

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)
(<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) (<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 の対数尤度関数の -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
 - 1.1 平均汎化誤差と平均二乗誤差の関係
- 2 最尤推定量
- 3 Ridge正則化とStein推定量
 - 3.1 $n \geq 3$ という仮定からの帰結
 - 3.2 第1項
 - 3.3 第2項
 - 3.4 第3項
 - 3.5 Stein推定量
 - 3.6 すべての μ_{i0} が0の場合の平均二乗誤差
 - 3.7 正則化と事前分布の関係
- 4 数値的検証
 - 4.1 すべての μ_{i0} が0の場合
 - 4.2 雑多な場合

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関数を

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

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。 $X = (X_1, \dots, X_n)$ は確率密度関数

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

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 $x = (x_1, \dots, x_n)$ に関する確率密度関数

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

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

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

ゆえに、もしも μ_i が $X = (X_1, \dots, X_n)$ の関数 $\mu_i(X)$ 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、

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

となる。

3 Ridge正則化とStein推定量

以下では $n \geq 3$ 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 $n \geq 3$ という仮定からの帰結

$n \geq 3$ 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

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

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

となる. これは $n \geq 3$ 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 $\varphi(w)$ と尤度関数の積の対数の -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

$n \geq 3$ 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。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]: 1 using Distributions
2 using LinearAlgebra
3
4 mu_hat(X) = X
5 alpha(X) = (length(X)-2)/dot(X,X)
6 mu_tilde(X) = (1 - alpha(X))*X
7 mu_check(X) = max(0, 1 - alpha(X))*X
8 square_error(mu, mu0) = sum((mu[i] - mu0[i])^2 for i in 1:length(mu))
9
10 function sim_stein(;
11     mu0 = rand(Normal(), 10),
12     niters = 10^5,
13 )
14 n = length(mu0)
15 square_error_mu_hat = Array{Float64, 1}(undef, niters)
16 square_error_mu_tilde = Array{Float64, 1}(undef, niters)
17 square_error_mu_check = Array{Float64, 1}(undef, niters)
18 for l in 1:niters
19     X = rand(MvNormal(mu0, I))
20     square_error_mu_hat[l] = square_error(mu_hat(X), mu0)
21     square_error_mu_tilde[l] = square_error(mu_tilde(X), mu0)
22     square_error_mu_check[l] = square_error(mu_check(X), mu0)
23 end
24 @show mean(square_error_mu_hat)
25 @show mean(square_error_mu_tilde)
26 @show mean(square_error_mu_check)
27 @show mean(square_error_mu_tilde.< square_error_mu_hat)
28 @show mean(square_error_mu_hat.< square_error_mu_tilde)
29 @show mean(square_error_mu_check.< square_error_mu_hat)
30 @show mean(square_error_mu_hat.< square_error_mu_check)
31 @show mean(square_error_mu_check.< square_error_mu_tilde)
32 @show mean(square_error_mu_tilde.< square_error_mu_check)
33 square_error_mu_hat, square_error_mu_tilde, square_error_mu_check
34 end
```

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

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になっていることを確認できる。

```

In [2]: 1 n = 3
        2 se_hat, se_tilde, se_check = sim_stein(mu0 = zeros(n));

mean(square_error_mu_hat) = 2.9884165039096073
mean(square_error_mu_tilde) = 2.0217859743179063
mean(square_error_mu_check) = 1.5916468513822986
mean(square_error_mu_tilde < square_error_mu_hat) = 0.91909
mean(square_error_mu_hat < square_error_mu_tilde) = 0.08091
mean(square_error_mu_check < square_error_mu_hat) = 1.0
mean(square_error_mu_hat < square_error_mu_check) = 0.0
mean(square_error_mu_check < square_error_mu_tilde) = 0.19874
mean(square_error_mu_tilde < square_error_mu_check) = 0.0

```

