overline{X}\right) a_{ij} = (1-\alpha) Y_j.$$

そして, $a_{in} = 1/\sqrt{n}$ より,

$$\sum_{i=1}^{n}\left((1-\alpha)X_i+\alpha\overline{X}\right)a_{in}=(1-\alpha)\sqrt{n}\,\overline{X}+\alpha\overline{X}\sqrt{n}=\sqrt{n}\,\overline{X}=Y_n.$$

これで, $Y_1,\ldots,Y_{n-1},Y_n$ の平均のStein推定量にちょうど上の $\overline{\tilde{\mu}}_i$ が対応していることがわかる.

4 数値的検証

$n\geqq 3$ であると仮定する. 以上で扱ったStein推定量

$$\tilde{\mu}_i=\left(1-\frac{n-2}{X^2}\right)X_i,$$
$$\overline{\tilde{\mu}}_i=\overline{X}+\left(1-\frac{n-3}{(X-\overline{X})^2}\right)(X_i-\overline{X})$$

を少しモディファイした推定量

$$\check{\mu}_i=\max\left(0,1-\frac{n-2}{X^2}\right)X_i,$$
$$\overline{\check{\mu}}_i=\overline{X}+\max\left(0,1-\frac{n-3}{(X-\overline{X})^2}\right)(X_i-\overline{X})$$

も定義しておこう.

以下では最尤推定量 $\hat{\mu}_i=X_i$ とこれらの推定量を比較する.

In [1]:
```
using Distributions
using LinearAlgebra
using Printf
```

In [2]:
```
mu_hat(X) = X
norm2(X) = dot(X,X)
alpha(X) = (length(X) - 2)/norm2(X)
mu_tilde(X) = (1 - alpha(X))*X
mu_check(X) = max(0, 1 - alpha(X))*X
alpha(X, c) = c/norm2(X)
mu_tilde_bar(X) = let M = mean(X), Y = X .- M; M .+ (1 - alpha(Y,length(X)-3))*Y end
mu_check_bar(X) = let M = mean(X), Y = X .- M; M .+ max(0, 1 - alpha(Y,length(X)-3))*Y end
square_error(mu, mu0) = sum((mu[i] - mu0[i])^2 for i in 1:length(mu))
```

Out[2]:
```
square_error (generic function with 1 method)
```

In [3]:
```
function sim_stein(;
        mu0 = rand(Normal(), 10),
        niters = 10^5,
    )
    n = length(mu0)
    square_error_mu_hat   = Array{Float64, 1}(undef, niters)
    square_error_mu_tilde = Array{Float64, 1}(undef, niters)
    square_error_mu_check = Array{Float64, 1}(undef, niters)
    square_error_mu_tilde_bar = Array{Float64, 1}(undef, niters)
    square_error_mu_check_bar = Array{Float64, 1}(undef, niters)
    for l in 1:niters
        X = rand(MvNormal(mu0, I))
        square_error_mu_hat[l]   = square_error(mu_hat(X),   mu0)
        square_error_mu_tilde[l] = square_error(mu_tilde(X), mu0)
        square_error_mu_check[l] = square_error(mu_check(X), mu0)
        square_error_mu_tilde_bar[l] = square_error(mu_tilde_bar(X), mu0)
        square_error_mu_check_bar[l] = square_error(mu_check_bar(X), mu0)
    end

    s = (
        square_error_mu_hat,
        square_error_mu_tilde,
        square_error_mu_check,
        square_error_mu_tilde_bar,
        square_error_mu_check_bar
    )

    print_estimator(s)
    println()
    print_comparison(s)
    s
end
```

Out[3]: sim_stein (generic function with 1 method)

In [4]:
```
function print_estimator(s)
    @printf("average of square errors of mu_hat       = %10.3f\n", mean(s[1]))
    @printf("average of square errors of mu_tilde     = %10.3f\n", mean(s[2]))
    @printf("average of square errors of mu_check     = %10.3f\n", mean(s[3]))
    @printf("average of square errors of mu_tilde_bar = %10.3f\n", mean(s[4]))
    @printf("average of square errors of mu_check_bar = %10.3f\n", mean(s[5]))
end
```

