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

$$N(x|a) = \frac{1}{\mu} e^{-(x-a)^2/2}$$

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 $X=(X_1,\dots,X_n)$ 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
$\mu(X)=(\mu_1(X),\dots,\mu_n(X))$ 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

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

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

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

3.5 Stein推定量

ゆえに, $\tilde{\mu}_i$ の平均二乗誤差は

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

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

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

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

$\mu_0 = (\mu_{10},\dots,\mu_{n0}) = (0,\dots,0)$ 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

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

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.

4 数値的検証

$n \geqq 3$ であると仮定する. Stein推定量 $\tilde{\mu}_i = (1 - (n-2)/X^2)X_i$ を少しモディファイした推定量

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

も定義しておこう. さらに, X^2 を X_i 達の平均との差の二乗和で置き換えた以下の推定量も考える:

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

ここで,

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

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

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))

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

    @show mean(square_error_mu_hat)
    @show mean(square_error_mu_tilde)
    @show mean(square_error_mu_check)
    @show mean(square_error_mu_tilde_bar)
    @show mean(square_error_mu_check_bar)
    println()

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

    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], "  ---  ")

    s
end

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[1]: comparison (generic function with 1 method)
```

```
In [2]:
1  s = sim_stein();
```

```
mean(square_error_mu_hat) = 10.00562220516786
mean(square_error_mu_tilde) = 5.094083326620445
mean(square_error_mu_check) = 4.824902607322814
mean(square_error_mu_tilde_bar) = 5.70912460546358
mean(square_error_mu_check_bar) = 5.426119847208289

count(<)/n   ¦ mu_hat       ¦ mu_tilde     ¦ mu_check     ¦ mu_tilde_bar ¦ mu_check_bar ¦
-------------+--------------+--------------+--------------+--------------+--------------¦
mu_hat       ¦        ---   ¦      0.09260 ¦      0.07364 ¦      0.10089 ¦      0.08038 ¦
mu_tilde     ¦      0.90740 ¦        ---   ¦      0.00140 ¦      0.73422 ¦      0.71645 ¦
mu_check     ¦      0.92636 ¦      0.09395 ¦        ---   ¦      0.77860 ¦      0.73553 ¦
mu_tilde_bar ¦      0.89911 ¦      0.26578 ¦      0.22140 ¦        ---   ¦      0.00174 ¦
mu_check_bar ¦      0.91962 ¦      0.28355 ¦      0.26447 ¦      0.09921 ¦        ---   ¦
```

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 [3]:
1  n = 3
2  sim_stein(mu0 = zeros(n));
```

```
mean(square_error_mu_hat) = 2.9964192223421287
mean(square_error_mu_tilde) = 1.9507781500665806
mean(square_error_mu_check) = 1.5975855109278558
mean(square_error_mu_tilde_bar) = 2.9964192223421287
mean(square_error_mu_check_bar) = 2.9964192223421287

count(<)/n   ¦ mu_hat       ¦ mu_tilde     ¦ mu_check     ¦ mu_tilde_bar ¦ mu_check_bar ¦
-------------+--------------+--------------+--------------+--------------+--------------¦
mu_hat       ¦        ---   ¦      0.07999 ¦      0.00000 ¦      0.04411 ¦      0.04411 ¦
mu_tilde     ¦      0.92001 ¦        ---   ¦      0.00000 ¦      0.92001 ¦      0.92001 ¦
mu_check     ¦      1.00000 ¦      0.19768 ¦        ---   ¦      1.00000 ¦      1.00000 ¦
mu_tilde_bar ¦      0.04452 ¦      0.07999 ¦      0.00000 ¦        ---   ¦      0.00000 ¦
mu_check_bar ¦      0.04452 ¦      0.07999 ¦      0.00000 ¦      0.00000 ¦        ---   ¦
```

