STAT 5361: Statistical Computing, Spring 2018

Jun Yan

Department of Statistics
University of Connecticut

August 27, 2018

Outline I

Outline II

Outline III

- Penalized Least Squares
- Introduction to Lasso
- Orthogonal Design and Geometry
- Computation by Coordinate Descent
- Penalty Parameter Selection
- Least Angle Regression

Lecture 1: Non-Stochastic Optimization (Newton-like Methods)

Non-stochastic Optimization

What we will learn:

- Some basics about statistical inference
- Univariate problems
  - Newton's method
  - Fisher scoring
  - Secant method
  - Fixed-point method
  - Connections of the above
- Multivariate problems
  - Newton's method
  - Newton-like methods

Background: likelihood inference

- Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ be an i.i.d. sample from

$$f(\boldsymbol{x} \,|\, \boldsymbol{\theta}^*),$$

  with the true parameter value $\boldsymbol{\theta}^*$ being unknown.
- The likelihood function is

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} f(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}),$$

- The *maximum likelihood estimator* (MLE) of the parameter value is the maximizer of $L(\boldsymbol{\theta})$. MLE has the invariate property.
- Usually it is easier to work with the *log likelihood function*

$$l(\boldsymbol{\theta}) = \log L(\boldsymbol{\theta}).$$

- Typically, maximization of $l(\boldsymbol{\theta})$ is done by solving

$$l'(\boldsymbol{\theta}) = 0.$$

  - $l'(\boldsymbol{\theta})$ is called the *score function*.
  - For each possible parameter value $\boldsymbol{\theta}$, $l(\boldsymbol{\theta})$ is a random variable, because it depends on the observed values of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$.

- For any $\boldsymbol{\theta}$,

$$\mathbb{E}_{\boldsymbol{\theta}} \left\{ l'(\boldsymbol{\theta}) \right\} = 0,$$
$$\mathbb{E}_{\boldsymbol{\theta}} \left\{ l'(\boldsymbol{\theta}) l'(\boldsymbol{\theta})^T \right\} = -\mathbb{E}_{\boldsymbol{\theta}} \left\{ l''(\boldsymbol{\theta}) \right\}.$$

  where $\mathbb{E}_{\boldsymbol{\theta}}$ is the expectation with respect to $f(\boldsymbol{x} \,|\, \boldsymbol{\theta})$. The equality holds true under mild regularity conditions.

- $$I(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\theta}} \left\{ l'(\boldsymbol{\theta}) l'(\boldsymbol{\theta})^T \right\}$$

  is known as the *Fisher information*.

- The importance of the Fisher information $I(\boldsymbol{\theta})$ is that it sets the limit on how accurate an unbiased estimate of $\boldsymbol{\theta}$ can be.
- As $n \to \infty$, the asymptotic distribution of $\sqrt{n}(\widehat{\boldsymbol{\theta}}_{\mathrm{MLE}} - \boldsymbol{\theta}^*)$ is $N_p(\mathbf{0}, nI(\boldsymbol{\theta}^*)^{-1})$. Since θ^* is unknown, the asymptotic covariance matrix $I(\boldsymbol{\theta}^*)^{-1})$ needs to be estimated.
  - If $\dim(\boldsymbol{\theta}) = 1$, $I(\boldsymbol{\theta})$ is a nonnegative number.
  - If $\dim(\boldsymbol{\theta}) > 1$, $I(\boldsymbol{\theta})$ is a nonnegative definite matrix.
- The *observed Fisher information* is

$$-l''(\boldsymbol{\theta}).$$

The expected Fisher information $I(\boldsymbol{\theta})$ may not be easily computed. The observed Fisher information is a good approximation to $I(\boldsymbol{\theta})$ that improves as n gets bigger.

Besides MLEs, there are other likelihoods for parameter estimation. Suppose $\boldsymbol{\theta}$ has two parts: $\boldsymbol{\theta} = (\boldsymbol{\phi}, \boldsymbol{\mu})$ and we are only interested in $\boldsymbol{\phi}$. The *profile likelihood* for $\boldsymbol{\phi}$ is

$$L(\boldsymbol{\phi}) := \max_{\boldsymbol{\mu}} L(\boldsymbol{\phi}, \boldsymbol{\mu}).$$

- $\boldsymbol{\mu}$ is the nuisance parameter.
- Need to maximize $L(\boldsymbol{\phi}, \boldsymbol{\mu})$ for every fixed $\boldsymbol{\phi}$. Also need to maximize $L(\boldsymbol{\phi})$.

A general method

Suppose $g(\boldsymbol{x})$ is a differentiable function, where

$$\boldsymbol{x} = (x_1, \ldots, x_n).$$

To find its maximum (or minimum), one method is to solve the equation

$$g'(\boldsymbol{x}) = 0$$

$$\text{where} \quad g'(\boldsymbol{x}) = \left(\frac{\partial g}{\partial x_1}, \ldots, \frac{\partial g}{\partial x_n} \right)^T.$$

Then the maximization is equivalent to solving

$$f(\boldsymbol{x}) = 0,$$

where $f = g'$.
For maximum likelihood estimation, g is the log likelihood function l, and $\boldsymbol{x}$ is the corresponding parameter vector $\boldsymbol{\theta}$.

Univariate Problems

Optimization: Univariate case

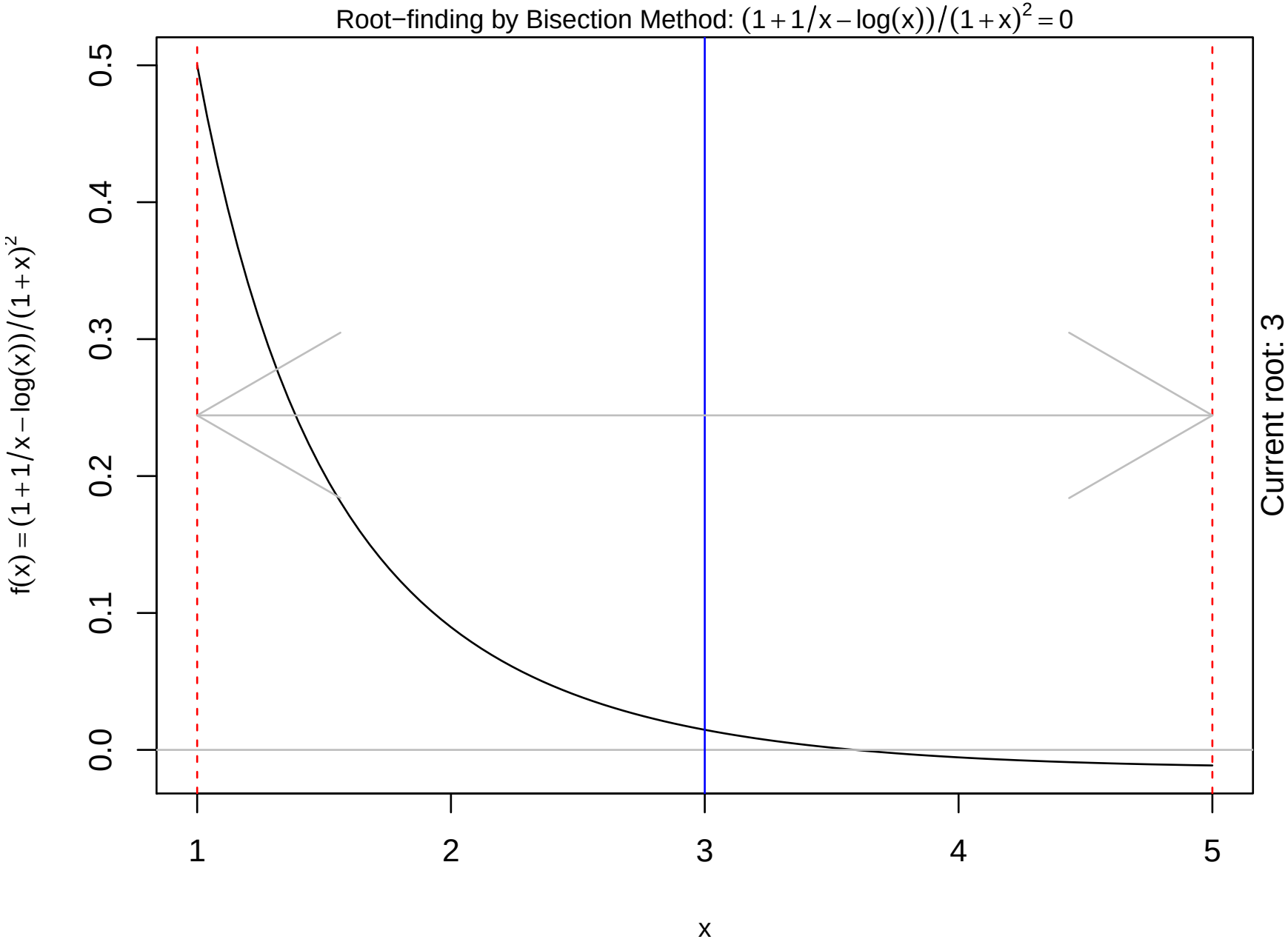
- Goal: optimize a real-valued function g with respect to its argument, a p dimensional vector $\boldsymbol{x}$. We will first consider the case $p = 1$.
- We will limit consideration mainly to smooth and differentiable functions.
- Root finding methods. Solving unconstrained nonlinear equations.
- Iterative algorithms: starting value, updating equation, stopping rule/convergence criterion.

Using bisection method to maximize

$$g(x) = \frac{\log x}{1 + x}.$$

or to sovle

$$f(x) = g'(x) = \frac{1 + \frac{1}{x} - \log x}{(1 + x)^2} = 0.$$


Root−finding by Bisection Method: $(1+1/x-\log(x))/(1+x)^2=0$
$f(x)=(1+1/x-\log(x))/(1+x)^2$
0.0
0.1
0.2
0.3
0.4
0.5
1
2
3
4
5
x
Current root: 3

Newton's method

This is a fast approach to finding roots of a differentiable function $f(x)$. First, set initial value x_0. Then for $t = 0, 1, \ldots$, compute

$$x_{t+1} = x_t + h_t, \text{ with } h_t = -\frac{f(x_t)}{f'(x_t)}.$$

Continue the iterations until x_t converges.

- Also known as Newton-Raphson iteration.
- Need to specify x_0.
- If $f(x) = 0$ has multiple solutions, the end result depends on x_0.

Newton's method require computing the derivative of a function. Algorithms are available to find root(s) of a univariate function without having to compute its derivative. For example, in R, one can use `uniroot`. Newton's method can be applied to optimize g by applying to

$$f = g'.$$

- Both g' (*gradient*) and g'' (*Hessian*) are needed.
- Many variants of Newton's method avoid the computation of the Hessian, which can be difficult especially for multivariate functions.

To maximize

$$g(x) = \frac{\log x}{1+x},$$

first find

$$f(x) = g'(x) = \frac{1 + \frac{1}{x} - \log x}{(1+x)^2},$$

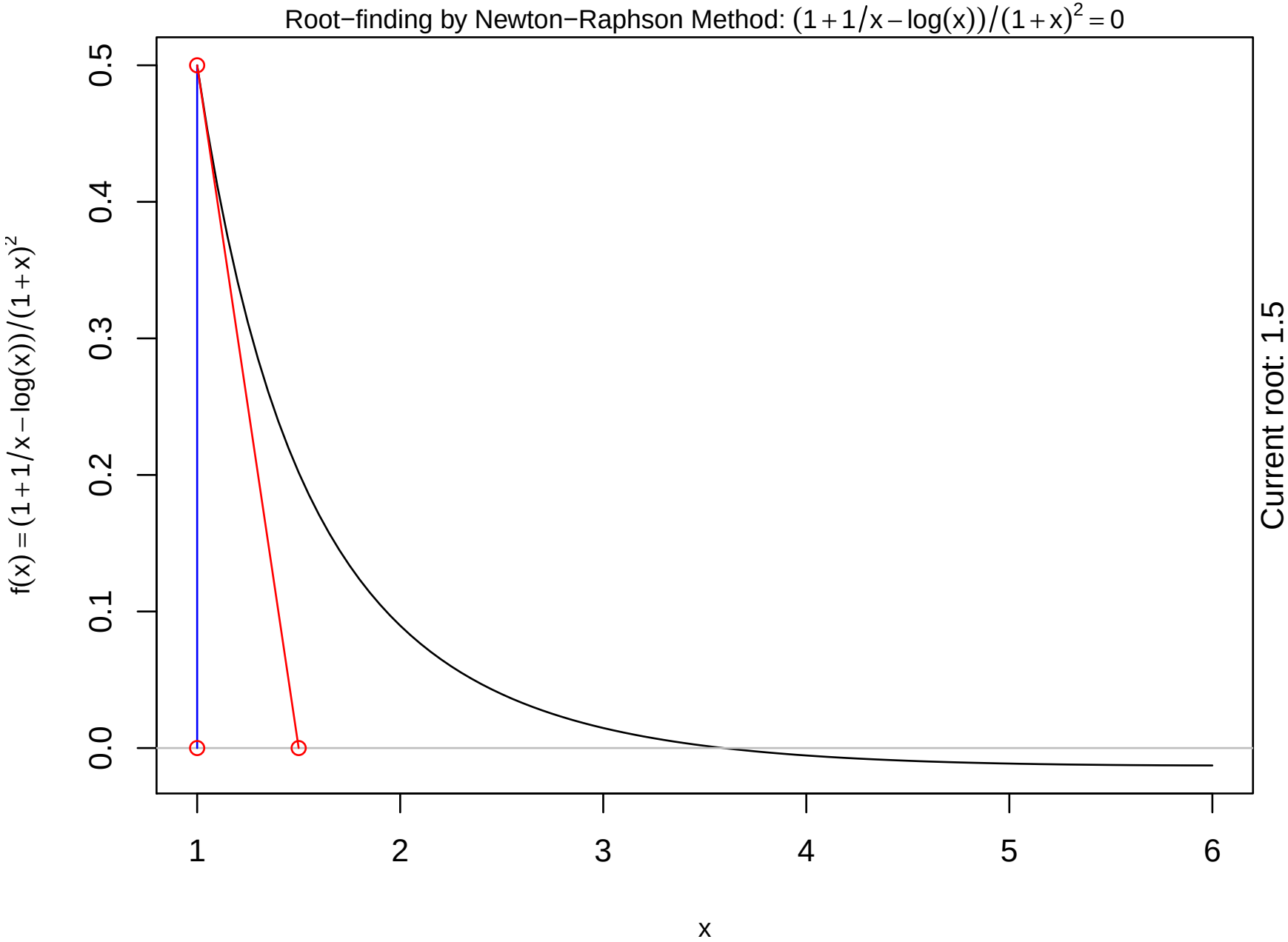
$$f'(x) = g''(x) = -\frac{3 + 4/x + 1/x^2 - 2\log x}{(1+x)^3}.$$

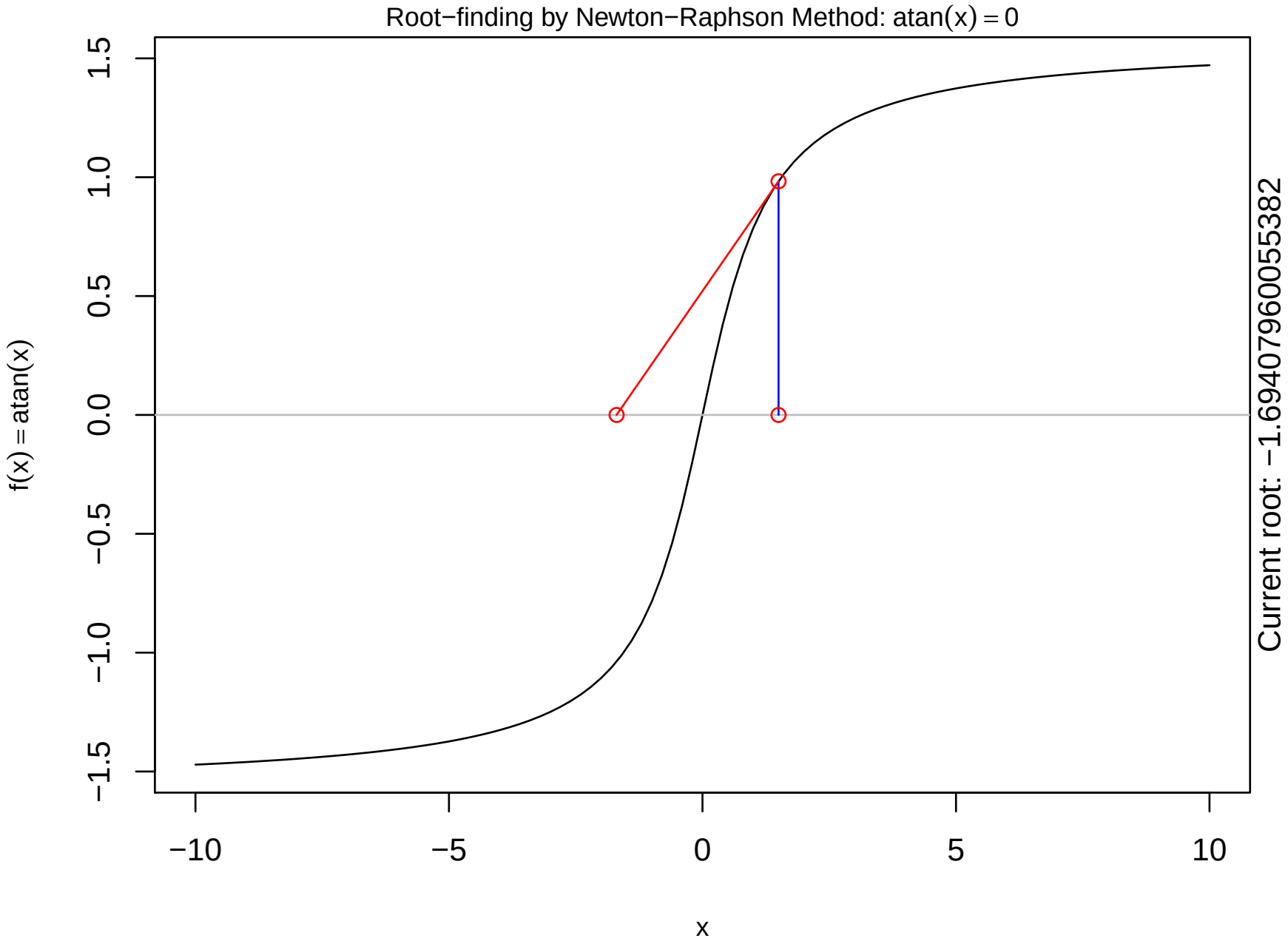
So in the Newton's method,

$$h_t = \frac{(x_t+1)(1+1/x_t - \log x_t)}{3 + 4/x_t + 1/x_t^2 - 2\log x_t}.$$

Note that to solve $f(x) = 0$, one can instead solve $1 + 1/x - \log x = 0$. Treat this as a new f function. Then in the Newton's method,

$$h_t = x_t - \frac{x_t^2 \log x_t}{1+x_t} \quad \Longrightarrow \quad x_{t+1} = 2x_t - \frac{x_t^2 \log x_t}{1+x_t}.$$


Root−finding by Newton−Raphson Method: $(1+1/x-\log(x))/(1+x)^2=0$
$f(x)=(1+1/x-\log(x))/(1+x)^2$
0.0
0.1
0.2
0.3
0.4
0.5
1
2
3
4
5
6
x
Current root: 1.5


Root−finding by Newton−Raphson Method: atan(x) = 0
f(x) = atan(x)
x
Current root: −1.69407960055382
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−10
−5
0
5
10

To maximize the log likelihood $l(\theta)$, Newton's method computes

$$\theta_{t+1} = \theta_t - \frac{l'(\theta_t)}{l''(\theta_t)}$$

Example: consider $x_1, \ldots, x_n \sim$ i.i.d. $N(\mu, \sigma^2)$.

Example: consider the model on shift

$$p(x \mid \theta) = p(x - \theta).$$

Given observations $x_1, \ldots, x_n$ i.i.d. $\sim p(x \mid \theta)$

$$l(\theta) = \sum_{i=1}^{n} \log p(x_i - \theta)$$

$$l'(\theta) = -\sum_{i=1}^{n} \frac{p'(x_i - \theta)}{p(x_i - \theta)}$$

$$l''(\theta) = \sum_{i=1}^{n} \frac{p''(x_i - \theta)}{p(x_i - \theta)} - \sum_{i=1}^{n} \left\{ \frac{p'(x_i - \theta)}{p(x_i - \theta)} \right\}^2.$$

Note that we need to update θ in the iterations, not $x_1, \ldots, x_n$.

In R, to *minimize* a function, one can use

```
z = nlminb(x0, g, gr.g, hess.g)
```

here `x0` is the initial value, `g` is the function being minimized, `gr.g` its gradient and `hess.g` its Hessian. (Unconstrained and box-constrained optimization using PORT routines).
In the above function, the derivatives g' and g'' have to be analytically calculated. One can also use

```
z = nlminb(x0, g, gr.g)
```

without inputing the analytic expression of g'', or, even simpler,

```
z = nlminb(x0, g)
```

without inputing the analytic expressions of either derivatives. In these cases, numerical approximations of derivatives will be computed during the iterations.
Other functions for optimization in R include `optim`, `optimize`, `nlm` and `constrOptim`.

For profile likelihood

$$L(\phi) = \max_{\mu} L(\phi, \mu)$$

we need to optimize $L(\phi, \mu)$ for each given ϕ.
In R, in order to get $\min_x g(x, y)$ for each fixed y, one can do the following. First, define `g(x,y)`, `gr.g(x,y)`, etc. Then call, say,

```
nlminb(x0, g, gr.g, hess.g, y=1),   or
nlminb(x0, g, gr.g, y=.3),   or
nlminb(x0, g, y=-1)
```

If one only wants minimization for $x \in [a, b]$, use,

```
nlminb(x0, ..., lower=a, upper=b)
```

Fisher scoring

This is a variant of Newton's method specific for MLE. Recall $-l''(\theta)$ is the observed Fisher information at θ. To maximize $l(\theta)$, an alternative is to replace $-l''(\theta_t)$ with $I(\theta_t)$ to get

$$\theta_{t+1} = \theta_t + \frac{l'(\theta_t)}{I(\theta_t)}.$$

To *minimize* $-l(\theta)$, one still can use

```
z = nlminb(x0, f, grf, fs)
```

where `f` is $-l(\theta)$, `grf` is $-l'(\theta)$, and `fs` is $I(\theta)$ instead of $-l''(\theta)$. Generally, use Fisher scoring in the beginning to make rapid improvements, and Newton's method for refinement near the end.

Continuing the example on $p(x \mid \theta) = p(x - \theta)$, to use Fisher scoring, we need to compute

$$I(\theta) = -\mathbb{E}_\theta[l''(\theta)].$$

We already know

$$l''(\theta) = \sum_{i=1}^{n} \frac{p''(x_i - \theta)}{p(x_i - \theta)} - \sum_{i=1}^{n} \left\{ \frac{p'(x_i - \theta)}{p(x_i - \theta)} \right\}^2.$$

Therefore

$$I(\theta) = -n\mathbb{E}_\theta \left[\frac{p''(X - \theta)}{p(X - \theta)} - \left\{ \frac{p'(X - \theta)}{p(X - \theta)} \right\}^2 \right].$$

Since under parameter θ, X has density $p(x-\theta)$, the last $\mathbb{E}_\theta[...]$ equals

$$\begin{aligned}
&\int \left[\frac{p''(x-\theta)}{p(x-\theta)} - \left\{ \frac{p'(x-\theta)}{p(x-\theta)} \right\}^2 \right] p(x-\theta)\, \mathrm{d}x \\
&= \int p''(x-\theta)\, \mathrm{d}x - \int \frac{[p'(x-\theta)]^2}{p(x-\theta)}\, \mathrm{d}x \\
&= \frac{\mathrm{d}^2}{\mathrm{d}\theta^2} \int p(x-\theta)\, \mathrm{d}x - \int \frac{\{p'(x)\}^2}{p(x)}\, \mathrm{d}x \\
&= \frac{\mathrm{d}^2}{\mathrm{d}\theta^2} 1 - \int \frac{\{p'(x)\}^2}{p(x)}\, \mathrm{d}x = -\int \frac{\{p'(x)\}^2}{p(x)}\, \mathrm{d}x.
\end{aligned}$$

Therefore,

$$I(\theta) = n \int \frac{\{p'(x)\}^2}{p(x)}\, \mathrm{d}x.$$

So, in this case, Fisher information is a constant.

Secant method

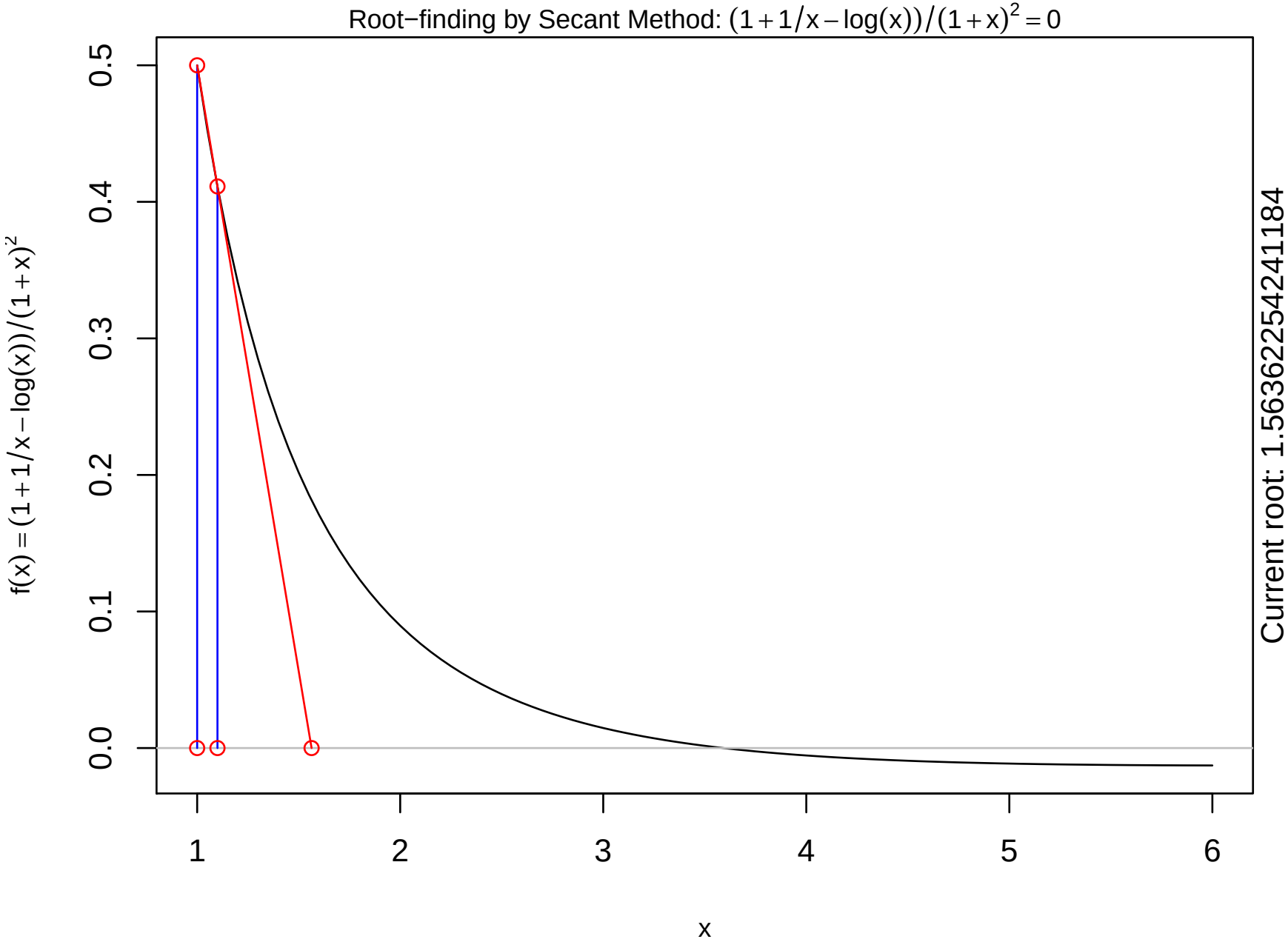
Approximating $f'(x_t)$ by

$$\frac{f(x_t) - f(x_{t-1})}{x_t - x_{t-1}},$$

the Newton's method turns into the secant method

$$x_{t+1} = x_t - \frac{f(x_t)(x_t - x_{t-1})}{f(x_t) - f(x_{t-1})}.$$

- Need to specify x_0 and x_1.


Root−finding by Secant Method: $(1+1/x-\log(x))/(1+x)^2=0$
$f(x)=(1+1/x-\log(x))/(1+x)^2$
0.0
0.1
0.2
0.3
0.4
0.5
1
2
3
4
5
6
x
Current root: 1.563622542411184

Fixed point iteration

Compute

$$x_{t+1} = x_t + \alpha f(x_t), \quad t = 0, 1, \ldots,$$

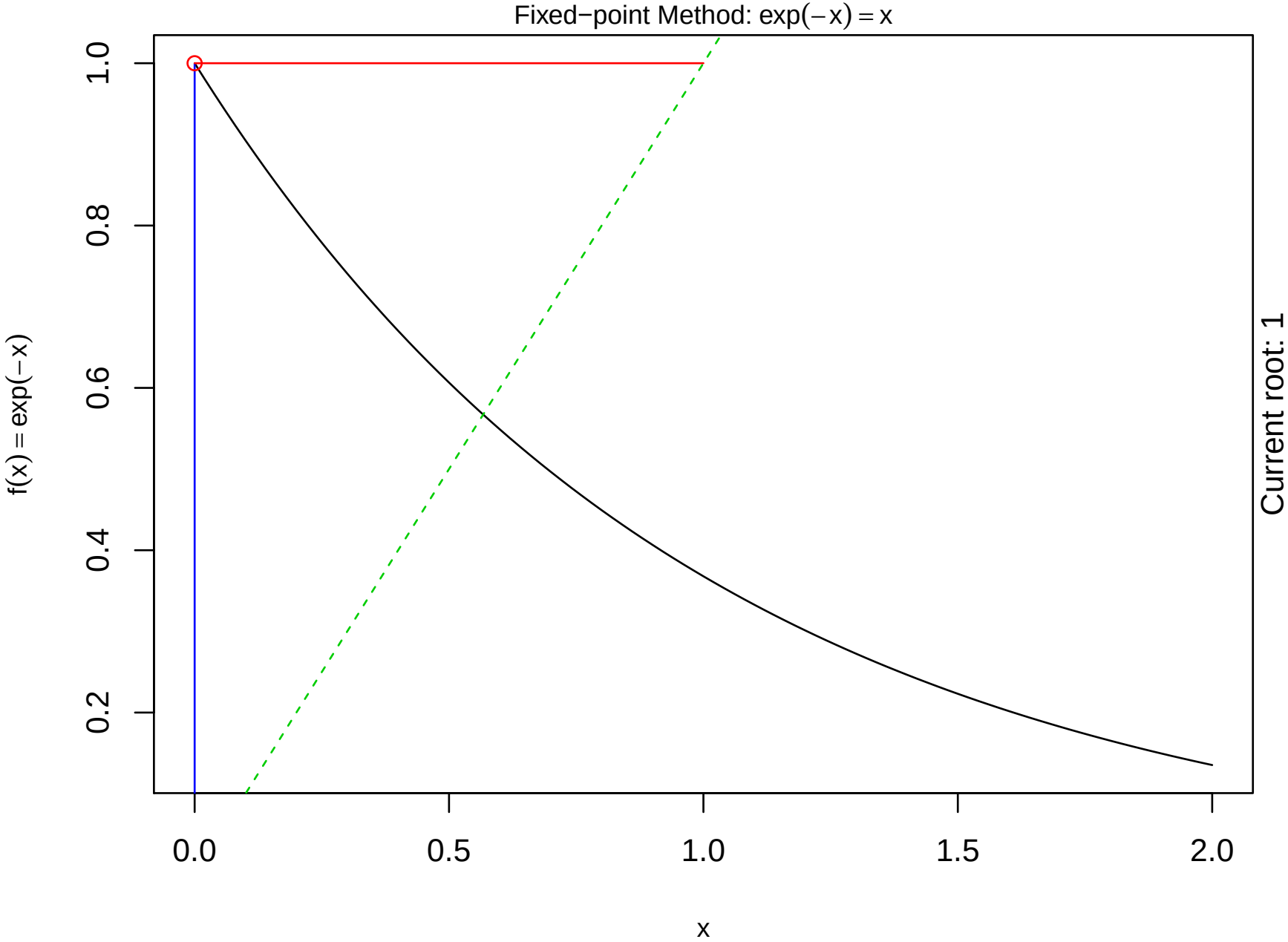
where $\alpha \neq 0$ is a tuning parameter so that

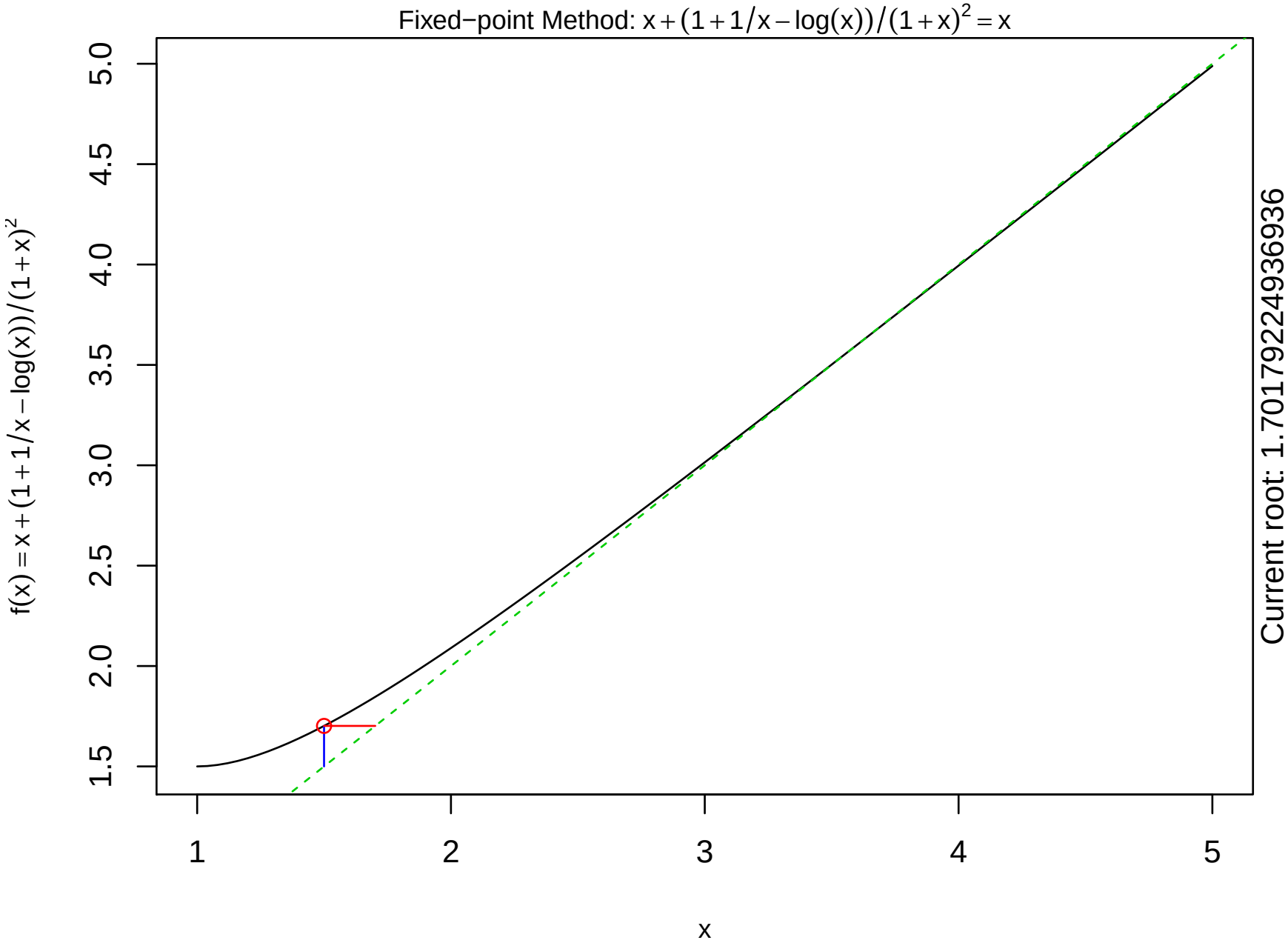
$$|1 + \alpha f'(x)| \leq \lambda < 1, \text{ all } x$$

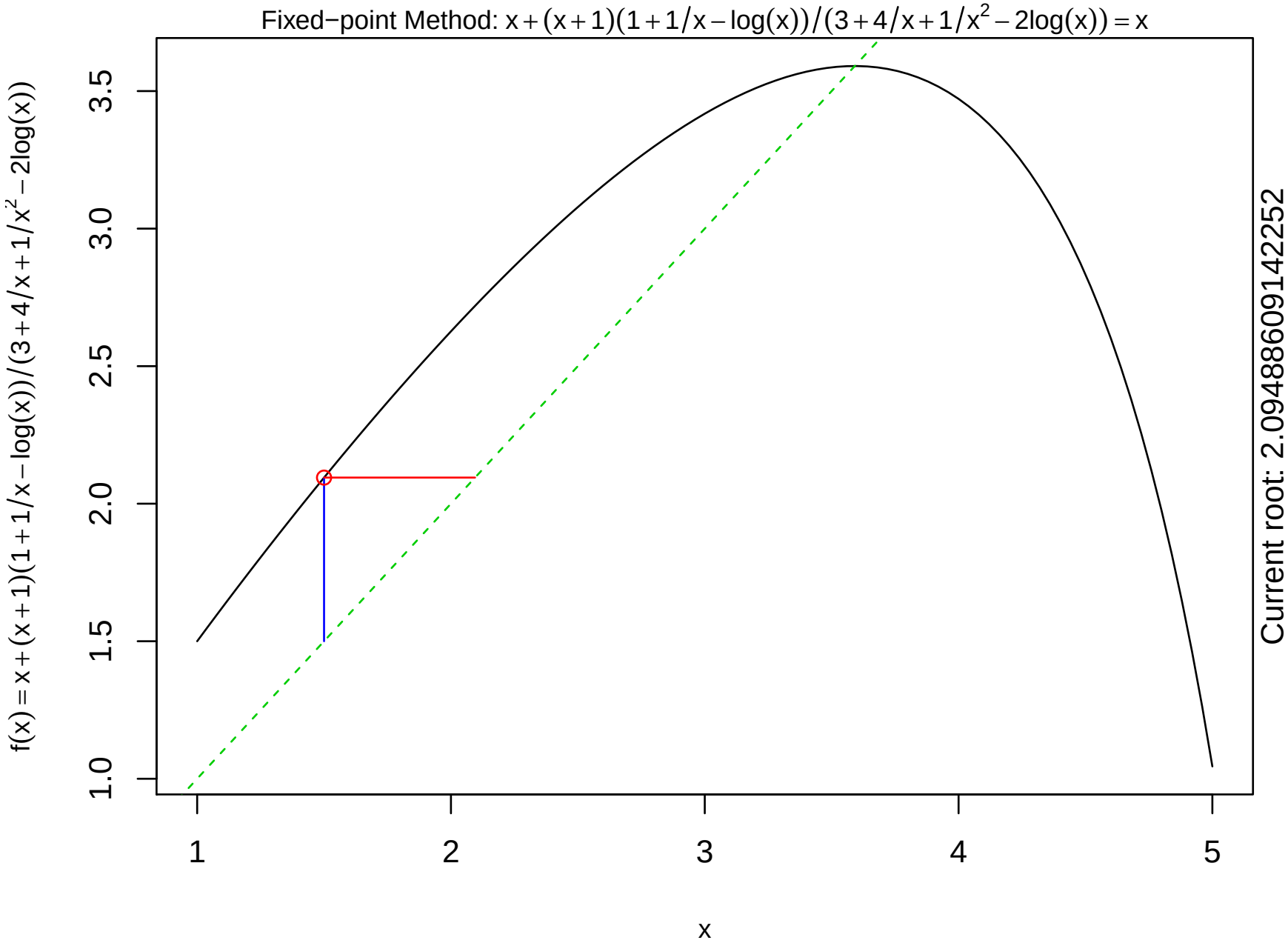
or more generally,

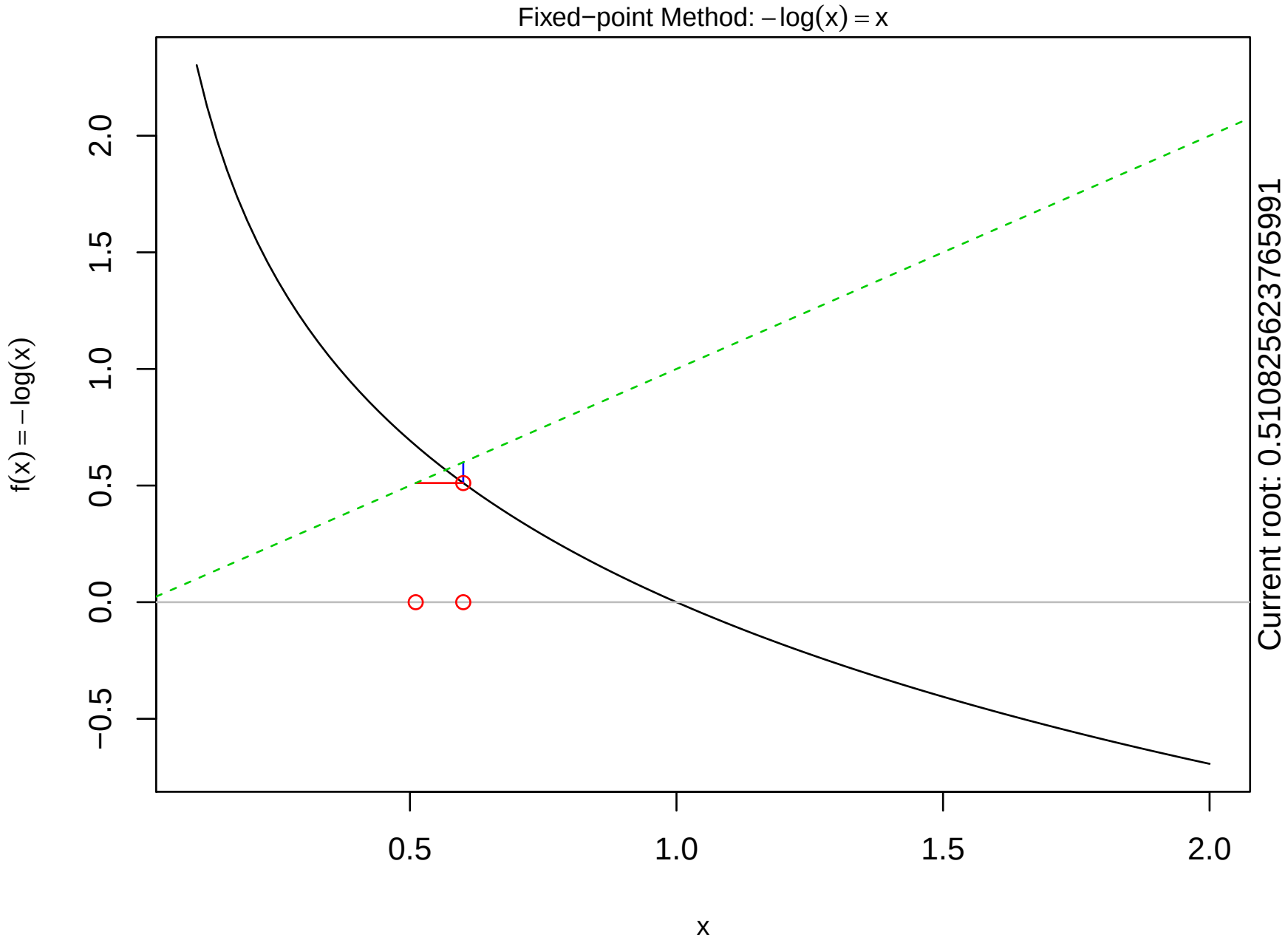
$$|x - y + \alpha[f(x) - f(y)]| \leq \lambda |x - y|, \quad \text{any } x, y$$

with $0 \leq \lambda < 1$ a constant, i.e., $F(x) = x + \alpha f(x)$ is a *contraction*.
If such α exists, the speed of convergence depends on λ. The smaller λ is, the faster the convergence.