Out[4]: print_estimator (generic function with 1 method)

In [5]:
```
function comparison(s)
    n = length(s)
    c = zeros(n, n)
    for i in 1:n, j in 1:n
        if i != j
            c[i,j] = mean(s[i] .< s[j])
        end
    end
    c
end
```

Out[5]: comparison (generic function with 1 method)

In [6]:
```
function print_comparison(s)
    c = comparison(s)
    @printf("%-12s | %-12s | %-12s | %-12s | %-12s | %-12s |\n", "count(<)/n", "mu_hat", "mu_tilde", "mu_check", "mu_tilde_bar", "mu_check_bar")
    println("-"^13 * ("+"*"-"^14)^5 * "|")
    @printf("%-12s | %12s | %12.5f | %12.5f | %12.5f | %12.5f |\n", "mu_hat", "  ---  ", c[1,2], c[1,3], c[1,4], c[1,5])
    @printf("%-12s | %12.5f | %12s | %12.5f | %12.5f | %12.5f |\n", "mu_tilde", c[2,1], "  ---  ", c[2,3], c[2,4], c[2,5])
    @printf("%-12s | %12.5f | %12.5f | %12s | %12.5f | %12.5f |\n", "mu_check", c[3,1], c[3,2], "  ---  ", c[3,4], c[3,5])
    @printf("%-12s | %12.5f | %12.5f | %12.5f | %12s | %12.5f |\n", "mu_tilde_bar", c[4,1], c[4,2], c[4,3], "  ---  ", c[4,5])
    @printf("%-12s | %12.5f | %12.5f | %12.5f | %12.5f | %12s |\n", "mu_check_bar", c[5,1], c[5,2], c[5,3], c[5,4], "  ---  ")
end
```

Out[6]:
```
print_comparison (generic function with 1 method)
```

In [7]:
```
s = sim_stein();
```

```
average of square errors of mu_hat       =     10.001
average of square errors of mu_tilde     =      6.033
average of square errors of mu_check     =      5.915
average of square errors of mu_tilde_bar =      6.747
average of square errors of mu_check_bar =      6.644

count(<)/n   | mu_hat       | mu_tilde     | mu_check     | mu_tilde_bar | mu_check_bar |
-------------+--------------+--------------+--------------+--------------+--------------|
mu_hat       |        ---   |      0.11221 |      0.10871 |      0.12912 |      0.12677 |
mu_tilde     |      0.88779 |        ---   |      0.00019 |      0.81243 |      0.80570 |
mu_check     |      0.89129 |      0.03864 |        ---   |      0.83169 |      0.83037 |
mu_tilde_bar |      0.87088 |      0.18757 |      0.16831 |        ---   |      0.00018 |
mu_check_bar |      0.87323 |      0.19430 |      0.16963 |      0.03386 |        ---   |
```

例えば, 以上の表の mu_hat の行の数値(muhat の右側の数値)は, 最尤推定量 $\hat{\mu}$ の方が他の推定量よりも二乗誤差が小さい場合の割合である. その割合は大きいほどよい. この場合には(サンプルが標準正規分布のサイズ 10), 最尤推定量 $\hat{\mu}$ は他の推定量よりも二乗誤差が大きな場合の割合がどれも88%を超えている.

- mu_hat = $\hat{\mu}$ = 最尤推定量
- mu_tilde = $\tilde{\mu}$ = Stein推定量
- mu_check = $\check{\mu}$ = $1-\alpha$ を $\max(0, 1-\alpha)$ に置き換えたStein推定量
- mu_tilde_bar = $\bar{\tilde{\mu}}$ = 平均の方向に縮小するStein推定量
- mu_check_bar = $\bar{\hat{\mu}}$ = 平均の方法に縮小するStein推定量で$1-\alpha$ を $\max(0, 1-\alpha)$ に置き換えたもの

4.1 すべての μ_{i0} が0の場合

この場合には, 最尤推定量 $\hat{\mu}_i = X_i$ の平均二乗誤差は n になり, Stein推定量 $\tilde{\mu}_i = (1-(n-2)/X^2)X_i$ の平均二乗誤差は 2 になる. 以下の計算でも実際にそうなっていることを確認できる.