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

```
mean(square_error_mu_hat) = 3.998888372097366
mean(square_error_mu_tilde) = 1.982670514709086
mean(square_error_mu_check) = 1.4682042682255385
mean(square_error_mu_tilde_bar) = 2.9829946148228994
mean(square_error_mu_check_bar) = 2.6014811205752197

count(<)/n   ¦ mu_hat       ¦ mu_tilde     ¦ mu_check     ¦ mu_tilde_bar ¦ mu_check_bar ¦
-------------+--------------+--------------+--------------+--------------+--------------¦
mu_hat       ¦        ---   ¦      0.08939 ¦      0.00000 ¦      0.08080 ¦      0.00000 ¦
mu_tilde     ¦      0.91061 ¦        ---   ¦      0.00000 ¦      0.86996 ¦      0.85533 ¦
mu_check     ¦      1.00000 ¦      0.26220 ¦        ---   ¦      1.00000 ¦      1.00000 ¦
mu_tilde_bar ¦      0.91920 ¦      0.13004 ¦      0.00000 ¦        ---   ¦      0.00000 ¦
mu_check_bar ¦      1.00000 ¦      0.14467 ¦      0.00000 ¦      0.19996 ¦        ---   ¦
```

In [5]:
```
n = 10
sim_stein(mu0 = zeros(n));
```

```
mean(square_error_mu_hat) = 9.98431026164017
mean(square_error_mu_tilde) = 1.9991432760068886
mean(square_error_mu_check) = 1.250841943899265
mean(square_error_mu_tilde_bar) = 3.005396855462041
mean(square_error_mu_check_bar) = 2.2692593852165808

count(<)/n   ¦ mu_hat       ¦ mu_tilde     ¦ mu_check     ¦ mu_tilde_bar ¦ mu_check_bar ¦
-------------+--------------+--------------+--------------+--------------+--------------¦
mu_hat       ¦        ---   ¦      0.05322 ¦      0.00000 ¦      0.05953 ¦      0.00000 ¦
mu_tilde     ¦      0.94678 ¦        ---   ¦      0.00000 ¦      0.82848 ¦      0.77330 ¦
mu_check     ¦      1.00000 ¦      0.37263 ¦        ---   ¦      1.00000 ¦      1.00000 ¦
mu_tilde_bar ¦      0.94047 ¦      0.17152 ¦      0.00000 ¦        ---   ¦      0.00000 ¦
mu_check_bar ¦      1.00000 ¦      0.22670 ¦      0.00000 ¦      0.36450 ¦        ---   ¦
```

In [6]:
```
n = 100
sim_stein(mu0 = zeros(n));
```

```
mean(square_error_mu_hat) = 100.03974192322661
mean(square_error_mu_tilde) = 2.0105287987588234
mean(square_error_mu_check) = 1.084531187322945
mean(square_error_mu_tilde_bar) = 3.0041206218235916
mean(square_error_mu_check_bar) = 2.079647042605977

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 ¦      0.85608 ¦      0.71710 ¦
mu_check     ¦      1.00000 ¦      0.46143 ¦        ---   ¦      1.00000 ¦      1.00000 ¦
mu_tilde_bar ¦      1.00000 ¦      0.14392 ¦      0.00000 ¦        ---   ¦      0.00000 ¦
mu_check_bar ¦      1.00000 ¦      0.28290 ¦      0.00000 ¦      0.46073 ¦        ---   ¦
```

In [7]:
```
n = 1000
sim_stein(mu0 = zeros(n));
```

```
mean(square_error_mu_hat) = 999.9810016048416
mean(square_error_mu_tilde) = 2.0128327921327833
mean(square_error_mu_check) = 1.032927681852483
mean(square_error_mu_tilde_bar) = 3.005309957186467
mean(square_error_mu_check_bar) = 2.0254574530613483

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 ¦      0.90721 ¦      0.70046 ¦
mu_check     ¦      1.00000 ¦      0.48863 ¦        ---   ¦      1.00000 ¦      1.00000 ¦
mu_tilde_bar ¦      1.00000 ¦      0.09279 ¦      0.00000 ¦        ---   ¦      0.00000 ¦
mu_check_bar ¦      1.00000 ¦      0.29954 ¦      0.00000 ¦      0.48842 ¦        ---   ¦
```