Fixed−point Method: exp(−x) = x
f(x) = exp(−x)
1.0
0.8
0.6
0.4
0.2
0.0
0.5
1.0
1.5
2.0
x
Current root: 1


Fixed−point Method: x + (1 + 1/x − log(x))/(1 + x)^2 = x
f(x) = x + (1 + 1/x − log(x))/(1 + x)^2
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
1
2
3
4
5
x
Current root: 1.70179224936936


Fixed−point Method: x + (x + 1)(1 + 1/x − log(x))/(3 + 4/x + 1/x^2 − 2log(x)) = x
f(x) = x + (x + 1)(1 + 1/x − log(x))/(3 + 4/x + 1/x^2 − 2log(x))
1.0
1.5
2.0
2.5
3.0
3.5
1
2
3
4
5
x
Current root: 2.094886091422 52


Fixed−point Method: −log(x) = x
f(x) = −log(x)
x
Current root: 0.510825623765991
1
0.5
1.0
1.5
2.0
−0.5
0.0

Some Details on Fixed point iteration

Definition: A fixed point of a function is a point whose evaluation by that function equals to itself, i.e., $x = G(x)$.
Fixed point iteration: the natural way of hunting for a fixed point is to use $x_{t+1} = G(x_t)$.
Definition: A function G is contractive on $[a, b]$ if

(1). $G(x) \in [a, b]$ whenever $x \in [a, b]$,
(2). $|G(x_1) - G(x_2)| \leq \lambda |x_1 - x_2|$ for all $x_1, x_2 \in [a, b]$ and some $\lambda \in [0, 1)$.

Theorem: if G is contractive on $[a, b]$, then there is a unique fixed point $x^* \in [a, b]$, and the fixed point iteration convergence to it when starting inside the interval.
Convergence:
$|x_{t+1} - x_t| = |G(x_t) - G(x_{t-1})| \leq \lambda |x_t - x_{t-1}| \leq \lambda^t |x_1 - x_0| \to 0$, as $t \to \infty$. It follows that $\{x_t\}$ convergent to a limit x^*.

Connection between fixed-point method and Newton methods

Root-finding problems using fixed point iteration: for solving $f(x) = 0$, we can simply let $G(x) = x + \alpha f(x)$, where $\alpha \neq 0$ is a constant.
Required Lipschitz condition: $|x - y + \alpha[f(x) - f(y)]| \leq \lambda |x - y|$, for some $\lambda \in [0, 1)$ and for all $x, y \in [a, b]$. This holds if $|G'(x)| \leq \lambda$ for some $\lambda \in [0, 1)$ and for all $x \in [a, b]$, i.e., $|1 + \alpha f'(x)| \leq \lambda$. (use mean value theorem.)
Newton methods: $G(x) = x - f(x)/f'(x)$. So it is as if α_t is chosen adaptively as $\alpha_t = -1/f'(x_t)$. This leads to a faster convergence order (quadratic).

Convergence Order

Define $\varepsilon_t = x_t - x^*$. A method has convergence order β if

$$\lim_{t\to\infty} \varepsilon_t = 0, \qquad \lim_{t\to\infty} \frac{|\varepsilon_{t+1}|}{|\varepsilon_t|^\beta} = c,$$

for some constant $c \neq 0$ and $\beta > 0$.

- For Newton's method, $\beta = 2$. (If it converges.)
- For Secant method, $\beta \approx 1.62$.
- For fixed point iteration, $\beta = 1$.

Convergence Order

For Newton's method: suppose f has two continuous derivatives and $f'(x^*) \neq 0$. There then exists a neighborhood of x^* within which $f'(x) \neq 0$ for all x. By Taylor expansion,

$$0 = f(x^*) = f(x_t) + f'(x_t)(x^* - x_t) + \frac{1}{2}f''(q)(x^* - x_t)^2,$$

for some q between x^* and x_t. Rearranging terms, we find that

$$\frac{\varepsilon_{t+1}}{\varepsilon_t^2} = \frac{f''(q)}{2f'(x_t)} \to \frac{f''(x^*)}{2f'(x^*)}.$$

Convergence Order

For fixed point iteration: let G be a continuous function on the closed interval $[a, b]$ with $G : [a, b] \to [a, b]$ and suppose that G' is continuous on the open interval (a, b) with $|G'(x)| \leq k < 1$ for all $x \in (a, b)$. If $g'(x^*) \neq 0$, then for any $x_0 \in [a, b]$, the fixed point iteration converges linearly to the fixed point x^*. This is because

$$|x_{t+1} - x^*| = |G(x_t) - G(x^*)| = |G'(q)||x_t - x^*|,$$

for some q between x_t and x^*. This implies that

$$\frac{|\varepsilon_{t+1}|}{|\varepsilon_t|} \to |G'(x^*)|.$$

Stopping Rules

- Absolute convergence criterion:

$$\|x_{t+1} - x_t\| < \epsilon,$$

  where ϵ is a chosen tolerance.

- Relative convergence criterion

$$\frac{\|x_{t+1} - x_t\|}{\|x_t\|} < \epsilon.$$

  If x_t close to zero,

$$\frac{\|x_{t+1} - x_t\|}{\|x_t\| + \epsilon} < \epsilon.$$

- Many other rules depending on the algorithm.

Multivariate Problems

Newton's method

Let g now be a function in $\boldsymbol{x} = (x_1, \ldots, x_p)^T$.
The generalization is straightforward: to maximize $g(\boldsymbol{x})$, set

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - [g''(\boldsymbol{x}_t)]^{-1} g'(\boldsymbol{x}_t).$$

- $g''(\boldsymbol{x})$ is a $p \times p$ matrix with the (i, j)th entry equal to

$$\frac{\partial^2 g(\boldsymbol{x})}{\partial x_i \partial x_j};$$

- $g'(\boldsymbol{x})$ is a $p \times 1$ vector with the ith entry equal to

$$\frac{\partial g(\boldsymbol{x})}{\partial x_i}.$$

Newton-like methods

For MLE, the generalization is straightforward. Simply use

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + I(\boldsymbol{\theta}_t)^{-1} l'(\boldsymbol{\theta}_t).$$

These methods in each iteration approximate $g''(\boldsymbol{x}_t)$ by some $p \times p$ matrix $\boldsymbol{M}_t = \boldsymbol{M}_t(\boldsymbol{x}_t)$ to get

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \boldsymbol{M}_t^{-1} g'(\boldsymbol{x}_t). \tag{1}$$

Of course, $\boldsymbol{M}_t$ should be easier to compute than $g''(\boldsymbol{x}_t)$.
Fisher scoring is a Newton-like method, because it uses

$$\boldsymbol{M}_t = -I(\boldsymbol{\theta}_t)$$

in place of $-l''(\boldsymbol{\theta}_t)$.

Steepest ascent methods

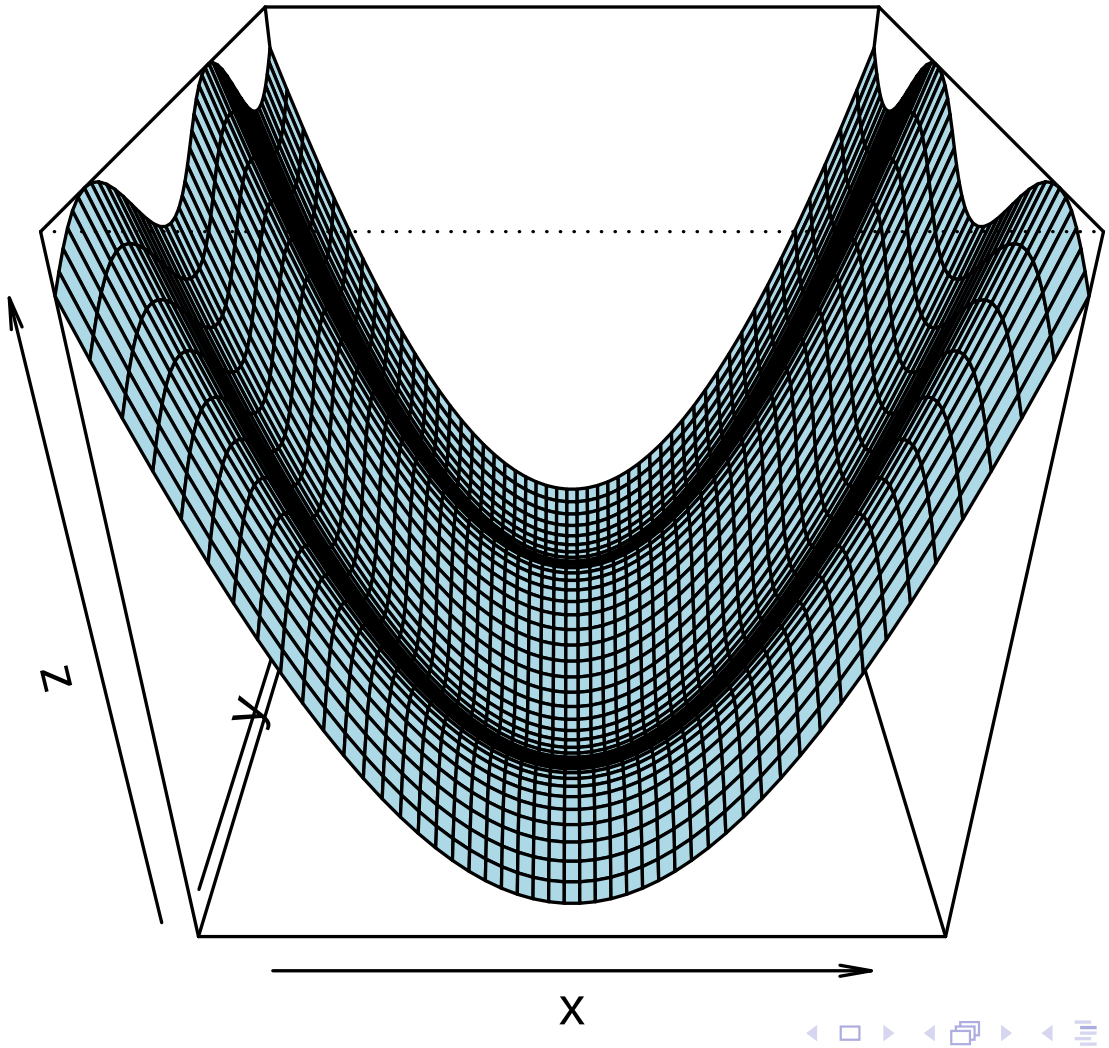
In (1), set

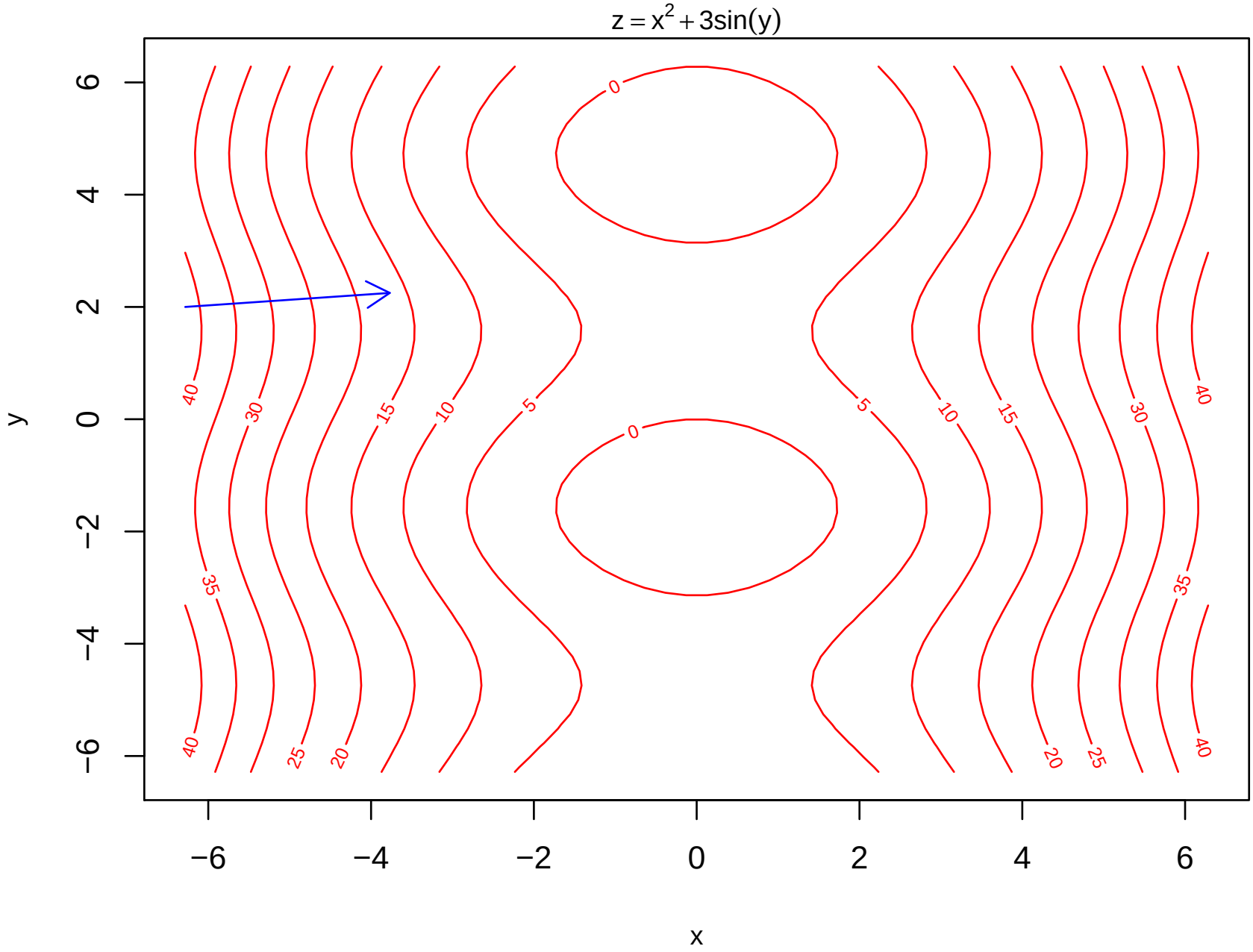
$$\boldsymbol{M}_t = -\alpha_t^{-1}\boldsymbol{I}_p,$$

so that

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \alpha_t g'(\boldsymbol{x}_t),$$

where $\alpha_t > 0$ is the step size at t which can shrink to ensure ascent. If at step t, the original step turns out to be downhill, the updating can backtrack by halving α_t.


Z
Y
X


$z = x^2 + 3\sin(y)$
y
x
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6
0
0
5
5
10
10
15
15
20
20
25
25
30
30
35
35
40
40
40
40

Discrete Newton and Fixed-point methods

In fixed-point methods, usually $\boldsymbol{M}_t$ is fixed to be $\boldsymbol{M}$, e.g., $g''(\boldsymbol{x}_0)$. This amounts to applying univariate scaled fixed-point iteration to each component.

In discrete Newton methods, $g''(\boldsymbol{x}_t)$ is approximated as follows. Compute matrix $\boldsymbol{M}_t$ with the $(i,j)^{\text{th}}$ entry equal to

$$\boldsymbol{M}_t(i,j) = \frac{g_i'(\boldsymbol{x}_t + h_t(i,j)\boldsymbol{e}_j) - g_i'(\boldsymbol{x}_t)}{h_t(i,j)}$$

for some constant $h_t(i,j) \neq 0$, where

$$g_i'(\boldsymbol{x}) = \frac{\partial g(\boldsymbol{x})}{\partial x_i}.$$

If $h_t(i,j)$ is small, then

$$\boldsymbol{M}_t(i,j) \approx \frac{\partial^2 g(\boldsymbol{x}_t)}{\partial x_i \partial x_j}$$

and so $\boldsymbol{M}_t$ approximates $g''(\boldsymbol{x}_t)$.

However, since $g''(\boldsymbol{x}_t)$ is symmetric, instead of using $\boldsymbol{M}_t$ directly, use the symmetric

$$(\boldsymbol{M}_t + \boldsymbol{M}_t^T)/2$$

as the approximation of $g''(\boldsymbol{x}_t)$.

- No need to calculate second order derivatives
- Inefficient: all p^2 entries have to be updated each time.

Quasi-Newton methods

These methods aim to achieve the following goals.

- Each step satisfies the secant condition

$$g'(\boldsymbol{x}_{t+1}) - g'(\boldsymbol{x}_t) = \boldsymbol{M}_{t+1}(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t).$$

- No need to compute second order derivatives.
- Maintain symmetry of $\boldsymbol{M}_t$.
- Aim to update $\boldsymbol{M}_t$ efficiently.

There is a unique rank-one method that satisfies the secant condition and maintains the symmetry of $\boldsymbol{M}_t$: after getting

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \boldsymbol{M}_t^{-1} g'(\boldsymbol{x}_t)$$

compute

$$\boldsymbol{z}_t = \boldsymbol{x}_{t+1} - \boldsymbol{x}_t, \quad \boldsymbol{y}_t = g'(\boldsymbol{x}_{t+1}) - g'(\boldsymbol{x}_t),$$
$$\boldsymbol{v}_t = \boldsymbol{y}_t - \boldsymbol{M}_t \boldsymbol{z}_t, \quad c_t = \frac{1}{\boldsymbol{v}_t^T \boldsymbol{z}_t}.$$

Then update

$$\boldsymbol{M}_{t+1} = \boldsymbol{M}_t + c_t \boldsymbol{v}_t \boldsymbol{v}_t^T.$$

Note $\boldsymbol{M}_{t+1} - \boldsymbol{M}_t$ is of rank one. We can verify the secant condition is satisfied by multiplying both sides by $\boldsymbol{z}_t$.

There are several rank-two method satisfying the secant condition and the symmetry requirement. The Broyden class updates $\boldsymbol{M}_t$ as follows. After getting $\boldsymbol{x}_{t+1}$ and $\boldsymbol{z}_t$, $\boldsymbol{y}_t$ as above, compute

$$\boldsymbol{d}_t = \frac{\boldsymbol{y}_t}{\boldsymbol{z}_t^T \boldsymbol{y}_t} - \frac{\boldsymbol{M}_t \boldsymbol{z}_t}{\boldsymbol{z}_t^T \boldsymbol{M}_t \boldsymbol{z}_t}.$$

Then update

$$\boldsymbol{M}_{t+1} = \boldsymbol{M}_t - \frac{\boldsymbol{M}_t \boldsymbol{z}_t (\boldsymbol{M}_t \boldsymbol{z}_t)^T}{\boldsymbol{z}_t^T \boldsymbol{M}_t \boldsymbol{z}_t} + \frac{\boldsymbol{y}_t \boldsymbol{y}_t^T}{\boldsymbol{z}_t^T \boldsymbol{y}_t} + \delta_t (\boldsymbol{z}_t^T \boldsymbol{M}_t \boldsymbol{z}_t) \boldsymbol{d}_t \boldsymbol{d}_t^T,$$

where δ_t is a parameter.
A popular method to solve unconstrained nonlinear optimization problems is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method, with $\delta_t \equiv 0$.
In R, `optim` can be called to minimize a function, using the BFGS method or its variant, the "L-BFGS-B" method. "L" stands for limited memory, and "B" stands for box constraint.

Approximating Hessian for MLE

For MLE, the Hessian $l''(\widehat{\boldsymbol{\theta}}_{\mathrm{MLE}})$ is critical because it provides estimates of standard error and covariance.

- Quasi-Newton methods may provide poor approximation because it is based on the idea of using poor approximation of the Hessian to find the root for $l'(\boldsymbol{\theta}) = 0$.
- Use the discrete multivariate Newton method with

$$\boldsymbol{M}_t(i,j) = \frac{l_i'(\boldsymbol{\theta}_t + h_{ij}\boldsymbol{e}_j) - l_i'(\boldsymbol{\theta}_t - h_{ij}\boldsymbol{e}_j)}{2h_{ij}}.$$

Gauss-Newton method

Example: nonlinear regression. In many cases, we want to maximize

$$g(\boldsymbol{\theta}) = -\sum_{i=1}^{n}(y_i - f_i(\boldsymbol{\theta}))^2,$$

where each $f_i(\boldsymbol{\theta})$ is differentiable. Recall that for linear regression:

$$y_i = \boldsymbol{x}_i^T\boldsymbol{\theta} + \varepsilon, \quad i = 1, \ldots, n$$

the LS estimator of $\boldsymbol{\theta}$ maximizes $g(\boldsymbol{\theta})$ with $f_i(\boldsymbol{\theta}) = \boldsymbol{x}_i^T\boldsymbol{\theta}$ and equals

$$\widehat{\boldsymbol{\theta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

where

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_n^T \end{pmatrix}, \quad \boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

The Gauss-Newton method applies a similar idea to the nonlinear case. Let $\boldsymbol{\theta}^*$ be the (unknown) maximizer of $g(\boldsymbol{\theta})$. Given any candidate $\boldsymbol{\theta}$, the function

$$h(\boldsymbol{u}) = -\sum_{i=1}^{n}(y_i - f_i(\boldsymbol{\theta} + \boldsymbol{u}))^2.$$

is maximized by $\boldsymbol{\beta} = \boldsymbol{\theta}^* - \boldsymbol{\theta}$. Of course, $\boldsymbol{\beta}$ is unknown. However, if $\boldsymbol{\theta}$ is near $\boldsymbol{\theta}^*$, then $\boldsymbol{\beta} \approx 0$, so by Taylor expansion of $h(\boldsymbol{u})$, it may be close to the maximizer of

$$-\sum_{i=1}^{n}(y_i - f_i(\boldsymbol{\theta}) - f_i'(\boldsymbol{\theta})^T\boldsymbol{u})^2.$$

Treat $y_i - f_i(\boldsymbol{\theta})$ the same way as y_i in the linear regression, and $f_i'(\boldsymbol{\theta})$ the same way as $\boldsymbol{x}_i$, then

$$\boldsymbol{\theta}^* - \boldsymbol{\theta} \approx (\boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{z}$$

where

$$\boldsymbol{z} = \boldsymbol{z}(\boldsymbol{\theta}) = \begin{pmatrix} y_1 - f_1(\boldsymbol{\theta}) \\ \vdots \\ y_n - f_n(\boldsymbol{\theta}) \end{pmatrix}, \quad \boldsymbol{A} = \boldsymbol{A}(\boldsymbol{\theta}) = \begin{pmatrix} f_1'(\boldsymbol{\theta})^T \\ \vdots \\ f_n'(\boldsymbol{\theta})^T \end{pmatrix}.$$

The update rule thus is

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + (\boldsymbol{A}_t^T \boldsymbol{A}_t)^{-1} \boldsymbol{A}_t^T \boldsymbol{z}_t,$$

where $\boldsymbol{z}_t = \boldsymbol{z}(\boldsymbol{\theta}_t)$ and $\boldsymbol{A}_t = \boldsymbol{A}(\boldsymbol{\theta}_t)$.
In R, the Gauss-Newton method is the default method of the function `nls`.

Lecture 2: Combinatorial Optimization

Combinatorial Optimization

Motivation:

- There are hard optimization problems for which most methods including what we have discussed so far are useless.
- These problems are usually combinatorial in nature. Maximization requires a discrete search of a very large space.
- Problem: maximize $f(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)$, where $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ and $\boldsymbol{\Theta}$ consists of N elements.
  - Each $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ is called a candidate solution.
  - Usually N is a very large number depending on problem size p.
  - The difficulty of a particular size-p problem can be characterized by the number of operations required to solve it in the worst case scenario using the best known algorithm.
  - Suppose a problem is $\mathcal{O}(p!)$. If it requires 1 minute to solve for $p = 20$, it would take 12.1 years for $p = 25$ and 207 million years for $p = 30$.

What can we do:

- We have to take some sacrifices: we will abandon the global algorithms and focus on algorithms that can find a good local solution.
- Heuristic strategies.

What we will learn:

- Local search methods
- Simulated annealing
- Genetic algorithms

Motivating Example: Subset regression

Suppose $x_1, \ldots, x_p$ are predictors that can be used to construct a linear model for Y. Each candidate model is

$$Y = \sum_{j=1}^{s} \beta_{i_j} x_{i_j} + \varepsilon,$$

where $1 \leq i_1 < i_2 < \ldots < i_s \leq p$, $s \geq 0$. The Akaike information criterion (AIC) for the candidate model is

$$\text{AIC} = n \log \frac{\text{RSS}}{n} + 2s$$

where n is the sample size and RSS is the residual sum of squares of model. The best model is the one that minimizes AIC.
To parameterize the model, set $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)$, such that $\theta_i = 1$ if x_i is included as a predictor, and 0 otherwise. The set $\boldsymbol{\Theta}$ of all possible $\boldsymbol{\theta}$ has 2^p values.

Local search

First, define a neighborhood $\mathcal{N}(\boldsymbol{\theta})$ for each $\boldsymbol{\theta}$, so that it

- contains candidate solutions that are "near" $\boldsymbol{\theta}$, and
- reduces the number of changes to the current $\boldsymbol{\theta}$.

For example, if $\boldsymbol{\Theta} = \{\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p) : \text{each } \theta_i = 0, 1\}$, one may define $\mathcal{N}(\boldsymbol{\theta})$ to be the set of $\boldsymbol{\theta}'$ which are different from $\boldsymbol{\theta}$ in at most one coordinate.

After neighborhoods are defined, at iteration t, choose $\boldsymbol{\theta}_{t+1}$ from $\mathcal{N}(\boldsymbol{\theta}_t)$ according to a certain rule. For example,

- steepest ascent: $\boldsymbol{\theta}_{t+1} = \arg\max_{\boldsymbol{\theta} \in \mathcal{N}(\boldsymbol{\theta}_t)} f(\boldsymbol{\theta})$;
- ascent algorithm: $\boldsymbol{\theta}_{t+1} \in \mathcal{N}(\boldsymbol{\theta}_t)$ uphill from $\boldsymbol{\theta}_t$, i.e.,

$$f(\boldsymbol{\theta}_{t+1}) \geq f(\boldsymbol{\theta}_t).$$

To avoid trapping into a local maximum, two often used variants are

- random-starts local search: repeatedly run an ascent algorithm to termination from a large number of randomly chosen starting points.
- steepest ascent/mildest descent: set to the least unfavorable $\boldsymbol{\theta}_{t+1} \in \mathcal{N}(\boldsymbol{\theta}_t)$;
  - if $\boldsymbol{\theta}_t$ is a local maximum $\boldsymbol{\theta}_{t+1}$ is the one with least decrease;
  - otherwise, $\boldsymbol{\theta}_{t+1}$ is the one with the largest increase.

To select a linear model to minimize AIC (or maximize $-$AIC),

- randomly select a set of predictors
- at iteration t, decrease the AIC by adding a predictor to the set of selected predictors or deleting a predictor that has been selected; continue the iterations until the AIC can not be decreased;
- repeat Steps 1 and 2 many times, and choose the set of predictors at termination that has the lowest AIC.

A salesman must visit each of p cities exactly once and return to his starting city, using the shortest total travel distance.
A candidate solution is

$$\boldsymbol{\theta} = (i_1, i_2, \ldots, i_p, i_1)$$

where $i_1, \ldots, i_p$ is a permutation of 1, ..., p. The objective function to be minimized is

$$f(\boldsymbol{\theta}) = d(i_1, i_2) + d(i_2, i_3) + \cdots + d(i_{p-1}, i_p) + d(i_p, i_1).$$

There are $(p-1)!/2$ all possible routes, since the point of origin and direction of travel are arbitrary. For a traveling plan to visit 20 cities, that amounts to more than 6×10^{16} possible routes.

One could define $\mathcal{N}(\boldsymbol{\theta})$ as the set of sequences that only differ from $\boldsymbol{\theta}$ at two entries. In other words, each sequence in $\mathcal{N}(\boldsymbol{\theta})$ is obtained by exchanging two entries of $\boldsymbol{\theta}$.
For example, if

$$\boldsymbol{\theta} = (1, 2, 5, 4, 6, 3, 1)$$

then

$$(1, 2, 3, 4, 6, 5, 1)$$

is a neighbor of $\boldsymbol{\theta}$, because it only has 3 and 5 exchanged; while

$$(1, 2, 4, 3, 6, 5, 1)$$

is not a neighbor of $\boldsymbol{\theta}$, because it has 5, 4, and 3 rotated.

Simulated annealing

In most cases, the simulated annealing algorithm can be thought of as a randomized local search algorithm. It uses a "temperature" parameter to control the randomness of the search. The algorithm starts with a high temperature and cools down gradually so that a global optimum may be reached. This is analogous to the annealing of metal or glass.
In the following description, a global *minimum* of f is being searched. The algorithm is run in stages, such that for the iterations within a stage, the temperature is a constant τ_j.

Suppose iteration t belongs to stage j.

1. Sample a candidate $\boldsymbol{\theta}^* \in \mathcal{N}(\boldsymbol{\theta})$ according to a *proposal density* $g_t(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}_t)$; for different t, the proposal density g_t can be different.
2. Let $\Delta = f(\boldsymbol{\theta}^*) - f(\boldsymbol{\theta}_t)$.
   a) If $\Delta \leq 0$, then set $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}^*$.
   b) If $\Delta > 0$, then set $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}^*$ with probability $e^{-\Delta/\tau_j}$, and $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t$ otherwise. This can be done as follows. Sample $U \sim \text{Unif}(0, 1)$. Then

$$\boldsymbol{\theta}_{t+1} = \begin{cases} \boldsymbol{\theta}^* & \text{if } \; U \leq e^{-\Delta/\tau_j} \\ \boldsymbol{\theta}_t & \text{otherwise.} \end{cases}$$

3. Repeat steps 1 and 2 a total of m_j times.
4. Update $\tau_{j+1} = \alpha(\tau_j)$, $m_{j+1} = \beta(m_j)$ and move to stage $j + 1$, where α and β are two deterministic functions that govern how to cool down the temperature and how long to stay at a given temperature.

Suppose I is an image consisting of "pixels" $I(i,j)$, $i,j = 1,\ldots,N$. The image is corrupted by noise $Z(i,j)$ and only $J = I + Z$ is observed. To reconstruct I, one way is to minimize a function

$$f(I) = \sum_{i,j} |J(i,j) - I(i,j)|^2 + K(I),$$

where $K(I)$ is a function that has large values if I has many "irregularities". The idea is that, as long as the noise is not too strong, the real image should be similar to J; on the other hand, irregularities observed in J are likely to be due to noise and should be reduced.
One way to use the simulated annealing to minimize $f(I)$ is as follows. In each iteration, only one pixel of I can be updated. That means two I and I' are neighbors only when they are different at just one pixel. Then at each iteration, choose a pixel $I(i,j)$ and select a candidate value for it. Update $I(i,j)$ according to the rule in step 2 and move to the next iteration.

Some guidelines:

- R function

  `optim`

  can implement some simple versions of simulated annealing;
- the temperature τ_j should slowly decrease to 0;
- the number m_j of iterations at each temperature τ_j should be large and increasing in j;
- reheating strategies that allow sporadic, systematic, or interactive temperature increases to prevent getting stuck in a local minimum at low temperatures can be effective.

Genetic algorithms

First, each $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ is a string of alphabets:

$$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_C),$$

where each θ_i is a symbol from a finite set, such as $\{0, 1\}$, $\{0, 1, 2\}$, $\{$'a', 'b', ...$\}$.

Genetic algorithms regard the maximization of $f(\boldsymbol{\theta})$ over $\boldsymbol{\Theta}$ as a process of natural selection, with $f(\boldsymbol{\theta})$ being a measure of fitness and $\boldsymbol{\theta}$ the genetic code, or "chromosome". It assumes that fitness is a result of some "good" pieces of $\boldsymbol{\theta}$ and by inheriting these pieces plus some mutations fitness is enhanced.

In a genetic algorithm, at each iteration t, at least two candidate solutions $\boldsymbol{\theta}_{t,1}$, ..., $\boldsymbol{\theta}_{t,P}$ have to be tracked. Each iteration consists of several steps.

1. **Selection.** Randomly select from $\boldsymbol{\theta}_{t,1}, \ldots, \boldsymbol{\theta}_{t,P}$ to form a set of pairs. The selection should be based on a *fitness function* $\phi(\boldsymbol{\theta})$. In general, larger values of $f(\boldsymbol{\theta})$ result in larger values of $\phi(\boldsymbol{\theta})$, and higher chance for $\boldsymbol{\theta}$ to be selected.
   There are many selection mechanisms:
   - **a)** select one parent $\boldsymbol{\theta}$ with probability proportional to $\phi(\boldsymbol{\theta})$ and the other parent completely at random;
   - **b)** select each parent independently with probability proportional to $\phi(\boldsymbol{\theta})$;

   ϕ must be selected carefully: using $\phi(\boldsymbol{\theta}_{t,i}) = f(\boldsymbol{\theta}_{t,i})$ may result in rapid convergence into a local maximum. A common choice is

$$\phi(\boldsymbol{\theta}_{t,i}) = \frac{2r_i}{P(P+1)}$$

   where r_i is the *rank* of $f(\boldsymbol{\theta}_{t,i})$ in $f(\boldsymbol{\theta}_{t,1}), \ldots, f(\boldsymbol{\theta}_{t,P})$.

2. **Breeding.** For each pair, $(\boldsymbol{\theta}_a, \boldsymbol{\theta}_b)$, generate one or more "offspring" $\boldsymbol{\theta}' = c(\boldsymbol{\theta}_a, \boldsymbol{\theta}_b) \in \boldsymbol{\Theta}$, where c is a random operator.
Typically, c is a "crossover". If

$$\boldsymbol{\theta}_a = (\theta_{a1}, \ldots, \theta_{aC}),$$
$$\boldsymbol{\theta}_b = (\theta_{b1}, \ldots, \theta_{bC}),$$

then a crossover works as follows,

a) randomly choose a position $1 \leq d \leq C$; and
b) combine $(\theta_{a1}, \ldots, \theta_{ad})$ and $(\theta_{b,d+1}, \ldots, \theta_{bC})$ to form

$$\boldsymbol{\theta}' = (\theta_{a1}, \ldots, \theta_{ad}, \theta_{b,d+1}, \ldots, \theta_{bC}).$$

If necessary, also take

$$\boldsymbol{\theta}'' = (\theta_{a,d+1}, \ldots, \theta_{aC}, \theta_{b1}, \ldots, \theta_{bd})$$

to be another offspring.

3. **Mutation.** Make some random modifications to each offspring. Typically, if an offspring chromosome is

$$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_C)$$

then for $i = 1, \ldots, C$, with a small probability $0 < p < 1$ (mutation rate), change θ_i to a different value.

The offspring produced at iteration t are taken as the candidate solutions for iteration $t + 1$.

Initialization.

Usually the first generation consists of completely random individuals.