In [8]:
```
n = 3
sim_stein(mu0 = zeros(n));
```

```
average of square errors of mu_hat       =      2.991
average of square errors of mu_tilde     =      2.000
average of square errors of mu_check     =      1.595
average of square errors of mu_tilde_bar =      2.991
average of square errors of mu_check_bar =      2.991

count(<)/n   | mu_hat       | mu_tilde     | mu_check     | mu_tilde_bar | mu_check_bar |
-------------+--------------+--------------+--------------+--------------+--------------|
mu_hat       |        ---   |      0.08172 |      0.00000 |      0.04344 |      0.04344 |
mu_tilde     |      0.91828 |        ---   |      0.00000 |      0.91828 |      0.91828 |
mu_check     |      1.00000 |      0.19871 |        ---   |      1.00000 |      1.00000 |
mu_tilde_bar |      0.04402 |      0.08172 |      0.00000 |        ---   |      0.00000 |
mu_check_bar |      0.04402 |      0.08172 |      0.00000 |      0.00000 |        ---   |
```

In [9]:
```
n = 4
sim_stein(mu0 = zeros(n));
```

```
average of square errors of mu_hat       =      4.006
average of square errors of mu_tilde     =      1.986
average of square errors of mu_check     =      1.477
average of square errors of mu_tilde_bar =      2.989
average of square errors of mu_check_bar =      2.608

count(<)/n   | mu_hat       | mu_tilde     | mu_check     | mu_tilde_bar | mu_check_bar |
-------------+--------------+--------------+--------------+--------------+--------------|
mu_hat       |         ---  |      0.08954 |      0.00000 |      0.08104 |      0.00000 |
mu_tilde     |      0.91046 |        ---   |      0.00000 |      0.86904 |      0.85374 |
mu_check     |      1.00000 |      0.26480 |        ---   |      1.00000 |      1.00000 |
mu_tilde_bar |      0.91896 |      0.13096 |      0.00000 |        ---   |      0.00000 |
mu_check_bar |      1.00000 |      0.14626 |      0.00000 |      0.19774 |        ---   |
```

In [10]:
```
n = 10
sim_stein(mu0 = zeros(n));
```

```
average of square errors of mu_hat       =      9.992
average of square errors of mu_tilde     =      1.985
average of square errors of mu_check     =      1.252
average of square errors of mu_tilde_bar =      2.991
average of square errors of mu_check_bar =      2.270

count(<)/n   | mu_hat       | mu_tilde     | mu_check     | mu_tilde_bar | mu_check_bar |
-------------+--------------+--------------+--------------+--------------+--------------|
mu_hat       |         ---  |      0.05237 |      0.00000 |      0.05836 |      0.00000 |
mu_tilde     |      0.94763 |        ---   |      0.00000 |      0.82910 |      0.77475 |
mu_check     |      1.00000 |      0.37208 |        ---   |      1.00000 |      1.00000 |
mu_tilde_bar |      0.94164 |      0.17090 |      0.00000 |        ---   |      0.00000 |
mu_check_bar |      1.00000 |      0.22525 |      0.00000 |      0.36343 |        ---   |
```

In [11]:
```
n = 100
sim_stein(mu0 = zeros(n));
```

```
average of square errors of mu_hat       =    100.060
average of square errors of mu_tilde     =      2.012
average of square errors of mu_check     =      1.094
average of square errors of mu_tilde_bar =      3.007
average of square errors of mu_check_bar =      2.090

count(<)/n   | mu_hat       | mu_tilde     | mu_check     | mu_tilde_bar | mu_check_bar |
-------------+--------------+--------------+--------------+--------------+--------------|
mu_hat       |         ---  |      0.00003 |      0.00000 |      0.00003 |      0.00000 |
mu_tilde     |      0.99997 |        ---   |      0.00000 |      0.85675 |      0.71665 |
mu_check     |      1.00000 |      0.46319 |        ---   |      1.00000 |      1.00000 |
mu_tilde_bar |      0.99997 |      0.14325 |      0.00000 |        ---   |      0.00000 |
mu_check_bar |      1.00000 |      0.28335 |      0.00000 |      0.46285 |        ---   |
```