4.2 雑多な場合

In [8]:
```
mu0 = 3ones(100)
sim_stein(mu0 = mu0);
```

```
mean(square_error_mu_hat) = 99.95041187453639
mean(square_error_mu_tilde) = 90.32844721388072
mean(square_error_mu_check) = 90.32844721388072
mean(square_error_mu_tilde_bar) = 3.0004723663092308
mean(square_error_mu_check_bar) = 2.0666125673906586

count(<)/n   ¦ mu_hat       ¦ mu_tilde     ¦ mu_check     ¦ mu_tilde_bar ¦ mu_check_bar ¦
-------------+--------------+--------------+--------------+--------------+--------------¦
mu_hat       ¦        ---   ¦      0.05759 ¦      0.05759 ¦      0.00003 ¦      0.00000 ¦
mu_tilde     ¦      0.94241 ¦        ---   ¦      0.00000 ¦      0.00002 ¦      0.00000 ¦
mu_check     ¦      0.94241 ¦      0.00000 ¦        ---   ¦      0.00002 ¦      0.00000 ¦
mu_tilde_bar ¦      0.99997 ¦      0.99998 ¦      0.99998 ¦        ---   ¦      0.00000 ¦
mu_check_bar ¦      1.00000 ¦      1.00000 ¦      1.00000 ¦      0.46312 ¦        ---   ¦
```

In [9]:
```
mu0 = 3 .+ randn(100)
sim_stein(mu0 = mu0);
```

```
mean(square_error_mu_hat) = 100.11651084558751
mean(square_error_mu_tilde) = 91.7159715579375
mean(square_error_mu_check) = 91.7159715579375
mean(square_error_mu_tilde_bar) = 50.30816320333466
mean(square_error_mu_check_bar) = 50.30816320333466

count(<)/n   ¦ mu_hat       ¦ mu_tilde     ¦ mu_check     ¦ mu_tilde_bar ¦ mu_check_bar ¦
-------------+--------------+--------------+--------------+--------------+--------------¦
mu_hat       ¦        ---   ¦      0.07013 ¦      0.07013 ¦      0.00021 ¦      0.00021 ¦
mu_tilde     ¦      0.92987 ¦        ---   ¦      0.00000 ¦      0.00049 ¦      0.00049 ¦
mu_check     ¦      0.92987 ¦      0.00000 ¦        ---   ¦      0.00049 ¦      0.00049 ¦
mu_tilde_bar ¦      0.99979 ¦      0.99951 ¦      0.99951 ¦        ---   ¦      0.00000 ¦
mu_check_bar ¦      0.99979 ¦      0.99951 ¦      0.99951 ¦      0.00000 ¦        ---   ¦
```

In [10]:
```
mu0 = collect(range(0, 6, length=100))
sim_stein(mu0 = mu0);
```

```
mean(square_error_mu_hat) = 99.94891364099948
mean(square_error_mu_tilde) = 92.56479110294975
mean(square_error_mu_check) = 92.56479110294975
mean(square_error_mu_tilde_bar) = 76.53385099001721
mean(square_error_mu_check_bar) = 76.53385099001721

count(<)/n   ¦ mu_hat       ¦ mu_tilde     ¦ mu_check     ¦ mu_tilde_bar ¦ mu_check_bar ¦
-------------+--------------+--------------+--------------+--------------+--------------¦
mu_hat       ¦        ---   ¦      0.08371 ¦      0.08371 ¦      0.00742 ¦      0.00742 ¦
mu_tilde     ¦      0.91629 ¦        ---   ¦      0.00000 ¦      0.01700 ¦      0.01700 ¦
mu_check     ¦      0.91629 ¦      0.00000 ¦        ---   ¦      0.01700 ¦      0.01700 ¦
mu_tilde_bar ¦      0.99258 ¦      0.98300 ¦      0.98300 ¦        ---   ¦      0.00000 ¦
mu_check_bar ¦      0.99258 ¦      0.98300 ¦      0.98300 ¦      0.00000 ¦        ---   ¦
```