```

In [3]: 1 n = 4
        2 se_hat, se_tilde, se_check = sim_stein(mu0 = zeros(n));

mean(square_error_mu_hat) = 4.029065719496039
mean(square_error_mu_tilde) = 2.011987435613537
mean(square_error_mu_check) = 1.4907367863628262
mean(square_error_mu_tilde < square_error_mu_hat) = 0.91207
mean(square_error_mu_hat < square_error_mu_tilde) = 0.08793
mean(square_error_mu_check < square_error_mu_hat) = 1.0
mean(square_error_mu_hat < square_error_mu_check) = 0.0
mean(square_error_mu_check < square_error_mu_tilde) = 0.26006
mean(square_error_mu_tilde < square_error_mu_check) = 0.0

```

```

In [4]: 1 n = 10
        2 se_hat, se_tilde, se_check = sim_stein(mu0 = zeros(n));

mean(square_error_mu_hat) = 10.000695534768926
mean(square_error_mu_tilde) = 1.9979661261377688
mean(square_error_mu_check) = 1.258272199906412
mean(square_error_mu_tilde < square_error_mu_hat) = 0.94752
mean(square_error_mu_hat < square_error_mu_tilde) = 0.05248
mean(square_error_mu_check < square_error_mu_hat) = 1.0
mean(square_error_mu_hat < square_error_mu_check) = 0.0
mean(square_error_mu_check < square_error_mu_tilde) = 0.37125
mean(square_error_mu_tilde < square_error_mu_check) = 0.0

```

```

In [5]: 1 n = 100
        2 se_hat, se_tilde, se_check = sim_stein(mu0 = zeros(n));

mean(square_error_mu_hat) = 99.96525480174655
mean(square_error_mu_tilde) = 2.0140129372828657
mean(square_error_mu_check) = 1.0780383074157502
mean(square_error_mu_tilde < square_error_mu_hat) = 0.99999
mean(square_error_mu_hat < square_error_mu_tilde) = 1.0e-5
mean(square_error_mu_check < square_error_mu_hat) = 1.0
mean(square_error_mu_hat < square_error_mu_check) = 0.0
mean(square_error_mu_check < square_error_mu_tilde) = 0.46306
mean(square_error_mu_tilde < square_error_mu_check) = 0.0

```

```

In [6]: 1 n = 1000
        2 se_hat, se_tilde, se_check = sim_stein(mu0 = zeros(n));

mean(square_error_mu_hat) = 999.9049387643802
mean(square_error_mu_tilde) = 2.006865287316881
mean(square_error_mu_check) = 1.025900728112256
mean(square_error_mu_tilde < square_error_mu_hat) = 1.0
mean(square_error_mu_hat < square_error_mu_tilde) = 0.0
mean(square_error_mu_check < square_error_mu_hat) = 1.0
mean(square_error_mu_hat < square_error_mu_check) = 0.0
mean(square_error_mu_check < square_error_mu_tilde) = 0.49088
mean(square_error_mu_tilde < square_error_mu_check) = 0.0

```

4.2 雑多な場合

```

In [7]: 1 mu0 = rand(100)
        2 se_hat, se_tilde, se_check = sim_stein(mu0 = mu0);

mean(square_error_mu_hat) = 99.95992698307617
mean(square_error_mu_tilde) = 28.18277951780004
mean(square_error_mu_check) = 28.139705090076866
mean(square_error_mu_tilde .< square_error_mu_hat) = 0.99998
mean(square_error_mu_hat .< square_error_mu_tilde) = 2.0e-5
mean(square_error_mu_check .< square_error_mu_hat) = 1.0
mean(square_error_mu_hat .< square_error_mu_check) = 0.0
mean(square_error_mu_check .< square_error_mu_tilde) = 0.01274
mean(square_error_mu_tilde .< square_error_mu_check) = 0.0

```

```

In [8]: 1 mu0 = collect(range(-1, 1, length=100))
        2 se_hat, se_tilde, se_check = sim_stein(mu0 = mu0);

mean(square_error_mu_hat) = 100.00394342840605
mean(square_error_mu_tilde) = 26.971349342291784
mean(square_error_mu_check) = 26.914545491835018
mean(square_error_mu_tilde .< square_error_mu_hat) = 0.99996
mean(square_error_mu_hat .< square_error_mu_tilde) = 4.0e-5
mean(square_error_mu_check .< square_error_mu_hat) = 1.0
mean(square_error_mu_hat .< square_error_mu_check) = 0.0
mean(square_error_mu_check .< square_error_mu_tilde) = 0.01705
mean(square_error_mu_tilde .< square_error_mu_check) = 0.0

```