In [12]:
```
n = 1000
sim_stein(mu0 = zeros(n));
```

```
average of square errors of mu_hat       =   1000.011
average of square errors of mu_tilde     =      2.000
average of square errors of mu_check     =      1.025
average of square errors of mu_tilde_bar =      3.007
average of square errors of mu_check_bar =      2.031

count(<)/n   | mu_hat       | mu_tilde     | mu_check     | mu_tilde_bar | mu_check_bar |
-------------+--------------+--------------+--------------+--------------+--------------|
mu_hat       |         ---  |      0.00000 |      0.00000 |      0.00000 |      0.00000 |
mu_tilde     |      1.00000 |        ---   |      0.00000 |      0.90802 |      0.70213 |
mu_check     |      1.00000 |      0.48835 |        ---   |      1.00000 |      1.00000 |
mu_tilde_bar |      1.00000 |      0.09198 |      0.00000 |        ---   |      0.00000 |
mu_check_bar |      1.00000 |      0.29787 |      0.00000 |      0.48855 |        ---   |
```

4.2 雑多な場合

In [13]:
```
mu0 = 3ones(100)
sim_stein(mu0 = mu0);
```

```
average of square errors of mu_hat       =    100.056
average of square errors of mu_tilde     =     90.423
average of square errors of mu_check     =     90.423
average of square errors of mu_tilde_bar =      3.012
average of square errors of mu_check_bar =      2.088

count(<)/n   | mu_hat       | mu_tilde     | mu_check     | mu_tilde_bar | mu_check_bar |
-------------+--------------+--------------+--------------+--------------+--------------|
mu_hat       |          ---    |      0.05751 |      0.05751 |      0.00000 |      0.00000 |
mu_tilde     |      0.94249 |        ---   |      0.00000 |      0.00000 |      0.00000 |
mu_check     |      0.94249 |      0.00000 |        ---   |      0.00000 |      0.00000 |
mu_tilde_bar |      1.00000 |      1.00000 |      1.00000 |        ---   |      0.00000 |
mu_check_bar |      1.00000 |      1.00000 |      1.00000 |      0.45818 |        ---   |
```

In [14]:
```
mu0 = 3 .+ randn(100)
sim_stein(mu0 = mu0);
```

```
average of square errors of mu_hat       =    100.042
average of square errors of mu_tilde     =     91.155
average of square errors of mu_check     =     91.155
average of square errors of mu_tilde_bar =     46.518
average of square errors of mu_check_bar =     46.518

count(<)/n   | mu_hat       | mu_tilde     | mu_check     | mu_tilde_bar | mu_check_bar |
-------------+--------------+--------------+--------------+--------------+--------------|
mu_hat       |          ---    |      0.06500 |      0.06500 |      0.00019 |      0.00019 |
mu_tilde     |      0.93500 |        ---   |      0.00000 |      0.00030 |      0.00030 |
mu_check     |      0.93500 |      0.00000 |        ---   |      0.00030 |      0.00030 |
mu_tilde_bar |      0.99981 |      0.99970 |      0.99970 |        ---   |      0.00000 |
mu_check_bar |      0.99981 |      0.99970 |      0.99970 |      0.00001 |        ---   |
```