In [11]:
```
mu0 = collect(range(-3, 3, length=100))
sim_stein(mu0 = mu0);
```

```
mean(square_error_mu_hat) = 100.00158149925917
mean(square_error_mu_tilde) = 76.10606441738278
mean(square_error_mu_check) = 76.10606441738278
mean(square_error_mu_tilde_bar) = 76.5316653863651
mean(square_error_mu_check_bar) = 76.5316653863651

count(<)/n   ¦ mu_hat       ¦ mu_tilde     ¦ mu_check     ¦ mu_tilde_bar ¦ mu_check_bar ¦
-------------+--------------+--------------+--------------+--------------+--------------¦
mu_hat       ¦        ---   ¦      0.00681 ¦      0.00681 ¦      0.00712 ¦      0.00712 ¦
mu_tilde     ¦      0.99319 ¦        ---   ¦      0.00000 ¦      0.85182 ¦      0.85182 ¦
mu_check     ¦      0.99319 ¦      0.00000 ¦        ---   ¦      0.85182 ¦      0.85182 ¦
mu_tilde_bar ¦      0.99288 ¦      0.14818 ¦      0.14818 ¦        ---   ¦      0.00000 ¦
mu_check_bar ¦      0.99288 ¦      0.14818 ¦      0.14818 ¦      0.00000 ¦        ---   ¦
```

In [12]:
```
mu0 = randn(10)
sim_stein(mu0 = mu0);
```

```
mean(square_error_mu_hat) = 10.024476567375109
mean(square_error_mu_tilde) = 5.764361664139967
mean(square_error_mu_check) = 5.602475289096201
mean(square_error_mu_tilde_bar) = 6.500022804278585
mean(square_error_mu_check_bar) = 6.355826292896643

count(<)/n   ¦ mu_hat       ¦ mu_tilde     ¦ mu_check     ¦ mu_tilde_bar ¦ mu_check_bar ¦
-------------+--------------+--------------+--------------+--------------+--------------¦
mu_hat       ¦        ---   ¦      0.10522 ¦      0.09916 ¦      0.11958 ¦      0.11491 ¦
mu_tilde     ¦      0.89478 ¦        ---   ¦      0.00043 ¦      0.81218 ¦      0.80361 ¦
mu_check     ¦      0.90084 ¦      0.05276 ¦        ---   ¦      0.83899 ¦      0.83175 ¦
mu_tilde_bar ¦      0.88042 ¦      0.18782 ¦      0.16101 ¦        ---   ¦      0.00045 ¦
mu_check_bar ¦      0.88509 ¦      0.19639 ¦      0.16825 ¦      0.04676 ¦        ---   ¦
```

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

```
mean(square_error_mu_hat) = 100.0088851052276
mean(square_error_mu_tilde) = 45.38828408302389
mean(square_error_mu_check) = 45.38822288947407
mean(square_error_mu_tilde_bar) = 46.15080448522736
mean(square_error_mu_check_bar) = 46.150761849515575

count(<)/n   ¦ mu_hat       ¦ mu_tilde     ¦ mu_check     ¦ mu_tilde_bar ¦ mu_check_bar ¦
-------------+--------------+--------------+--------------+--------------+--------------¦
mu_hat       ¦        ---   ¦      0.00020 ¦      0.00020 ¦      0.00020 ¦      0.00020 ¦
mu_tilde     ¦      0.99980 ¦        ---   ¦      0.00000 ¦      0.82302 ¦      0.82302 ¦
mu_check     ¦      0.99980 ¦      0.00002 ¦        ---   ¦      0.82304 ¦      0.82302 ¦
mu_tilde_bar ¦      0.99980 ¦      0.17698 ¦      0.17696 ¦        ---   ¦      0.00000 ¦
mu_check_bar ¦      0.99980 ¦      0.17698 ¦      0.17698 ¦      0.00002 ¦        ---   ¦
```