- Large values of the size of a generation, P, are preferred. For binary encoding of $\boldsymbol{\theta}$, one suggestion is to have $C \leq P \leq 2C$, where C is the chromosome length. In most real applications, population sizes have ranged between 10 and 200.
- Mutation rates are typically very low, in the neighborhood of 1%.

Termination.

A genetic algorithm is usually terminated after a maximum number of iterations. Alternatively, the algorithm can be stopped once the genetic diversity in the current generation is sufficiently low.

In many problems, like the traveling salesman problem, it is natural to write $\boldsymbol{\theta}$ as a permutation of $1, 2, \ldots, p$.

- Standard crossover usually produces invalid sequences

For example, if

$$\boldsymbol{\theta}_a = \{7, 5, 2, 6, 3, 1, 9, 4, 8\}$$
$$\boldsymbol{\theta}_b = \{9, 1, 2, 3, 8, 6, 7, 5, 4\}$$

and if the crossover point is between 2nd and 3rd positions, then it produces

$$\{7, 5, 2, 3, 8, 6, 7, 5, 4\}$$

an invalid traveling route.

A remedy is order crossover: pick a number of positions from one parent and get the values at those positions, say $i_1, \ldots, i_s$. Next identify those values from the 2nd parent. Suppose the set of values are found at $j_1 < j_2 < \ldots < j_s$. Put i_1 at j_1, i_2 at j_2, $\ldots$, i_s at j_s while keeping the values of the 2nd on other locations unchanged.
Example: Consider parents (752631948) and (912386754) and random positions $(4, 6, 7)$. The offsprings are (612389754) and (352671948).
The drawback of the operation is that it destroys some important structures of the parent routes, in particular, the links.

Edge-recombination crossover uses a different idea. It generates an offspring whose edges belong to those of the parent routes. By edge it means a link into or out of a city in a route. For example, the above two parents have the following links, the order of numbers in each link is unimportant.

$$(1,2),(1,3),(1,9),(2,3),(2,5),(2,6),(3,6),(3,8),$$
$$(4,5),(4,8),(4,9),(5,7),(6,7),(6,8),(7,8)$$

First, choose one of the initial cities of the parents as the initial city of the offspring. At k-th step, if $i_1, \ldots, i_k$ have been chosen as the first k cities on the route, then choose among cities that are linked to i_k *and* are different from $i_1, \ldots, i_{k-1}$ as the $(k+1)$st city of the offspring.

EM and MM Algorithms

EM Optimizations

What it is for

- inference when there is missing information
  - incomplete observations
  - auxiliary parameters that are unobserved but can facilitate inference on the main parameter.
  - Some important applications of the EM algorithm are in problems where one has what we might call pseudo missing data. We never had any chance of obtaining such data, but we could pretend they are missing and use EM algorithm to facilitate the computation of MLEs.

Why it is important

- widely useful
- simple to implement
- numerically stable

Suppose a population consists of several sub-populations, each following a distribution $p(x \mid \theta_k)$ and having population fraction q_k. Typically,

$$\boldsymbol{q} = (q_1, \ldots, q_K), \quad \boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$$

are unknown, and the goal is to estimate them.
Let $x_1, \ldots, x_n$ be a sample. If for each x_i we know it comes from the z_i^{th} sub-population, then, letting

$$\boldsymbol{x} = (x_1, \ldots, x_n), \quad \boldsymbol{z} = (z_1, \ldots, z_n),$$

the likelihood function of $(\boldsymbol{q}, \boldsymbol{\theta})$ is

$$L(\boldsymbol{q}, \boldsymbol{\theta} \mid \boldsymbol{x}, \boldsymbol{z}) = \prod_{i=1}^{n} [q_{z_i} \times p(x_i \mid \theta_{z_i})].$$

However, in many cases, $\boldsymbol{z}$ is unobservable:

- cluster analysis
- the "sub-populations" may be constructs that facilitate inference or decision making, so they are nonexistent in reality.

The likelihood function then becomes

$$L(\boldsymbol{q}, \boldsymbol{\theta} \mid \boldsymbol{x}) = \sum_{\boldsymbol{z}} L(\boldsymbol{q}, \boldsymbol{\theta} \mid \boldsymbol{x}, \boldsymbol{z}).$$

The question is how to get the MLE of $\boldsymbol{q}$ and $\boldsymbol{\theta}$ efficiently.
In this example, $\boldsymbol{y} = (\boldsymbol{x}, \boldsymbol{z})$ is the complete data. When only $\boldsymbol{x}$ is observable, $\boldsymbol{z}$ is called the missing data.

Suppose a population follows a distribution $p(y \mid \theta)$, where θ is the unknown parameter. Suppose a random sample is collected from $p(y \mid \theta)$. However, when an observation y_i is greater than a cut-off C, it is truncated and recorded as C, such as in clinical trials. In this case, we only know $y_i \geq C$ but not its exact value. On the other hand, if $y_i \leq C$, then it is fully observed. Therefore, the observed data is

$$x_1 = \min(y_1, C), \ldots, x_n = \min(y_n, C).$$

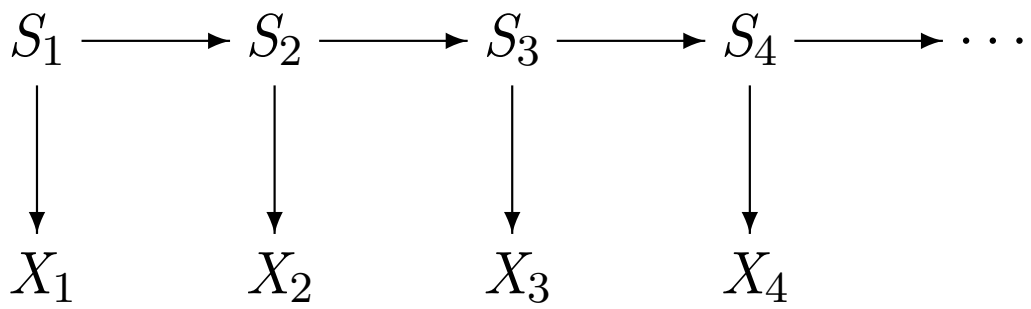
In this case, $\boldsymbol{y} = (y_1, \ldots, y_n)$ is the complete data and $\boldsymbol{x} = (x_1, \ldots, x_n)$. The question is how to estimate θ based on $\boldsymbol{x}$.

Suppose $Z_1, \ldots, Z_n$ and $X_1, \ldots, X_n$ are time series related by

$$X_k = f(Z_k \,|\, \theta),$$

where the transformation f is parametrized by θ. Z_k may include not only variables of interest, but also some "noise" that interact with the variables. Suppose the joint distribution of $Z_1, \ldots, Z_n$ has the parametric form $h(z_1, \ldots, z_n \,|\, \gamma)$, however, the values of θ and γ are both unknown. If $X_1, \ldots, X_n$ are observed but $Z_1, \ldots, Z_n$ are only partially observed or completely hidden, how to estimate θ and γ?

The Hidden Markov Model (HMM) is a typical case. Let $S_1, \ldots, S_n$ be a discrete Markov chain with a finite number of states. Conditional on $S_1, \ldots, S_n$, the observations $X_1, \ldots, X_n$ are independent, such that, given $S_k = s$, $X_k \sim N(\mu_s, \sigma_s^2)$.



In this case,

$$Z_k = (S_k, W_k),$$

where $W_1, \ldots, W_n \sim N(0, 1)$ are independent of $Z_1, \ldots, Z_n$, such that $X_k = \mu_{S_k} + \sigma_{S_k} W_k$. Typically, μ_k, σ_k and the transition probabilities of S_k are unknown and only X_k are observed.

Positron emission tomography (PET) is a tool to study the metabolic activity in organs. To generate a PET scan, some radioactive material is administered into the organ. Then the emissions of the material are recorded using a PET scanner, which consists of an array of detectors surrounding the patient's body. The region of interest is divided into "voxels". The number of photons Y_{ij} coming from voxel i and received by detector j is assumed to follow a Poisson distribution

$$Y_{ij} \sim \text{Poisson}(\theta_i a_{ij})$$

where the coefficient a_{ij} can be determined accurately beforehand, and θ_i is the emission density of the radioactive material in voxel i, which is used to quantify the metabolic activity therein.

If for each pair of voxel and detector, we can observe Y_{ij}, then θ_j's can be estimated by MLE. However, in reality, each detector j receives photons from all the voxels, and hence only the sum

$$X_j = \sum_i Y_{ij}$$

is observable. The goal is to use $\boldsymbol{X} = (X_j)$ instead of $\boldsymbol{Y} = (Y_{ij})$ to estimate $\boldsymbol{\theta} = (\theta_j)$.

Suppose the complete data

$$\boldsymbol{Y} \sim p_{\boldsymbol{Y}}(\boldsymbol{y} \,|\, \boldsymbol{\theta})$$

while the observed data is $\boldsymbol{X} = M(\boldsymbol{Y})$, where M is a many-to-one mapping. Then

$$\underbrace{p_{\boldsymbol{X}}(\boldsymbol{x} \,|\, \boldsymbol{\theta})}_{L(\boldsymbol{\theta} \,|\, \boldsymbol{x})} = \int_{M(\boldsymbol{y})=\boldsymbol{x}} \underbrace{p_{\boldsymbol{Y}}(\boldsymbol{y} \,|\, \boldsymbol{\theta})}_{L(\boldsymbol{\theta} \,|\, \boldsymbol{y})} \, \mathrm{d}\boldsymbol{y}.$$

If only $\boldsymbol{X} = \boldsymbol{x}$ is observed, then the MLE $\widehat{\theta}$ satisfies

$$L'(\widehat{\theta} \,|\, \boldsymbol{x}) = 0$$

i.e.,

$$\int_{M(\boldsymbol{y})=\boldsymbol{x}} L'(\widehat{\theta} \,|\, \boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} = 0.$$

Write log-likelihood function $l(\boldsymbol{\theta} \mid \boldsymbol{y}) = \ln L(\boldsymbol{\theta} \mid \boldsymbol{y})$, so that

$$[L(\boldsymbol{\theta} \mid \boldsymbol{y})]' = [e^{l(\boldsymbol{\theta} \mid \boldsymbol{y})}]' = [l(\boldsymbol{\theta} \mid \boldsymbol{y})]' e^{l(\boldsymbol{\theta} \mid \boldsymbol{y})} = [l(\boldsymbol{\theta} \mid \boldsymbol{y})]' L(\boldsymbol{\theta} \mid \boldsymbol{y}).$$

Therefore,

$$\int_{M(\boldsymbol{y})=\boldsymbol{x}} L'(\widehat{\theta} \mid \boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} = 0$$

is the same as

$$\int_{M(\boldsymbol{y})=\boldsymbol{x}} [l(\widehat{\theta} \mid \boldsymbol{y})]' L(\widehat{\theta} \mid \boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} = 0.$$

In principle, Newton's method can be used to find a solution. However, since $\widehat{\theta}$ appears in two places in the integration, the implementation is often messy and because of that, numerically unstable.

We may try to "decouple" the two $\boldsymbol{\theta}$'s in the integration to get simpler iterations. An iterative procedure could go as follows. At step t, if we have $\boldsymbol{\theta}_t$, then find $\boldsymbol{\theta}_{t+1}$ to solve

$$\int_{M(\boldsymbol{y})=\boldsymbol{x}} [l(\boldsymbol{\theta}_{t+1} \mid \boldsymbol{y})]' L(\boldsymbol{\theta}_t \mid \boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} = 0,$$

or, to optimize the function

$$\int_{M(\boldsymbol{y})=\boldsymbol{x}} l(\boldsymbol{\theta} \mid \boldsymbol{y}) L(\boldsymbol{\theta}_t \mid \boldsymbol{y}) \, \mathrm{d}\boldsymbol{y}.$$

This is a function in $\boldsymbol{\theta}$. Since $\boldsymbol{\theta}_t$ is different at each step, the function is different. If $\boldsymbol{\theta}_t$ converge, then the limit is a solution to the original MLE.

The explicit numerical integration in the above function is inconvenient. To replace it, note that, since $\boldsymbol{\theta}_t$ is fixed, the above way to find $\boldsymbol{\theta}_{t+1}$ is equivalent to optimizing

$$\int_{M(\boldsymbol{y})=\boldsymbol{x}} l(\boldsymbol{\theta} \,|\, \boldsymbol{y}) \frac{L(\boldsymbol{\theta}_t \,|\, \boldsymbol{y})}{L(\boldsymbol{\theta}_t \,|\, \boldsymbol{x})} \, \mathrm{d}\boldsymbol{y}.$$

For any $\boldsymbol{\theta}_t$,

$$\frac{L(\boldsymbol{\theta}_t \,|\, \boldsymbol{y})}{L(\boldsymbol{\theta}_t \,|\, \boldsymbol{x})} = \frac{p_{\boldsymbol{Y}}(\boldsymbol{y} \,|\, \boldsymbol{\theta}_t)}{p_{\boldsymbol{X}}(\boldsymbol{x} \,|\, \boldsymbol{\theta}_t)} = p_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y} \,|\, \boldsymbol{x}, \boldsymbol{\theta}_t),$$

the conditional density of $\boldsymbol{Y}$ given $\boldsymbol{X} = \boldsymbol{x}$ under $p_{\boldsymbol{Y}}(\boldsymbol{y} \,|\, \boldsymbol{\theta}_t)$. Then the integral is simply $\mathbb{E}\left[l(\boldsymbol{\theta} \,|\, \boldsymbol{Y}) \,|\, \boldsymbol{x}, \boldsymbol{\theta}_t\right]$. In turns out that in order to find suitable $\boldsymbol{\theta}_{t+1}$, we need to maximize this function.

EM (Expectation-Maximization) algorithm.

Define

$$Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}') = \mathbb{E}\left[l(\boldsymbol{\theta} \mid \boldsymbol{Y}) \,|\, \boldsymbol{x}, \boldsymbol{\theta}'\right] = \mathbb{E}\left[\ln L(\boldsymbol{\theta} \mid \boldsymbol{Y}) \,|\, \boldsymbol{x}, \boldsymbol{\theta}'\right].$$

- Initialize $\boldsymbol{\theta}_0$.
- At iteration $t = 1, 2, \ldots$, with $\boldsymbol{\theta}_t$ given, set $\boldsymbol{\theta}_{t+1}$ to maximize $Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}_t)$.

The "E step" of the algorithm refers to the computation of $Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}_t)$, and the "M step" refers to the maximization of $Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}_t)$. Stopping criteria are usually based upon $|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t|$ or $|Q(\boldsymbol{\theta}_{t+1} \,|\, \boldsymbol{\theta}_t) - Q(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_t)|$.

If $\boldsymbol{Y} = (\boldsymbol{X}, \boldsymbol{Z})$, where $\boldsymbol{Z}$ is the missing data, then, given $\boldsymbol{X} = \boldsymbol{x}$, the only unknown part of $\boldsymbol{Y}$ is $\boldsymbol{Z}$, so

$$\begin{aligned} Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}') &= \mathbb{E}\left[\ln L(\boldsymbol{\theta} \,|\, \boldsymbol{Y}) \,|\, \boldsymbol{x}, \boldsymbol{\theta}'\right] \\ &= \mathbb{E}\left[\ln L(\boldsymbol{\theta} \,|\, \boldsymbol{x}, \boldsymbol{Z}) \,|\, \boldsymbol{x}, \boldsymbol{\theta}'\right] \\ &= \begin{cases} \sum_{\boldsymbol{z}} p(\boldsymbol{z} \,|\, \boldsymbol{x}, \boldsymbol{\theta}') \ln p(\boldsymbol{x}, \boldsymbol{z} \,|\, \boldsymbol{\theta}), & \text{if } \boldsymbol{Z} \text{ is discrete} \\ \int p(\boldsymbol{z} \,|\, \boldsymbol{x}, \boldsymbol{\theta}') \ln p(\boldsymbol{x}, \boldsymbol{z} \,|\, \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{z}, & \text{if } \boldsymbol{Z} \text{ is continuous} \end{cases} \end{aligned}$$

Gaussian mixture clustering

Suppose a population is a mixture of Gaussian distributions $N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ with population fractions q_k, $k = 1, \ldots, K$. The goal is to estimate $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, q_k.

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$ be an iid sample from the population. Let z_i be the index of the Gaussian distribution from which $\boldsymbol{x}_i$ is sampled. Suppose that

$$\boldsymbol{z} = (z_1, \ldots, z_n)$$

cannot be observed. The observed data thus is

$$\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n).$$

Suppose at iteration t of the EM algorithm, one has obtained

$$\boldsymbol{\theta}_t = (\boldsymbol{\mu}_{t1}, \ldots, \boldsymbol{\mu}_{tK}, \boldsymbol{\Sigma}_{t1}, \ldots \boldsymbol{\Sigma}_{tK}, q_{t1}, \ldots, q_{tK}).$$

Then, since $(\boldsymbol{x}_i, z_i)$ are independent, and z_i are discrete, for any candidate

$$\boldsymbol{\theta} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K, q_1, \ldots, q_K),$$

one has

$$\begin{aligned} Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}_t) &= \sum_{\boldsymbol{z}} p(\boldsymbol{z} \,|\, \boldsymbol{x}, \boldsymbol{\theta}_t) \ln p(\boldsymbol{x}, \boldsymbol{z} \,|\, \boldsymbol{\theta}) \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} \underbrace{p(z_i = k \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_t)}_{w_{ik}} \ln p(z_i = k, \boldsymbol{x}_i \,|\, \boldsymbol{\theta}). \end{aligned}$$

To maximize $Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}_t)$, note that w_{ik} has nothing to do with $\boldsymbol{\theta}$ so it should be computed in the first place. By Bayes rule,

$$w_{ik} = \frac{p(z_i = k, \boldsymbol{x}_i \,|\, \boldsymbol{\theta}_t)}{p(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}_t)} = \frac{p(z_i = k, \boldsymbol{x}_i \,|\, \boldsymbol{\theta}_t)}{\sum_{s=1}^{K} p(z_i = s, \boldsymbol{x}_i \,|\, \boldsymbol{\theta}_t)},$$

with

$$\begin{aligned} p(z_i = k, \boldsymbol{x}_i \,|\, \boldsymbol{\theta}_t) &= p(z_i = k \,|\, \boldsymbol{\theta}_t) p(\boldsymbol{x}_i \,|\, z_i = k, \boldsymbol{\theta}_t) \\ &= q_{tk} \phi(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_{tk}, \boldsymbol{\Sigma}_{tk}), \end{aligned} \tag{1}$$

where $\phi(\boldsymbol{x} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the density of $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ at $\boldsymbol{x}$. Once w_{ik}'s are computed,

$$Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}_t) = \sum_{k=1}^{K} \sum_{i=1}^{n} w_{ik} \ln p(z_i = k, \boldsymbol{x}_i \,|\, \boldsymbol{\theta}),$$

which looks a lot simpler.

As in (1), $p(z_i = k, \boldsymbol{x}_i \,|\, \boldsymbol{\theta}) = q_k n(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. Since

$$n(\boldsymbol{x}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{(2\pi)^{-d/2}}{\sqrt{|\boldsymbol{\Sigma}_k|}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)'\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)\right\},$$

then

$$\begin{aligned} Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}_t) &= \sum_{k=1}^{K}\sum_{i=1}^{n} w_{ik} \ln[(2\pi)^{-d/2} q_k] - \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{n} w_{ik} \ln|\boldsymbol{\Sigma}_k| \\ &\quad - \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{n} w_{ik}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)'\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k) \\ &= I_1 - \frac{I_2}{2} - \frac{I_3}{2}. \end{aligned}$$

From last page,

$$I_1 = \sum_{k=1}^{K}\sum_{i=1}^{n} w_{ik} \ln[(2\pi)^{-d/2} q_k], \quad I_2 = \sum_{k=1}^{K}\sum_{i=1}^{n} w_{ik} \ln |\boldsymbol{\Sigma}_k|$$

$$I_3 = \sum_{k=1}^{K}\sum_{i=1}^{n} w_{ik}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)'\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)$$

First, only I_3 contains $\boldsymbol{\mu}_k$, which is a quadratic form. To *minimize* it, observe that I_3 is the sum of K quadratic forms, each involving a single $\boldsymbol{\mu}_k$

$$I_{3k} = \sum_{i=1}^{n} w_{ik}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)'\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k).$$

We only need to find $\boldsymbol{\mu}_k$ to minimize each I_{3k}. From the property of sample mean, $\boldsymbol{\mu}_k$ must be the mean of a weighted sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, each $\boldsymbol{x}_i$ having weight w_{ik}. So

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{n} w_{ik}\boldsymbol{x}_i}{\sum_{i=1}^{n} w_{ik}}.$$

Next, only I_2 and I_3 contain $\mathbf{\Sigma}_k$, and $I_2 + I_3$ is the sum of K terms of the following form, each involving a single $\mathbf{\Sigma}_k$,

$$S_k = \sum_{i=1}^{n} w_{ik} \ln |\mathbf{\Sigma}_k| + \sum_{i=1}^{n} w_{ik} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)' \mathbf{\Sigma}_k^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)$$

and so we only need to find $\mathbf{\Sigma}_k$ to minimize each S_k. Now that $\boldsymbol{\mu}_k$ is equal to the weighted mean of $\boldsymbol{x}_i$, to *minimize* S_k, $\mathbf{\Sigma}_k$ must be the sample variance of the weighted sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. So

$$\mathbf{\Sigma}_k = \frac{\sum_{i=1}^{n} w_{ik} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)'}{\sum_{i=1}^{n} w_{ik}}.$$

Finally, only I_1 contains q_k. Since

$$\begin{aligned} I_1 &= \sum_{k=1}^{K}\sum_{i=1}^{n} w_{ik} \ln[(2\pi)^{-d/2} q_k] \\ &= \sum_{k=1}^{K}\sum_{i=1}^{n} w_{ik} \ln[(2\pi)^{-d/2}] + \sum_{k=1}^{K}\sum_{i=1}^{n} w_{ik} \ln q_k \\ &= -(d/2)\ln(2\pi)\sum_{k=1}^{K}\sum_{i=1}^{n} w_{ik} + \sum_{k=1}^{K}\left(\sum_{i=1}^{n} w_{ik}\right) \ln q_k \end{aligned}$$

it suffices to minimize

$$\sum_{k=1}^{K} \underbrace{\left(\sum_{i=1}^{n} w_{ik}\right)}_{W_k} \ln q_k$$

Note $q_1, \ldots, q_K$ must satisfy $q_1 + \cdots + q_K = 1$. Therefore, the maximization can be obtained by finding a solution for $\mathcal{L}'(q_1, \ldots, q_K) = 0$, i.e.

$$\frac{\partial \mathcal{L}(q_1, \ldots, q_K)}{\partial q_k} = 0$$

where

$$\mathcal{L}(q_1, \ldots, q_K) = \sum_{k=1}^{K} W_k \ln q_k - \lambda \left\{ \sum_{k=1}^{K} q_k - 1 \right\}$$

with λ a Lagrange multiplier. Then

$$q_k = \frac{W_k}{\sum_{k=1}^{K} W_k}.$$

Since for each i, $\sum_{k=1}^{K} w_{ik} = 1$,

$$q_k = \frac{1}{n} \sum_{i=1}^{n} w_{ik}.$$

The above steps consist a basic EM Gaussian clustering algorithm. It can be summarized as follows.

- Given

$$\boldsymbol{\theta}_t = (\boldsymbol{\mu}_{t1}, \ldots, \boldsymbol{\mu}_{tn}, \boldsymbol{\Sigma}_{t1}, \ldots \boldsymbol{\Sigma}_{tK}, q_{t1}, \ldots, q_{tK}),$$

for $k = 1, \ldots, K$, compute

$$w_{ik} = p(z_i = k \,|\, \boldsymbol{x}_i, \boldsymbol{\theta}_t), \quad i = 1, \ldots, n$$

then set $q_{t+1,k}$ equal to the mean of w_{1k}, ..., w_{nk}, $\boldsymbol{\mu}_{t+1,k}$ and $\boldsymbol{\Sigma}_{t+1,k}$ equal to the weighted average and weighted covariance of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, with weights w_{1k}, ..., w_{nk}, respectively.

For high dimensional data, there are many variants of the algorithm which put constraints on $\mathbf{\Sigma}_k$ to improve computational and estimation efficiency. The strongest constraint (and computationally the simplest) is

$$\mathbf{\Sigma}_1 = \cdots = \mathbf{\Sigma}_K = \sigma^2 \boldsymbol{I},$$

with σ^2 being the only parameter. Others include

1. $\mathbf{\Sigma}_k = \sigma_k^2 \boldsymbol{I}$, $k = 1, \ldots, K$, with σ_k being the parameters;
2. $\mathbf{\Sigma}_k$ are diagonal matrices that may be equal to or different from each other;
3. $\mathbf{\Sigma}_1 = \sigma_k^2 \mathbf{\Sigma}$, with the parameters being σ_k^2 and $\mathbf{\Sigma}$.

If Newton's method is used to directly maximize the log likelihood function $l(\boldsymbol{\theta} \mid \boldsymbol{x}) = \ln L(\boldsymbol{\theta} \mid \boldsymbol{x})$, then one would have to differentiate the function

$$l(\boldsymbol{\theta} \mid \boldsymbol{x}) = \sum_{i=1}^{n} l_i(\boldsymbol{\theta})$$

where, for every $i = 1, \ldots, n$,

$$\begin{aligned} l_i(\boldsymbol{\theta}) = \ln p(\boldsymbol{x}_i \mid \boldsymbol{\theta}) &= \ln \left\{ \sum_{z_i} p(\boldsymbol{x}_i, z_i \mid \boldsymbol{\theta}) \right\} \\ &= \ln \left\{ \sum_{k=1}^{K} p(\boldsymbol{x}_k \mid z_i = k, \boldsymbol{\theta}) p(z_i = k \mid \boldsymbol{\theta}) \right\} \\ &= \ln \left\{ \sum_{k=1}^{K} \frac{(2\pi)^{-\frac{d}{2}} q_k}{\sqrt{|\boldsymbol{\Sigma}_k|}} e^{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)' \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)} \right\}. \end{aligned}$$

The iterations apparently are quite involved.

Computation

The R package

`mclust`

provides a large number of clustering methods, including EM based methods. Use

```
install.packages("mclust")
```

to install it on your computer if it is not yet installed. To use, first execute

```
library("mclust")
```

which loads all the functions in the package into an R session.

`mclustBIC`

computes the Bayes information criterion associated with different Gaussian mixture models, which are characterized by number of clusters and constraints on the covariance matrices.

`mclustModel`

estimate the parameters for the best model determined via `mclustBIC`.

`mclustModelNames`

lists the names of models used in the `mclust` package.

`mclust2Dplot`

plots 2-D data given parameters of a Gaussian mixture model for the data. Some MATLAB routines, such as `EM-GM.m` can perform some of the functions of the R package.

EM for Exponential Families

Many important parametric distributions belong to an exponential family, which has the form

$$p_{\boldsymbol{Y}}(\boldsymbol{y} \,|\, \boldsymbol{\theta}) = c_1(\boldsymbol{y}) c_2(\boldsymbol{\theta}) \exp\left\{\boldsymbol{\theta}^T \boldsymbol{s}(\boldsymbol{y})\right\},$$

where $\boldsymbol{\theta}$ is a vector of parameters and $\boldsymbol{s}(\boldsymbol{y})$ a vector of sufficient statistics.

1. The normal densities $N_d(\boldsymbol{\theta}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma}$ being fixed consist an exponential family, because

$$
\begin{aligned}
p_{\boldsymbol{Y}}(\boldsymbol{y} \,|\, \boldsymbol{\theta}) &= \frac{e^{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\theta})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\boldsymbol{\theta})}}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \\
&= \underbrace{\frac{e^{-\frac{1}{2}\boldsymbol{y}^T\boldsymbol{y}}}{(2\pi)^{d/2}}}_{c_1(\boldsymbol{y})} \underbrace{\frac{e^{-\frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\theta}}}{|\boldsymbol{\Sigma}|^{1/2}}}_{c_2(\boldsymbol{\theta})} \exp\Big\{\boldsymbol{\theta}^T \underbrace{(\boldsymbol{\Sigma}^{-1}\boldsymbol{y})}_{\boldsymbol{s}(\boldsymbol{y})}\Big\}.
\end{aligned}
$$

1. Poisson distribution Poisson(θ), where $\theta > 0$, has distribution function

$$p(y \mid \theta) = \frac{e^{-\theta} y^{\theta}}{y!}, \quad y = 0, 1, \ldots$$

If we define

$$c_1(y) = \frac{1}{y!} \quad c_2(\theta) = e^{-\theta} \quad s(y) = \ln y,$$

then $p(y \mid \theta) = c_1(y) c_2(\theta) e^{\theta s(y)}$. Therefore, Poisson($\theta$) also form an exponential family.

1. Gamma distribution Gamma(a, r), where $a > 0$, $r > 0$, has density

$$p(y \mid a, r) = \begin{cases} \dfrac{e^{-y/r} y^{a-1}}{r^a \Gamma(a)} & y > 0 \\ 0 & y \le 0. \end{cases}$$

Indeed, if we define $b = 1/r$, $\theta = (a, b)$, and

$$c_1(y) = \frac{1}{y} \quad c_2(\boldsymbol{\theta}) = c_2(a, b) = \frac{b^a}{\Gamma(\theta_1)} \quad \boldsymbol{s}(y) = (\ln y, -y)$$

Then

$$c_1(y) c_2(\boldsymbol{\theta}) \exp(\boldsymbol{\theta}^T \boldsymbol{s}(y)) = \frac{1}{y} \frac{1}{b^a \Gamma(a)} e^{a \ln y - by} = p(y \mid a, r).$$

1. Beta distribution Beta(a, b), where $a > 0$, $b > 0$, has density

$$p(y \mid a, b) = \begin{cases} \dfrac{y^{a-1}(1-y)^{b-1}}{B(a,b)}, & 0 < y < 1 \\ 0 & \text{otherwise} \end{cases}$$

If we let $\boldsymbol{\theta} = (a, b)$,

$$c_1(y) = \frac{1}{y(1-y)} \quad c_2(\boldsymbol{\theta}) = \frac{1}{B(a,b)} \quad \boldsymbol{s}(y) = (\ln y, \ln(1-y))$$

then

$$c_1(y)c_2(\boldsymbol{\theta}) \exp(\boldsymbol{\theta}^T \boldsymbol{s}(y)) = \frac{1}{y(1-y)} \frac{1}{B(a,b)} e^{a \ln y + b \ln(1-y)} = p(y \mid a, b).$$

Suppose $\boldsymbol{x} = M(\boldsymbol{y})$ is observed and we want to estimate $\boldsymbol{\theta}$. The EM algorithm in this case is significantly simplified, with no explicit evaluation of $Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}')$. It works as follows. Given $\boldsymbol{\theta}_t$, set $\boldsymbol{\theta}_{t+1}$ such that

$$\mathbb{E}\left[s(\boldsymbol{Y}) \,|\, \boldsymbol{\theta}_{t+1}\right] = \mathbb{E}\left[s(\boldsymbol{Y}) \,|\, \boldsymbol{x}, \boldsymbol{\theta}_t\right]. \tag{2}$$

To get the result, we start with the evaluation

$$
\begin{aligned}
Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}_t) &= \int [\ln p_{\boldsymbol{Y}}(\boldsymbol{y} \,|\, \boldsymbol{\theta})] p(\boldsymbol{y} \,|\, \boldsymbol{x}, \boldsymbol{\theta}_t) \, \mathrm{d}\boldsymbol{y} \\
&= \int [\ln c_1(\boldsymbol{y}) + \ln c_2(\boldsymbol{\theta}) + \boldsymbol{\theta}^T \boldsymbol{s}(\boldsymbol{y})] p(\boldsymbol{y} \,|\, \boldsymbol{x}, \boldsymbol{\theta}_t) \, \mathrm{d}\boldsymbol{y} \\
&= \int [\ln c_1(\boldsymbol{y})] p(\boldsymbol{y} \,|\, \boldsymbol{x}, \boldsymbol{\theta}_t) \, \mathrm{d}\boldsymbol{y} \\
&\quad + \int [\ln c_2(\boldsymbol{\theta})] p(\boldsymbol{y} \,|\, \boldsymbol{x}, \boldsymbol{\theta}_t) \, \mathrm{d}\boldsymbol{y} + \int \boldsymbol{\theta}^T \boldsymbol{s}(\boldsymbol{y}) p(\boldsymbol{y} \,|\, \boldsymbol{x}, \boldsymbol{\theta}_t) \, \mathrm{d}\boldsymbol{y} \\
&= k(\boldsymbol{\theta}_t) + \ln c_2(\boldsymbol{\theta}) + \boldsymbol{\theta}^T \mathbb{E}\left[s(\boldsymbol{Y}) \,|\, \boldsymbol{x}, \boldsymbol{\theta}_t \right].
\end{aligned}
$$

To maximize $Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}_t)$ as a function of $\boldsymbol{\theta}$, it suffices to solve

$$(\ln c_2(\boldsymbol{\theta}))' + \mathbb{E}\left[s(\boldsymbol{Y}) \,|\, \boldsymbol{x}, \boldsymbol{\theta}_t\right] = 0,$$

or

$$\frac{c_2'(\boldsymbol{\theta})}{c_2(\boldsymbol{\theta})} + \mathbb{E}\left[s(\boldsymbol{Y}) \,|\, \boldsymbol{x}, \boldsymbol{\theta}_t\right] = 0.$$

The following equality is basic for exponential families. For *any* $\boldsymbol{\theta}$,

$$\frac{c_2'(\boldsymbol{\theta})}{c_2(\boldsymbol{\theta})} = -\mathbb{E}\left[\boldsymbol{s}(\boldsymbol{Y}) \,|\, \boldsymbol{\theta}\right] = -\int \boldsymbol{s}(\boldsymbol{y}) p_{\boldsymbol{Y}}(\boldsymbol{y} \,|\, \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{y}. \tag{3}$$

Assuming the result is correct for now, it is seen that $\boldsymbol{\theta}_{t+1}$ should be set such that $\mathbb{E}\left[s(\boldsymbol{Y}) \,|\, \boldsymbol{\theta}_{t+1}\right] = \mathbb{E}\left[s(\boldsymbol{Y}) \,|\, \boldsymbol{x}, \boldsymbol{\theta}_t\right]$, i.e., as in (2).

To get (3), notice $\int p_{\boldsymbol{Y}}(\boldsymbol{y} \,|\, \boldsymbol{\theta}) = 1$ for any $\boldsymbol{\theta}$, i.e.

$$c_2(\boldsymbol{\theta}) \int c_1(\boldsymbol{y}) \exp\{\boldsymbol{\theta}^T \boldsymbol{s}(\boldsymbol{y})\} \, \mathrm{d}\boldsymbol{y} = 1.$$

Then

$$\ln c_2(\boldsymbol{\theta}) = -\ln \left\{ \int c_1(\boldsymbol{y}) \exp\{\boldsymbol{\theta}^T \boldsymbol{s}(\boldsymbol{y})\} \, \mathrm{d}\boldsymbol{y} \right\}.$$

Differentiate both sides to get

$$\begin{aligned} \frac{c_2'(\boldsymbol{\theta})}{c_2(\boldsymbol{\theta})} &= -\frac{\int c_1(\boldsymbol{y}) \boldsymbol{s}(\boldsymbol{y}) \exp\{\boldsymbol{\theta}^T \boldsymbol{s}(\boldsymbol{y})\} \, \mathrm{d}\boldsymbol{y}}{\int c_1(\boldsymbol{y}) \exp\{\boldsymbol{\theta}^T \boldsymbol{s}(\boldsymbol{y})\} \, \mathrm{d}\boldsymbol{y}} \\ &= -c_2(\boldsymbol{\theta}) \int c_1(\boldsymbol{y}) \boldsymbol{s}(\boldsymbol{y}) \exp\{\boldsymbol{\theta}^T \boldsymbol{s}(\boldsymbol{y})\} \, \mathrm{d}\boldsymbol{y} \\ &= -\int \boldsymbol{s}(\boldsymbol{y}) p_{\boldsymbol{Y}}(\boldsymbol{y} \,|\, \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{y}, \end{aligned}$$

as claimed.

Convergence of EM

We first show that each step of the algorithm increases the observed-data log likelihood $l(\boldsymbol{\theta} \mid \boldsymbol{x}) = \ln p_{\boldsymbol{X}}(\boldsymbol{x} \mid \boldsymbol{\theta})$, that is

$$l(\boldsymbol{\theta}_{t+1} \mid \boldsymbol{x}) \geq l(\boldsymbol{\theta}_t \mid \boldsymbol{x}).$$

Once this is done, one can say $l(\boldsymbol{\theta}_t \mid \boldsymbol{x})$ always climbs up to a local maximum of $l(\boldsymbol{\theta} \mid \boldsymbol{x})$.

- In general, there is no guarantee that the limit is the (global) maximum.
- EM has slower convergence than the Newton's method, sometime substantially.
- Nevertheless, the ease of implementation and the stable ascent of EM are often very attractive.

Convergence of EM

By $\boldsymbol{X} = M(\boldsymbol{Y})$, $p_{\boldsymbol{Y}}(\boldsymbol{y} \,|\, \boldsymbol{\theta}) = p(\boldsymbol{x}, \boldsymbol{y} \,|\, \boldsymbol{\theta}) = p_{\boldsymbol{X}}(\boldsymbol{x} \,|\, \boldsymbol{\theta}) p_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y} \,|\, \boldsymbol{x}, \boldsymbol{\theta})$, so

$$p_{\boldsymbol{X}}(\boldsymbol{x} \,|\, \boldsymbol{\theta}) = \frac{p_{\boldsymbol{Y}}(\boldsymbol{y} \,|\, \boldsymbol{\theta})}{p_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y} \,|\, \boldsymbol{x}, \boldsymbol{\theta})}.$$

Because $\boldsymbol{x}$ is fixed, $p_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y} \,|\, \boldsymbol{x}, \boldsymbol{\theta})$ only depends on $\boldsymbol{\theta}$. To simplify notation, write $f(\boldsymbol{y} \,|\, \boldsymbol{\theta}) = p_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y} \,|\, \boldsymbol{x}, \boldsymbol{\theta})$. Then

$$\ln p_{\boldsymbol{X}}(\boldsymbol{x} \,|\, \boldsymbol{\theta}) = \ln p_{\boldsymbol{Y}}(\boldsymbol{y} \,|\, \boldsymbol{\theta}) - \ln f(\boldsymbol{y} \,|\, \boldsymbol{\theta}).$$

Convergence of EM

Integrate both sides with respect to the density $f(\boldsymbol{y} \mid \boldsymbol{\theta}_t)$. From

$$\begin{aligned} &\int [\ln p_{\boldsymbol{Y}}(\boldsymbol{y} \mid \boldsymbol{\theta})] f(\boldsymbol{y} \mid \boldsymbol{\theta}_t) \, \mathrm{d}\boldsymbol{y} \\ &= \int [\ln p_{\boldsymbol{Y}}(\boldsymbol{y} \mid \boldsymbol{\theta})] \, p_{\boldsymbol{Y} \mid \boldsymbol{X}}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}_t) \, \mathrm{d}\boldsymbol{y} = Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}_t), \end{aligned}$$

It follows

$$\ln p_{\boldsymbol{X}}(\boldsymbol{x} \mid \boldsymbol{\theta}_t) = Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}_t) - H(\boldsymbol{\theta} \mid \boldsymbol{\theta}_t), \tag{4}$$

where

$$H(\boldsymbol{\theta} \mid \boldsymbol{\theta}_t) = \int [\ln f(\boldsymbol{y} \mid \boldsymbol{\theta})] f(\boldsymbol{y} \mid \boldsymbol{\theta}_t) \, \mathrm{d}\boldsymbol{y}.$$

Convergence of EM

To show $\ln p_{\boldsymbol{X}}(\boldsymbol{x} \,|\, \boldsymbol{\theta}_{t+1}) \geq \ln p_{\boldsymbol{X}}(\boldsymbol{x} \,|\, \boldsymbol{\theta}_t)$, by (4), their difference equals

$$[Q(\boldsymbol{\theta}_{t+1} \,|\, \boldsymbol{\theta}_t) - Q(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_t)] - [H(\boldsymbol{\theta}_{t+1} \,|\, \boldsymbol{\theta}_t) - H(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_t)].$$

By the M-step, $Q(\boldsymbol{\theta}_{t+1} \,|\, \boldsymbol{\theta}_t) \geq Q(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_t)$. On the other hand, by the Jensen's inequality,

$$\begin{aligned} H(\boldsymbol{\theta}_{t+1} \,|\, \boldsymbol{\theta}_t) - H(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_t) &= \int \ln \left[\frac{f(\boldsymbol{y} \,|\, \boldsymbol{\theta}_{t+1})}{f(\boldsymbol{y} \,|\, \boldsymbol{\theta}_t)} \right] f(\boldsymbol{y} \,|\, \boldsymbol{\theta}_t) \, \mathrm{d}\boldsymbol{y} \\ &\leq \ln \left[\int \frac{f(\boldsymbol{y} \,|\, \boldsymbol{\theta}_{t+1})}{f(\boldsymbol{y} \,|\, \boldsymbol{\theta}_t)} f(\boldsymbol{y} \,|\, \boldsymbol{\theta}_t) \, \mathrm{d}\boldsymbol{y} \right] = \ln \left[\int f(\boldsymbol{y} \,|\, \boldsymbol{\theta}_{t+1}) \, \mathrm{d}\boldsymbol{y} \right] = 0, \end{aligned}$$

thus showing the likelihood increases at each iteration.