In [15]:
```
mu0 = collect(range(0, 6, length=100))
sim_stein(mu0 = mu0);
```

```
average of square errors of mu_hat       =     99.995
average of square errors of mu_tilde     =     92.611
average of square errors of mu_check     =     92.611
average of square errors of mu_tilde_bar =     76.569
average of square errors of mu_check_bar =     76.569

count(<)/n   | mu_hat       | mu_tilde     | mu_check     | mu_tilde_bar | mu_check_bar |
-------------+--------------+--------------+--------------+--------------+--------------|
mu_hat       |          ---    |      0.08330 |      0.08330 |      0.00713 |      0.00713 |
mu_tilde     |      0.91670 |        ---   |      0.00000 |      0.01783 |      0.01783 |
mu_check     |      0.91670 |      0.00000 |        ---   |      0.01783 |      0.01783 |
mu_tilde_bar |      0.99287 |      0.98217 |      0.98217 |        ---   |      0.00000 |
mu_check_bar |      0.99287 |      0.98217 |      0.98217 |      0.00000 |        ---   |
```

In [16]:
```
mu0 = collect(range(-3, 3, length=100))
sim_stein(mu0 = mu0);
```

```
average of square errors of mu_hat       =     99.953
average of square errors of mu_tilde     =     76.137
average of square errors of mu_check     =     76.137
average of square errors of mu_tilde_bar =     76.562
average of square errors of mu_check_bar =     76.562

count(<)/n   | mu_hat       | mu_tilde     | mu_check     | mu_tilde_bar | mu_check_bar |
-------------+--------------+--------------+--------------+--------------+--------------|
mu_hat       |          ---    |      0.00708 |      0.00708 |      0.00752 |      0.00752 |
mu_tilde     |      0.99292 |        ---   |      0.00000 |      0.84951 |      0.84951 |
mu_check     |      0.99292 |      0.00000 |        ---   |      0.84951 |      0.84951 |
mu_tilde_bar |      0.99248 |      0.15049 |      0.15049 |        ---   |      0.00000 |
mu_check_bar |      0.99248 |      0.15049 |      0.15049 |      0.00000 |        ---   |
```

In [17]:
```
mu0 = randn(10)
sim_stein(mu0 = mu0);
```

```
average of square errors of mu_hat       =      9.999
average of square errors of mu_tilde     =      5.237
average of square errors of mu_check     =      4.987
average of square errors of mu_tilde_bar =      5.994
average of square errors of mu_check_bar =      5.765

count(<)/n   | mu_hat       | mu_tilde     | mu_check     | mu_tilde_bar | mu_check_bar |
-------------+--------------+--------------+--------------+--------------+--------------|
mu_hat       |         ---  |      0.09480 |      0.07974 |      0.10659 |      0.09411 |
mu_tilde     |      0.90520 |        ---   |      0.00099 |      0.80349 |      0.78930 |
mu_check     |      0.92026 |      0.08388 |        ---   |      0.84623 |      0.82581 |
mu_tilde_bar |      0.89341 |      0.19651 |      0.15377 |        ---   |      0.00118 |
mu_check_bar |      0.90589 |      0.21070 |      0.17419 |      0.07799 |        ---   |
```

In [18]:
```
mu0 = randn(100)
sim_stein(mu0 = mu0);
```

```
average of square errors of mu_hat       =     99.981
average of square errors of mu_tilde     =     54.670
average of square errors of mu_check     =     54.670
average of square errors of mu_tilde_bar =     55.288
average of square errors of mu_check_bar =     55.288

count(<)/n   | mu_hat       | mu_tilde     | mu_check     | mu_tilde_bar | mu_check_bar |
-------------+--------------+--------------+--------------+--------------+--------------|
mu_hat       |         ---  |      0.00045 |      0.00045 |      0.00049 |      0.00049 |
mu_tilde     |      0.99955 |        ---   |      0.00000 |      0.73767 |      0.73767 |
mu_check     |      0.99955 |      0.00000 |        ---   |      0.73767 |      0.73767 |
mu_tilde_bar |      0.99951 |      0.26233 |      0.26233 |        ---   |      0.00000 |
mu_check_bar |      0.99951 |      0.26233 |      0.26233 |      0.00000 |        ---   |
```