In [14]:
```
mu0 = randn(1000)
sim_stein(mu0 = mu0);
```

```
mean(square_error_mu_hat) = 999.859979346132
mean(square_error_mu_tilde) = 498.74623501265467
mean(square_error_mu_check) = 498.74623501265467
mean(square_error_mu_tilde_bar) = 499.15179545651836
mean(square_error_mu_check_bar) = 499.15179545651836

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 ¦      0.46880 ¦      0.46880 ¦
mu_check     ¦      1.00000 ¦      0.00000 ¦        ---   ¦      0.46880 ¦      0.46880 ¦
mu_tilde_bar ¦      1.00000 ¦      0.53120 ¦      0.53120 ¦        ---   ¦      0.00000 ¦
mu_check_bar ¦      1.00000 ¦      0.53120 ¦      0.53120 ¦      0.00000 ¦        ---   ¦
```

In [15]:
```
mu0 = 10 .+ randn(10)
sim_stein(mu0 = mu0);
```

```
mean(square_error_mu_hat) = 9.984404426266675
mean(square_error_mu_tilde) = 9.913816529372525
mean(square_error_mu_check) = 9.913816529372525
mean(square_error_mu_tilde_bar) = 7.366883646073828
mean(square_error_mu_check_bar) = 7.328793661647324

count(<)/n   ¦ mu_hat       ¦ mu_tilde     ¦ mu_check     ¦ mu_tilde_bar ¦ mu_check_bar ¦
-------------+--------------+--------------+--------------+--------------+--------------¦
mu_hat       ¦        ---   ¦      0.43387 ¦      0.43387 ¦      0.15308 ¦      0.15271 ¦
mu_tilde     ¦      0.56613 ¦        ---   ¦      0.00000 ¦      0.15575 ¦      0.15533 ¦
mu_check     ¦      0.56613 ¦      0.00000 ¦        ---   ¦      0.15575 ¦      0.15533 ¦
mu_tilde_bar ¦      0.84692 ¦      0.84425 ¦      0.84425 ¦        ---   ¦      0.00002 ¦
mu_check_bar ¦      0.84729 ¦      0.84467 ¦      0.84467 ¦      0.01173 ¦        ---   ¦
```

In [16]:
```
mu0 = 10 .+ randn(100)
sim_stein(mu0 = mu0);
```

```
mean(square_error_mu_hat) = 100.03789466205295
mean(square_error_mu_tilde) = 99.08892362895115
mean(square_error_mu_check) = 99.08892362895115
mean(square_error_mu_tilde_bar) = 55.08856937797487
mean(square_error_mu_check_bar) = 55.08856937797487

count(<)/n   ¦ mu_hat       ¦ mu_tilde     ¦ mu_check     ¦ mu_tilde_bar ¦ mu_check_bar ¦
-------------+--------------+--------------+--------------+--------------+--------------¦
mu_hat       ¦        ---   ¦      0.30941 ¦      0.30941 ¦      0.00054 ¦      0.00054 ¦
mu_tilde     ¦      0.69059 ¦        ---   ¦      0.00000 ¦      0.00056 ¦      0.00056 ¦
mu_check     ¦      0.69059 ¦      0.00000 ¦        ---   ¦      0.00056 ¦      0.00056 ¦
mu_tilde_bar ¦      0.99946 ¦      0.99944 ¦      0.99944 ¦        ---   ¦      0.00000 ¦
mu_check_bar ¦      0.99946 ¦      0.99944 ¦      0.99944 ¦      0.00000 ¦        ---   ¦
```

In [17]:
```
mu0 = 10 .+ randn(1000)
sim_stein(mu0 = mu0);
```

```
mean(square_error_mu_hat) = 1000.0622201963936
mean(square_error_mu_tilde) = 990.2926723598086
mean(square_error_mu_check) = 990.2926723598086
mean(square_error_mu_tilde_bar) = 480.65459565551816
mean(square_error_mu_check_bar) = 480.65459565551816

count(<)/n   ¦ mu_hat       ¦ mu_tilde     ¦ mu_check     ¦ mu_tilde_bar ¦ mu_check_bar ¦
-------------+--------------+--------------+--------------+--------------+--------------¦
mu_hat       ¦        ---   ¦      0.06023 ¦      0.06023 ¦      0.00000 ¦      0.00000 ¦
mu_tilde     ¦      0.93977 ¦        ---   ¦      0.00000 ¦      0.00000 ¦      0.00000 ¦
mu_check     ¦      0.93977 ¦      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 [ ]:
```

```