Variants of EM

- E step is hard: Monte Carlo (MCEM)
- M step is hard: Instead of choosing $\boldsymbol{\theta}_{t+1}$ to maximize $Q(\boldsymbol{\theta} \,|\, \boldsymbol{\theta}_t)$, one simply chooses any $\boldsymbol{\theta}_{t+1}$ for which

$$Q(\boldsymbol{\theta}_{t+1} \,|\, \boldsymbol{\theta}_t) > Q(\boldsymbol{\theta}_t \,|\, \boldsymbol{\theta}_t),$$

  then the resulting algorithm is called generalized EM, or GEM.

  - In GEM, $l(\boldsymbol{\theta}_t \,|\, \boldsymbol{x})$ is still increasing.
  - An example: ECM algorithm, which replaces the M-step of the EM algorithm with several computationally simpler conditional maximization (CM) steps.

Variance Estimation

- Louis' method (1982, JRSSB)
- Supplemented EM (SEM) algorithm (Meng and Rubin, 1991, JASA)
- Oakes' method (1999, JRSSB)
- Bootstrapping
- Empirical information
- Numerical Differentiation

Example: Hidden Markov Model

- Hidden state $\mathbf{s} = (s_t)$, $t = 1, \ldots, T$, follows a Markov chain with initial state distribution π_k, $k = 1, \ldots, M$, and transition probability matrix $\{a_{k,l} = \Pr(s_{t+1} = l | s_t = k)\}$. (Note this matrix is row stochastic.)
- Given the state s_t, the observation u_t is independent of other observations and states
- Given state k, the observation follows distribution $b_k(u)$. For example, $b_k(u|\gamma_k)$ can be $N(\mu_k, \Sigma_k)$; or discrete.
- The observed data is $\mathbf{u} = (u_t)$, $t = 1, \ldots, T$.

Likelihood

- Parameters: θ contains π_k's, a_{kl}'s, γ_k's
- Missing (latent, unobserved) states: s_t, $t = 1, \ldots, T$.
- Complete data: $(\mathbf{s}, \mathbf{u})$
- Complete data likelihood

$$pi_{s_1} \prod_{t=2}^{T} a_{s_{t-1}, s_t} \prod_{t=1}^{T} b_{s_t}(u_t|\gamma_{s_t})$$

- Complete data log-likelihood

$$\log p(\mathbf{s}, \mathbf{u}|\theta) = \log pi_{s_1} + \sum_{t=2}^{T} \log a_{s_{t-1}, s_t} + \sum_{t=2}^{T} b_{s_t}(u_t|\theta_{s_t})$$

E-Step

Taking expectation of $\log p(\mathbf{s}, \mathbf{u}|\theta')$ with respect to $p(\mathbf{s}|\mathbf{u}, \theta)$:

$$
\begin{aligned}
& E\Big(\log p(\mathbf{s}, \mathbf{u})|\theta')|\mathbf{u}, \theta) \Big) \\
&= \sum_{\mathbf{s}} p(\mathbf{s}|\mathbf{u}, \theta_t) \log p(\mathbf{s}, \mathbf{u}|\theta) \\
&= \sum_{k=1}^{M} L_k(1) \log \pi'_k + \sum_{t=2}^{T} \sum_{k=1}^{M} \sum_{l=1}^{M} H_{k,l}(t) \log a'_{k,l} \\
&\quad + \sum_{t=1}^{T} \sum_{k=1}^{M} L_k(t) \log b_{s_t}(u_t|\theta'_k)
\end{aligned}
$$

where

$$
L_k(t) = \Pr(s_t = k|\mathbf{u}) = \sum_{\mathbf{s}: s_t = k} p(\mathbf{s}|\mathbf{u})
$$

and

$$
H_{k,l}(t) = \Pr(s_t = k, s_{k+1} = l|\mathbf{u}) = \sum_{\mathbf{s}: s_t = k, s_{t+1} = l} \Pr(\mathbf{s}|\mathbf{u}).
$$

M-step

If b_k's are multivariate normals, the M-step has closed-form solution:

$$\mu_k = \frac{\sum_{t=1}^{T} L_k(t) u_t}{\sum_{t=1}^{T} L_k(t)}$$

$$\Sigma_k = \frac{\sum_{t=1}^{T} L_k(t)(u_t - \mu_k)(u_t - \mu_k)^\top}{\sum_{t=1}^{T} L_k(t)}$$

$$a_{k,l} = \frac{\sum_{t=1}^{T-1} H_{k,l}(t)}{\sum_{t=1}^{T-1} L_k(t)}$$

$$q_{k,j} = \frac{\sum_{t=1}^{T} L_k(t) I(u_j = j)}{\sum_{t=1}^{T} L_k(t)}$$

$$\pi_k = \frac{\sum_{t=1}^{T} L_k(t)}{\sum_{k=1}^{M} \sum_{t=1}^{T} L_k(t)}$$

Forward and Backward Probability

- This is for efficient computation of $L_k(t)$'s and $H_{k,l}(t)$'s
- Computation of order M^2T and memory requirement of order MT.
- Forward probability

$$\alpha_k(t) = \mathsf{Pr}(u_1, \ldots, u_t, s_t = k)$$

- Backword probability

$$\beta_k(t) = \mathsf{Pr}(u_{t+1}, \ldots, u_T | s_t = k)$$

Forward and Backward Algorithms

- Forward probability recursion

$$\alpha_k(1) = \pi_k b_k(u_1), \qquad 1 \le k \le M,$$

$$\alpha_k(t) = b_k(u_t) \sum_{l=1}^{M} \alpha_l(t-1) a_{l,k}, \qquad 1 < t \le T, \quad 1 \le k \le M.$$

- Backward probability recursion

$$\beta_k(T) = 1, \qquad 1 \le k \le M,$$

$$\beta_k(t) = \sum_{l=1}^{M} a_{k,l} b_l(u_{t+1}) \beta_l(t+1), \qquad 1 \le t < T, \quad 1 \le k \le M.$$

Fast Algorithm for the E-step Quantities

$$L_k(t) = \mathsf{Pr}(s_t = k|\mathbf{u}) = \frac{\alpha_k(t)\beta(t)}{p(\mathbf{u})}$$

$$H_{k,l}(t) = \mathsf{Pr}(s_t = k, s_{k+1} = l|\mathbf{u}) = \frac{\alpha_k(t)a_{k,l}b_l(u_{t+1})\beta_l(t+1)}{p(\mathbf{u})}$$

$$p(\mathbf{u}) = \sum_{k=1}^{M} \alpha_k(t)\beta(t), \qquad \forall t$$

Majorization-Minimization Algorithm

Majorization-Minimization Algorithm

- The MM algorithm is not an algorithm, but a prescription for constructing optimization algorithms.
- The EM algorithm is a special case.
- An MM algorithm operates by creating a surrogate function that majorizes (minorizes) the objective function. When the surrogate function is optimized, the objective function is driven downhill (uphill).
- Majorization-Minimization, Minorization-Maximization.

Majorization-Minimization Algorithm

- It may generate an algorithm that avoids matrix inversion.
- It can linearize an optimization problem.
- It can conveniently deal with equality and inequality constraints.
- It can solve a non-differentiable problem by iteratively solving smooth problems.

Majorization-Minimization Algorithm

- A function $g(\theta \mid \theta^s)$ is said to majorize the function $f(\theta)$ at θ^s if

$$f(\theta^s) = g(\theta^s \mid \theta^s),$$

and

$$f(\theta) \leq g(\theta \mid \theta^s)$$

for all θ.

- Let

$$\theta^{s+1} = \arg\min_{\theta} g(\theta \mid \theta^s).$$

- It follows that

$$f(\theta^{s+1}) \leq g(\theta^{s+1} \mid \theta^s) \leq g(\theta^s \mid \theta^s) = f(\theta^s).$$

The strict inequality holds unless $g(\theta^{s+1} \mid \theta^s) = g(\theta^s \mid \theta^s)$ and $f(\theta^{s+1}) = g(\theta^{s+1} \mid \theta^s)$.

- Therefore, by alternating between the majorization and the minimization steps, the objective function is monotonically decreasing and thus its convergence is guaranteed.

Majorization-Minimization Algorithm

- We can use any inequalities to construct the desired majorized/minorized version of the objective function. There are several typical choices:
  - Jensen's inequality: for a convex function $f(x)$,

$$f(E(X)) \leq E(f(X)).$$

  - Convexity inequality: for any $\lambda \in [0, 1]$,

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$$

  - Cauchy-Schwarz inequality.
  - Supporting hyperplanes.
  - Arithmetic-Geometric Mean Inequality.

Example: Penalized AFT model with induced smoothing

AFT model

- Let $\{T_i, C_i, \mathbf{x}_i\}$, $i = 1, \ldots, n$ be n independent copies of $\{T, C, \mathbf{X}\}$, where T_i and C_i are log-transformed failure time and log transformed censoring time, $\mathbf{x}_i$ is a $p \times 1$ covariate vector, and given $\mathbf{X}$, C and T are independent.
- The Accelerated failure time model has the form

$$T_i = \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i, \qquad i = 1, \ldots, n.$$

- The observed data are $(Y_i, \delta_i, \mathbf{x}_i)$, where $Y_i = \min(T_i, C_i)$, $\delta_i = I(T_i < C_i)$.

AFT model

- For a case-cohort study, The weight-adjusted estimating equation with induced smoothing is

$$\tilde{U}_n(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\Delta_i(\mathbf{x}_i - \mathbf{x}_j)\Phi\left(\frac{e_j(\boldsymbol{\beta}) - e_i(\boldsymbol{\beta})}{r_{ij}}\right) = 0,$$

  where $e_j(\boldsymbol{\beta}) = Y_j - \mathbf{x}_j^{\mathrm{T}}\boldsymbol{\beta}$ and
  $r_{ij}^2 = n^{-1}(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j) = n^{-1}\|\mathbf{x}_i - \mathbf{x}_j\|^2$.
- Define $H(x) = x\Phi(x) + \phi(x)$, and

$$\tilde{L}_n(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\Delta_i r_{ij} H\left(\frac{e_j(\boldsymbol{\beta}) - e_i(\boldsymbol{\beta})}{r_{ij}}\right).$$

  Then solving the estimating equation $\tilde{U}_n(\boldsymbol{\beta}) = 0$ is equivalent to minimizing the objective function $\tilde{L}_n(\boldsymbol{\beta})$.
- The objective function $\tilde{L}_n(\boldsymbol{\beta})$ is convex in $\boldsymbol{\beta}$.

AFT model

- We propose the penalized AFT method, which amounts to minimize

$$\tilde{PL}_n(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\Delta_i r_{ij} H\left(\frac{e_j(\boldsymbol{\beta}) - e_i(\boldsymbol{\beta})}{r_{ij}}\right) + \lambda P(\boldsymbol{\beta}),$$

  where $P(\boldsymbol{\beta})$ is some penalty function and λ is the tuning parameter.

AFT model

- To solve this problem, the key is to identity a surrogate function of $\tilde{L}_n(\boldsymbol{\beta})$ that is easy to work with. In view of the form of the objective function, an immediate idea is to majorize $H(x)$.
- Because $H(x)$ is twice differentiable and $H''(x) = \phi(x) < \phi(0) \approx 0.4$ for any $x \in \mathbb{R}$, it follows that for any $x^0 \in R$,

$$H(x \mid x^0) \leq G(x \mid x^0) \equiv H(x^0) + (x - x^0)\Phi(x^0) + \frac{\phi(0)}{2}(x - x^0)^2.$$

- Note that in a standard optimization method based on quadratic approximation, e.g., Newton-Raphson, the $\phi(0)$ term is replaced with $\phi(x^0)$, which, however, does not guarantee the majorization.

AFT model

- Write

$$\tilde{L}_n(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} H(a_{ij} + \mathbf{z}_{ij}^{\mathrm{T}}\boldsymbol{\beta}) + \lambda P(\boldsymbol{\beta}),$$

where $w_{ij} = \Delta_i r_{ij}$, $a_{ij} = (Y_j - Y_i)/r_{ij}$ and $\mathbf{z}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)/r_{ij}$. Then

$$\begin{aligned}\tilde{L}_n(\boldsymbol{\beta} \mid \boldsymbol{\beta}^s) &\le \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} G(a_{ij} + \mathbf{z}_{ij}^{\mathrm{T}}\boldsymbol{\beta} \mid a_{ij} + \mathbf{z}_{ij}^{\mathrm{T}}\boldsymbol{\beta}^s) + \lambda P(\boldsymbol{\beta}) \\ &= \frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\mathbf{A}\boldsymbol{\beta} - \mathbf{b}^{s^{\mathrm{T}}}\boldsymbol{\beta} + \lambda P(\boldsymbol{\beta}) + \text{const} \\ &\equiv \tilde{L}_n^*(\boldsymbol{\beta} \mid \boldsymbol{\beta}^s) + \text{const},\end{aligned}$$

where

$$\mathbf{A} = \frac{\phi(0)}{n}\sum_{i=1}^{n}\sum_{i=1}^{n} w_{ij}\mathbf{z}_{ij}\mathbf{z}_{ij}^{\mathrm{T}}, \qquad \mathbf{b}^s = \mathbf{A}\boldsymbol{\beta}^s - \frac{1}{n}\sum_{i=1}^{n}\sum_{i=1}^{n} w_{ij}\Phi(a_{ij} + \mathbf{z}_{ij}^{\mathrm{T}}\boldsymbol{\beta}^s)\mathbf{z}_{ij}$$

- Minimizing $\tilde{L}_n^*(\boldsymbol{\beta} \mid \boldsymbol{\beta}^s)$ is a Lasso-type problem, for which a coordinate descent algorithm can be applied.

MM for Penalized AFT

Initialize $\boldsymbol{\beta}^0 \in \mathbb{R}^p$. Compute

$$\mathbf{A} = \frac{\phi(0)}{n} \sum_{i=1}^{n} \sum_{i=1}^{n} w_{ij} \mathbf{z}_{ij} \mathbf{z}_{ij}^{\mathrm{T}}.$$

Set $k \leftarrow 0$.
repeat
(1). Majorization step:

$$\mathbf{b}^{k+1} \leftarrow \mathbf{A}\boldsymbol{\beta}^k - \frac{1}{n} \sum_{i=1}^{n} \sum_{i=1}^{n} w_{ij} \Phi(a_{ij} + \mathbf{z}_{ij}^{\mathrm{T}} \boldsymbol{\beta}^k) \mathbf{z}_{ij}.$$

(2). Minimization step:
Initialize $\tilde{\boldsymbol{\beta}}^0 = \boldsymbol{\beta}^k$.
while $\|\tilde{\boldsymbol{\beta}}^s - \tilde{\boldsymbol{\beta}}^{s+1}\| / \|\tilde{\boldsymbol{\beta}}^s\| \geq \epsilon$ and $s < s_{max}$ **do**
(i) For $j = 1, \ldots, p$,

$$\tilde{\beta}_j^s \leftarrow \frac{1}{a_{jj}} \mathcal{S}(b_j^{k+1} - \mathbf{a}_j^{\mathrm{T}} \tilde{\boldsymbol{\beta}}^s + a_{jj} \tilde{\beta}_j^s, \lambda).$$

(ii) $\tilde{\boldsymbol{\beta}}^{s+1} \leftarrow \tilde{\boldsymbol{\beta}}^s$.
(iii) $s \leftarrow s + 1$.
end while

$$\boldsymbol{\beta}^{k+1} \leftarrow \tilde{\beta}^{s+1}.$$

Set $k \leftarrow k + 1$.
until convergence, i.e., $\|\boldsymbol{\beta}^{k+1} - \boldsymbol{\beta}^k\| / \|\boldsymbol{\beta}^k\| \leq \epsilon$.

Simulation Basics

Sampling Random Variables

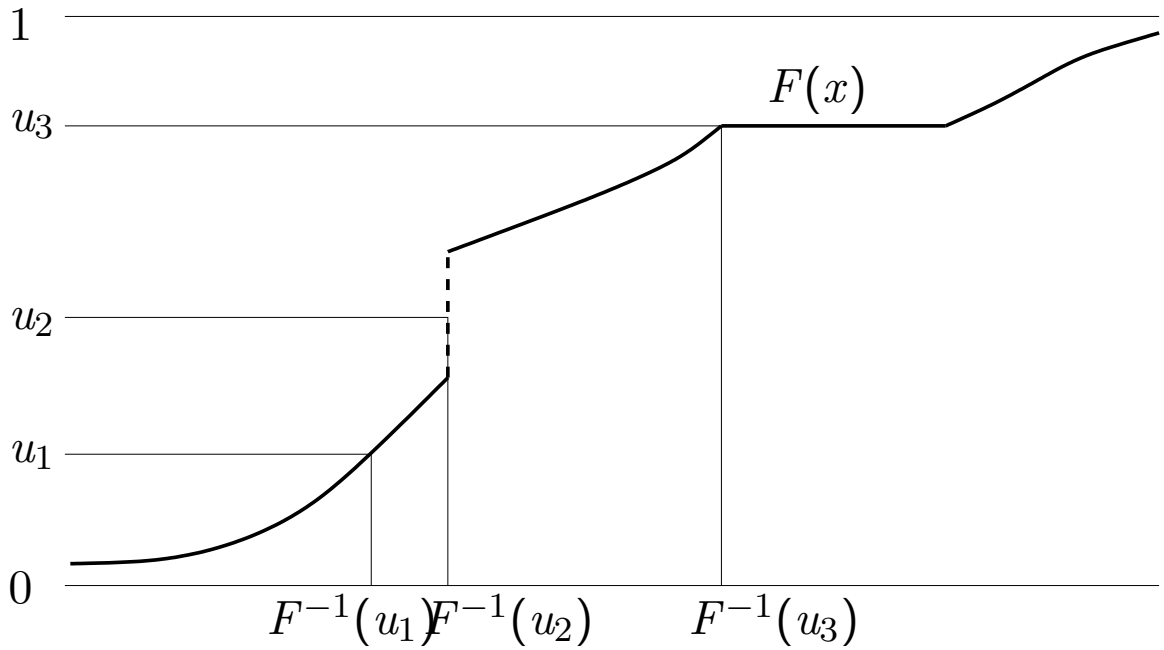
The entire area of random sampling is built up the fact that we can use computing device to generate, deterministically, long sequences of numbers that by many accounts are very similar to long sequences of iid random variables following Unif$(0, 1)$.
In our discussion, we assume that we can generate ideal sequence $U_1, U_2, \ldots$ iid $\sim$ Unif$(0, 1)$. The question is how to use them to simulate other random variables for vectors.

Inverse transform method

This method only applies to random variables taking values in $\mathbb{R}$. Let F be a distribution function. Recall that F is nondecreasing and right-continuous with $0 \leq F(x) \leq 1$. However, F may be discontinuous at some x and may be constant in certain intervals.
For $0 \leq u \leq 1$, define

$$F^{-1}(u) = \inf\{x : F(x) \geq u\}.$$

The inverse can be thought as follows. Plot the graph of F. If F has a jump at x, the graph of F will have a gap at x. Fill the gap with a vertical line. Draw a horizontal line at height u starting from $-\infty$. Then $F^{-1}(u)$ is the x-coordinate of the first point that the horizontal line meets with the gap-filled graph of F.



Theorem

If $U \sim$ *Unif*$(0,1)$, *then* $F^{-1}(U) \sim F$.

Proof.

For any x, $\mathbb{P}F^{-1}(U) \leq x = \mathbb{P}U \leq F(x) = F(x)$. □

Discrete Random Variable

Let X be a random variable taking values in $\{c_1, \ldots, c_n\}$. Note that c_i may not be sorted or even not real numbers. Let

$$q_0 = 0, \quad q_i = \sum_{j=1}^{i} \mathbb{P}\{X = c_j\}, \quad i = 1, \ldots, n$$

To sample X,

1. sample $U \sim \text{Unif}(0, 1)$;
2. find $K \in \{1, \ldots, n\}$, such that $q_{K-1} < U \leq q_K$;
3. set $X = c_K$.

□

Exponential Variable

The exponential distribution with parameter θ, usually denoted by $\text{Exp}(\theta)$, has

$$F(x) = 1 - e^{-x/\theta}, \quad x \geq 0.$$

Since $F^{-1}(u) = -\theta \ln(1-u)$ for $u \in (0,1)$,

$$-\theta \ln(1-U) \sim \text{Exp}(\theta), \quad U \sim \text{Unif}(0,1).$$

Since $1 - U \sim \text{Unif}(0,1)$, there is also $-\theta \ln U \sim \text{Exp}(\theta)$.

A standard Brownian motion $W(t)$ is a process with continuous sample paths that satisfies the following conditions

(i) $W(0) = 0$; and

(ii) for $0 \leq t_1 < \cdots < t_k$, the increments

$$W(t_2) - W(t_1), \ldots, W(t_k) - W(t_{k-1})$$

are independent and each $W(t_i) - W(t_{i-1}) \sim N(0, t_i - t_{i-1})$.

Example Let T be the time at which a standard Brownian motion $W(t)$ attains its maximum over the time interval $[0, 1]$:

$$T = \inf\{t \in [0, 1] : W(t) = \sup_{x \in [0,1]} W(x)\}.$$

Then T follows the *arcsine law* whose distribution is

$$F(x) = \frac{2}{\pi} \arcsin(\sqrt{x}), \quad x \in [0, 1]$$

and hence

$$\frac{1 - \cos(U\pi)}{2} \sim T.$$

□

Proof.

For $x \in [0,1]$,

$$\begin{aligned}
\mathbb{P}\{T \le x\} &= \mathbb{P}\left\{ \max_{t\in[0,x]} W(t) - W(x) \ge \max_{t\in[x,1]} W(t) - W(x) \right\} \\
&= \mathbb{P}\left\{ \max_{t\in[-x,0]} W(t) \ge \max_{t\in[0,1-x]} W(t) \right\} \\
&= \mathbb{P}\left\{ \max_{t\in[0,x]} \tilde{W}(t) \ge \max_{t\in[0,1-x]} W(t) \right\}
\end{aligned}$$

where $\tilde{W}$ is an independent copy of W. It is known (reflection principle) that $\max_{t\in[0,x]} W(t) \sim x|Z|$, where $Z \sim N(0,1)$. It is also known that if Z and $\tilde{Z}$ are iid $N(0,1)$, then $Z/\tilde{Z}$ is Cauchy with $F(x) = \pi^{-1}\arctan x + 1/2$. Therefore, $\mathbb{P}\{T \le x\} = \mathbb{P}\left\{(|Z/\tilde{Z}| \le \sqrt{x/(1-x)}\right\} = \frac{2}{\pi}\arctan\sqrt{x/(1-x)} = \frac{2}{\pi}\arcsin(\sqrt{x})$. □

Example Let

$$M = \sup_{t \in [0,1]} W(t)$$

Then, conditioning on $W(1) = b$, M follows the *Rayleigh distribution*

$$\mathbb{P}\{M \le x \mid W(1) = b\} = F_b(x) = 1 - e^{-2x(x-b)}, \quad x > \max(0, b).$$

Solving $u = 1 - e^{-2x(x-b)}$ gives

$$x = \frac{b \pm \sqrt{b^2 - 2\ln(1-u)}}{2}$$

Since $F_b(x) > 0$ only for $x > b$,

$$F_b^{-1}(u) = \frac{b + \sqrt{b^2 - 2\ln(1-u)}}{2}.$$

Then

$$\frac{b + \sqrt{b^2 - 2\ln U}}{2} \sim F_b.$$

□

Proof.

Let $T = \inf\{t : W(t) = x\}$. For $t \in (0, 1)$, $f_{T|W(1)}(t \mid b) \propto f_{W(1)|T}(b \mid t) f_T(t)$. By strong property of W,

$$f_{W(1)\mid T}(b \mid t) = \frac{e^{-(b-x)^2/2(1-t)}}{\sqrt{2\pi(1-t)}}.$$

and by scaling $\mathbb{P}\{T \le t\} = \mathbb{P}\left\{\max_{s\in[0,t]} W(s) \ge x\right\} = \mathbb{P}\left\{M \ge x/\sqrt{t}\right\}$.

□

Sampling a conditional distribution

Let $X \sim F$. To sample X conditional on $a < X \leq b$, recall

$$\mathbb{P}\{a < X \leq b\} = F(b) - F(a).$$

In order for the sampling to make sense, assume $F(a) < F(b)$. Let $A = F(a)$ and $B = F(b)$. Then

$$F^{-1}(A + (B - A)U)$$

follows the conditional distribution, since

$$\begin{aligned}\mathbb{P}\left\{F^{-1}(A + (B - A)U) \leq x\right\} &= \mathbb{P}\{A + (B - A)U \leq F(x)\} \\ &= \mathbb{P}\left\{U \leq \frac{F(x) - F(a)}{F(b) - F(a)}\right\} = \frac{F(x) - F(a)}{F(b) - F(a)}.\end{aligned}$$

The right hand side is just $\mathbb{P}\{X \leq x \,|\, a < X \leq b\}$.

Example Let $c > 0$ and $X \sim \text{Exp}(\theta)$. To sample X conditional on $X \geq c$, recall the exponential distribution is memoryless in the sense that for any $c > 0$,

$$X - c \sim \text{Exp}(\theta) \quad \text{given} \quad X > c.$$

Therefore, the conditional distribution can be sampled by $-\theta \ln U + c$ with $U \sim \text{Unif}(0, 1)$. □

Numerical evaluation of F^{-1}

If F^{-1} is not known explicitly, the inverse transform method is still applicable through numerical evaluation of F^{-1}. If F is continuous, computing $F^{-1}(u)$ is equivalent to finding a root x of the equation $F(x) - u = 0$. For a distribution F with density f, Newton's method procedures a sequence

$$x_{n+1} = x_n - \frac{F(x_n) - u}{f(x_n)}$$

iteratively, given a starting point x_0. Under suitable conditions, $x_n \to F^{-1}(u)$.

R provides many sampling functions. In each of the following functions, `n` is the number of observations to be sampled.

- `runif(n, min=0, max=1)`: Unif(a, b);
- `rnorm(n, mean=0, sd=1)`: $N(\mu, \sigma^2)$;
- `rpois(n, lambda)`: Poisson(λ)

$$p_n = e^{-\lambda}\lambda^n/n!.$$

- `rbeta(n, shape1, shape2)`: Beta(α, β)

$$f(x) = x^{\alpha-1}(1-x)^{\beta-1}/B(\alpha,\beta), \quad 0 < x < 1;$$

note α and β must be positive.

- `rgamma(n, shape, rate=1)`: Gamma$(\alpha, 1/r)$

$$f(x) = r^{\alpha}x^{\alpha-1}e^{-rx}/\Gamma(\alpha). \quad x > 0;$$

note α must be positive.

Rejection sampling

The method was introduced by Von Neumann and is among the most widely applicable mechanisms. In standard statistical software, the method is implemented to randomly sample many important distributions, such as normal distributions, Gamma distributions, Beta distributions, Poisson distributions. Much more detail on specific developments can be found in the following books.

1. Wolfgang Hörmann, Josef Leydold, and Gerhard Derflinger, *Automatic nonuniform random variate generation*, Springer-Verlag, 2004.
2. Luc Devroye, *Nonuniform random variate generation*, Springer-Verlag, 1986.

Let $\mathscr{X}$ be a discrete set or a Euclidean space. Suppose the goal is to sample from a target density f defined on $\mathscr{X}$. Suppose we know that $f(x)$ is proportional to a function $q(x)$, i.e.,

$$f(x) = q(x)/C, \quad x \in \mathscr{X}.$$

In order for f to be a probability density, C must satisfy

$$C = \begin{cases} \int_{\mathscr{X}} q(x)\,\mathrm{d}x & \text{if } \mathscr{X} \text{ is continuous} \\ \sum_{x\in\mathscr{X}} q(x) & \text{if } \mathscr{X} \text{ is discrete} \end{cases}$$

Note that C is often intractable, especially when $\mathscr{X}$ or q is complicated.

Let g be an "instrumental" density on $\mathscr{X}$ from which we know how to generate samples (easily). Suppose g has the property that for some constant $\alpha > 0$,

$$q(x) \leq \alpha g(x), \qquad \text{for all } x \in \mathscr{X}.$$

The function $\alpha g(x)$ is known as an *envelope*.

Rejection sampling

1. Sample $X \sim g$ and $U \sim \text{Unif}(0, 1)$
2. If

$$U > \frac{q(X)}{\alpha g(X)}$$

then go to Step 1; otherwise return X

The returned value is a random sample from the density $f(x)$.

Even when the complete form of $f(x)$ is known, it is often still advantageous to use rejection sampling, by simply using $q(x) = f(x)$. As can be seen, the number of iterations until a sample is accepted follows a geometric distribution, and in each iteration, the probability of acceptance is

$$p_a = \frac{1}{\alpha} \int_{\mathscr{X}} q(x) \, \mathrm{d}x.$$

To reduce the expected number of iterations, p_a should be large. Equivalently, α should be small. Because α must satisfy $q(x) \leq \alpha g(x)$, given $g(x)$, the smallest possible α is

$$\alpha = \sup \frac{q(x)}{g(x)}.$$

To further reduce α, one has to choose appropriate g. Of course, one has to make sure g is easy to sample at the same time.

Sometimes, even though the instrument density g is easy to sample, it is tedious to write down the complete form of $g(x)$. Suppose $g(x) = Ch(x)$, where C is the known (but complicated) normalizing constant. Then the algorithm can be modified as follows. Let β be a constant such that

$$\frac{q(x)}{h(x)} \leq \beta$$

for all x with $q(x) > 0$. (Actually, $\beta = \alpha/C$.) Then in Step 2, replace $q(X)/\alpha g(X)$ with

$$\frac{q(X)}{\beta h(X)}.$$

Sometimes it is easier to express $f(x)$ as a function proportional to the sum of a set of functions,

$$f(x) \propto \sum_{k=1}^{m} q_k(x).$$

Superpose that we can find densities $g_1(x)$, ..., $g_m(x)$ that are easy to sample, and constants $\alpha_1, \ldots, \alpha_m$, such that

$$q_k(x) \leq \alpha_k g_k(x).$$

Then the rejection sampling can be modified as follows. Note that the complete form of each $g_k(x)$ has to be known.

Rejection sampling

1. Sample k from $\{1, \ldots, m\}$ with probabilities $p_k \propto \alpha_k$.
2. Sample $X \sim g_k$ and $U \sim \text{Unif}(0, 1)$
3. If

$$U > \frac{q_k(X)}{\alpha_k g_k(X)}$$

   then go to Step 1; otherwise return X

The returned value is a random sample from the density $f(x)$.

Proof: Let Y be a sampled value. To show that Y has density f, it suffices to show that for any set $A \subset \mathscr{X}$, $\mathbb{P}\{Y \in A\} = \int_A f(x)\,\mathrm{d}x$. From the procedure,

$$\begin{aligned}\mathbb{P}\{Y \in A\} &= \mathbb{P}\{X \in A \,|\, X \text{ is accepted}\} \\ &= \frac{\mathbb{P}\{X \in A \text{ and } X \text{ is accepted}\}}{\mathbb{P}\{X \text{ is accepted}\}}.\end{aligned}$$

Let $C = \alpha_1 + \cdots + \alpha_m$. Then

$$\begin{aligned}&\mathbb{P}\{X \in A \text{ and } X \text{ is accepted}\} \\ &= \sum_{k=1}^{m} p_k \mathbb{P}\{X \in A \text{ and } X \text{ is accepted} \,|\, X \text{ is from } g_k\} \\ &= C^{-1} \sum_{k=1}^{m} \alpha_k \mathbb{P}\{X \in A \text{ and } X \text{ is accepted} \,|\, X \text{ is from } g_k\}\end{aligned}$$

For each k, let

$$r(x) = \frac{q_k(x)}{\alpha_k g_k(x)}.$$

Then

$$\begin{aligned}
&\mathbb{P}\{X \in A \text{ and } X \text{ is accepted} \mid X \text{ is from } g_k\} \\
=&\mathbb{P}\{X \in A, U \leq r(X) \mid X \text{ is from } g_k\}
\end{aligned}$$

Given $X = x$, the probability that $U \leq r(X)$ is simply $r(x)$. Therefore,

$$\begin{aligned}
\mathbb{P}\{X \in A, U \leq r(X) \mid X \text{ is from } g_k\} &= \int \mathbf{1}\{x \in A\} r(x) g_k(x) \, \mathrm{d}x \\
&= \int_A \frac{q_k(x)}{\alpha_k g_k(x)} g_k(x) \, \mathrm{d}x \\
&= \frac{1}{\alpha_k} \int_A q_k(x) \, \mathrm{d}x.
\end{aligned}$$

Note the factor $1/\alpha_k$. As a result,

$$\mathbb{P}\{X \in A \text{ and } X \text{ is accepted}\} = C^{-1} \sum_{k=1}^{m} \int_A q_k(x)\,\mathrm{d}x.$$

In particular, applying the formula to $A = \mathscr{X}$,

$$\mathbb{P}\{X \text{ is accepted}\} = C^{-1} \sum_{k=1}^{m} \int q_k(x)\,\mathrm{d}x,$$

As a result,

$$\mathbb{P}\{Y \in A\} = \frac{\sum_{k=1}^{m} \int_A q_k(x)\,\mathrm{d}x}{\sum_{k=1}^{m} \int q_k(x)\,\mathrm{d}x} = \int_A f(x)\,\mathrm{d}x.$$

Example: Beta Variable

For $a, b > 0$, Beta(a, b) has a density

$$f(x) = \begin{cases} \dfrac{x^{a-1}(1-x)^{b-1}}{B(a,b)}, & 0 \le x \le 1 \\ 0 & \text{elsewhere} \end{cases}$$

and can be sampled by using the fact for independent $Y \sim$ Gamma$(a, 1)$ and $Z \sim$ Gamma$(b, 1)$,

$$\frac{Y}{Y+Z} \sim \text{Beta}(a, b).$$

One can try some easy alternatives without sampling Gamma distributions.

Method 1

This method works only when $a, b \geq 1$. If $a = b = 1$, then Beta(a, b) is simply Unif$(0, 1)$. Let $a > 1$ and $b > 1$. Then $f(x)$ is maximized

$$x_0 = \frac{a-1}{a+b-2}.$$

Choose g to be the uniform density on $[0, 1]$. Then the optimal α is

$$\alpha = \sup \frac{f(x)}{g(x)} = f(x_0).$$

The algorithm is implemented as follows.

- Draw X, $U \sim$ Unif$(0, 1)$ until $f(x_0)U \leq f(X)$ and then return X.

Method 2

Let $U \sim \text{Unif}(0,1)$. Observe that for any $a,\ b > 0$,

$$U^{1/a} \sim \text{Beta}(a,1), \quad 1 - U^{1/b} \sim \text{Beta}(1,b).$$

The densities of the distributions are

$$f_a(x) = ax^{a-1}, \quad f_b(x) = b(1-x)^{b-1}.$$

Therefore, if $g(x)$ is a mixture of $f_a(x)$ and $f_b(x)$,

$$g(x) = (1-\lambda)f_a(x) + \lambda f_b(x),$$

where $\lambda \in (0,1)$, then $f(x)/g(x)$ is bounded.
To sample from $g(x)$, one can run the following steps.

1. Sample U and $V \sim \text{Unif}(0,1)$
2. If $V > \lambda$, then return $X = U^{1/a}$; otherwise return $X = 1 - U^{1/b}$. □

Conditional distributions

Let A be a subset of $\mathscr{X}$. To sample X conditional on $X \in A$ one can use the following crude rejection sampling procedure

- Sample X until $X \in A$ and then return X.

However, sometimes more carefully designed rejection sampling is needed to make efficient sampling.

Example: Sampling from Normal Tails

Let $c > 0$. When c is large, to sample X from $N(0,1)$ conditional on $X \geq c$, the simple rejection sampling is very inefficient. To improve the efficiency, an exponential distribution can be used as the envelope.
It suffices to sample $X - c$. Given $X \geq c$, the conditional density of $X - c$ for $X \sim N(0,1)$ is

$$f(x) = \frac{\phi(x+c)}{1-\Phi(c)}, \quad x \geq 0,$$

where $\phi(x)$ is the standard normal density.
On the other hand, for $X \sim \text{Exp}(\lambda)$, the conditional density of $X - c$ is

$$g(x) = \lambda e^{-\lambda x}.$$

Given λ, the optimal

$$\alpha = \sup_{x \geq 0} \frac{\phi(x+c)/[1-\Phi(c)]}{\lambda e^{-\lambda x}} = \frac{\exp(\frac{1}{2}\lambda^2 - \lambda c)}{\sqrt{2\pi}\lambda[1-\Phi(c)]}.$$

The value of λ that minimizes the last expression is

$$\lambda = \frac{c + \sqrt{c^2+4}}{2}.$$

Then the optimal

$$p_a = \frac{1}{\alpha} = \frac{\sqrt{\pi}(c + \sqrt{c^2+4})[1-\Phi(c)]}{\sqrt{2e}}.$$

□

Gamma distributions

For r, $\lambda > 0$, Gamma(r, λ) has a density

$$f(x) = \frac{x^{r-1} e^{-x/\lambda}}{\lambda^r \Gamma(r)}, \quad x > 0.$$

1. Exp(λ) = Gamma$(1, \lambda)$. If $U \sim$ Unif$(0, 1)$, then $-\lambda \ln U \sim$ Exp(λ).
2. χ^2_k = Gamma$(\frac{k}{2}, \frac{1}{2})$.
3. If $X \sim$ Gamma(r, λ) and $c > 0$, then $cX \sim$ Gamma$(r, c\lambda)$.
4. If $X_i \sim$ Gamma(r_i, λ), $i = 1, \ldots, n$, are independent, then

$$X_1 + \cdots + X_n \sim \text{Gamma}(r_1 + \cdots + r_n, \lambda).$$

Some simple ways to sample special Gamma distributions.

1. If r is an integer, then Gamma(r, λ) can be sampled by

$$-\lambda \sum_{i=1}^{r} \ln U_i$$

where $U_1, \ldots, U_r$ are iid $\sim$ Unif$(0, 1)$

2. If r is an half integer $n/2$, then Gamma(r, λ) can be sampled by

$$2\lambda \sum_{i=1}^{n} X_i^2$$

where $X_1, \ldots, X_n$ are iid $\sim N(0, 1)$.