In [19]:
```
mu0 = randn(1000)
sim_stein(mu0 = mu0);
```

```
average of square errors of mu_hat       =   1000.082
average of square errors of mu_tilde     =    501.837
average of square errors of mu_check     =    501.837
average of square errors of mu_tilde_bar =    502.579
average of square errors of mu_check_bar =    502.579

count(<)/n   | mu_hat       | mu_tilde     | mu_check     | mu_tilde_bar | mu_check_bar |
-------------+--------------+--------------+--------------+--------------+--------------|
mu_hat       |         ---  |      0.00000 |      0.00000 |      0.00000 |      0.00000 |
mu_tilde     |      1.00000 |        ---   |      0.00000 |      0.89765 |      0.89765 |
mu_check     |      1.00000 |      0.00000 |        ---   |      0.89765 |      0.89765 |
mu_tilde_bar |      1.00000 |      0.10235 |      0.10235 |        ---   |      0.00000 |
mu_check_bar |      1.00000 |      0.10235 |      0.10235 |      0.00000 |        ---   |
```

In [20]:
```
mu0 = 10 .+ randn(10)
sim_stein(mu0 = mu0);
```

```
average of square errors of mu_hat       =      9.996
average of square errors of mu_tilde     =      9.925
average of square errors of mu_check     =      9.925
average of square errors of mu_tilde_bar =      7.098
average of square errors of mu_check_bar =      7.038

count(<)/n   | mu_hat       | mu_tilde     | mu_check     | mu_tilde_bar | mu_check_bar |
-------------+--------------+--------------+--------------+--------------+--------------|
mu_hat       |         ---  |      0.43457 |      0.43457 |      0.14198 |      0.14111 |
mu_tilde     |      0.56543 |        ---   |      0.00000 |      0.14404 |      0.14315 |
mu_check     |      0.56543 |      0.00000 |        ---   |      0.14404 |      0.14315 |
mu_tilde_bar |      0.85802 |      0.85596 |      0.85596 |        ---   |      0.00008 |
mu_check_bar |      0.85889 |      0.85685 |      0.85685 |      0.01939 |        ---   |
```

In [21]:
```
mu0 = 10 .+ randn(100)
sim_stein(mu0 = mu0);
```

```
average of square errors of mu_hat       =    100.088
average of square errors of mu_tilde     =     99.145
average of square errors of mu_check     =     99.145
average of square errors of mu_tilde_bar =     54.798
average of square errors of mu_check_bar =     54.798

count(<)/n   | mu_hat       | mu_tilde     | mu_check     | mu_tilde_bar | mu_check_bar |
-------------+--------------+--------------+--------------+--------------+--------------|
mu_hat       |        ---   |      0.30972 |      0.30972 |      0.00043 |      0.00043 |
mu_tilde     |      0.69028 |        ---   |      0.00000 |      0.00047 |      0.00047 |
mu_check     |      0.69028 |      0.00000 |        ---   |      0.00047 |      0.00047 |
mu_tilde_bar |      0.99957 |      0.99953 |      0.99953 |        ---   |      0.00000 |
mu_check_bar |      0.99957 |      0.99953 |      0.99953 |      0.00000 |        ---   |
```

In [22]:
```
mu0 = 10 .+ randn(1000)
sim_stein(mu0 = mu0);
```

```
average of square errors of mu_hat       =    999.873
average of square errors of mu_tilde     =    990.044
average of square errors of mu_check     =    990.044
average of square errors of mu_tilde_bar =    514.212
average of square errors of mu_check_bar =    514.212

count(<)/n   | mu_hat       | mu_tilde     | mu_check     | mu_tilde_bar | mu_check_bar |
-------------+--------------+--------------+--------------+--------------+--------------|
mu_hat       |        ---   |      0.05757 |      0.05757 |      0.00000 |      0.00000 |
mu_tilde     |      0.94243 |        ---   |      0.00000 |      0.00000 |      0.00000 |
mu_check     |      0.94243 |      0.00000 |        ---   |      0.00000 |      0.00000 |
mu_tilde_bar |      1.00000 |      1.00000 |      1.00000 |        ---   |      0.00000 |
mu_check_bar |      1.00000 |      1.00000 |      1.00000 |      0.00000 |        ---   |
```

In [ ]:
```

```