```

In [9]: 1 mu0 = randn(100)
        2 se_hat, se_tilde, se_check = sim_stein(mu0 = mu0);

mean(square_error_mu_hat) = 99.99448668844144
mean(square_error_mu_tilde) = 50.294799198411965
mean(square_error_mu_check) = 50.294799198411965
mean(square_error_mu_tilde .< square_error_mu_hat) = 0.99964
mean(square_error_mu_hat .< square_error_mu_tilde) = 0.00036
mean(square_error_mu_check .< square_error_mu_hat) = 0.99964
mean(square_error_mu_hat .< square_error_mu_check) = 0.00036
mean(square_error_mu_check .< square_error_mu_tilde) = 0.0
mean(square_error_mu_tilde .< square_error_mu_check) = 0.0

```

```

In [10]: 1 mu0 = randn(1000)
         2 se_hat, se_tilde, se_check = sim_stein(mu0 = mu0);

mean(square_error_mu_hat) = 1000.1038280762532
mean(square_error_mu_tilde) = 485.10303941793546
mean(square_error_mu_check) = 485.10303941793546
mean(square_error_mu_tilde .< square_error_mu_hat) = 1.0
mean(square_error_mu_hat .< square_error_mu_tilde) = 0.0
mean(square_error_mu_check .< square_error_mu_hat) = 1.0
mean(square_error_mu_hat .< square_error_mu_check) = 0.0
mean(square_error_mu_check .< square_error_mu_tilde) = 0.0
mean(square_error_mu_tilde .< square_error_mu_check) = 0.0

```

```

In [11]: 1 mu0 = 10 .+ randn(10)
         2 se_hat, se_tilde, se_check = sim_stein(mu0 = mu0);

mean(square_error_mu_hat) = 10.014952274038158
mean(square_error_mu_tilde) = 9.958338280423055
mean(square_error_mu_check) = 9.958338280423055
mean(square_error_mu_tilde .< square_error_mu_hat) = 0.55892
mean(square_error_mu_hat .< square_error_mu_tilde) = 0.44108
mean(square_error_mu_check .< square_error_mu_hat) = 0.55892
mean(square_error_mu_hat .< square_error_mu_check) = 0.44108
mean(square_error_mu_check .< square_error_mu_tilde) = 0.0
mean(square_error_mu_tilde .< square_error_mu_check) = 0.0

```

```
In [12]: 1 mu0 = 10 .+ randn(100)
2 se_hat, se_tilde, se_check = sim_stein(mu0 = mu0);

mean(square_error_mu_hat) = 99.94468710829554
mean(square_error_mu_tilde) = 99.0242590511661
mean(square_error_mu_check) = 99.0242590511661
mean(square_error_mu_tilde .< square_error_mu_hat) = 0.68881
mean(square_error_mu_hat .< square_error_mu_tilde) = 0.31119
mean(square_error_mu_check .< square_error_mu_hat) = 0.68881
mean(square_error_mu_hat .< square_error_mu_check) = 0.31119
mean(square_error_mu_check .< square_error_mu_tilde) = 0.0
mean(square_error_mu_tilde .< square_error_mu_check) = 0.0
```

```
In [13]: 1 mu0 = 10 .+ randn(1000)
2 se_hat, se_tilde, se_check = sim_stein(mu0 = mu0);

mean(square_error_mu_hat) = 1000.2443723311364
mean(square_error_mu_tilde) = 990.5328474630195
mean(square_error_mu_check) = 990.5328474630195
mean(square_error_mu_tilde .< square_error_mu_hat) = 0.9396
mean(square_error_mu_hat .< square_error_mu_tilde) = 0.0604
mean(square_error_mu_check .< square_error_mu_hat) = 0.9396
mean(square_error_mu_hat .< square_error_mu_check) = 0.0604
mean(square_error_mu_check .< square_error_mu_tilde) = 0.0
mean(square_error_mu_tilde .< square_error_mu_check) = 0.0
```

```
In [ ]: 1
```