Note that the above methods take longer time as r increases, because they require sampling of more and more variables.

There are many rejection sampling algorithms for Gamma distributions Gamma$(r, 1)$ that are *universally fast*. That is, for such an algorithm, one can find a constant $M > 0$, such that for any $r > 0$, the expected number of iterations is no greater than M.
Typically, the cases $r \geq 1$ and $r < 1$ are treated separately. For $r \geq 1$, one universally fast algorithm, which is due to Best (1978), is as follows. Let

$$b = r - 1, \quad c = 3r - 3/4.$$

1. Draw U, V iid $\sim$ Unif$(0, 1)$ and set $W = U(1 - U)$, $Y = \sqrt{c/W}(U - 1/2)$, $X = b + Y$.
2. Set $Z = 64\, W^3\, V^2$. If $X \geq 0$, and either $Z \leq 1 - 2\, Y^2/X$ or $\ln(Z) \leq 2[b \ln(X/b) - Y]$, then return X; otherwise go to Step 1.

For $r \leq 1$, the following algorithm of Ahrens and Dieter (1974) is universally fast. Let

$$p = \frac{e}{r + e}$$

and the instrument density

$$g(x) = \begin{cases} prx^{r-1}, & x \in [0, 1] \\ (1-p)e^{-x+1} & x > 1. \end{cases}$$

That is, g is a mixture of Beta$(r, 1)$ and Exp(1) conditional on $(1, \infty)$. Then keep sampling $X \sim g$ and $U \sim$ Unif$(0, 1)$ until $U \leq f(X)/[Cg(X)]$, where $f(X) = x^{r-1}e^{-x}/\Gamma(a)$ is the density of Gamma$(r, 1)$ and C can be set equal to 1.39. In principle, the expected number of iterations is no greater than 1.39.

Box–Müller method.

Let Φ be the distribution function of $N(0,1)$. To sample $X \sim N(0,1)$, a straightforward method is to apply $\Phi^{-1}(U)$ with $U \sim \text{Unif}(0,1)$. There are two other often used methods.
This method does not need rejection sampling. However, the computation of sine and cosine functions can be time consuming.

1. Sample $U_1,\ U_2$ iid $\sim \text{Unif}(0,1)$
2. Set $R = \sqrt{-2 \ln U_1}$
3. Return

$$Z_1 = R\cos(2\pi U_2), \quad Z_2 = R\sin(2\pi U_2).$$

Then $Z_1,\ Z_2$ are iid $\sim N(0,1)$.

Proof: For any $r > 0$,

$$\begin{aligned}\mathbb{P}\{R \le r\} &= \mathbb{P}\left\{-2\ln U_1 \le r^2\right\} \\ &= \mathbb{P}\left\{U_1 \ge e^{-r^2/2}\right\} = 1 - e^{-r^2/2}\end{aligned}$$

So R has density $re^{-r^2/2}$. Recall that if $\boldsymbol{X}$ is a random vector of d dimension and has a density $f(\boldsymbol{x})$, while

$$\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots, \phi_d(\boldsymbol{x}))$$

is a 1-to-1 differentiable function, then $\boldsymbol{Y} = \boldsymbol{\phi}(\boldsymbol{X})$ is a random vector with density

$$g(\boldsymbol{y}) = \frac{f(\boldsymbol{x})}{|\det[\boldsymbol{\phi}'(\boldsymbol{x})]|}, \text{ with } \boldsymbol{x} = \boldsymbol{\phi}^{-1}(\boldsymbol{y}).$$

Let $\boldsymbol{X} = (R, U_2)$ and

$$\boldsymbol{\phi}(\boldsymbol{x}) = (r\cos(2\pi u), r\sin(2\pi u)).$$

Then $\boldsymbol{Y} = (Z_1, Z_2) = \boldsymbol{\phi}(\boldsymbol{X})$.

First, $\boldsymbol{X} = (R, U_2)$ has a joint density $f(r, u) = re^{-r^2/2}$. Second,

$$\boldsymbol{\phi}'(\boldsymbol{x}) = \begin{pmatrix} \cos(2\pi u) & -2\pi r \sin(2\pi u) \\ \sin(2\pi u) & 2\pi r \cos(2\pi u) \end{pmatrix}$$

and so $|\det[\boldsymbol{\phi}'(\boldsymbol{x})]| = 2\pi r$. So $\boldsymbol{Y} = (Z_1, Z_2)$ has density

$$g(z_1, z_2) = \frac{re^{-r^2/2}}{2\pi r} = \frac{e^{-r^2/2}}{2\pi},$$

with (r, u) satisfying

$$z_1 = r\cos(2\pi u_2), \quad z_2 = r\sin(2\pi u_2).$$

Then $r^2 = z_1^2 + z_2^2$ and so

$$g(z_1, z_2) = \frac{e^{-(z_1^2+z_2^2)/2}}{2\pi},$$

showing (Z_1, Z_2) are iid $\sim N(0, 1)$.

Marsaglia–Bray method.

Let D be the disk centered at $(0, 0)$ with radius 1. Then the method goes as follows:

1. use rejection sampling to sample (X, Y) from Unif(D)
2. set $R = \sqrt{X^2 + Y^2}$ and

$$T = \frac{2\sqrt{-\ln R}}{R}$$

3. return $Z_1 = TX$, $Z_2 = TY$

Again, Z_1 and Z_2 are iid $\sim N(0, 1)$.

Multivariate normal distributions

Let $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ be nonnegative definite. Recall that for $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\boldsymbol{A} \in \mathbb{R}^{n \times d}$,

$$\boldsymbol{A}\boldsymbol{X} \sim N(\boldsymbol{A}\boldsymbol{\mu}, \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^t).$$

Based on this, to sample $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, a method is to first compute the *Cholesky factorization* of $\boldsymbol{\Sigma}$, i.e.,

$$\boldsymbol{\Sigma} = \boldsymbol{A}\boldsymbol{A}^t,$$

with $\boldsymbol{A}$ being lower triangular, and then sample

$$\boldsymbol{\mu} + \boldsymbol{A}\boldsymbol{Z},$$

with all the coordinates of $\boldsymbol{Z}$ being iid $\sim N(0, 1)$.

Conditioning formula

Let $\boldsymbol{X}_1 \in \mathbb{R}^m$, $\boldsymbol{X}_2 \in \mathbb{R}^n$ be jointly normal, such that

$$\mathbb{E}\boldsymbol{X}_i = \boldsymbol{\mu}_i, \quad \mathrm{Cov}(\boldsymbol{X}_i, \boldsymbol{X}_j) = \boldsymbol{\Sigma}_{ij}.$$

If $\boldsymbol{\Sigma}_{22}$ is of full rank, then, conditional on $\boldsymbol{X}_2 = \boldsymbol{x}$, $\boldsymbol{X}_1 \sim N(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$, where

$$\begin{aligned}\boldsymbol{\mu}_c &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_2),\\ \boldsymbol{\Sigma}_c &= \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}.\end{aligned}$$

Note that $\boldsymbol{\Sigma}_c$ is independent of $\boldsymbol{x}$.

Squeezed rejection sampling

Recall that in the ordinary rejection sampling of $f(x)$, we have $q(x)$, $g(x)$, and α available, such that $f(x) \propto q(x)$, where the proportionality factor may be intractable, and $g(x)$ a density function such that $q(x) \leq \alpha g(x)$. Every time $X \sim g$ is drawn, $q(X)$ needs to be evaluated. If the evaluation is computationally costly, it slows down the simulation.

1. Squeezed rejection sampling can be used to preempt the evaluation of $q(X)$, hence improving simulation speed.
2. The sampling is still exact.

Choose a *squeezing function* s that is easy to compute, while satisfying

$$s(x) \leq q(x) \text{ for all } x$$

Squeezed rejection sampling

1. Sample $X \sim g$, $U \sim \text{Unif}(0,1)$, and compute $z = \alpha U g(X)$
2. If $z \leq s(X)$, then return X
   otherwise, if $z \leq q(X)$, then return X
   otherwise, go to Step 1.

In contrast, the ordinary rejection sampling directly goes from Step 1 to 3. Because of Step 2, the proportion of iterations in which evaluation of q is avoided is

$$\mathbb{P}\left\{U \leq \frac{s(X)}{\alpha g(X)}\right\} = \frac{1}{\alpha}\int s(x)\,\mathrm{d}x.$$

Adaptive rejection sampling

The method developed by Gilks and Wild works well for density functions on an interval of $\mathbb{R}$ which are continuous, differentiable, and *log-concave*. Log-concave densities are common in applications, for example, all $N(\mu, \sigma^2)$, Gamma(α, β) with $\alpha \geq 1$, and Beta(α, β) with α, $\beta \geq 1$ are log-concave.

Suppose the density f is a density defined on (a, b), where $-\infty \leq a < b \leq \infty$, such that

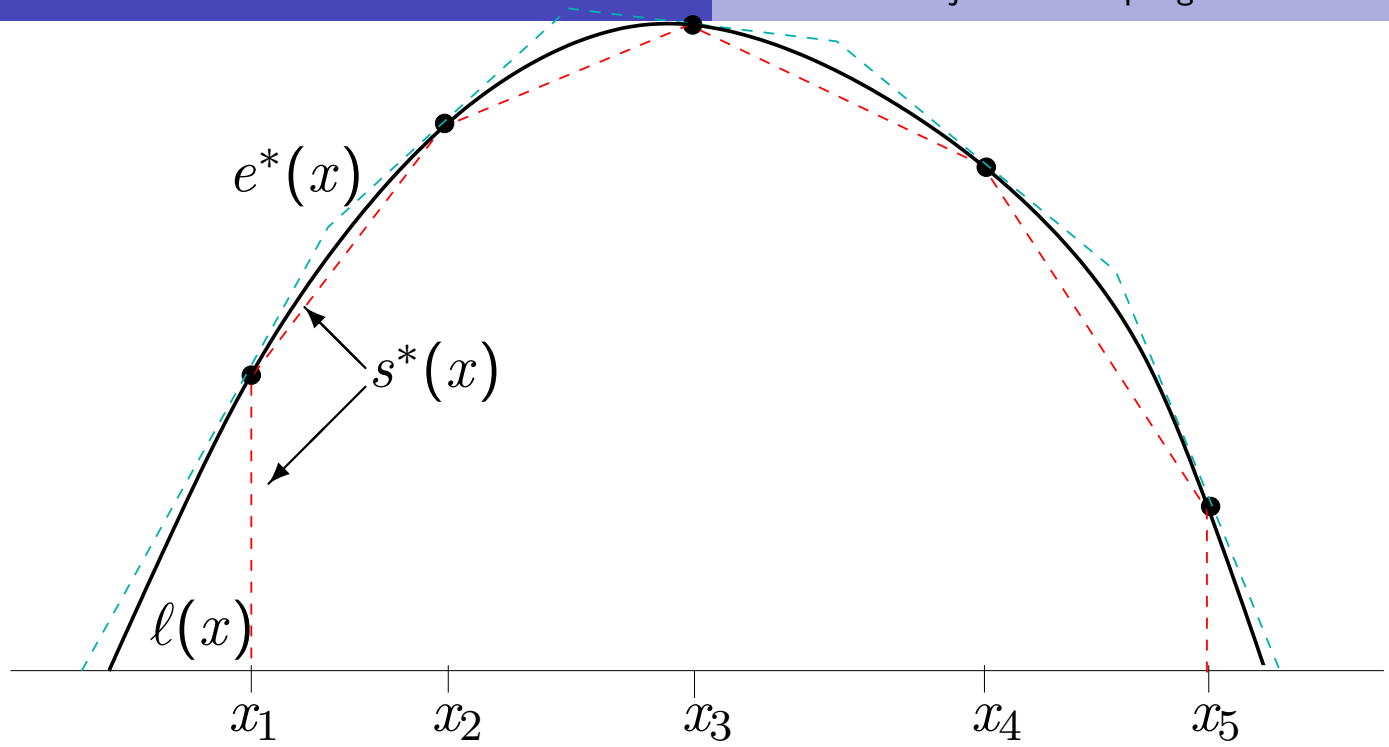
$$f(x) \propto q(x) = e^{\ell(x)}$$

where $\ell(x)$ is concave. The idea of the method is to construct two piecewise linear functions $e^*(x)$ and $s^*(x)$, such that

$$s^*(x) \leq \ell(x) \leq e^*(x)$$

Then $e(x) = \exp(e^*(x))$ and $s(x) = \exp(s^*(x))$ are envelope and squeezing functions respectively,

$$s(x) \leq q(x) \leq e(x).$$



The two piecewise linear functions s^* and e^* are shown above. First, select a set of points $a < x_1 < \cdots < x_n < b$. Define $s^*(x)$ to be $-\infty$ outside $[x_1, x_n]$ and to be linear on each $[x_i, x_{i+1}]$, such that its graph is the line segment connecting $(x_i, \ell(x_i))$ and $(x_{i+1}, \ell(x_{i+1}))$. To construct $e^*(x)$, draw tangent lines to the graph of ℓ at each $(x_i, \ell(x_i))$ and cut off the parts that are beyond the intersection points of the tangent lines. Since $\ell(x)$ is concave, it is guaranteed that $s^*(x) \leq \ell(x) \leq e^*(x)$. The "corner" points of $e^*(x)$ can be derived in close form. We omit the detail for brevity.

Clearly, $s(x) = \exp(s^*(x))$ can be used directly as a squeezing function. On the other hand, the $e(x)$ is not a density function. Instead, letting

$$C = \int e(x)\,\mathrm{d}x,$$

$g(x) = e(x)/C$ is a density function and satisfies $q(x) \leq Cg(x)$. The rest of the method is then the standard squeezed rejection sampling.
To see why the method often works so well, note the following.

- First, $g(x)$ is an exponential function on each of the intervals $(-\infty, x_1)$, $[x_1, x_2]$, $\ldots$, $[x_{n-1}, x_n]$, (x_n, ∞). Therefore, it can be sampled using the sampling for exponential distributions, which is easy.
- Second, C can be evaluated easily as well, again because $e(x)$ is piecewise exponential. Third, since $s^*(x)$ is piecewise linear, it is very easy to evaluate. As a result, $s(x) = \exp(s^*(x))$ is also easy to evaluate.

Sampling importance resampling (SIR)

Like rejection sampling, SIR uses an auxiliary density g, this time called an importance sampling function. After getting a (large) sample from g, it is resampled according to appropriate weights. This *approximately* generates a sample from the target density f.
The weights are called the standardized importance weights. Given $X_1, \ldots, X_m \sim g$, the weight for X_i is $w(X_i) \propto f(X_i)/g(X_i)$, specifically,

$$w(X_i) = \frac{f(X_i)/g(X_i)}{\sum_{k=1}^{m} f(X_k)/g(X_k)}.$$

- If $f = Cq$ for some unknown $C > 0$ but known q, the same $w(X_i)$ can be computed with $f(X_i)$ being replaced by $q(X_i)$.

SIR

Fix m and n.

1. Sample $X_1, \ldots, X_m$ iid from g.
2. Calculate $w(X_1), \ldots, w(X_m)$.
3. Draw n draws with replacement from $X_1, \ldots, X_m$ with probabilities $w(X_1), \ldots, w(X_m)$.

- The returned values $Y_1, \ldots, Y_n$ consist a random sample *approximately* following the density $f(x)$.
- In order for $Y_1, \ldots, Y_n$ to behave like an iid sample from $f(x)$, there should be $m \gg 1$ and $n/m \to 0$ as $m \to \infty$.

Let $h(x)$ be a function proportional to $f(x)/g(x)$. The above SIR is a procedure to sample from

$$f(x) \propto h(x)g(x).$$

It can be extended to

$$f(x) \propto \sum_{k=1}^{K} h_k(x)g_k(x)$$

where $g_k(x)$ are probability densities and $h_k(x) \geq 0$ positive functions. However, here the complete form of $g_k(x)$ has to be known. The procedure can be modified as follows.

SIR

Fix m and n.

1. For $k = 1, \ldots, K$, sample $X_{k1}, \ldots, X_{km}$ iid from g_k.
2. Calculate the weights

$$w_{kj} \propto h_k(X_{kj})$$

   for $k = 1, \ldots, K$, $j = 1, \ldots, m$, such that their total is 1.
3. Make n draws with replacement from $\{X_{kj}\}$ with probabilities w_{kj}.

To justify SIR, we show that when $m \gg 1$, Y_1 (or Y_2, Y_3, etc) has a distribution approximately equal to f. Let

$$C = \sum_{k=1}^{K} \int h_k(x) g_k(x) \, \mathrm{d}x.$$

Then

$$f(x) = C^{-1} \sum_{k=1}^{K} h_k(x) g_k(x).$$

We need to show that, if A is an arbitrary region, then, as $m \to \infty$,

$$\mathbb{P}\{ Y_1 \in A \} \to \int_A f(x) \, \mathrm{d}x.$$

Denote by $\boldsymbol{X}$ the set of X_{kj}, $k = 1, \ldots, K$, $j = 1, \ldots, m$. *Given* $\boldsymbol{X}$, in the resampling step, the probability that X_{kj} is drawn to be Y_1 is

$$w_{kj} = \frac{h_k(X_{kj})}{\sum_{k=1}^{K} \sum_{j=1}^{m} h_k(X_{kj})}.$$

Then

$$\begin{aligned}
\mathbb{P}\{Y_1 \in A \,|\, \boldsymbol{X}\} &= \sum_{k=1}^{K} \sum_{j=1}^{m} \mathbb{P}\{X_{kj} \text{ is drawn to be } Y_1\} \mathbf{1}\{X_{kj} \in A\} \\
&= \frac{\frac{1}{m} \sum_{k=1}^{K} \sum_{j=1}^{m} \mathbf{1}\{X_{kj} \in A\} h_k(X_{kj})}{\frac{1}{m} \sum_{k=1}^{K} \sum_{j=1}^{m} h_k(X_{kj})}.
\end{aligned}$$

By the Strong Law of Large Numbers, with probability 1, as $m \to \infty$, for each k

$$\frac{1}{m}\sum_{j=1}^{m} \mathbf{1}\left\{X_{kj} \in A\right\} h_k(X_{kj}) \to \int \mathbf{1}\left\{x \in A\right\} h_k(x) g_k(x)\,\mathrm{d}x$$

and so

$$\frac{1}{m}\sum_{k=1}^{K}\sum_{j=1}^{m} \mathbf{1}\left\{X_{kj} \in A\right\} h_k(X_{kj})$$

$$\to \int \mathbf{1}\left\{x \in A\right\} \sum_{k=1}^{K} h_k(x) g_k(x)\,\mathrm{d}x = C \int \mathbf{1}\left\{x \in A\right\} f(x)\,\mathrm{d}x$$

In particular, let A be the entire domain of possible values of X_{jk}. Then

$$\frac{1}{m}\sum_{k=1}^{K}\sum_{j=1}^{m} h_k(X_{kj}) \to C.$$

So with probability one, as $m \to \infty$,

$$\mathbb{P}\{Y_1 \in A \mid \boldsymbol{X}\} \to \int_A f(x)\,\mathrm{d}x.$$

Since we are only interested in the distribution of Y_1, we need to remove the dependence on $\boldsymbol{X}$. This requires finding the limit of $\mathbb{P}\{Y_1 \in A\}$ as $m \to \infty$. Note

$$\mathbb{P}\{Y_1 \in A\} = \mathbb{E}[\mathbb{P}\{Y_1 \in A \mid X_1, \ldots, X_m\}].$$

By the Dominated Convergence Theorem,

$$\lim_{m\to\infty} \mathbb{P}\{Y_1 \in A\} = \mathbb{E}\left[\lim_{m\to\infty} \mathbb{P}\{Y_1 \in A \mid \boldsymbol{X}\}\right] = \mathbb{E}\left[\int_A f(x)\,\mathrm{d}x\right] = \int_A f(x)\,\mathrm{d}x$$

Sampling using Markov chains

The sampling using Markov chains is a very important method. It is highly adaptive to various difficult sampling problems, especially high-dimensional problems. Although in most applications it is an approximate sampling method, it often has good performance. Another advantage of the method is the ease of its implementation.

The method is often used to estimate $\mathbb{E}[h(X)]$ for a function $h(X)$ of random variable X. In this context, the method is known as Markov chain Monte Carlo (MCMC). Here we will use this term, but only consider the sampling aspect.

Let $\mathscr{X}$ be a discrete or Euclidean space. Recall that a stochastic process $\{X_t, t = 0, 1, 2, \ldots\}$ taking values $\mathscr{X}$ is called a Markov chain if

$$\mathbb{P}\{X_t \in A \,|\, X_{t-1}, \ldots, X_1, X_0\} = \mathbb{P}\{X_t \in A \,|\, X_{t-1}\} \text{ a.s.}$$

for all $t > 0$ and (measurable) $A \subset \mathscr{X}$. The marginal distribution of X_0 is called the initial distribution of of the Markov chain (X_t). If every X_t has a conditional density given X_s, $0 \leq s < t$, then the condition can be written as

$$p(x_t \,|\, x_{t-1}, \ldots, x_1, x_0) = p(x_t \,|\, x_{t-1}).$$

The chain is homogeneous if for any a and b,

$$p(x_t = b \,|\, x_{t-1} = a)$$

is the same for all $t > 0$. Denote by $q(b \,|\, a)$ to be the conditional density and call $q(\cdot \,|\, \cdot)$ the transition kernel of the Markov chain.

A homogeneous Markov chain $(X_t, t = 0, 1, 2, \ldots)$ in general is not stationary, because the density p_t of X_t is given by

$$p_t(x) = \int q(x \mid y) p_{t-1}(y) \, \mathrm{d}y$$

and hence is not necessarily the same as p_{t-1}. However, if X_0 follows a distribution π that satisfies

$$\pi(x) = \int q(x \mid y) \pi(y) \, \mathrm{d}y,$$

then $p_t \equiv \pi$ for all $t \geq 0$. This then implies the stationarity of (X_t). Thus π is called a *stationary* distribution of (X_t).

Importantly, under certain conditions, no matter the initial distribution p_0, as $t \to \infty$, $p_t \to \pi$. The existence and uniqueness of stationary distribution, and the convergence of p_t are true in a large number of cases, in particular, when $\mathscr{X}$ is finite and $p(y \mid x) > 0$ for all x, $y \in \mathscr{X}$.

The idea of MCMC is to construct a Markov chain (X_t) such that it has the target distribution f as the unique stationary distribution. If this can be done, then one has a good reason to expect that X_t is a random sample from a distribution very close to f, as long as n is large enough. Generally speaking, it is relatively easy to specify the transition kernel of a Markov chain, however, it is much harder to determine the corresponding stationary distribution(s). So a basic question is, how to choose $q(\cdot \mid \cdot)$ for (X_t), such that f is *a* stationary distribution of (X_t)? If this question is answered, then one can proceed to find conditions to make sure that (X_t) has a unique stationary distribution and p_t converges to it.

Suppose f is known up to a multiplicative factor, i.e

$$f(x) = h(x)/C,$$

where $C > 0$ is a constant that is possibly intractable. If $q(\cdot \mid \cdot)$ satisfies the following *detailed balance* condition,

$$h(x)q(y \mid x) = h(y)q(x \mid y) \quad \text{for all } x \neq y \in \mathscr{X},$$

then f is a stationary distribution of (X_t).

Proof.

From the condition, $f(x)q(y \mid x) = f(y)q(x \mid y)$ for all x and $y \in \mathscr{X}$. Then

$$\int f(x)q(y \mid x)\,\mathrm{d}x = \int f(y)q(x \mid y)\,\mathrm{d}x = f(y)\int q(x \mid y)\,\mathrm{d}x = f(y)$$

where the last equality is due to the fact that given y, $q(x \mid y)$ is a probability density. □

Metropolis-Hastings (MH) algorithm

Detailed balance is the guiding principle for many MCMC algorithms. However, it does not say how to construct a kernel. One answer is provided by the well-known MH algorithm. It uses a *proposal distribution* $k(y \mid x)$ to implicitly construct a transition kernel satisfying the detailed balance condition. The construction has some similarity to rejection sampling.

MH Algorithm

Set an arbitrary x_0 with $h(x_0) > 0$. Then, for $t \geq 0$, given x_t, sample x_{t+1} as follows.

1. Draw $x^* \sim k(\cdot \mid x_t)$ and $U \sim \text{Unif}(0, 1)$.
2. Compute the *MH ratio* $R(x_t, x^*)$, where

$$R(x, y) = \frac{h(y)k(x \mid y)}{h(x)k(y \mid x)}$$

3. If $U \leq \min\{R(x_t, x^*), 1\}$, then set $x_{t+1} = x^*$, else set $x_{t+1} = x_t$.

We shall show that the MH algorithm satisfies the detailed balance condition. But first, note that $R(x_t, x^*)$ is always well-defined, because x^* can be sampled only if $f(x_t) > 0$ and $k(x^* \mid x) > 0$.
Since each x_{t+1} is sampled solely based on x_t, then (x_t) is a Markov chain. Let $q(y \mid x)$ be its transition kernel. Observe that for $y \neq x$,

$$\begin{aligned} h(x)q(y \mid x) > 0 &\iff h(x) > 0 \text{ and } q(y \mid x) > 0 \\ &\iff h(x) > 0,\ k(y \mid x) > 0,\ h(y) > 0,\ k(x \mid y) > 0 \\ &\iff h(y) > 0 \text{ and } q(x \mid y) > 0 \\ &\iff h(y)q(y \mid x) > 0. \end{aligned}$$

Therefore, the detailed balance condition is satisfied if $h(x)q(y \mid x) = 0$.

Next, if $h(x)q(y \mid x) > 0$, then $h(y)q(x \mid y) > 0$. In this case,

$$\begin{aligned} q(y \mid x) &= k(y \mid x) \min\{R(x, y), 1\} \\ &= k(y \mid x) \min\left\{\frac{h(y)k(x \mid y)}{h(x)k(y \mid x)}, 1\right\} \end{aligned}$$

Then

$$\begin{aligned} h(x)q(y \mid x) &= h(x)k(y \mid x) \min\left\{\frac{h(y)k(x \mid y)}{h(x)k(y \mid x)}, 1\right\} \\ &= \min\left\{h(y)k(x \mid y),\ h(x)k(y \mid x)\right\}. \end{aligned}$$

Likewise, since $h(y)q(x \mid y) > 0$, it is also equal to the right hand side of the equation. Therefore, the detailed balance is also satisfied for $x \neq y$ with $h(x)k(y \mid x) > 0$.

Bayesian inference

MCMC methods like the MH algorithm are particularly popular tools for Bayesian inference. Let $f(x \mid \theta)$ be a parametric family of probability densities and $p(\theta)$ be a prior distribution of parameter θ. Given data x, the posterior distribution of θ is

$$p(\theta \mid x) \propto h(\theta) := f(x \mid \theta)p(\theta).$$

A simple strategy is to use the prior $p(\theta)$ as a proposal distribution; i.e., for any θ_t, $k(\theta^* \mid \theta_t) = p(\theta^*)$. Then the MH ratio is simply the likelihood ratio

$$R(\theta_t, \theta^*) = \frac{h(\theta^*)p(\theta^*)}{h(\theta_t)p(\theta_t)} = \frac{f(x \mid \theta^*)}{f(x \mid \theta_t)}.$$

and the M-H algorithm becomes

- Draw $\theta^* \sim p$ and $U \sim \text{Unif}(0, 1)$. If $f(x \mid \theta^*) \geq Uf(x \mid \theta_t)$, then set $x_{t+1} = x^*$; else set $x_{t+1} = x_t$.

Gibbs sampling

Gibbs sampling is specifically adapted for multidimensional target distributions. It works by sequentially sampling from the *conditional* distributions of parts of the random vector, which are often available in closed form.

Let $\boldsymbol{X}$ be a random vector. Suppose we can partition it into random subvectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_p$, such that, for each $i = 1, \ldots, p$, the conditional distribution of $\boldsymbol{X}_i$ given all the other $\boldsymbol{X}_j$'s is easy to sample. With a little abuse of notation, denote by

$$f(\boldsymbol{x}_i \,|\, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_p)$$

the conditional density of $\boldsymbol{X}_i$ given $\boldsymbol{X}_j = \boldsymbol{x}_j$ for $j \neq i$.

Gibbs sampler

Set $t = 0$ and $\boldsymbol{x}^{(0)} = (\boldsymbol{x}_1^{(0)}, \ldots, \boldsymbol{x}_p^{(0)})$. Then, for $t \geq 0$, given $\boldsymbol{x}^{(t)}$, sample $\boldsymbol{x}^{(t+1)}$ as follows.

1. Sample i uniformly from $\{1, \ldots, p\}$.
2. Draw $\boldsymbol{x}_i^{(t+1)}$ from $f(\boldsymbol{x}_i \,|\, \boldsymbol{x}_1^{(t)}, \ldots, \boldsymbol{x}_{i-1}^{(t)}, \boldsymbol{x}_{i+1}^{(t)}, \ldots, \boldsymbol{x}_p^{(t)})$
3. Set $\boldsymbol{x}_j^{(t+1)} = \boldsymbol{x}_j^{(t)}$ for all $j \neq i$.

Let $q(\boldsymbol{y} \mid \boldsymbol{x})$ be the transition kernel of the Markov chain in the Gibbs sampler. Note that for $\boldsymbol{x} \neq \boldsymbol{y}$, $q(\boldsymbol{y} \mid \boldsymbol{x}) > 0$ only when there is exactly one i such that $\boldsymbol{x}_i \neq \boldsymbol{y}_i$ and for all $j \neq i$, $\boldsymbol{x}_i = \boldsymbol{y}_i$. In this case, writing $\boldsymbol{x}_{-i} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_p)$,

$$\begin{aligned} f(\boldsymbol{x})q(\boldsymbol{y} \mid \boldsymbol{x}) &= \frac{1}{n} f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p) f(\boldsymbol{y}_i \mid \boldsymbol{x}_{-i}) \\ &= \frac{1}{n} f(\boldsymbol{x}_i \mid \boldsymbol{x}_{-i}) f(\boldsymbol{x}_{-i}) f(\boldsymbol{y}_i \mid \boldsymbol{x}_{-i}) \end{aligned}$$

where the factor $1/n$ is the probability that i is sampled. Because, $\boldsymbol{y}_{-i} = \boldsymbol{x}_{-i}$,

$$\begin{aligned} f(\boldsymbol{x})q(\boldsymbol{y} \mid \boldsymbol{x}) &= \frac{1}{n} f(\boldsymbol{x}_i \mid \boldsymbol{y}_{-i}) f(\boldsymbol{y}_{-i}) f(\boldsymbol{y}_i \mid \boldsymbol{y}_{-i}) \\ &= \frac{1}{n} f(\boldsymbol{x}_i \mid \boldsymbol{y}_{-i}) f(\boldsymbol{y}) \\ &= f(\boldsymbol{y})q(\boldsymbol{x} \mid \boldsymbol{y}). \end{aligned}$$

Therefore, the detailed balance condition is satisfied.

Usually, instead of randomly sampling the index i, the following variant is used. Note that for $p > 1$, the variant does not satisfy the detailed balance condition. However, under certain conditions, the Markov chain $\boldsymbol{x}^{(t)}$ generated by the sampler still has f as the stationary distribution.

Gibbs sampler with cycles

Set $t = 0$ and $\boldsymbol{x}^{(0)} = (\boldsymbol{x}_1^{(0)}, \ldots, \boldsymbol{x}_p^{(0)})$. Then, for $t \geq 0$, given $\boldsymbol{x}^{(t)}$, sample $\boldsymbol{x}^{(t+1)}$ as follows.

Draw $\boldsymbol{x}_1^{(t+1)}$ from $f(\boldsymbol{x}_1 \mid \boldsymbol{x}_2^{(t)}, \boldsymbol{x}_3^{(t)}, \boldsymbol{x}_4^{(t)}, \ldots, \boldsymbol{x}_p^{(t)})$.

Draw $\boldsymbol{x}_2^{(t+1)}$ from $f(\boldsymbol{x}_2 \mid \boldsymbol{x}_1^{(t+1)}, \boldsymbol{x}_3^{(t)}, \boldsymbol{x}_4^{(t)}, \ldots, \boldsymbol{x}_p^{(t)})$.

Draw $\boldsymbol{x}_3^{(t+1)}$ from $f(\boldsymbol{x}_3 \mid \boldsymbol{x}_1^{(t+1)}, \boldsymbol{x}_2^{(t+1)}, \boldsymbol{x}_4^{(t)}, \ldots, \boldsymbol{x}_p^{(t)})$.

.....

Draw $\boldsymbol{x}_{p-1}^{(t+1)}$ from $f(\boldsymbol{x}_{p-1} \mid \boldsymbol{x}_1^{(t+1)}, \boldsymbol{x}_2^{(t+1)}, \ldots, \boldsymbol{x}_{p-2}^{(t+1)}, \boldsymbol{x}_p^{(t)})$.

Draw $\boldsymbol{x}_p^{(t+1)}$ from $f(\boldsymbol{x}_{p-1} \mid \boldsymbol{x}_1^{(t+1)}, \boldsymbol{x}_2^{(t+1)}, \ldots, \boldsymbol{x}_{p-1}^{(t+1)})$.

Monte Carlo Integration

Basics of Monte Carlo Integration

Monte Carlo is in particular useful for multidimensional problems that require the estimation of

$$\mu = \mathbb{E}h(\boldsymbol{X}),$$

where $\boldsymbol{X}$ is a random vector and h a function.
The form of a MC method is simple. Basically, it will sample $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ iid $\sim \boldsymbol{X}$ and approximate μ by

$$\widehat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} h(\boldsymbol{X}_i).$$

1. $\widehat{\mu}_n$ is *consistent*: by the Strong Law of Large Numbers, with probability 1,

$$\widehat{\mu}_n \to \mathbb{E}h(\boldsymbol{X}), \quad \text{as} \quad n \to \infty.$$

2. $\widehat{\mu}_n$ is *unbiased*:

$$\mathbb{E}(\widehat{\mu}_n) = \mathbb{E}h(\boldsymbol{X}).$$

3. Variance

$$\mathrm{Var}(\widehat{\mu}_n) = \frac{\mathrm{Var}[h(\boldsymbol{X})]}{n} = \frac{\sigma^2}{n}$$

and σ^2 can be estimated by

$$\widehat{\mathrm{Var}}(\widehat{\mu}_n) = \frac{1}{n-1}\sum_{i=1}^{n}[h(\boldsymbol{X}_i) - \widehat{\mu}_n]^2.$$

MC for Evaluating Integrals

MC essentially is a method to numerically integrate functions. It therefore can be used in problems that have no obvious connection to random variables. A typical case is the evaluation of a "usual" integral, $\int H(x)\,\mathrm{d}x$, where x may be of multidimensional. A general idea is to factorize $H(x) = f(x)h(x)$, where $f(x)$ is a pdf, and then approximate the integral by

$$\int H(x)\,\mathrm{d}x \approx \frac{1}{n}\sum_{i=1}^{n} h(X_i), \quad X_1, \ldots, X_n \text{ iid} \sim f(x).$$

For example, to compute

$$I = \int_{-\infty}^{\infty} \ln|x| e^{-(x+1)^2/8} \, \mathrm{d}x,$$

set $h(x) = \sqrt{8\pi} \ln|x|$ and $f(x) = e^{-(x+1)^2/8}/\sqrt{8\pi}$. Since $f(x)$ is the density of $N(-1, 4)$,

$$I \approx \frac{\sqrt{8\pi}}{n} \sum_{i=1}^{n} \ln|X_i|, \quad X_i \sim N(-1, 4).$$

Generally speaking, the choice of the factorization $H(x) = f(x)h(x)$ is critical to how well the MC approximation works.

MC for European Call Option

Let

$$S(t) = \text{price of a stock at time } t.$$

A European call option grants its holder the right to buy the stock at a fixed price ("strike price") K at time T from now.

- If $S(T) > K$, the holder exercises the option to buy the stock at price K and immediately sell it to realize a profit of $S(T) - K$.
- If $S(T) \leq K$, the holder lets the option expire worthless.

The payoff at time T is

$$[S(T) - K]^+ = \max\{S(T) - K, 0\}.$$

If the continuously compounded interest rate is r, then the expected present value of the option is

$$P = \mathbb{E}\{e^{-rT}[S(T) - K]^+\}.$$

Under the Black-Scholes model for S,

$$\frac{\mathrm{d}S(t)}{S(t)} = r\,\mathrm{d}t + \sigma\,\mathrm{d}W(t),$$

where W is a standard Brownian motion, one has

$$S(T) = S(0)e^{(r-\frac{1}{2}\sigma^2)T+\sigma W(T)}.$$

Since

$$W(T) = \sqrt{T}Z, \quad \text{with } Z \sim N(0,1),$$

P can be evaluated explicitly. Nevertheless, as an illustration of the Monte Carlo method, P can be approximated by

$$\frac{e^{-rT}}{n}\sum_{i=1}^{n}\left[S(0)e^{(r-\frac{1}{2}\sigma^2)T+\sigma\sqrt{T}Z_i} - K\right]^+$$

with $Z_1, \ldots, Z_n$ iid $\sim N(0,1)$.

Path-dependent MC

To value more complicated derivative securities with more complicated models of $S(t)$, one often has to simulate $S(t)$ over multiple intermediate dates.

1. The payoff of a derivative security may depend explicitly on the values of $S(t)$ at multiple dates.
2. The transitions of $S(t)$ may not be known exactly and hence $[0, T]$ has to be divided into smaller subintervals to obtain a reasonable approximation to the distribution of $S(T)$.

Asian options are path-dependent options. Their payoffs depend on the average level of $S(t)$, for example, the payoff $(\overline{S} - K)^+$ with

$$\overline{S} = \frac{1}{m} \sum_{j=1}^{m} S(t_j)$$

for some fixed set of dates $0 = t_0 < t_1 < \cdots < t_m = T$.

To estimate $\mathbb{E}[(\overline{S} - K)^+]$, a simple MC procedure is as follows.

1. For $i = 1, \ldots, N$, simulate $S(t_1^{(i)}), \ldots, S(t_m^{(i)})$ and set

$$\overline{S}_i = \frac{1}{m} \sum_{j=1}^{m} S(t_j^{(i)})$$

2. The average of $(\overline{S}_1 - K)^+, \ldots, (\overline{S}_N - K)^+$ is taken as an estimate of $\mathbb{E}[(\overline{S} - K)^+]$.

Similarly, MC procedure can be used to price other path dependent options.

- Barrier options. A typical example is one that gets terminated worthless if $S(t)$ crosses a level specified beforehand. For instance, a *down-and-out call* with barrier b, strike K and expiration T has payoff

$$\mathbf{1}\{\text{all } S(t_i) \geq b\}\,(S(T) - K)^+.$$

- Lookback options. In calls expiring at $T = t_n$, the strike price is $\min S(t_i)$, so the payoff is then $S(t_n) - \min S(t_i)$. In lookback puts, the strike price is $\max S(t_i)$, so the payoff is $\max S(t_i) - S(t_n)$.

Evaluating Ratio of Normalizing Constants

Suppose a target density p is known up to a hard-to-know factor Z, i.e., $p(\boldsymbol{x}) = q(\boldsymbol{x})/Z$ with

$$Z = \int q(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}.$$

We have seen procedures that are designed to draw from p without having to evaluate Z. However, in many cases, Z has to be taken into consideration one way or another other. For example, in likelihood ratio test, one is interested in

$$\frac{p_2(\boldsymbol{x})}{p_1(\boldsymbol{x})}$$

where $p_i(\boldsymbol{x})$ are likelihoods of data $\boldsymbol{x}$ under two competing hypotheses. Again, in many cases, it is known that $p_i(\boldsymbol{x}) \propto q_i(\boldsymbol{x})$. Because

$$\frac{p_2(\boldsymbol{x})}{p_1(\boldsymbol{x})} = \frac{q_2(\boldsymbol{x})/Z_2}{q_1(\boldsymbol{x})/Z_1} = \frac{q_2(\boldsymbol{x})}{q_1(\boldsymbol{x})} \frac{Z_1}{Z_2}, \quad \text{where} \quad Z_i = \int q_i(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}.$$

it is necessary to evaluate the ratio Z_2/Z_1.

Ideally, a MC integration procedure will *not* evaluate Z_i separately. A relatively straightforward solution is as follows. We know that using the rejection method, one can sample $\boldsymbol{X}_i$ iid from p_1. Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ be an iid sample from p_1. Then a MC estimate of (Z_2/Z_1) is

$$\widehat{(Z_2/Z_1)} = \frac{1}{n}\sum_{i=1}^{n} \frac{q_2(\boldsymbol{X}_i)}{q_1(\boldsymbol{X}_i)}.$$

This follows from

$$\mathbb{E}\left[\frac{q_2(\boldsymbol{X})}{q_1(\boldsymbol{X})}\right] = \int \frac{q_2(\boldsymbol{x})}{q_1(\boldsymbol{x})} p_1(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \frac{Z_2}{Z_1}\int p_2(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \frac{Z_2}{Z_1}. \qquad \square$$

Bayesian Missing Data Problem

In a missing data problem, suppose $\boldsymbol{Y} \sim p_{\boldsymbol{Y}}(\boldsymbol{y} \,|\, \boldsymbol{\theta})$ but only $\boldsymbol{X} = M(\boldsymbol{Y})$ is observed, where M is a many-to-fewer mapping.
In the frequentist approach to the problem, in order to estimate θ by the MLE method, at least, one has to evaluate

$$L(\boldsymbol{\theta} \,|\, \boldsymbol{x}) = p_{\boldsymbol{X}}(\boldsymbol{x} \,|\, \boldsymbol{\theta}) = \int_{M(\boldsymbol{y})=\boldsymbol{x}} p_{\boldsymbol{Y}}(\boldsymbol{y} \,|\, \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{y}.$$

The integral is often difficult to compute analytically.
Using MC integration, one designs a suitable function h and a distribution P, then samples $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n$ from P to get

$$L(\boldsymbol{\theta} \,|\, \boldsymbol{x}) \approx \frac{1}{n} \sum_{i=1}^{n} h(\boldsymbol{Y}_i).$$

In the Bayesian approach to the problem, let $\boldsymbol{\theta}$ be assigned with a prior distribution $\pi(\boldsymbol{\theta})$. One is interested in the posterior distribution of $\boldsymbol{\theta}$ when the data is observed. If the full data $\boldsymbol{Y} = \boldsymbol{y}$ were observed, then we can evaluate the posterior distribution of $\boldsymbol{\theta}$ given $\boldsymbol{y}$. By Bayes formula,

$$f(\boldsymbol{\theta} \,|\, \boldsymbol{y}) \propto p_{\boldsymbol{Y}}(\boldsymbol{y} \,|\, \boldsymbol{\theta})\pi(\boldsymbol{\theta}).$$

If only $\boldsymbol{X} = \boldsymbol{x}$ is observed, then the posterior distribution is

$$f(\boldsymbol{\theta} \,|\, \boldsymbol{x}) = \int f(\boldsymbol{\theta} \,|\, \boldsymbol{y}) p(\boldsymbol{y} \,|\, M(\boldsymbol{y}) = \boldsymbol{x}) \,\mathrm{d}\boldsymbol{y}.$$

Provided that $f(\boldsymbol{\theta} \,|\, \boldsymbol{y})$ has a closed form, a MC integration method may sample

$$\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n \sim p(\boldsymbol{y} \,|\, M(\boldsymbol{y}) = \boldsymbol{x})$$

and approximate

$$f(\boldsymbol{\theta} \,|\, \boldsymbol{x}) \approx \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{\theta} \,|\, \boldsymbol{Y}_i). \qquad \square$$

Importance Sampling, A First Look

To evaluate $\mu = \mathbb{E}h(\boldsymbol{X})$, it often works better *not* to directly draw from the distribution of $\boldsymbol{X}$, but instead to draw from a different, auxiliary distribution. Assume $\boldsymbol{X} \sim p$. Let q be another distribution, such that $q(\boldsymbol{x}) > 0$ wherever $p(\boldsymbol{x}) > 0$. Then we can approximate μ by

$$\mu \approx \frac{1}{n} \sum_{i=1}^{n} \frac{h(\boldsymbol{Y}_i) p(\boldsymbol{Y}_i)}{q(\boldsymbol{Y}_i)},$$

with $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n \sim q$, because

$$\begin{aligned} \mathbb{E}\left[\frac{h(\boldsymbol{Y}) p(\boldsymbol{Y})}{q(\boldsymbol{Y})}\right] &= \int \frac{h(\boldsymbol{y}) p(\boldsymbol{y})}{q(\boldsymbol{y})} q(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} \\ &= \int h(\boldsymbol{y}) p(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} = \mathbb{E}[h(\boldsymbol{X})]. \end{aligned}$$

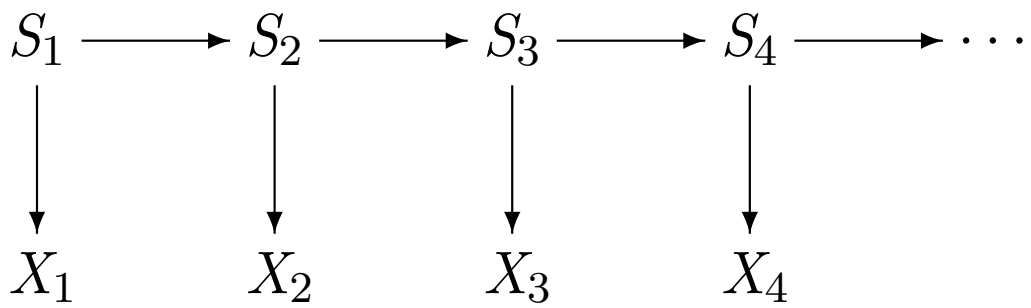
This method is called "*importance sampling*", which is an important method to reduce the variance of MC estimate.

State-Space Model

The state-space model is a dynamic system that consists of two parts:

$$\begin{aligned} \text{(state equation):} \quad & S_t \sim q_t(s_t \mid S_{t-1}, \theta) \\ \text{(observation equation):} \quad & X_t \sim f_t(x_t \mid S_t, \phi) \end{aligned}$$

1. S_t and X_t can be multi-dimensional.
2. Only $X_1, X_2, \ldots,$ $\ldots$ are observed, $S_1, S_2, \ldots$ are hidden.
3. θ and ϕ are system parameters.



Non-linear filtering aims to estimate, for $t \geq 1$, the value of S_t given the up-to-date history of the observations

$$X_1 = x_1, \ldots, X_t = x_t.$$

For example, if the system parameters are known, then the least-square-error "on-line" estimate of S_t is the Bayes solution

$$\widehat{s}_t = \mathbb{E}[S_t \mid x_1, \ldots, x_t] = \int s_t p(s_t \mid x_1, \ldots, x_t) \, \mathrm{d}s_t.$$

Depending on questions being asked, median, mode, etc. of the conditional distribution of S_t given $x_1, \ldots, x_t$ can also be used as estimators.
The question is how to *compute* a given estimator.

1. Linear state-space model: both f_t and q_t are linear Gaussian conditional distributions, i.e.,

$$S_t = A_t S_{t-1} + \varepsilon_t, \quad X_t = B_t S_t + \delta_t$$

   where A_t, B_t are matrices, ε_t and δ_t are independent Gaussian random vectors with mean 0.
   In this case, the estimate can be obtained analytically, and the resulting algorithm is the famous Kalman filter.

2. Hidden Markov model (HMM): S_t can only take a finite number of values, or 'states".
   Dynamic programming method can be used to evaluate the integrals.

Other than these two cases, the evaluation of the integrals is difficult and in most cases has to resort to MC. A popular algorithm based on MC integration is the so-called "particle filtering".

Partical Filtering

Henceforth, write

$$x_{1:t} = (x_1, \ldots, x_t), \quad s_{1:t} = (s_1, \ldots, s_t), \quad \text{etc.}$$

Particle filtering at each time t keeps a finite set $s_t^{(1)}, \ldots, s_t^{(m)}$ to approximately follow the analytically intractable $p(s_t \mid x_{1:t})$. Each $s_t^{(j)}$ is referred to as a "particle". Then mean, median, etc. can be estimated using the particles, for example,

$$\mathbb{E}[s_t \mid x_{1:t}] \approx \text{Mean of } s_t^{(1)}, \ldots, s_t^{(m)},$$
$$\text{Median}[s_t \mid x_{1:t}] \approx \text{Median of } s_t^{(1)}, \ldots, s_t^{(m)}.$$

For $t = 1$,

$$p(s_1 \mid x_1) \propto f_1(x_1 \mid s_1) q_1(s_1)$$

is typically tractable. So one can set

$$s_1^{(1)}, \ldots, s_1^{(m)} \text{ i.i.d.} \sim p(s_1 \mid x_1)$$

The question is how to update the set of particles to approximate $p(s_{t+1} \mid x_{1:t+1})$ when x_{t+1} comes in. First,

$$p(s_{t+1} \mid x_{1:t+1}) = \int p(s_t, s_{t+1} \mid x_{1:t}, x_{t+1}) \, \mathrm{d}s_t.$$

Apply Bayesian formula to s_t, s_{t+1} and x_{t+1}, while treating $x_{1:t}$ as a fixed parameter. Then

$$p(s_t, s_{t+1} \mid x_{1:t}, x_{t+1}) \propto f(x_{t+1} \mid s_t, s_{t+1}, x_{1:t}) p(s_{t+1} \mid s_t, x_{1:t}) p(s_t \mid x_{1:t}).$$

By the state equation and observation equation, the right hand side is

$$f(x_{t+1} \mid s_{t+1}) p(s_{t+1} \mid s_t) p(s_t \mid x_{1:t}).$$

Then

$$p(s_{t+1} \mid x_{1:t+1}) \propto f_{t+1}(x_{t+1} \mid s_{t+1}) \int q_{t+1}(s_{t+1} \mid s_t) p(s_t \mid x_{1:t}) \, \mathrm{d}s_t$$

If we write

$$p(s_{t+1} \mid x_{1:t}) = \int q_{t+1}(s_{t+1} \mid s_t) p(s_t \mid x_{1:t}) \, \mathrm{d}s_t, \qquad (\dagger)$$

then

$$p(s_{t+1} \mid x_{1:t+1}) \propto f_{t+1}(x_{t+1} \mid s_{t+1}) p(s_{t+1} \mid x_{1:t}).$$

Therefore, if we can sample from $p(s_{t+1} \mid x_{1:t})$, we can sample from $p(s_{t+1} \mid x_{1:t+1})$ using rejection sampling or SIR. Now if $p(s_t \mid x_{1:t})$ can be sampled, then $p(s_{t+1} \mid x_{1:t})$ can be sampled. This leads to

1) sample s_{t+1}^* from $p(s_{t+1} \mid x_{1:t})$ as follows
   (a) sample s_t from $p(s_t \mid x_{1:t})$;
   (b) given s_t, sample s_{t+1}^* from $q_{t+1}(s_{t+1} \mid s_t)$;
2) fixing a constant a such that $f_{t+1}(x_{t+1} \mid s_{t+1}) \leq a$ for all possible s_{t+1}, accept s_{t+1}^* with probability $f_{t+1}(x_{t+1} \mid s_{t+1}^*)/a$, otherwise go back to step 1.

The hope is that at each step t, $s_t^{(1)}, \ldots, s_t^{(m)}$ approximately follow $p(s_t \mid x_{1:t})$. Then one can repeat the above procedure, except that in step 1), s_t is sampled uniformly from $s_t^{(1)}, \ldots, s_t^{(m)}$. This will generate a new set of particles $s_{t+1}^{(1)}, \ldots, s_{t+1}^{(m)}$ that hopefully approximately follow $p(s_{t+1} \mid x_{1:t+1})$. The procedure can then continue.

Alternatively, SIR can be used:

1. For each $j = 1, \ldots, m$, sample $s_{t+1}^{(j*)}$ from $q_{t+1}(s_{t+1} \mid s_t^{(j)})$.
2. Generate $s_{t+1}^{(1)}, \ldots, s_{t+1}^{(m)}$ by resampling from $s_{t+1}^{(1*)}, \ldots, s_{t+1}^{(m*)}$ with replacement, with probabilities proportional to $f_{t+1}(x_{t+1} \mid s_{t+1}^{(j*)})$. □

The same procedure can be used for updating prediction. Here one needs to sample $p(s_{t+2} \mid x_{1:t+1})$. This can be done by

$$p(s_{t+2} \mid x_{1:t+1}) \propto \int p(s_{t+2} \mid s_{t+1}) p(s_{t+1} \mid x_{1:t+1}) \, \mathrm{d}s_{t+1}.$$

The formula has already occurred in (†).
If after estimating S_t or predicting S_{t+1} based on $X_{1:t}$, an action is done to modify the parameters of the state equation, then the system has feedback. In this case, the above procedure still applies.

Computing time for unbiased estimator

Three important considerations

- Computing time
- Bias
- Variance.

Suppose

$$\widehat{c}_n = \frac{1}{n}\sum_{i=1}^{n} C_i$$

is an unbiased estimator of c, with

$$C_i \quad \text{iid}, \quad \mathbb{E}(C_i) = c, \quad \text{Var}(C_i) = \sigma^2 < \infty.$$

By CLT, $\sqrt{n}(\widehat{c}_n - c) \Rightarrow N(0, \sigma^2)$, so roughly speaking, with n replications, the magnitude of error in the estimator $\widehat{c}_n$ is about $\sigma/\sqrt{n}$.

Suppose s is the available time budget to produce an estimate. Since generating each C_i takes some time τ_i, the number of C_i that can be generated is finite. The issue is how τ_i affects the accuracy of $\widehat{c}_n$.
Case 1: $\boldsymbol{\tau_i \equiv \tau}$ nonrandom. Since $n = \lfloor s/\tau \rfloor$, for $s \gg 0$, the magnitude of error in $\widehat{c}_n$ is about $(\sigma/\sqrt{s})\sqrt{\tau}$. So, given a fixed large s, the magnitude of error is proportional to $\sqrt{\tau}$.
Case 2: $\boldsymbol{\tau_i}$ is random. In many cases, τ_i can vary substantially across replications. With τ_i being random, n is random as well:

$$n = N(s) = \max\left\{n \geq 0 : \tau_1 + \cdots + \tau_n \leq s\right\}.$$

Since τ_i are iid $\sim \tau$, if $\mathbb{E}\tau < \infty$, then for $s \gg 0$,

$$N(s) \approx s/\mathbb{E}\tau$$

and hence the magnitude of error $\approx (\sigma/\sqrt{s})\sqrt{\mathbb{E}\tau}$.

Bias

Some MC are biased for all finite sample sizes but become asymptotically unbiased as $n \to \infty$. For these estimators, the bias is negligible.
Example: For random τ_i, in general $\mathbb{E}[\widehat{C}_{N(s)}] \neq c$, where s is the available running time. Since

$$\sqrt{s}(\widehat{C}_{N(s)} - C) \Rightarrow N(0, \sigma^2 \mathbb{E}\tau)$$

as $s \to \infty$, the bias decreases in the same rate as $1/\sqrt{s}$.

Example: Let (X_i, Y_i) be iid $\sim (X, Y)$. Then

$$\frac{\overline{X}_n}{\overline{Y}_n} \text{ is a biased estimator of } \frac{\mathbb{E}(X)}{\mathbb{E}(Y)}$$

for all n because

$$\mathbb{E}\left[\frac{\overline{X}_n}{\overline{Y}_n}\right] \neq \frac{\mathbb{E}(\overline{X}_n)}{\mathbb{E}(\overline{Y}_n)} = \frac{\mathbb{E}(X)}{\mathbb{E}(Y)}.$$

However, actually, as $n \to \infty$

$$\sqrt{n}\left[\frac{\overline{X}_n}{\overline{Y}_n} - \frac{\mathbb{E}(X)}{\mathbb{E}(Y)}\right]$$

is asymptotically normal.

Example: Let $f_a(x)$ be a set of functions parametrized by $a \in A$. In various optimization problems, it is of interest to find

$$a_0 = \arg\min_{a \in A} \mathbb{E}[f_a(X)].$$

A simple idea is to draw $X_1, \ldots, X_n$ iid, and estimate a_0 by

$$\widehat{a}_0 = \arg\min_{a \in A} \frac{1}{n} \sum_{i=1}^{n} f_a(X_i).$$

This has to be dealt with carefully because of the bias

$$\mathbb{E}\left[\min_{a \in A} \frac{1}{n} \sum_{i=1}^{n} f_a(X_i)\right] \leq \min_{a \in A} \mathbb{E}[f_a(X)].$$

As long as A is finite, the bias vanishes as $n \to \infty$. However, if $f_a(x)$ is difficult to evaluate, the potential bias may be difficult to remove practically as it is unpractical to evaluate $f_a(x)$ for a large number of X_i.

Often, however, the bias does not vanish as $n \to \infty$.
Example: In simulating

$$S(t_j),\ t_j = \frac{jT}{m}, \quad j = 0, 1, \ldots, m,$$

under the Black-Sholes model, if one uses the naive approximation

$$S(t_{j+1}) = S(t_j) + rS(t_j) + \sigma S(t_j)\sqrt{h}Z_{j+1},$$

the expected discounted payoff of an Asian call option will differ from the exact value. The bias typically vanishes as

$$h = T/m \to 0.$$

However, decreasing h leads to higher computational burden.

As the next example shows, if two unbiased estimators are combined to estimate another quantity, the composite estimator may be biased.
Example: Suppose A is a call option expiring at T with strike price K_A, and B is a (compound) option expiring at t with strike price K_B to buy A:

$$S(0) \longrightarrow \underset{B}{S(t)} \longrightarrow \underset{A}{S(T)}.$$

Let $P_A(x)$ denote the expected discounted payoff of A given that $S(t) = x$:

$$P_A(x) = \mathbb{E}\{e^{-r(T-t)}[S(T) - K_A]^+ \mid S(t) = x\}.$$

The expected discounted payoff of B is then

$$P_B = \mathbb{E}\{e^{-rt}[P_A(S(t)) - K_B]^+\}.$$

To estimate P_B, one can draw m replicates s_i of $S(t)$ to get an estimate

$$\frac{e^{-rt}}{n}\sum_{i=1}^{n}[P_A(s_i)-K_B]^+.$$

If, for each value s of $S(t)$, $P_A(s)$ can be evaluated exactly using a closed-form formula, then the estimate is unbiased.
However, if $P_A(s)$ does not have a closed-form formula, then one has to draw $x_j(s)$, $j=1,\ldots,k$, from the *conditional* distribution of $S(T)$ given $S(t)=s$, to get an unbiased estimate

$$\widehat{P}_A(s)=\frac{e^{-r(T-t)}}{k}\sum_{j=1}^{k}[x_j(s)-K_A]^+.$$

The estimator of P_B is a compound of the two unbiased estimators

$$\widehat{P}_B=\frac{e^{-rt}}{n}\sum_{i=1}^{n}[\widehat{P}_A(s_i)-K_B]^+,$$

often referred to as a "plug-in" estimator.

It turns out that $\widehat{P}_B$ has a positive bias,

$$\mathbb{E}(\widehat{P}_B) \geq \mathbb{E}(P_B),$$

with "=" only under special circumstances. This means if someone uses $\widehat{P}_B$ to price the compound option, he will overprice it. The bias can only be reduced by increasing k but not n.
To see how the positive bias comes, recall Jensen's inequality: given random variables X and Y and convex function h,

$$\mathbb{E}[h(X) \mid Y] \geq h[\mathbb{E}(X \mid Y)]$$

and hence

$$\mathbb{E}[h(X)] = \mathbb{E}\left\{\mathbb{E}[h(X) \mid Y]\right\} \geq \mathbb{E}\left\{h[\mathbb{E}(X \mid Y)]\right\}.$$

Let $X = \widehat{P}_A(S(t))$, $Y = S(t)$, $h(x) = e^{-rt}(x - K_B)^+$. Then

1. $h(x)$ is convex,

$$P_B = \mathbb{E}[h(P_A(S(t)))].$$

and

$$\widehat{P}_B = \frac{1}{n}\sum_{i=1}^{n} h(\widehat{P}_A(s_i)) = \frac{1}{n}\sum_{i=1}^{n} h(X_i),$$

where $s_1, \ldots, s_n$ iid $\sim S(t)$ and $X_i = \widehat{P}_A(s_i)$.

2. For each s_i, $\widehat{P}_A(s_i)$ is an unbiased estimator of $P_A(s)$:

$$\mathbb{E}[X_i] = P_A(s_i).$$

Then

$$\mathbb{E}[\widehat{P}_B] = \mathbb{E}[h(X)] \geq \mathbb{E}\left\{h[\mathbb{E}(X \mid Y)]\right\} = \mathbb{E}[h(P_A(S(t)))] = P_B. \qquad \square$$

Mean squared error

A standard measure of performance is mean squared error (MSE). If $\widehat{c}$ is an estimator of C, then

$$\begin{aligned}\mathsf{MSE}(\widehat{c}) &= \mathbb{E}[(\widehat{c}-C)^2] \\ &= [\mathbb{E}(\widehat{c})-C]^2 + \mathbb{E}[(\widehat{c}-\mathbb{E}(\widehat{c}))^2] \\ &= \mathsf{Bias}^2(\widehat{c}) + \mathrm{Var}(\widehat{c}).\end{aligned}$$

Importance Sampling

Let $f(\boldsymbol{x})$ be a probability density. To estimate

$$\mu = \int h(\boldsymbol{x}) f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x},$$

an ordinary MC estimator will do:

1. generate $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_n$ iid $\sim$ density f, and
2. estimate μ by

$$\widehat{\mu} = \frac{1}{n} \sum_{i=1}^{n} h(\boldsymbol{Z}_i).$$

Observe that if g is any probability density such that

$$g(\boldsymbol{x}) > 0 \text{ whenever } f(\boldsymbol{x}) > 0,$$

then

$$\mu = \int h(\boldsymbol{x}) \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})} \, g(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}.$$

Let $\boldsymbol{X}$ be a random vector with density g. Then

$$\mu = \mathbb{E}\left[h(\boldsymbol{X}) w(\boldsymbol{X})\right], \quad \text{with} \quad w(\boldsymbol{x}) = \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})}.$$

Consider the following steps

1. Generate $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ iid $\sim$ density g
2. Estimate μ by

$$\widehat{\mu}_g = \frac{1}{n} \sum_{i=1}^{n} h(\boldsymbol{X}_i) w(\boldsymbol{X}_i)$$

where $w(\boldsymbol{X}_i)$ are referred to as *importance ratios*.

$\widehat{\mu}_g$ is the importance sampling (IS) estimator of μ associated with g.

1. $\widehat{\alpha}_g$ is the weighted sum of $h(\boldsymbol{X}_i)$.
2. If $g = f$, then $\widehat{\alpha}_g$ is the ordinary MC estimator.

Suppose f is known up to a constant factor, i.e.

$$f(\boldsymbol{x}) = Cq(\boldsymbol{x}),$$

where $C > 0$ is unknown. Then

$$\mu = \frac{\mathbb{E}\left[h(\boldsymbol{X})w(\boldsymbol{X})\right]}{\mathbb{E}\left[w(\boldsymbol{X})\right]}, \quad \text{with} \quad w(\boldsymbol{x}) = \frac{q(\boldsymbol{x})}{g(\boldsymbol{x})}.$$

In this case, the standardized weights have to be used in the IS.

1. Generate $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ iid $\sim$ density g
2. Estimate μ by

$$\widehat{\mu}_g = \frac{1}{n}\sum_{i=1}^{n} h(\boldsymbol{X}_i)w^*(\boldsymbol{X}_i)$$

where

$$w^*(\boldsymbol{X}_i) = \frac{w(\boldsymbol{X}_i)}{\sum_{i=1}^{n} w(\boldsymbol{X}_i)}.$$

Output analysis

Remember that we want to estimate

$$\mu = \mathbb{E}[h(\boldsymbol{Y})], \quad \text{with} \quad \boldsymbol{Y} \sim f$$

and we use $\boldsymbol{X} \sim g$ and IS to do this:

$$\mu = \mathbb{E}[h(\boldsymbol{X})w(\boldsymbol{X})], \quad \text{with} \quad w(\boldsymbol{x}) = \frac{f(\boldsymbol{x})}{g(\boldsymbol{x})}.$$

- $\widehat{\mu}_g$ is unbiased: $\mathbb{E}[\widehat{\mu}_g] = \mathbb{E}\left[h(\boldsymbol{X})w(\boldsymbol{X})\right] = \mu$.
- To reduce $\mathrm{Var}[\widehat{\mu}_g]$, we have to reduce $\mathbb{E}\left\{[h(\boldsymbol{X})w(\boldsymbol{X})]^2\right\}$.

The claim is

- to reduce the variance of $\widehat{\mu}_g$, $g(\boldsymbol{x})$ should be in proportion to $h(\boldsymbol{x})f(\boldsymbol{x})$ as much as possible.

To see this,

$$\mathbb{E}\left\{[h(\boldsymbol{X})w(\boldsymbol{X})]^2\right\} \geq \{\mathbb{E}\left[h(\boldsymbol{X})w(\boldsymbol{X})\right]\}^2 = \mu^2,$$

with "=" if and only if

$$h(\boldsymbol{X})w(\boldsymbol{X}) = h(\boldsymbol{X})\frac{f(\boldsymbol{X})}{g(\boldsymbol{X})} \equiv \text{constant},$$

that is,

$$g(\boldsymbol{x}) \propto h(\boldsymbol{x})f(\boldsymbol{x}).$$

Clearly, strict proportionality is possible only when $h(\boldsymbol{x}) \geq 0$.

Discrete random variables

If X is discrete with probability function $p(x)$, i.e.

$$p(x) = \mathbb{P}\{X = x\},$$

then IS still works if g is replaced with a probability function $q(x)$ such that

$$q(x) > 0 \quad \text{whenever} \quad p(x) > 0$$

and the importance ratio is changed accordingly.

Application in Rare Event Evaluation

IS is very useful for estimating quantities related to rare events.
Example: Suppose a network between A and B has N edges. A failure of the network occurs when some of the edges are broken so that there is no path connecting A and B. When a failure occurs, depending on which edges are broken, there is a cost incurred.
Let $\boldsymbol{X} = (X_1, \ldots, X_N)$, such that

$$X_i = \begin{cases} 0 & \text{edge } i \text{ is connected} \\ 1 & \text{otherwise} \end{cases}$$

and the cost is a function $c(\boldsymbol{X})$. Suppose we are interested in estimating $\mathbb{E}[c(\boldsymbol{X})]$. If each edge only has a small probability to fail and the entire network is well designed, then it is very rare to have a network failure. Thus, if a naive MC is used, a huge number of network configurations may need to be simulated in order to estimate $\mathbb{E}[c(\boldsymbol{X})]$ with sufficient precision.

The idea of IS is to "boost" the probability for an edge to fail. This will increase the chance to obtain network failures during the MC. Meanwhile, we have to compensate for the artificial "failure boosting" through assignment of importance weights.
Suppose the edges fail independently with probability p. On the other hand, in the IS, we simulate a network such that the edges fail independently with probability $p^* > p$. Then the weight for each sampled $\boldsymbol{x} = (x_1, \ldots, x_N)$ is

$$\begin{aligned} w(\boldsymbol{x}) &= \frac{\prod_{i=1}^N p^{x_i}(1-p)^{1-x_i}}{\prod_{i=1}^N p^{*x_i}(1-p^*)^{1-x_i}} \\ &= \left(\frac{1-p}{1-p^*}\right)^N \left[\frac{p(1-p^*)}{p^*(1-p)}\right]^{\sum x_i}. \end{aligned}$$

□

Processes

Let $S_1, S_2, \ldots$ be a process, e.g. the asset prices at time $t_1, t_2, \ldots$. To estimate $\mu = \mathbb{E}[h(S_1, \ldots, S_n)]$, suppose $\boldsymbol{S} = (S_1, \ldots, S_n)$ has density $f(\boldsymbol{x})$. Then IS says μ can be estimated by

$$\widehat{\mu}_g = \frac{1}{R} \sum_{i=1}^{R} h(\boldsymbol{X}_i) w(\boldsymbol{X}_i),$$

where $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_R$ are iid $\sim \boldsymbol{X} \sim g$ and

$$g(\boldsymbol{x}) > 0 \text{ whenever } f(\boldsymbol{x}) > 0.$$

Think of $\boldsymbol{X} = (X_1, \ldots, X_n)$ as a process. From this point, IS uses a process different from $\boldsymbol{S}$ to estimate μ.
An important issue is, if the process $S_1, S_2, \ldots$ has some special dependence structure, how to design the auxiliary process to take advantage of it.

Markov processes

Suppose $S_1, \ldots, S_n$ is a Markov process, i.e., we can decompose their joint density as

$$f(x_1, \ldots, x_n) = f_1(x_1) \prod_{j=2}^{n} f_j(x_j \mid x_{j-1}),$$

where f_1 is the density of the initial distribution, and $f_j(x_j \mid x_{j-1})$ the transition density at step j. To utilize the Markov property in IS, the auxiliary density $g(x_1, \ldots, x_n)$ may also have a similar structure:

$$g(x_1, \ldots, x_n) = g_1(x_1) \prod_{j=2}^{n} g_j(x_j \mid x_{j-1}).$$

Now in the IS, the weight for $\boldsymbol{x} = (x_1, \ldots, x_n)$ is

$$w(\boldsymbol{x}) = \frac{f_1(x_1)}{g_1(x_1)} \prod_{j=2}^{n} \frac{f_j(x_j \mid x_{j-1})}{g_j(x_j \mid x_{j-1})}.$$

The IS in this case is as follows.
For $i = 1, \ldots, R$,

1. draw $X_1 \sim g_1$;
2. for $j = 2, \ldots, n$, draw $X_j \sim g_j(x_j \mid X_{j-1})$;
3. set

$$h_i = h(X_1, \ldots, X_n), \qquad W_i = \frac{f_1(X_1)}{g_1(X_1)} \prod_{j=2}^{n} \frac{f_j(X_j \mid X_{j-1})}{g_j(X_j \mid X_{j-1})}.$$

The IS estimate is

$$\widehat{\mu}_g = \frac{1}{R} \sum_{j=1}^{R} h_j W_j.$$

Estimating ruin probabilities using IS

Let the following hold for an insurance firm:

1. it has a reserve $x \geq 0$;
2. it earns premiums at a constant rate p per unit time;
3. the payments $Y_1, Y_2, \ldots$ for the claims are iid;
4. on average it receives λ claims per unit time;
5. premiums flow in at a faster rate than claims are paid out, i.e.

$$\lambda \mathbb{E}\, Y_i < p.$$

Let ξ_i be the interarrival times of claims. Then

$$Z_i = Y_i - p\xi_i$$

is the loss during the period that starts immediately after the $(i-1)$th claim and ends the ith claim and

$$L_n = Z_1 + \cdots + Z_n$$

is the total loss by the time of the nth claim. So

$$\text{ruin ever occurs} \iff L_n > x \text{ for some } n.$$

Let

$$N = \text{first time } L_n \text{ passes } x.$$

Then

$$\{L_n \text{ ever passes } x\} = \{N < \infty\}$$

So the probability of ruin is $\mathbb{P}\{N < \infty\}$.

To proceed, assume that the arrival times of claims form a Poisson process with rate λ. Under this assumption, $Z_1, Z_2, \ldots$ are iid. Let their common density be f. Since

$$\mathbb{E}Z_i = \mathbb{E}\,Y_i - p/\lambda < 0,$$

by the LLN, $L_n \to -\infty$ as $n \to \infty$. Since $x \geq 0$, $\mathbb{P}\{N < \infty\} < 1$.
Proposal 1: run the following routine many times and count how often $L_n > x$

- keep sampling $Z_i \sim f$ until $L_n > x$.

Problem: how to get out of routines that never have $L_n > x$?
Proposal 2: set $M \gg 1$, run the following routine many times and count how often $L_n > x$

- keep sampling $Z_i \sim f$ until $L_n > x$ or M Z_i have been sampled.

Problem: Bias.

We need a procedure that guarantees to stop and yields an unbiased estimate of $\mathbb{P}\{N < \infty\}$. Suppose instead of sampling $Z_i \sim f$, we sample X_i from some g such that

$$\mathbb{E}X_i > 0, \quad g(x) > 0 \text{ whenever } f(x) > 0.$$

By the LLN, $G_n = X_1 + \cdots + X_n \to \infty$ w.p. 1.

- Unlike L_n, G_n is guaranteed to pass x.

Let

$$\tau = \text{first time } G_n \text{ passes } x.$$

Then,

$$\begin{aligned}\mathbb{P}\{N < \infty\} &= \mathbb{E}[\mathbf{1}\{N < \infty\}] \\ &= \mathbb{E}\left[\prod_{i=1}^{\tau} \frac{f(X_i)}{g(X_i)} \mathbf{1}\{\tau < \infty\}\right].\end{aligned}$$

The basic IS procedure is as follows

1. Fix g that satisfies

$$g(x) > 0 \text{ whenever } f(x) > 0,$$
$$\mathbb{E}X > 0 \text{ for } X \sim g.$$

2. For $i = 1, \ldots, n$
   1. keep sampling $X_i \sim g$ until $G_n > x$;
   2. if τ X's are sampled, then set

$$V_i = \prod_{j=1}^{\tau} \frac{f(X_i)}{g(X_i)}.$$

The IS estimate is $\widehat{\mu}_g = (V_1 + \cdots + V_n)/n$. What remains to be done is to find a convenient g.

A popular choice is the *exponentially tilted* f:

$$g(x) = e^{\theta x - \psi(\theta)} f(x),$$

with $\psi(\theta)$ the *cumulant generating function* of f, i.e.

$$\psi(\theta) = \ln \int e^{\theta x} f(x)\, dx.$$

It is easy to see that for any θ, $g(x) > 0$ whenever $f(x) > 0$. It remains to choose θ so that $\mathbb{E}X > 0$ for $X \sim g$. Now

$$\mathbb{E}X = \mathbb{E}[Ze^{\theta Z - \psi(\theta)}] = \frac{\mathbb{E}(Ze^{\theta Z})}{\mathbb{E}(e^{\theta Z})} = \psi'(\theta).$$

Since $\psi(\theta)$ is analytically available, we can choose θ such that

$$\mathbb{E}X = \psi'(\theta) > 0.$$

With θ chosen as above,

$$\mathbb{P}\{N < \infty\} = \mathbb{E}\left[\prod_{i=1}^{\tau} e^{\psi(\theta)-\theta X_i}\right] = \mathbb{E}\left[e^{-\theta G_\tau + \psi(\theta)\tau}\right].$$

Recall that $G_\tau = X_1 + \cdots + X_\tau$. In particular, if $\theta = \theta_*$ is the (unique) positive solution to $\psi(\theta) = 0$, then

$$\mathbb{P}\{N < \infty\} = e^{-\theta_* x}\mathbb{E}\left[e^{-\theta_*(G_\tau - x)}\right],$$

where $G_\tau - x$ is the amount of overshoot the first time $X_1 + \cdots + X_n$ passes x.

The existence and uniqueness of θ_* is due to the following facts:

1. $\psi(\theta)$ is strictly convex
2. $\psi(0) = 0$ and $\psi'(0) = \mathbb{E}X_i < 0$
3. $\psi(\theta) \to \infty$ as $\theta \to \infty$.

An example in population genetics

In evolutionary biology, one question is to estimate the time of the most recent common ancestor of a set of populations.
A very simplified demographic model employed in the study is as follows.

1. The populations evolve in non-overlapping generations.
2. Each individual in a population is a chromosome segment, small enough to exclude the possibility of recombinations.

Under this model, an individual can be treated as a descendent of a single parent segment.

Each segment has a genetic type. If a parent has type a, then its offspring is of type a with probability $1-\mu$, and of type $b \neq a$ with probability μP_{ab}.

1. μ is the mutation rate per chromosome per generation.
2. P_{ab} is the mutation transition matrix.
3. P_{ab} is assumed to have a unique stationary distribution.

Suppose we get a random sample from the current population. Any possible ancestral relationship among the individuals in the sample can be described by a genealogical tree. The tree is rooted with a single individual (i.e., a single type a). The root generates several offspring, all of them type a. This can be thought of as branching at the root. The branches evolve independently of each other and at any time one of the two things can happen at a branch (1) two or more offspring are generated with the same type as their parent, each evolving independently afterward; or (2) one offspring is generated with a mutation.

Suppose there are some parameter θ important to the evolution that needs to be estimated. To do this, denote by H_0 the observed sample of types, H_{-1} the types of the sample's ancestors when the most recent change occurred, H_{-2} those when the next most recent change occurred, and so on, until H_{-k}, which only consists of the type of the root.
A natural assumption is that under p_θ, H_{-k}, ..., H_0 form a Markov chain:

$$p_\theta(H_{-k}, \ldots, H_{-1}, H_0) = p_\theta(H_{-k}) p_\theta(H_{-k+1} \mid H_{-k}) \cdots p_\theta(H_{-1} \mid H_0).$$

Then, to estimate θ, we need to evaluate

$$p_\theta(H_0) = \sum_{H_{-k} \to \cdots \to H_{-1} \to H_0} p_\theta(H_{-k}, \ldots, H_{-1}, H_0),$$

where $H_{-i} \to H_{-i+1}$ means H_{-i} and H_{-i+1} are "compatible", i.e., it is possible to generate H_{-i+1} from H_{-i}. Only H_0 is observable.

A straightforward idea to evaluate $p_\theta(H_0)$ is to run the Markov chain in a forward fashion. That is, pick a random $-k$, which is a random guess of the time of the most recent common ancestor, then sample H_{-k}, then sample H_{-k+1} conditional on H_{-k}, and so on. Finally, if the chain at the k step ends up with H_0, then include the whole H_{-k}, ..., H_0 in the sum. It is easy to see this is a very difficult thing to do.

The IS runs the chain "backward" as follows. It samples H_{-1} *conditional* on H_0, and H_{-2} conditional on H_{-1}, and so on, until H_{-i} *coalesces*, i.e., becomes a single type. Then

$$\widehat{p}_\theta(H_0) = \frac{1}{n}\sum_{j=1}^{n} \frac{p_\theta(\mathcal{H}_j)}{g(\mathcal{H}_j)},$$

where $\mathcal{H}_j$ is the collection of H_0, H_{-1}, ..., H_{-k} in the jth run of the IS, and g the distribution of the backward Markov chain.
One choice of g is as follows. Pick a value θ_0, which can be θ itself, then let the "backward" transition probability from H_{-i+1} to H_{-i} be

$$g(H_{-i} \mid H_{-i+1}) = \frac{p_{\theta_0}(H_{-i+1} \mid H_{-i})}{\sum_{\text{all possible } a} p_{\theta_0}(H_{-i+1} \mid H_{-i} = a)}.$$

Since H_0 is already known,

$$g(H_0, H_{-1}, \ldots, H_{-k}) = g(H_{-1} \mid H_0) g(H_{-2} \mid H_{-1}) \cdots g(H_{-k} \mid H_{-k+1}).$$

Control Variates

Suppose we are interested in $\mu = \mathbb{E}[h(\boldsymbol{X})]$. The simple MC estimator for μ is

$$\widehat{\mu}_{\mathrm{MC}} = \frac{1}{n}\sum_{i=1}^{n} h(\boldsymbol{X}_i),$$

Suppose $\theta = \mathbb{E}[c(\boldsymbol{Y})]$ is a quantity somewhat related to μ. The simple MC estimator of θ is

$$\widehat{\theta}_{\mathrm{MC}} = \frac{1}{n}\sum_{i=1}^{n} c(\boldsymbol{Y}_i).$$

If θ is available analytically, then there is no point to compute $\widehat{\theta}_{\mathrm{MC}}$ if we just want to know θ. However, $\widehat{\theta}_{\mathrm{MC}}$ can be useful for the estimation of μ!

Imagine that we can sample $\boldsymbol{X}$ and $\boldsymbol{Y}$ together, such that $h(\boldsymbol{X})$ and $c(\boldsymbol{Y})$ are positively correlated. If we see $\widehat{\theta}_{\mathrm{MC}} > \theta$, then $\widehat{\mu}_{\mathrm{MC}}$ is somewhat more likely to have a positive difference from μ, so we may reduce $\widehat{\mu}_{\mathrm{MC}}$ according to the positive difference between $\widehat{\theta}_{\mathrm{MC}}$ and θ. Likewise, if we see $\widehat{\theta}_{\mathrm{MC}} < \theta$, then we may increase $\widehat{\mu}_{\mathrm{MC}}$ accordingly.
To be specific, suppose we can sample

$$(\boldsymbol{X}_1, \boldsymbol{Y}_1), \ldots, (\boldsymbol{X}_n, \boldsymbol{Y}_n) \text{ i.i.d.}$$

The control variate estimator for μ is

$$\widehat{\mu}_{\mathrm{CV}} = \widehat{\mu}_{\mathrm{MC}} - b(\widehat{\theta}_{\mathrm{MC}} - \theta).$$

First, $\widehat{\mu}_{\mathrm{CV}}$ is unbiased.

$$\begin{aligned}\mathbb{E}(\widehat{\mu}_{\mathrm{CV}}) &= \mathbb{E}[\widehat{\mu}_{\mathrm{MC}} - b(\widehat{\theta}_{\mathrm{MC}} - \theta)] \\ &= \mathbb{E}(\widehat{\mu}_{\mathrm{MC}}) - b\mathbb{E}(\widehat{\theta}_{\mathrm{MC}} - \theta) = \mu.\end{aligned}$$

Second, $\widehat{\mu}_{\mathrm{CV}}$ is consistent: since MC estimators are consistent, as $n \to \infty$,

$$\widehat{\mu}_{\mathrm{CV}} = \widehat{\mu}_{\mathrm{MC}} - b(\widehat{\theta}_{\mathrm{MC}} - \theta) \to \mu - b(\theta - \theta) = \mu.$$

1. If $b = 0$, $\widehat{\mu}_{\mathrm{CV}} = \widehat{\mu}_{\mathrm{MC}}$, so the optimal control variate estimator is at least as good as $\widehat{\mu}_{\mathrm{MC}}$.
2. Give a control variate $c(\boldsymbol{Y})$, need to choose b appropriately.
3. A harder problem is how to choose a control variate $c(\boldsymbol{Y})$.

Choosing b

For each b,

$$\mathrm{Var}(\widehat{\mu}_{\mathrm{CV}}) = \mathrm{Var}(\widehat{\mu}_{\mathrm{MC}}) + b^2\mathrm{Var}(\widehat{\theta}_{\mathrm{MC}}) - 2b\mathrm{Cov}(\widehat{\mu}_{\mathrm{MC}}, \widehat{\theta}_{\mathrm{MC}}).$$

To minimize $\mathrm{Var}(\widehat{\mu}_{\mathrm{CV}})$,

$$b^* = \frac{\mathrm{Cov}(\widehat{\mu}_{\mathrm{MC}}, \widehat{\theta}_{\mathrm{MC}})}{\mathrm{Var}(\widehat{\theta}_{\mathrm{MC}})} = \frac{\mathrm{Cov}[h(\boldsymbol{X}), c(\boldsymbol{Y})]}{\mathrm{Var}[c(\boldsymbol{Y})]}.$$

Plug b^* into the formula for $\mathrm{Var}(\widehat{\mu}_{\mathrm{CV}})$ to get the variance reduction factor

$$\frac{\mathrm{Var}(\widehat{\mu}_{\mathrm{CV}}^{\mathrm{opt}})}{\mathrm{Var}(\widehat{\mu}_{\mathrm{MC}})} = 1 - \rho^2,$$

where ρ is the correlation coefficient between $h(\boldsymbol{X})$ and $c(\boldsymbol{Y})$. Indeed, b^* is just the coefficient b_1 in the least square regression of $h(\boldsymbol{X})$ on $c(\boldsymbol{Y})$

$$\widehat{h}(\boldsymbol{X}) = b_0 + b_1 \times c(\boldsymbol{Y}).$$

Since the moments of $h(\boldsymbol{X}_i)$ and $c(\boldsymbol{Y}_i)$ are unknown, the optimal b can be estimated by using

$$\widehat{b}_n = \frac{\widehat{\mathrm{Cov}}(\widehat{\mu}_{\mathrm{MC}}, \widehat{\theta}_{\mathrm{MC}})}{\widehat{\mathrm{Var}}(\widehat{\theta}_{\mathrm{MC}})}$$

where

$$\widehat{\mathrm{Var}}(\widehat{\theta}_{\mathrm{MC}}) = \frac{1}{n(n-1)} \sum_{i=1}^{n} [c(\boldsymbol{Y}_i) - \widehat{\theta}_{\mathrm{MC}}]^2,$$

and

$$\widehat{\mathrm{Cov}}(\widehat{\mu}_{\mathrm{MC}}, \widehat{\theta}_{\mathrm{MC}}) = \frac{1}{n(n-1)} \sum_{i=1}^{n} [h(\boldsymbol{X}_i) - \widehat{\mu}_{\mathrm{MC}}][c(\boldsymbol{Y}_i) - \widehat{\theta}_{\mathrm{MC}}].$$

1. $\widehat{b}_n \to b^*$ with probability 1 as $n \to \infty$.
2. $\widehat{b}_n$ is the slope of the least-square regression line for $(h(\boldsymbol{X}_i), c(\boldsymbol{Y}_i))$, $i = 1, \ldots, n$.

From the above results, a general idea for CV selection is to search $c(\boldsymbol{Y})$, such that

1. $\mathbb{E}[c(\boldsymbol{Y})]$ is known, and
2. the scatter plot of $(h(\boldsymbol{X}_i), c(\boldsymbol{Y}_i))$, $i = 1, \ldots, n$ indicates strong correlation.

In practice, $\widehat{\mu}_{\mathrm{MC}}$ and $\widehat{\theta}_{\mathrm{MC}}$ often depend on the same random variables, so $\boldsymbol{Y}_i = \boldsymbol{X}_i$ in $c(\boldsymbol{Y}_i)$.

It is also possible to use more than one control variate. For example, suppose $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are jointly distributed and we have decided to use $c(\boldsymbol{Y})$ and $d(\boldsymbol{Z})$ as control variates, with

$$\theta = \mathbb{E}[c(\boldsymbol{Y})], \quad \gamma = \mathbb{E}[d(\boldsymbol{Z})].$$

In this case, we first sample

$$(\boldsymbol{X}_1, \boldsymbol{Y}_1, \boldsymbol{Z}_1), \ldots, (\boldsymbol{X}_n, \boldsymbol{Y}_n, \boldsymbol{Z}_n),$$

and then form

$$\widehat{\mu}_{\mathrm{CV}} = \widehat{\mu}_{\mathrm{MC}} - b_1(\widehat{\theta}_{\mathrm{MC}} - \theta) - b_2(\widehat{\gamma}_{\mathrm{MC}} - \gamma).$$

Use underlying assets as control variates

Suppose we are working in the risk-neutral measure with constant interest rate r. Let

$$S(t) = \text{asset price}$$
$$P = \text{discounted payoff of an option on } S \text{ with maturity } T$$

Recall $\mathbb{E}[S(T)] = e^{rT}S(0)$. Then we can form the control variate estimator

$$\widehat{P} = \frac{1}{n}\sum_{i=1}^{n}\Big\{P_i - \widehat{b}_n[S_i(T) - e^{rT}S(0)]\Big\}$$

For the European call option,

$$P = e^{-rT}[S(T) - K]^+, \quad K \geq 0$$

Therefore, the closer K is to 0, the larger the gain in variance reduction. (Why?)

Using tractable options as control variates

Let an Asian option have payoff $(\overline{S}_A - K)^+$, with

$$\overline{S}_A = \frac{1}{n}\sum_{i=1}^{n} S(t_i).$$

To estimate $\mathbb{E}[(\overline{S}_A - K)^+]$, one idea is to use an artificial option with payoff $(\overline{S}_G - K)^+$, with

$$\overline{S}_G = \left[\prod_{i=1}^{n} S(t_i)\right]^{1/n}$$

As seen next, this artificial option serves as an excellent control.

First, $\mathbb{E}[(\overline{S}_G - K)^+]$ has a closed form solution. This is because, by

$$\begin{aligned}\frac{\overline{S}_G}{S(0)} &= \left\{\prod_{i=1}^{n} \exp\left[\left(r - \tfrac{1}{2}\sigma^2\right) t_i + \sigma W(t_i)\right]\right\}^{\frac{1}{n}} \\ &= \exp\left[\frac{r - \frac{1}{2}\sigma^2}{n}\sum_{i=1}^{n} t_i + \frac{\sigma}{n}\sum_{i=1}^{n} W(t_i)\right] \\ &\sim LN\left(\left(r - \tfrac{1}{2}\sigma^2\right)\overline{t},\, \overline{\sigma}^2\overline{t}\right),\end{aligned}$$

with

$$\overline{t} = \frac{1}{n}\sum_{i=1}^{n} t_i, \quad \overline{\sigma}^2 = \frac{\sigma^2}{n^2\overline{t}}\sum_{i=1}^{n}(2i-1)t_{n+1-i}.$$

- Property of lognormal: recall that, if $A > 0$ and $K > 0$ are constants, and

$$\frac{X}{A} \sim LN(\mu, \sigma^2),$$

then

$$\mathbb{E}[(X - K)^+] = Ae^{\mu + \frac{1}{2}\sigma^2}\Phi(d) - K\Phi(d - \sigma),$$

where Φ is the distribution function of $N(0, 1)$ and

$$d = \frac{1}{\sigma}\left(\ln\frac{A}{K} + \mu + \sigma^2\right)$$

- In our example, for $\overline{S}_G$, we have

$$A = S(0),\ \mu = (r - \tfrac{1}{2}\sigma^2)\overline{t},\ \sigma^2 = \overline{\sigma}^2\overline{t}.$$

Therefore, using the above results, $\mathbb{E}[(\overline{S}_G - K)^+]$ can be evaluated with a close form as well.

So far, to estimate the price of a derivative of $S(t)$, $S(t)$ itself has been used as the control variate. This is natural. But what if $S(t)$ is difficult to simulate precisely?

Use as a control variate a different process $\tilde{S}(t)$ that

1. can be simulated precisely, and
2. is coupled with $S(t)$, i.e. $\tilde{S}(t)$ is driven by the same Brownian motion underlying $S(t)$.

Note the goal is still to estimate the price of the derivative on $S(t)$, not on $\tilde{S}(t)$.

Suppose

$$\frac{\mathrm{d}S(t)}{S(t)} = r\,\mathrm{d}t + \sigma(t, S(t))\,\mathrm{d}\,W(t),$$

or, more generally,

$$\frac{\mathrm{d}S(t)}{S(t)} = r\,\mathrm{d}t + \sigma(t, S(t), R(t))\,\mathrm{d}\,W(t),$$

where $R(t)$ is a stochastic process described by another SDE. Just write $\sigma(t)$ for either case.

To estimate $\mathbb{E}[(S(T)-K)^+]$, simulate both $S(t)$ and

$$\frac{\mathrm{d}\tilde{S}(t)}{\tilde{S}(t)} = r\,\mathrm{d}t + \tilde{\sigma}\,\mathrm{d}W(t)$$

1. using the *same* $W(t)$
2. $\tilde{\sigma}$: should be close to a typical value of $\sigma(t)$
3. $\tilde{S}(0) = S(0)$
4. the simulation of $S(t)$ in general is approximate, whereas that of $\tilde{S}(t)$ is precise.

Given a fixed partition of $[0, T]$,

$$0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = T,$$

set $\Delta_i = t_{i+1} - t_i$.
Use

$$S(t_{i+1}) = S(t_i) \exp\left\{\left[r - \tfrac{1}{2}\sigma(t_i)^2\right]\Delta_i + \sigma(t_i)\sqrt{\Delta_i} Z_{i+1}\right\}$$

to approximately simulate $S(t)$, where

$$Z_1, \ldots, Z_n \text{ iid } \sim N(0, 1).$$

Meanwhile, use the *same* Z_i to draw

$$\tilde{S}(t_{i+1}) = \tilde{S}(t_i) \exp\left\{\left[r - \tfrac{1}{2}\tilde{\sigma}^2\right]\Delta_i + \tilde{\sigma}\sqrt{\Delta_i} Z_{i+1}\right\}.$$

The point is that

$$\mathbb{E}\left[(\tilde{S}(T)-K)^{+}\right]$$

is analytically available by Black-Scholes. Then we can form a controlled estimator using independent replications of

$$[S(T)-K]^{+}-b\left\{[\tilde{S}(T)-K]^{+}-\mathbb{E}\left[(\tilde{S}(T)-K)^{+}\right]\right\}.$$

Antithetic variates

This method attempts to reduce variance by introducing *negative* correlation.
Suppose X is symmetric about $\mu = \mathbb{E}X$, i.e.

$$\mathbb{P}(X \leq \mu - x) = P(X \geq \mu + x), \text{ any } x.$$

Denote by F the cumulative distribution function of X and suppose it is invertible. Let $U \sim \text{Unif}(0, 1)$. By inverse transformation, $F^{-1}(U) \sim X$. On the other hand, since $1 - U \sim U$, $F^{-1}(1 - U) \sim X$ as well. Then

$$\widehat{\mu}_{\text{AV}} := \frac{F^{-1}(U) + F^{-1}(1 - U)}{2}$$

could be used as an estimator for μ.

- How good is $\widehat{\mu}_{\text{AV}}$?

Write

$$F^{-1}(U) = \mu - x.$$

Then

$$U = F(\mu - x) = \mathbb{P}(X \leq \mu - x) = \mathbb{P}(X \geq \mu + x) = 1 - F(\mu + x),$$

giving

$$F^{-1}(1 - U) = \mu + x.$$

As a result,

$$\widehat{\mu}_{\text{AV}} = \frac{F^{-1}(U) + F^{-1}(1 - U)}{2} \equiv \mu,$$

a perfect estimator of μ!

For an arbitrary $X \sim F$, to estimate $\mu = \mathbb{E}X$, one can draw

$$U_1, \ldots, U_n \text{ iid } \sim \text{Unif}(0,1)$$

and let

$$X_i = F^{-1}(U_i), \quad \tilde{X}_i = F^{-1}(1 - U_i).$$

1. $(X_1, \tilde{X}_1), \ldots, (X_n, \tilde{X}_n)$ are iid
2. for each i, $X_i \sim \tilde{X}_i$ but *negatively* correlated.

To see why the correlation is negative, note that

$$\begin{aligned} X_1 \le X_2 &\iff U_1 \le U_2 \\ &\iff F^{-1}(1-U_1) \ge F^{-1}(1-U_2) \iff \tilde{X}_1 \ge \tilde{X}_2. \end{aligned}$$

So $(X_1 - X_2)(\tilde{X}_1 - \tilde{X}_2) \le 0$ *always*. Take expectation to get

$$\underbrace{\mathbb{E}(X_1\tilde{X}_1) + \mathbb{E}(X_2\tilde{X}_2)}_{=2\mathbb{E}(X_1\tilde{X}_1)} \le \underbrace{\mathbb{E}(X_1\tilde{X}_2) + \mathbb{E}(\tilde{X}_1 X_2)}_{=2\mathbb{E}X_1\mathbb{E}\tilde{X}_1}.$$

Let

$$\widehat{\mu}_{\mathrm{AV}} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{X_i + \tilde{X}_i}{2}\right)$$

Then

$$\mathrm{Var}(\widehat{\mu}_{\mathrm{AV}}) = \frac{1}{n}\mathrm{Var}\left(\frac{X_1 + \tilde{X}_1}{2}\right) = \frac{\sigma_X^2 + \mathrm{Cov}(X_1, \tilde{X}_1)}{2n},$$

with $\mathrm{Cov}(X_1, \tilde{X}_1) \leq 0$. Then

$$\mathrm{Var}(\widehat{\mu}_{\mathrm{AV}}) \leq \frac{\sigma_X^2}{2n}.$$

On the other hand, if we use $X_1, \ldots, X_n$ to form a MC estimator, then

$$\mathrm{Var}(\widehat{\mu}_{\mathrm{MC}}) = \frac{\sigma_X^2}{n}.$$

- Comparing to $\widehat{\mu}_{\mathrm{MC}}$, the variance of $\widehat{\mu}_{\mathrm{AV}}$ is reduced by at least half.

We can also use $Z \sim N(0,1)$ for antithetic sampling. By $\Phi(Z) \sim \text{Unif}(0,1)$, with $\Phi(x)$ the distribution function of $N(0,1)$ and $Z \sim N(0,1)$,

$$X = F^{-1}(\Phi(Z))$$

with $F^{-1}(\Phi(x))$ increasing,

$$X_i = F^{-1}(U_i) = F^{-1}(\Phi(Z_i)),$$
$$\tilde{X}_i = F^{-1}(1 - U_i) = F^{-1}(\Phi(-Z_i)).$$

Generalizations

Suppose we want to estimate $\mu = \mathbb{E}X$.
If X can be written as

$$X = f(U), \text{ with } U \sim \text{Unif}(0,1),$$

then, one can form

$$X_i = f(U_i), \quad \tilde{X}_i = f(1-U_i), \quad \widehat{\mu}_{\text{AV}} = \frac{1}{2n}\sum_{i=1}^{n}(X_i + \tilde{X}_i),$$

with $U_1, \ldots, U_n$ iid $\sim \text{Unif}(0,1)$. As usual,

$$\widehat{\mu}_{\text{MC}} = \frac{1}{n}\sum_{i=1}^{n} X_i.$$

Likewise, if X can be written as

$$X = g(Z), \text{ with } Z \sim N(0,1),$$

then, one can form

$$X_i = g(Z_i), \quad \tilde{X}_i = g(-Z_i), \quad \widehat{\mu}_{\text{AV}} = \frac{1}{2n}\sum_{i=1}^{n}(X_i + \tilde{X}_i),$$

with $Z_1, \ldots, Z_n$ iid $\sim N(0,1)$. In either case, $(X_1, \tilde{X}_1), \ldots, (X_n, \tilde{X}_n)$ are iid and $X_i \sim \tilde{X}_i$. Notice that it is *always* true that for $X \sim \tilde{X}$, $\text{Cov}(X, \tilde{X}) \leq \text{Var}(X)$. Thus

$$\text{Var}(\widehat{\mu}_{\text{AV}}) \leq \text{Var}(\widehat{\mu}_{\text{MC}})$$

- The amount of variance reduction depends on f or g.

Variance decomposition

Suppose $X = g(Z)$ with $Z \sim N(0,1)$. It is important to note $g(Z) \sim g(-Z)$. Let

$$s(z) = \frac{g(z) + g(-z)}{2}, \quad a(z) = \frac{g(z) - g(-z)}{2}.$$

Clearly $g(z) = s(z) + a(z)$. Since $s(z) = s(-z)$ and $a(z) = -a(-z)$, s and a are called the *symmetric* and *antisymmetric* parts of g, respectively. Observe that

$$\widehat{\mu}_{\mathrm{AV}} = \frac{1}{n}\sum_{i=1}^{n} s(Z_i).$$

Then

$$\mathrm{Var}(\widehat{\mu}_{\mathrm{MC}}) - \mathrm{Var}(\widehat{\mu}_{\mathrm{AV}}) = \frac{\mathrm{Var}(X) - \mathrm{Var}[s(Z)]}{n}.$$

As $X = s(Z) + a(Z)$,

$$\mathrm{Var}(X) = \mathrm{Var}[s(Z)] + \mathrm{Var}[a(Z)] + 2\mathrm{Cov}[s(Z), a(Z)].$$

Importantly, $s(Z)$ and $a(Z)$ are uncorrelated:

$$\mathrm{Cov}[s(Z), a(Z)] = \mathbb{E}[s(Z)a(Z)] = \frac{1}{4}\mathbb{E}[g^2(Z) - g^2(-Z)] = 0,$$

So the reduction in variance is

$$\mathrm{Var}(\widehat{\mu}_{\mathrm{MC}}) - \mathrm{Var}(\widehat{\mu}_{\mathrm{AV}}) = \frac{\mathrm{Var}[a(Z)]}{n}.$$

To see how large $\mathrm{Var}[a(Z)]$ can be,

$$\mathrm{Var}[a(Z)] = \frac{\mathrm{Var}[g(Z)] - \mathrm{Cov}[g(Z), g(-Z)]}{2}.$$

We thus have the following conclusions

1. If g is monotone, then $\mathrm{Cov}[g(Z), g(-Z)] \leq 0$ and the variance of $\widehat{\mu}_{\mathrm{AV}}$ is reduced by at least one half comparing to $\widehat{\mu}_{\mathrm{MC}}$.
2. If g is symmetric, no reduction.
3. If g is antisymmetric, $\mathrm{Var}(\widehat{\mu}_{\mathrm{AV}}) = 0$.

Rao-Blackwellization

Like control covariate, Rao-Blackwellization uses information of correlated random variables to reduce the variance of estimation. Recall that for any jointly distributed random variables X and $\boldsymbol{Y}$,

$$\begin{aligned} \mathbb{E}X &= \mathbb{E}[\mathbb{E}(X \mid \boldsymbol{Y})], \\ \mathrm{Var}(X) &= \mathrm{Var}[\mathbb{E}(X \mid \boldsymbol{Y})] + \mathbb{E}[\mathrm{Var}(X \mid \boldsymbol{Y})]. \end{aligned} \tag{1}$$

That is, the random variable,

$$f(\boldsymbol{Y}) = \mathbb{E}(X \mid \boldsymbol{Y}),$$

which is a function is $\boldsymbol{Y}$, satisfies

$$\mathbb{E}[f(\boldsymbol{Y})] = \mathbb{E}X, \quad \mathrm{Var}[f(\boldsymbol{Y})] \leq \mathrm{Var}(X). \tag{2}$$

To estimate $\mu = \mathbb{E}[h(\boldsymbol{X})]$, the usual MC estimation will draw $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ iid $\sim \boldsymbol{X}$, and get

$$\widehat{\mu}_{\mathrm{MC}} = \frac{1}{n} \sum_{i=1}^{n} h(\boldsymbol{X}_i).$$

Suppose that each $\boldsymbol{X}_i$ can jointly distributed with some $\boldsymbol{Y}_i$, such that

$$f_i(\boldsymbol{Y}_i) = \mathbb{E}[h(\boldsymbol{X}_i) \mid \boldsymbol{Y}_i]$$

is analytically available. Here $\boldsymbol{Y}_i$ are independent but *not* necessarily identically distributed. $\boldsymbol{Y}_i$ should be a sufficient statistic so that the conditional expectation is full of μ. In this case, it is enough to *only* sample $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n$. Then the *Rao-Blackwellized* estimator of μ is

$$\widehat{\mu}_{\mathrm{RB}} = \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{Y}_i).$$

To see that $\widehat{\mu}_{\mathrm{RB}}$ is unbiased and has smaller variance than $\widehat{\mu}_{\mathrm{MC}}$, by (2)

$$\mathbb{E}(\widehat{\mu}_{\mathrm{RB}}) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[f_i(\boldsymbol{Y}_i)] = \mu,$$

$$\mathrm{Var}(\widehat{\mu}_{\mathrm{RB}}) = \frac{1}{n^2}\sum_{i=1}^{n}\mathrm{Var}[f_i(\boldsymbol{Y}_i)] \le \frac{\mathrm{Var}(X)}{n} = \mathrm{Var}(\widehat{\mu}_{\mathrm{MC}}).$$

The reduction in variance, by (1), is

$$\begin{aligned}\mathrm{Var}(\widehat{\mu}_{\mathrm{MC}}) - \mathrm{Var}(\widehat{\mu}_{\mathrm{RB}}) &= \frac{1}{n^2}\sum_{i=1}^{n}\{\mathrm{Var}(X) - \mathrm{Var}[f_i(\boldsymbol{Y}_i)]\} \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}[\mathrm{Var}(X_i \mid \boldsymbol{Y}_i)].\end{aligned}$$

In a more complex situation, suppose we want to estimate the value of a quantity μ. Suppose $\boldsymbol{Y}$ is known to jointly distributed with some random variables $X_1, \ldots, X_n$. The random variables X_i have the same mean μ but are *not* necessarily independent or identically distributed; however, they are conditionally independent given $\boldsymbol{Y}$, or, even pairwise conditionally negatively correlated given $\boldsymbol{Y}$. Provided that for each i, $f_i(\boldsymbol{Y}) = \mathbb{E}[X_i \mid \boldsymbol{Y}]$ are analytically available,

$$\widehat{\mu}_{\mathrm{RB}} = \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{Y}).$$

Stratified Sampling

Suppose we want to estimate

$$\mu = \mathbb{E}\, Y.$$

Let $A_1, \ldots, A_K$ be disjoint sets of $\mathbb{R}$ such that $\mathbb{P}(Y \in \cup A_i) = 1$. Suppose we can control the fraction of observations in A_i to be exactly p_i. More precisely, we can draw $Y_{i1}, \ldots, Y_{in_i}$ i.i.d. $\sim \mathbb{P}(Y \in \cdot \mid Y \in A_i)$, such that $n_i = p_i n$ with $n = n_1 + \cdots + n_K$. Let

$$\overline{Y}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} Y_{ij}.$$

- A_i: a stratum, a "subpopulation"; $\overline{Y}_i$: group average.

Then $\mathbb{E}\,Y$ can be estimated by the "grand" average

$$\widehat{\mu}_{\mathrm{SS}} = \sum_{i=1}^{K} p_i \overline{Y}_i = \frac{1}{n} \sum_{i=1}^{K} \sum_{j=1}^{n_i} Y_{ij}.$$

The above is the simplest stratified sampling method, known as proportional samling.
$\widehat{\mu}_{\mathrm{SS}}$ is unbiased

$$\begin{aligned}\mathbb{E}(\widehat{\mu}_{\mathrm{SS}}) &= \sum_{i=1}^{K} p_i \mathbb{E}(\overline{Y}_i) = \sum_{i=1}^{n} p_i \mathbb{E}(Y \mid Y \in A_i) \\ &= \sum_{i=1}^{K} \mathbb{P}(Y \in A_i) \mathbb{E}(Y \mid Y \in A_i) = \mathbb{E}(Y).\end{aligned}$$

How about its variance? Since

$$\mathrm{Var}(\widehat{\mu}_{\mathrm{SS}}) = \sum_{i=1}^{K} p_i^2 \mathrm{Var}(\overline{Y}_i) = \sum_{i=1}^{K} p_i^2 \frac{1}{n_i} \mathrm{Var}(Y \mid Y \in A_i),$$

with $n_i = p_i n$,

$$\mathrm{Var}(\widehat{\mu}_{\mathrm{SS}}) = \frac{1}{n} \sum_{i=1}^{K} p_i \mathrm{Var}(Y \mid Y \in A_i).$$

On the other hand, the sample mean $\overline{Y}$ has

$$\mathrm{Var}(\overline{Y}) = \frac{1}{n} \mathrm{Var}(Y).$$

Which one is smaller?

Define a random variable dependent on Y as

$$X = i \quad \text{if} \quad Y \in A_i, \quad i = 1, \ldots, K.$$

Then

$$\mathrm{Var}(\widehat{\mu}_{\mathrm{SS}}) = \frac{1}{n} \sum_{i=1}^{K} \mathbb{P}(X = i) \mathrm{Var}(Y \mid X = i) = \frac{1}{n} \mathbb{E}[\mathrm{Var}(Y \mid X)].$$

On the other hand,

$$\mathrm{Var}(\overline{Y}) = \frac{1}{n} \times \{\mathbb{E}[\mathrm{Var}(Y \mid X)] + \mathrm{Var}[\mathbb{E}(Y \mid X)]\}.$$

Therefore,

$$\mathrm{Var}(\overline{Y}) - \mathrm{Var}(\widehat{\mu}_{\mathrm{SS}}) = \frac{1}{n} \mathrm{Var}[\mathbb{E}(Y \mid X)] \geq 0.$$

1. The reduction can be substantial.
2. X is known as a *stratification variable*.

Generalization one

Use a *stratification variable* X with known $p_i = \mathbb{P}(X \in A_i)$.

- X and Y are jointly distributed.

For each i, sample (X_{ij}, Y_{ij}), $j = 1, \ldots, n_i$, iid from the conditional distribution of (X, Y) given $X \in A_i$ such that $n_i = p_i n$. Then use

$$\widehat{\mu}_{\text{SS}} = \sum_{i=1}^{K} p_i \overline{Y}_i = \frac{1}{n} \sum_{i=1}^{K} \sum_{j=1}^{n_i} Y_{ij}.$$

Since $\mathbb{E}(\overline{Y}_i) = \mathbb{E}(Y \mid X \in A_i)$,

$$\begin{aligned} \mathbb{E}(\widehat{\mu}_{\text{SS}}) &= \sum_{i=1}^{K} p_i \mathbb{E}(Y \mid X \in A_i) \\ &= \sum_{i=1}^{K} \mathbb{P}(X \in A_i) \mathbb{E}(Y \mid X \in A_i) = \mathbb{E}(Y) = \mu. \end{aligned}$$

Generalization two

Allow $q_i = n_i/n$ to be arbitrary. Then use

$$\widehat{\mu}_{\mathrm{SS}} = \sum_{i=1}^{K} p_i \overline{Y}_i = \frac{1}{n} \sum_{i=1}^{K} \frac{p_i}{q_i} \sum_{j=1}^{n_i} Y_{ij}.$$

Two main issues for stratified sampling

1. How to choose the stratification variable X, the strata $A_1, \ldots, A_K$ and allocation $n_1, \ldots, n_K$.
2. How to draw samples from the distribution of (X, Y) conditional on $X \in A_i$.

Example

For any distribution F on $\mathbb{R}$, $F^{-1}(U) \sim X$, with $U \sim \text{Unif}(0,1)$.

- U can be used as the stratification variable.
- Easy to do.

For any K, set

$$A_i = \left(\frac{i-1}{K}, \frac{i}{K} \right], \quad i = 1, \ldots, K.$$

Since $p_i = \mathbb{P}(U \in A_i) = 1/K$, we can choose $n_i = 1$ (one sample per stratum) to get $n_i/n = p_i$. The sample in A_i is simply generated as

$$V_i = \frac{i-1}{K} + \frac{U_i}{K}, \quad i = 1, \ldots, K.$$

Then μ can be estimated by

$$\widehat{\mu}_{\text{SS}} = \frac{1}{K} \sum_{i=1}^{K} F^{-1}(V_i).$$

$A_i = (a_i, a_{i+1}]$ with unequal lengths can also be used. In this case, $p_i = a_{i+1} - a_i$. Let $n_1, \ldots, n_K$ be the allocation numbers which may not satisfy $n_i = p_i n$, where $n = \sum_{i=1}^K n_i$. For each $i = 1, \ldots, K$, we can sample

$$V_{ij} = a_i + U_{ij}(a_{i+1} - a_i), \quad j = 1, \ldots, n_i$$

and use

$$\widehat{\mu}_{\text{SS}} = \sum_{i=1}^K \frac{p_i}{n_i} \sum_{j=1}^{n_i} F^{-1}(V_{ij}) = \frac{1}{n} \sum_{i=1}^K \frac{p_i}{q_i} \sum_{j=1}^{n_i} F^{-1}(V_{ij})$$

where $q_i = n_i/n$.

Example

Suppose $Y \sim N(0,1)$. How to stratify $(-\infty, \infty)$ into *equiprobable* bins, i.e., bins with equal probability?

More generally, suppose $Y \sim F$ continuous. Given probabilities $p_1, \ldots, p_K$ totaling 1, let $a_0 = -\infty$ and

$$a_1 = F^{-1}(p_1), \;\; a_2 = F^{-1}(p_1 + p_2), \ldots, a_K = F^{-1}(1).$$

Define strata

$$A_i = (a_{i-1}, a_i], \quad i = 1, \ldots, K$$

or with $A_K = (a_{K-1}, \infty)$ if $a_K = \infty$. The

$$\mathbb{P}(Y \in A_i) = F(a_i) - F(a_{i-1}) = p_i.$$

Now $V_i = a_{i-1} + U(a_i - a_{i-1}) \sim \text{Unif}(A_i)$ and hence

$$F^{-1}(V_i) \sim F(\cdot \mid Y \in A_i).$$

□

Stratification through acceptance-rejection

Let X be a stratification variable.

- Given strata $A_1, \ldots, A_K$ and allocation numbers $n_1, \ldots, n_K$, for each A_i, sample (X, Y) until n_i pairs are generated with $X \in A_i$.

The efficiency of the method depends on the computational cost of generating (X, Y) and determining which stratum X falls into.

Output analysis

Let $A_1, \ldots, A_K$ denote the strata for a stratification variable X and let

$$Y_{ij}, \quad j = 1, \ldots, n_i$$

have the conditional distribution of Y given $X \in A_i$. Set

$$\mu_i = \mathbb{E}[Y_{ij}] = \mathbb{E}[Y \mid X \in A_i]$$
$$\sigma_i^2 = \mathrm{Var}[Y_{ij}] = \mathrm{Var}[Y \mid X \in A_i]$$
$$p_i = \mathbb{P}(X \in A_i), \quad q_i = \frac{n_i}{n}$$
$$\overline{Y}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} Y_{ij}$$

The stratified estimator is

$$\widehat{\mu}_{\mathrm{SS}} = \sum_{i=1}^{K} p_i \overline{Y}_i.$$

Asymptotic normality

$\widehat{\mu}_{\mathrm{SS}}$ is unbiased because

$$\mathbb{E}(\widehat{\mu}_{\mathrm{SS}}) = \sum_{i=1}^{K} p_i \mathbb{E}(\overline{Y}_i) = \sum_{i=1}^{K} p_i \mu_i = \mu.$$

Since Y_{ij} are independent,

$$\mathrm{Var}(\widehat{\mu}_{\mathrm{SS}}) = \mathrm{Var}\left[\sum_{i=1}^{K} p_i \overline{Y}_i\right] = \sum_{i=1}^{K} p_i^2 \mathrm{Var}(\overline{Y}_i) = \sum_{i=1}^{K} p_i^2 \frac{\sigma_i^2}{n_i} = \frac{\sigma^2(q)}{n}$$

where

$$\sigma^2(q) = \sum_{i=1}^{K} \frac{p_i^2}{q_i} \sigma_i^2.$$

As $n_1, \ldots, n_K \to \infty$ in such a way that $n_i/n \equiv q_i$,

$$\sqrt{n}(\widehat{\mu}_{\mathrm{SS}} - \mu) \xrightarrow{d} N(0, \sigma^2(q)).$$

Optimal allocation

By minimizing

$$\sigma^2(q) = \sum_{i=1}^{K} \frac{p_i^2}{q_i} \sigma_i^2$$

under the constraints $q_i \geq 0$ and $q_1 + \cdots + q_K = 1$, the optimal allocation is

$$q_i^* = \frac{p_i \sigma_i}{\sum_{j=1}^{K} p_j \sigma_j}, \quad i = 1, \ldots, K$$

with

$$\sigma^2(q^*) = \left(\sum_{i=1}^{K} p_i \sigma_i \right)^2$$

In practice, σ_i are rarely known. However, it is often practical to use pilot runs to get estimates of σ_i and thus of q_i^*. The estimated optimal fractions can then be used to allocate samples to strata in a second, often larger set of runs.

Variance decomposition

First, $\mathrm{Var}(\overline{Y}) = \mathrm{Var}(Y)/n$. Let $I = I(X)$ be the index of A_i that contains X. Then

$$\mathrm{Var}(Y) = \mathrm{Var}[\mathbb{E}(Y \mid I)] + \mathbb{E}[\mathrm{Var}(Y \mid I)]$$

For each $i = 1, \ldots, K$, $\mathbb{E}(Y \mid I = i) = \mu_i$. Thus

$$\mathbb{E}(Y \mid I) = \mu_I,$$

which is a function in I. Then

$$\begin{aligned}\mathrm{Var}[\mathbb{E}(Y \mid I)] = \mathrm{Var}(\mu_I) &= \sum_{i=1}^{K} \mathbb{P}(I = i)\mu_i^2 - \left[\sum_{i=1}^{K} \mathbb{P}(I = i)\mu_i^2\right]^2 \\ &= \sum_{i=1}^{K} p_i \mu_i^2 - \left(\sum_{i=1}^{K} p_i \mu_i\right)^2 \geq 0,\end{aligned}$$

with "=" only when μ_i are equal.

On the other hand, $\mathrm{Var}[\,Y \mid I] = \sigma_I^2$. So

$$\mathbb{E}[\mathrm{Var}(\,Y \mid I)] = \sum_{i=1}^{K} p_i \sigma_i^2 = \sigma^2(p).$$

Note that in proportioanl allocation, $\mathrm{Var}(\widehat{\mu}_{\mathrm{SS}}) = \sigma^2(q)/n$. As a result

$$\mathrm{Var}(\overline{Y}) - \mathrm{Var}(\widehat{\mu}_{\mathrm{SS}}) = \frac{\mathrm{Var}[\mathbb{E}(\,Y \mid I)]}{n}.$$

- Proportional allocation gets rid off the uncertainty about the random variable $\mathbb{E}(\,Y \mid I)$, as measured by $\mathrm{Var}[\mathbb{E}(\,Y \mid I)]$.

More examples

Recall, when we say sampling Y by stratifying X, we mean several things.

1. X and Y are jointly distributed.
2. We need to choose strata $A_1, \ldots, A_K$ and allocation $n_1, \ldots, n_K$ for K
3. For each $i = 1, \ldots, K$, sample

$$(X_{i1}, Y_{i1}), \ldots, (X_{in_i}, Y_{in_i}) \text{ iid } \sim$$
$$\text{conditional distribution of } (X, Y) \text{ given } X \in A_i.$$

The sampling can be done as follows

1. Generated a stratified sample X_{ij}, $i = 1, \ldots, K$, $j = 1, \ldots, n_i$.
2. For each X_{ij}, generate Y_{ij} from the conditional distribution of Y given $X = X_{ij}$.

Terminal stratification

In option pricing, the most important feature of the path of an underlying asset is often its value at the option expiration

- Stratifying terminal value

We consider the stratification of BM along its terminal value.
Suppose we want to sample $W(t_i)$, $i = 1, \ldots, m$ and want to stratify $W(t_m)$.

- Use Brownian bridge construction.
- Use equiprobable strata for $W(t_m)$ and a proportional allocation.

Recall $W(t_m) \sim \sqrt{t_m} Z$ with $Z \sim N(0,1)$,

1. Generate a stratified sample for Z, with K equiprobable strata. Set

$$V_i = \frac{i - 1 + U_i}{K}, \quad Y_i = \Phi^{-1}(V_i), \quad i = 1, \ldots, K$$

and where $U_1, \ldots, U_K$ are iid $\sim$ Unif$(0,1)$. Then

$$\sqrt{t_m}\Phi^{-1}(V_i), \quad i = 1, \ldots, K$$

form a stratified sample of $W(t_m)$ with strata $(\sqrt{t_m}a_{i-1}, \sqrt{t_m}a_i]$, where $a_i = \Phi^{-1}(i/K)$, and allocation $n_i = 1$.

2. Given a sample point of $W(t_m)$,
   1. draw $W(t_1)$ from the conditional distribution given $W(t_0) = 0$ and $W(t_m)$;
   2. draw $W(t_2)$ from the conditional distribution given $W(t_1)$ and $W(t_m)$, as so on.

For geometric BM S, given stratified sample of W corresponds to a stratified sample of S, because

$$S(t) = S(0)e^{(\mu-\sigma^2/2)t+\sigma W(t)}.$$

In pricing an option on S, instead of using equiprobable strata over $S(t_m)$, one can

1. combine all values of $S(t_m)$ that result in zero payoff into a single stratum, and
2. create a finer stratification of $S(t_m)$ that potentially produce a nonzero payoff.

Stratifying a linear projection

Suppose

$$\boldsymbol{Y} = (Y_1, \ldots, Y_d)^T \sim N(\mu, \Sigma)$$

and we want to generate Y with

$$X \equiv \boldsymbol{v}^T \boldsymbol{Y}$$

stratified for some fixed $\boldsymbol{v} \in \mathbb{R}^d$.

- Terminal stratification is a special case

The steps of the stratified sampling are

1. Generate a stratified sample of X
2. For each X_i in the sample, generate Y_i from the conditional distribution of Y given $X = X_i$.

Some detail: conditional sampling

Since we can first sample $\boldsymbol{Y} - \boldsymbol{\mu}$ with $\boldsymbol{v}^T(\boldsymbol{Y} - \boldsymbol{\mu})$ being stratified and then add μ back to get a stratified sample of $\boldsymbol{Y}$, we may assume from the beginning that $\boldsymbol{Y} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma})$. By scaling, we may assume $\boldsymbol{v}^T \boldsymbol{\Sigma} \boldsymbol{v} = 1$. By

$$\begin{pmatrix} X \\ \boldsymbol{Y} \end{pmatrix} \sim N\left(0, \begin{pmatrix} 1 & \boldsymbol{v}^T \boldsymbol{\Sigma} \\ \boldsymbol{\Sigma} \boldsymbol{v} & \boldsymbol{\Sigma} \end{pmatrix}\right)$$

conditional on $X = x$,

$$\boldsymbol{Y} \sim N(x \boldsymbol{\Sigma} \boldsymbol{v},\ \boldsymbol{\Sigma} - \boldsymbol{\Sigma} \boldsymbol{v} \boldsymbol{v}^T \boldsymbol{\Sigma})$$

Let $\boldsymbol{A}$ be a matrix such that $\boldsymbol{\Sigma} = \boldsymbol{A}\boldsymbol{A}^T$. Let

$$\boldsymbol{B} = \boldsymbol{A} - \boldsymbol{\Sigma}\boldsymbol{v}\boldsymbol{v}^T\boldsymbol{A}$$

Then

$$\boldsymbol{\Sigma} - \boldsymbol{\Sigma}\boldsymbol{v}\boldsymbol{v}^T\boldsymbol{\Sigma} = \boldsymbol{B}\boldsymbol{B}^T$$

So conditional on $X = x$, $\boldsymbol{Y}$ can be generated as

$$\boldsymbol{Y} = x\boldsymbol{\Sigma}\boldsymbol{v} + \boldsymbol{B}\boldsymbol{Z}, \quad \boldsymbol{Z} \sim N(0, \boldsymbol{I}).$$

- Note that $\boldsymbol{B}$ does not depend on the value of x.

Radial stratification

If $\boldsymbol{Z} \sim N(0, I_d)$, then $\boldsymbol{Z}$ can sampled with stratified norm

$$X = \|\boldsymbol{Z}\| = \sqrt{Z_1^2 + \cdots + Z_d^2}$$

- $X^2 \sim \chi_d^2$
- Given X, $\boldsymbol{Z}$ can be sampled as

$$Z_i = X \prod_{k=1}^{i-1} \sin(2\pi U_k) \times \cos(2\pi U_i), \quad \text{for } 1 \le i < d$$

$$Z_d = X \prod_{k=1}^{d-1} \sin(2\pi U_k).$$

Stratifying a Poisson process

Let X be a Poisson process on $[0, T]$ with rate $\lambda(t)$. It can be sampled by stratifying $N =$ number of points in $[0, T]$.

- Let

$$M = \int_0^T \lambda(t)\, dt$$

  Then $N \sim \text{Poisson}(M)$.

- Given N, the points of X in $[0, T]$ are

$$\tau_1, \ldots, \tau_N \text{ iid with density } f(t) = \frac{\lambda(t)}{M}.$$

Overview

- One of the most important statistical methods
- Efron (1979) is one of the breakthroughs in statistics and one of the most cited paper in statistics.
- Mainly used for uncertainty assessment in statistical inferences (estimation, prediction, variance, confidence intervals, hypothesis testing, etc.)
- A resampling method.
- Allows estimation of the sampling distribution of almost any statistic using random sampling methods
- Approximate the target distribution by the empirical distribution of the observed data.

Principle Explained in One-Sample Problem

- A random sample $\mathcal{X} = \{X_1, \ldots, X_n\}$ from distribution F.
- Quantity of interest $\theta = T(F)$, where T is a functional of F.
- Estimator $\hat{\theta} = T(\hat{F})$, where $\hat{F}$ is the ECDF from the sample $\mathcal{X}$.
- How do we, in statistical inferences, assess the uncertainty of $T(\hat{F})$ or $R(\mathcal{X}, F)$, whose distribution may be intractable or unknown?
- Example: $R(\mathcal{X}, F) = [T(\hat{F}) - T(F)]/S(\hat{F})$, where S is the standard deviation functional.
- A *bootstrap sample* or *pseudo dataset* of $\mathcal{X}$ is formed by a random sample of the same size from $\hat{F}$.

Steps of Bootstrapping in One-Sample Problem

1. Construct empirical distribution function $\hat{F}$, putting mass $1/n$ at each point in $\mathbf{x}$.
2. With $\hat{F}$ fixed, draw a random sample of size n from $\hat{F}$ (sample $\mathbf{x}$ with replacement) — bootstrap sample: $\mathbf{X}^* = (X_1^*, \ldots, X_n^*)$. (In contrast, the ordinary jackknife draws samples of size $n-1$ without replacement.)
3. Approximate the sampling distribution of $R(\mathcal{X}, F)$ by the bootstrap distribution of $R^* = R(\mathcal{X}^*, \hat{F})$.

An Insight on Justification

Chao M-T had an illustration using characteristic functions. Consider the characteristic functions of R and R^*, respectively,

$$Ee^{itR} = \int e^{itR(\mathbf{x},F)} \mathrm{d}F(\mathbf{x})$$

and

$$Ee^{itR^*} = \int e^{itR(\mathbf{x}^*,\hat{F})} \mathrm{d}\hat{F}(\mathbf{x}^*) = \int e^{itR(\mathbf{x},\hat{F})} \mathrm{d}\hat{F}(\mathbf{x}).$$

The two are the same if $F = \hat{F}$.

Example: Sample mean of Bernoulli variables

- Suppose that X_i follows Bernoulli with rate θ.
- Consider $R(\mathcal{X}, F) = \bar{X} - \theta(F)$.
- Then X_i^*'s are iid Bernoulli with rate $\bar{x}$.
- The mean and variance of $R^* = \bar{X}^* - \bar{x}$ is zero and $\bar{x}(1-\bar{x})/n$.
- Conincide with the usual inference.

Example: Sample Median

Estimating the median with sample median. Let $\theta(F)$ be the median of F and $T(\mathcal{X})$ be the sample median $t(\mathcal{X}) = X_{(m)}$, where $X_{(1)} \leq X_{(2)} \leq \cdots \leq X_{(n)}$ are the order statistics for an odd sample size $n = 2m - 1$. Consider $R(\mathcal{X}, F) = T(\mathcal{X}) - \theta(F)$.
Having observed $\mathcal{X} = \S$, the distribution of X^* can be derived analytically (Efron, 1979).
The quantity to bootstrap is

$$R(\mathcal{X}, F) = \frac{|T(\mathcal{X}) - \theta(F)|}{\sigma(F)}$$

the absolute error of the sample median relative to the population standard deviation, which is numerically more stable.

Parametric Bootstrap

- The classic bootstrap is nonparametric (no distributional assumption).
- Parametric bootstrap has a parametric model specified to the data.
- Parametric bootstrap samples are generated from the fitted model, and for each sample the quantity to be bootstrapped is calculated.
- Parametric bootstrap is often used to approximate the null distribution of a testing statistic which is otherwise unwieldy.
- The uncertainty of parameter estimation can be accounted for.
- Example: Goodness-of-fit test.

Bootstrapping Regression

- Consider regression model $Y_i = X_i^\top \beta + \epsilon_i$, where ϵ_i's are iid with mean zero.
- Bootstrapping the residuals:
  - Draw bootstrap sample of residuals $\hat{\epsilon}_i^*$'s
  - Create bootstrap set of pseudo-responses $Y_i^* = \hat{Y}_i + \hat{\epsilon}_i^*,\ i = 1, \ldots, n.$
- Bootstrapping the cases
  - Draw bootstrap sample of $\{ Y_i, X_i)$ jointly.

Bootstrap Bias Correction

- Consider $R(\mathcal{X}, F) = T(\hat{F}) - T(F) = \hat{\theta} - \theta$, which represents the bias of $\hat{\theta}$.
- The mean bias is $E(\hat{\theta}) - \theta$
- Bootstrap estimate of the bias is

$$\sum_{i=1}^{B} (\hat{\theta}_i - \hat{\theta})/B = \bar{\theta}^* - \hat{\theta}.$$

- Improved bias estimate is $\bar{\theta} - T(\bar{F}^*)$ where $\bar{F}^*(x) = \sum_{i=1}^{B} \hat{F}_i^*(x)/B$.

Bootstrap Inference

- Confidence intervals from empirical percentiles of $\hat{\theta}^*$.
- Hypothesis tests based on bootstrap confidence intervals may lose power; greater power is possible if bootstrap a sampling distribution under the null hypothesis.
- Works better if θ is a location parameter.
- Ideally, the bootstrap statistic should be approximately pivotal (whose distribution does not depend on the value of θ).
- Variance stabilization if often used for pivoting.

Reducing Monte Carlo Error

- Balanced Bootstrap
  - Each data point in the combined collection of bootstrap pseudo-data with the same frequency.
  - Ensures that the bootstrap bias estimate is zero if the estimator is indeed unbiased.
- Antithetic Bootstrap
  - Antithetic samples from $\hat{F}^{-}(U)$ and $\hat{F}^{-}(1-U+1/n)$ where U is discrete uniform over $\{1,\ldots,n\}/n$
  - The statistic from the two samples are negatively dependent
  - Similar to antithetic variable in variance reduction

Bootstrapping Dependent Data

- Validity of the ordinary bootstrap depends on iid assumption.
- Dependent data: time series, spatial data, clustered data, among others.
- Assumptions:
  - Stationary: the distribution does not depend on time/space
  - Weekly dependent: dependence dies off as distance gets bigger
- Consider time series: $X_t = \alpha X_{t-1} + \epsilon_t$.
- Model-based approach: parametric; needs a burn-in period; depending on correct model specification
- How to retain the dependence structure in the data?

Block Bootstrap

- Nonmoving block bootstrap
  - split the data into b nonoverlapping blocks of length l
  - sample the blocks with replacement
  - concatenate the resampled blocks
- Moving block bootstrap
  - All blocks of length l are resampled before concatenation
- Naive implementations will produce pseudo-dataset that are closer to white noise than the original data

Blocks-of-Blocks Bootstrap

- Many important parameters of interest pertain to the dependence structure.
- The serial correlation in the observed data is broken in pseudo-dataset at points where adjacent resampled blocks meet.
- Blocks-of-blocks bootstrap procedure
  1. Form multivariate series from the original series
  2. Apply block bootstrapping (nonmoving or moving) on the multivariate series.

Centering and Studentizing

- Moving and nonmoving block bootstrap yield different bootstrap distribution for $\hat{\theta}_n^*$.
- For example, consider $\theta = EX_t$ and $\hat{\theta} = \bar{X}$.
  - For nonmoving block bootstrap, $E^*\hat{\theta}^* = \bar{X}$.
  - For moving block bootstrap, there are edge effects

$$E^*\hat{\theta}* = \frac{1}{n-l+1}\left\{n\bar{X} - \frac{1}{l}\sum_{i=1}^{l-1}(l-i)(X_i + X_{n-i+1})\right\}$$

- Bias correction: center using $\hat{\theta}^* - E^*\hat{\theta}^*$.
  - $E^*\hat{\theta}^*$ may be hard to calculate
  - If $\hat{\theta} = \hat{\theta}(\bar{X})$ is a smooth function, then apply to $\theta(\bar{X}^*) - \theta(E^*\bar{X}^*)$.
- Studentizing: needs improved standard deviation of the bootstrap sample s^*.
- Circular block bootstrap
- Stationary block bootstrap

Block Size

- The performance of block bootstrap depends on block length l
- $l = 1$ means iid bootstrap; dependence is lost
- Too large an l retains the dependence structure but may give too few blocks to sample from
- Asymptotic results indicate that l should increase as n increases to produce consistent estimators of moments, correct coverage probability of confidence intervals, appropriate error rates for hypothesis tests.
- Two methods for moving block bootstrap in minimizing MSE
  - Subsampling plus bootstrapping
  - Jackknifing plus bootstrapping

Linear Regression

- Classical linear regression
- Best subset selection and L_0 penalization.
- Shrinkage estimation.
- Sparse estimation and variable selection.

Linear Regression Models

- $X^T = (X_1, ..., X_p)$: input or feature variables;
- Y: output or outcome variables;
- Model: $E(Y|X) = f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j$.

- The β_js are unknown parameters or coefficients.
- The variables X_j may come from various sources:
  - quantitative inputs;
  - transformations of original inputs, e.g., log or square-root transformation; length, area and volume;
  - basis expansions, e.g., X_1, X_1^2, X_1^3...; Fourier or wavelet basis functions.
  - "dummy" coding of qualitative (categorical) inputs, e.g. suppose $G =$ smoking status and let $X_1 = I$(non-smoker) and $X_2 = I$(smoker).
  - interactions between variables, e.g., $X_3 = X_1 X_2$.
- More generally, we model $E(Y|X)$ using linear combinations of fixed linear/nonlinear functions of the input variables. The key is that the model is linear in the parameters.

Linear Regression Models

- Data: a set of i.i.d. input-output pairs

$$\mathcal{T}_n = \left\{(\mathbf{x}_{(i)}, y_i); 1 \leq i \leq n\right\},$$

where $\mathbf{x}_{(i)} = (x_{i1}, ..., x_{ip})^\top \in R^p$ and $y_i \in R$.

Linear Model Form

$$y_i = \beta_0 + x_{i1}\beta_1 + ... + x_{ip}\beta_p + \epsilon_i, \qquad 1 \leq i \leq n$$

- Response: $\mathbf{y} = (y_1, ..., y_n)^\top$
- Predictors: $\mathbf{x}_j = (x_{1j}, ..., x_{nj})^\top$, for $j = 1, ..., p$.
- Residuals: $\epsilon = (\epsilon_1, ..., \epsilon_n)^\top$,
- Design matrix: $\mathbf{X}_{n\times(p+1)} = (\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_p)$ with $\mathbf{x}_0 = 1$.
- Regression coefficients: $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_p)^\top$.

Least Squares

The most popular estimation method is least squares, in which we pick the coefficients $\boldsymbol{\beta}$ to minimize the residual sum of squares

$$RSS(\boldsymbol{\beta}) = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

- Note that in supervised learning problems such as regression, we usually do not seek to model the distribution of the input variables. Thus the predictors always appear in the set of conditioning variables.
- The above criterion is reasonable as long as y_i's are conditionally independent given the inputs $\mathbf{x}_{(i)}$.
- In matrix form:

$$RSS(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

Least Squares

- The quantity $\mathbf{X}^+ = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top$ is known as the Moore-Penrose pseudo-inverse of the matrix $\mathbf{X}$. It can be regarded as a generalization of the notion of matrix inverse to nonsquare matrices. Indeed, if $\mathbf{X}$ is square and invertible, then $\mathbf{X}^+ = \mathbf{X}^{-1}$.
- The quantity $\mathbf{H} = \mathbf{X}(\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top$ is known as the "hat" matrix because $\hat{\mathbf{Y}} = \mathbf{HY}$. What is trace($\mathbf{H}$)?
- The resulting estimate $\widehat{\mathbf{y}}$ is the orthogonal projection of $\mathbf{y}$ onto the column space of $\mathbf{X}$.
- If $\mathbf{X}$ is not of full rank, e.g. $p > n$, then the LS solution is not unique.

Least Squares

Up to now we have made minimal assumptions about the true distribution of the data. In order to pin down the sampling properties of $\widehat{\beta}$, we now assume that

- the y_i's are uncorrelated and have constant variance σ^2, and that the $\mathbf{x}_{(i)}$ are fixed (non-random).

To draw inferences about the parameters and the model, we now assume

- $\epsilon_i \sim N(0, \sigma^2)$, i.i.d..

Maximum Likelihood Estimation

If we assume $\epsilon_i \sim N(0, \sigma^2)$ in the first place, we can conduct maximum likelihood estimation:
The Gauss-Markov Theorem:

Theorem

If $\boldsymbol{\alpha}^\top \boldsymbol{\beta}$ is estimable, its least squares estimator $\boldsymbol{\alpha}^\top \widehat{\boldsymbol{\beta}}$ is a linear unbiased estimator of $\boldsymbol{\alpha}^\top \boldsymbol{\beta}$, and the variance of $\boldsymbol{\alpha}^\top \widehat{\boldsymbol{\beta}}$ is uniformly less than that of any other linear unbiased estimator of $\boldsymbol{\alpha}^\top \boldsymbol{\beta}$.

Proof:

The Gauss-Markov Theorem

- Consider the mean squared error of an estimator $\tilde{\theta}$ in estimating θ:

$$MSE(\tilde{\theta}) = E(\tilde{\theta} - \theta)^2 = Var(\tilde{\theta}) + [E(\tilde{\theta}) - \theta]^2$$

- What is the implication of the Gauss-Markov theorem?
- It implies that the least squares estimator has the smallest mean squared error of all linear estimators with no bias.
- Question: Is it possible to achieve a smaller mean squared error or better prediction accuracy by considering biased estimation?

Computation via QR decomposition

Regression by Successive Orthogonalization:

1. Initialization: $\mathbf{z}_0 = \mathbf{x}_0 = 1$.
2. Successive Orthogonalization: For $j = 1, 2, ..., p$:
   Regress $\mathbf{x}_j$ on $\mathbf{z}_0, ..., \mathbf{z}_{j-1}$ to produce coefficients $\hat{\gamma}_{lj} = \frac{<\mathbf{z}_l, \mathbf{x}_l>}{<\mathbf{z}_l, \mathbf{z}_l>}$ for $l = 0, ..., j-1$ and the residual vector $\mathbf{z}_j = \mathbf{x}_j - \sum_{k=0}^{j-1} \hat{\gamma}_{lj} \mathbf{z}_k$.
3. Result: Regress $\mathbf{y}$ on the residual $\mathbf{z}_p$ gives the LS estimate $\hat{\beta}_p$.

Questions to think about:

- Why does this work?
- What does it imply?

Computation via QR decomposition

- The orthogonalization does not change the subspace spanned by the predictors.
- The $\mathbf{z}_j$'s are all orthogonal, and they form a basis for the column space of $\mathbf{X}$.
- Regressing $\mathbf{y}$ on $\mathbf{z}_j$'s also produces the orthogonal projection of $\mathbf{y}$ onto the column space of $\mathbf{X}$.
- Since $\mathbf{z}_p$ alone involves $\mathbf{x}_p$, we see that regressing $\mathbf{y}$ on $\mathbf{z}_p$ produces the LS estimates of β_p.
- Moreover, this implies that the LS estiamte $\hat{\beta}_j$ can be obtained by regressing $\mathbf{y}$ on $\mathbf{e}_j$, where $\mathbf{e}_j$ is the residual vector by regressing $\mathbf{x}_j$ on the other predictors.
- That is, the multiple regression coefficient $\hat{\beta}_j$ represents the additional contribution of $\mathbf{x}_j$ on $\mathbf{y}$, after $\mathbf{x}_j$ has been adjusted by the other predictors (including the intercept).

Computation via QR decomposition

- This is known as the Gram–Schmidt procedure for regression.
- From step 2, it can be easily shown that $\mathbf{X} = \mathbf{Z}\boldsymbol{\Gamma}$, where $\mathbf{Z} = (\mathbf{z}_0, \mathbf{z}_1, ..., \mathbf{z}_p)$, and $\boldsymbol{\Gamma}$ is the upper triangular matrix with entries $\hat{\gamma}_{lj}$.
- Let $\mathbf{D}$ be a $(p+1) \times (p+1)$ diagonal matrix with $\|\mathbf{z}_j\|$ on the diagonal, $\mathbf{Q} = \mathbf{Z}\mathbf{D}^{-1}$ and $\mathbf{R} = \mathbf{D}\boldsymbol{\Gamma}$, then $\mathbf{X} = \mathbf{Q}\mathbf{R}$ gives its QR decomposition of $\mathbf{X}$.
- We can show that

$$\widehat{\boldsymbol{\beta}} = \mathbf{R}^{-1}\mathbf{Q}^{\top}\mathbf{y}, \qquad \widehat{\mathbf{y}} = \mathbf{Q}\mathbf{Q}^{\top}\mathbf{y}.$$

Implications

- What happens if some predictors are highly correlated?
- What if we consider the singular value decomposition of $\mathbf{X}$?
- The least squares estimator has the smallest MSE of all unbiased linear estimators. However, it is possible to achieve a smaller mean squared error or better prediction accuracy by considering biased estimation. In other words, it is possible to trade a little bias for a large reduction in variance.
- In fact, biased estimates are commonly used in practice. One popular method is called ridge regression. It is in fact closely related to the SVD method.

What is Shrinkage Estimation

- What is shrinkage estimation?
- How does shrinkage estimation arise?
- What is the rational behind shrinkage estimation?
- Ridge regression.

Standardization

Without loss of generality, we assume that the response and predictors are centered and the predictors are standardized as follows.

$$\sum_{i=1}^{n} y_i = 0,$$

$$\sum_{i=1}^{n} x_{ij} = 0, \quad 1 \le j \le p,$$

$$\sum_{i=1}^{n} x_{ij}^2 = n, \quad 1 \le j \le p.$$

- Then there is no intercept in the model.
- Each predictor is standardized to have the same magnitude in L_2. So the corresponding regression coefficients are "comparable".
- After model fitting, the results can be readily transformed back to the original scale.

Linear Regression Model after Standardization

Model

$$y_i = x_{i1}\beta_1 + ... + x_{ip}\beta_p + \epsilon_i, \qquad 1 \leq i \leq n$$

- Response: $\mathbf{y} = (y_1, ..., y_n)^\top$
- Predictors: $\mathbf{x}_j = (x_{1j}, ..., x_{nj})^\top$, for $j = 1, ..., p$.
- Residuals: $\epsilon = (\epsilon_1, ..., \epsilon_n)^\top$,
- Design matrix: $\mathbf{X}_{n\times p} = (\mathbf{x}_1, ..., \mathbf{x}_p)$.
- Regression coefficients: $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)^\top$.

Ridge Regression: A Regularization Method

- Recall in decision theory that adding a regularization term to an error function helps to control over-fitting and achieve smaller prediction risk.
- In regression, The traditional penalized method is the ridge regression that uses the square of the L_2 norm of the coefficients as penalty.
- Originally, it was proposed to regularize ill-conditioned design matrices in linear regression.

Ridge Criterion

$$\widehat{\boldsymbol{\beta}}_{ridge}(\lambda) = \arg\min_{\boldsymbol{\beta}} \left\{ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|^2 \right\}$$

Ridge Regression: A Regularization Method

Ridge Estimator

$$\widehat{\boldsymbol{\beta}}_{ridge}(\lambda) = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}$$

Proof:

Ridge Regression: A Regularization Method

- Notice that with the choice of quadratic penalty $\boldsymbol{\beta}^\top \boldsymbol{\beta}$, the ridge estimator is again a linear function of $\mathbf{y}$.
- The solution adds a positive constant to the diagonal of $\mathbf{X}^\top \mathbf{X}$ before inversion. This makes the problem nonsingular, even if $\mathbf{X}$ is not of full column rank, and was the main motivation when ridge regression was first introduced.
- For $\lambda \in [\lambda_{\text{min}}, \lambda_{\text{max}}]$, we have a solution path

$$\{\widehat{\boldsymbol{\beta}}_{ridge}(\lambda) : \lambda \in [\lambda_{\text{min}}, \lambda_{\text{max}}]\}.$$

- The optimal λ and hence the optimal solution has to be selected by some information criteria or cross validation methods. We will get to this topic later.

Ridge Regression: Example

Simulation:

$$y_i = \sum_{j=1}^{p} x_{ij}\beta_j + \epsilon_i, \qquad 1 \le i \le n$$

where $\epsilon \sim N(0, \sigma^2)$.

- First generate z_{ij}'s and w independently from $N(0, \tau^2)$. Then compute

$$x_{ij} = z_{ij} + w, 1 \le j \le 4$$
$$x_{i5} = z_{i5} + 2w, x_{i6} = z_{i6} + w$$

- We set $n = 100$, $p = 100$, $\sigma = 1.5$, $\tau = 1$ and

$$(\beta_1, \beta_2, \beta_3) = (2, 1, -1)$$
$$\beta_j = 0, j \ge 4.$$

Ridge Regression: Example

```
n <- 100; p <- 100; sigma <- 1.5; tau <- 1

## generate X
Z <- matrix(nrow=n, ncol=p, rnorm(n*p, 0, tau))
w <- rnorm(n, 0, tau)
X <- matrix(nrow=n, ncol=p, NA)
for(i in 1:6){
    X[,i] <- Z[,i] + ifelse(i==5, 2, 1)*w
}
X[,7:100] <- Z[,7:100]

beta <- c(2, 1, -1, rep(0, 97)))
e <- rnorm(n, 0, sigma)
y <- X %*% beta + e

library(ncvreg)
fit <- ncvreg(X,y, gamma=1000, alpha=0.05)
plot(fit)
```

Ridge Regression: Bias and Variance Tradeoff

MSE of the Ridge Estimator

Let $\Sigma_\lambda = \mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}$. Then

$$E\|\widehat{\boldsymbol{\beta}}_{ridge}(\lambda) - \boldsymbol{\beta}\|^2 = \lambda^2 \boldsymbol{\beta}^\top \Sigma_\lambda^{-2} \boldsymbol{\beta} + \sigma^2 \text{trace}\{\Sigma_\lambda^{-1} \mathbf{X}^\top \mathbf{X} \Sigma_\lambda^{-1}\}$$

- How about the LS estimator?
- How do they compare?

Ridge Regression: Bias and Variance Tradeoff

MSE of the model fit:

$$E\|f - \widehat{f}_\lambda\|^2 = \sum_{i=1}^{n} E(f(\mathbf{x}_{(i)}) - \widehat{f}_\lambda(\mathbf{x}_{(i)}))^2$$

where $\widehat{f}_\lambda = \mathbf{S}_\lambda \mathbf{y} = \mathbf{U}\mathbf{W}_\lambda \mathbf{U}^\top \mathbf{y}$.

$$\begin{aligned} E\|f - \widehat{f}_\lambda\|^2 =& \|f - E\widehat{f}_\lambda\|^2 + E\|\widehat{f}_\lambda - E\widehat{f}_\lambda\|^2 \\ =& \|(\mathbf{I} - \mathbf{S}_\lambda)f\|^2 + E\|\mathbf{S}_\lambda \epsilon\|^2 \\ =& \|(\mathbf{I} - \mathbf{S}_\lambda)f\|^2 + \sigma^2 \text{trace}(\mathbf{S}_\lambda^\top \mathbf{S}_\lambda) \end{aligned}$$

Remarks

- Ridge regression protects against the potentially high variance of gradients estimated in the short directions. The implicit assumption is that the response will tend to vary most in the directions of high variance of the inputs. A related method is called principal component regression.
- We have demonstrated that using shrinkage estimation can alleviate overfitting and achieve better prediction performance.
- The shrinkage approach also has connection with kernel method and smoothing spline in nonparametric regression.
- Note that the methods considered so far are all linear.

Questions

- There are mainly two reasons why we are often not satisfied with the least squares estimates.
  - The first is prediction accuracy: the least squares estimates often have low bias but large variance.
  - The second reason is interpretation. The model is hard to interpret with a large number of predictors.
- Ridge regression addresses the first, but it does not perform variable selection.
- How to perform variable selection?

Penalized Empirical Risk Minimization

Consider continuous or convex relaxation of the L_0-regularization method

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\{ \widehat{R}(\boldsymbol{\beta}) + \sum_{j=1}^{p} p_\lambda(|\beta_j|) \right\}, \tag{5}$$

where $\widehat{R}(\boldsymbol{\beta})$ is the empirical risk function, $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_p)^\top$, and $p_\lambda(t)$, $t \geq 0$, is a nonnegative penalty function indexed by the regularization parameter $\lambda \geq 0$ with $p_\lambda(0) = 0$.

- For example, $\widehat{R}(\beta) = (2n)^{-1}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2$ for penalized least squares in the linear model.
- With an appropriately chosen penalty function, we hope to simultaneously select important variables and estimate their associated regression coefficients.

Choices of Penalty Function

- The L_q-penalty $p_\lambda(t) = \lambda t^q$ for $0 < q \leq 2$ in the bridge regression in **?**, which bridges the best subset selection (L_0-regularization) and ridge regression (L_2-regularization), including the L_1-penalty as a special case.
- The non-negative garrote introduced in **?** for shrinkage estimation and variable selection.
- The L_1-penalized least squares method was called the Lasso in **?**, and it is now collectively referred to as L_1-penalized empirical risk minimization method.
- There are many other choices of the penalty function to be introduced later; e.g., the SCAD in **?**, adaptive lasso in **?**, group lasso in **?**, etc.

Desirable Properties of Penalty Functions

Clearly the L_0-regularization method (best subset selection) possesses the variable selection feature (producing sparse models), while the L_2-regularization method (ridge regression) does not.

- What kind of penalty functions are capable of variable selection?
- What kind of penalty functions are good penalty functions for variable selection?

Desirable Properties of Penalty Functions

- *Sparsity*: The resulting estimator automatically sets small estimated coefficients to zero to accomplish variable selection and reduce model complexity.
- *Unbiasedness*: The resulting estimator is nearly unbiased, especially when the true coefficient β_j is large, to reduce model bias.
- *Continuity*: The resulting estimator is continuous in data to reduce instability in model prediction.

It is desirable to have $p'_\lambda(0+) > 0$ to ensure the sparsity of the regularized estimate; see their paper for more insights.

Summary

The two standard techniques for improving the OLS estimates, subset selection and ridge regression, both have drawbacks.

- Subset selection provides interpretable models but can be extremely variable because it is a discrete process. Small changes in the data can result in very different models being selected.
- Ridge regression is a continuous process that shrinks coefficients and hence is more stable: however, it does not set any coefficients to 0 and hence does not give an easily interpretable model.

Yes, we can do better!

Motivation

High-dimensional data arise in many diverse fields of scientific research. For example, in genomic/genetic studies, the following types of data are often encountered:

- Microarry gene expression data
- Array-based SNP data for genome wide association studies
- Copy number variation data
- .....
- Data from different platforms in one project.

An important and ubiquitous problem in such studies is to identify genetic determinants affecting a certain phenotype or clinical outcome. Applications in various other fields.....

High-dimensional Linear Regression

Model

$$y_i = x_{i1}\beta_1 + ... + x_{ip}\beta_p + \epsilon_i, \qquad 1 \le i \le n$$

- Response: $\mathbf{y} = (y_1, ..., y_n)^\top$
- Predictors: $\mathbf{x}_j = (x_{1j}, ..., x_{nj})^\top$, for $j = 1, ..., p$.
- Residuals: $\epsilon = (\epsilon_1, ..., \epsilon_n)^\top$,
- Design matrix: $\mathbf{X}_{n\times p} = (\mathbf{x}_1, ..., \mathbf{x}_p)$.
- Regression coefficients: $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)^\top$.
  True regression coefficients: $\boldsymbol{\beta}^o = (\beta_1^o, ..., \beta_p^o)^\top$.
- Oracle set: $\mathcal{O} = \{j : \beta_j^o \neq 0\}$.
- Underlying model dimension: $d^0 = \|\mathcal{O}\| = \#\{j : \beta_j^o \neq 0\}$

Centering and Standardization

Without loss of generality, we assume that the response and predictors are centered and the predictors are standardized as follows.

$$\sum_{i=1}^{n} y_i = 0,$$
$$\sum_{i=1}^{n} x_{ij} = 0, 1 \leq j \leq p,$$
$$\sum_{i=1}^{n} x_{ij}^2 = n, 1 \leq j \leq p.$$

- Then there is no intercept in the model.
- Each predictor is standardized to have the same magnitude in L_2. So the corresponding regression coefficients are "comparable".
- After model fitting, the results can be readily transformed back to the original scale.

Penalized Least Squares

We consider the penalized least squares (PLS) method

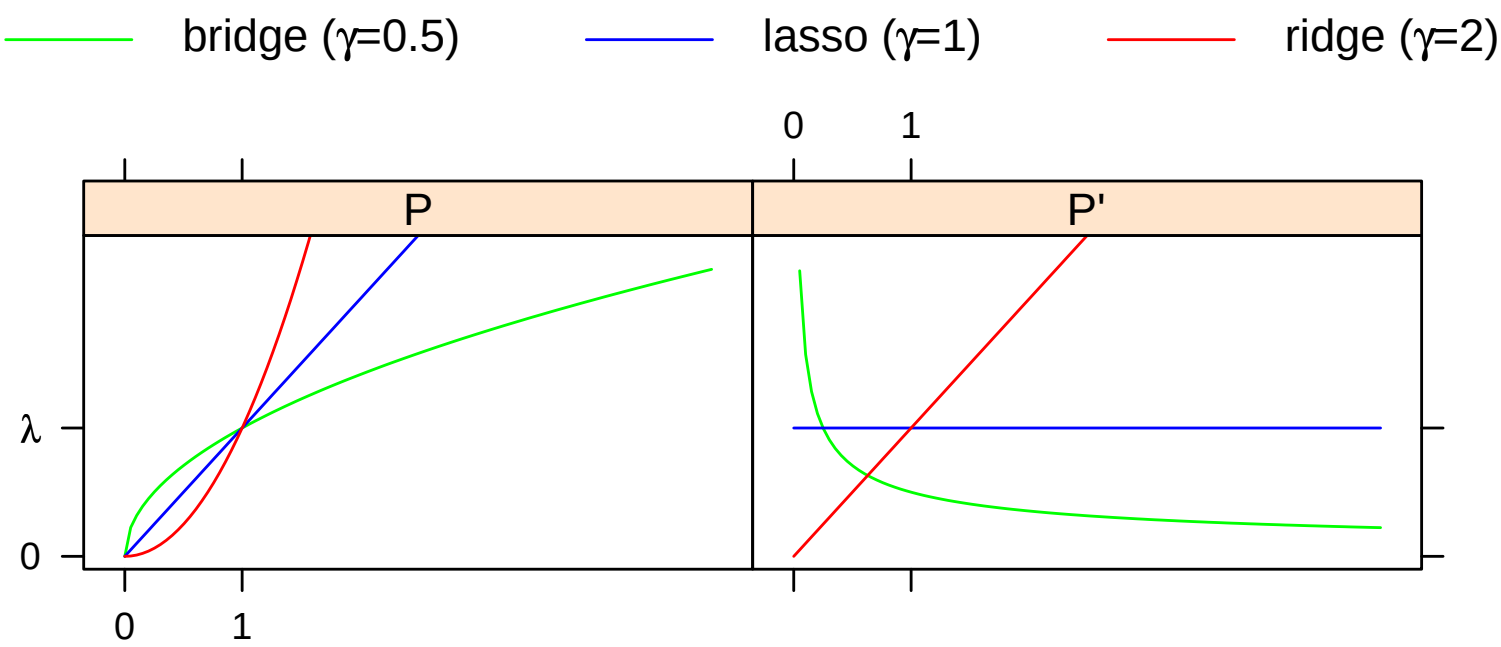
$$\min_{\boldsymbol{\beta}\in R^p}\{\frac{1}{2n}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2+\sum_{j=1}^{p}p_\lambda(|\beta_j|)\},$$

where $\|.\|$ denotes the L_2-norm. $p_\lambda(.)$ is a penalty function indexed by the regularization parameter $\lambda \geq 0$.
Common penalties:

- L_0 penalty (subset selection) and L_2 penalty (ridge regression).
- Bridge or L_γ penalty, $\gamma > 0$ (**?**).
- L_1 penalty or Lasso (**?**).
- SCAD penalty (**?**).
- MCP penalty (**?**).
- Group penalties (**?**) and bilevel penalties (**??**).

Bridge Penalties

$$f_\lambda(\theta) = \lambda|\theta|^\gamma, \gamma > 0.$$



Introduction to Lasso

Lasso

Lasso stands for "least absolute shrinkage and selection operator". There are two equivalent definitions.

- minimize the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant:

$$\hat{\beta} = \arg\min \left\{ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 \right\} \text{ subject to } \sum_j |\beta_j| \leq t.$$

- minimize the penalized sum of squares:

$$\hat{\beta} = \arg\min \{ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \sum_j |\beta_j| \}.$$

  - Because of the nature of L_1 penalty or constraint, Lasso is able to estimate some coefficients as exactly 0 and hence performs variable selection.
  - The lasso enjoys some of the favorable properties of both subset selection and ridge regression. It produces interpretable models and exhibits the stability of ridge regression.

Lasso

The motivation for the Lasso came from an interesting proposal of **?**. Breiman's *non-negative garotte* minimizes

$$\sum_{i=1}^{n}(y_i - \sum_j c_j \widehat{\beta}_j^{LS} x_{ij})^2 \ subject\ to\ c_j \geq 0,\ \sum c_j \leq t.$$

- The garotte starts with the OLS estimates and shrinks them by non-negative factors whose sum is constrained.
- It depends on both the sign and the magnitude of OLS. In contrast, the lasso avoids the explicit use of the OLS estimates.

Lasso is also closely related to the wavelet soft-thresholding method by Donoho and Johnstone, forward stagewise regression, and boosting methods.

Solution Path

- For each given λ, we solve the PLS problem. Therefore, for $\lambda \in [\lambda_{\min}, \lambda_{\max}]$, we have a solution path

$$\{\widehat{\boldsymbol{\beta}}_n(\lambda) : \lambda \in [\lambda_{\min}, \lambda_{\max}]\}.$$

- To examine the solution path, we can plot each component of $\widehat{\boldsymbol{\beta}}_n(\lambda)$ versus λ.
- In practice, we usually need to determine a value of λ, say, λ_*, and use $\widehat{\boldsymbol{\beta}}_n(\lambda^*)$ as the final estimator. This model selection step is usually done using some information criterion or cross validation techniques.
- Thus it is important to have fast algorithms for computing the whole solution path (or for a grid of λ values).
- There are multiple packages in R for computing the Lasso path: *ncvreg*, *glmnet*, *lars*....
- Note that the solution path can also be indexed by the constraint value t.

Comparing Lasso and Ridge: Example

Simulation:

$$y_i = \sum_{j=1}^{p} x_{ij}\beta_j + \epsilon_i, \qquad 1 \le i \le n$$

where $\epsilon \sim N(0, \sigma^2)$.

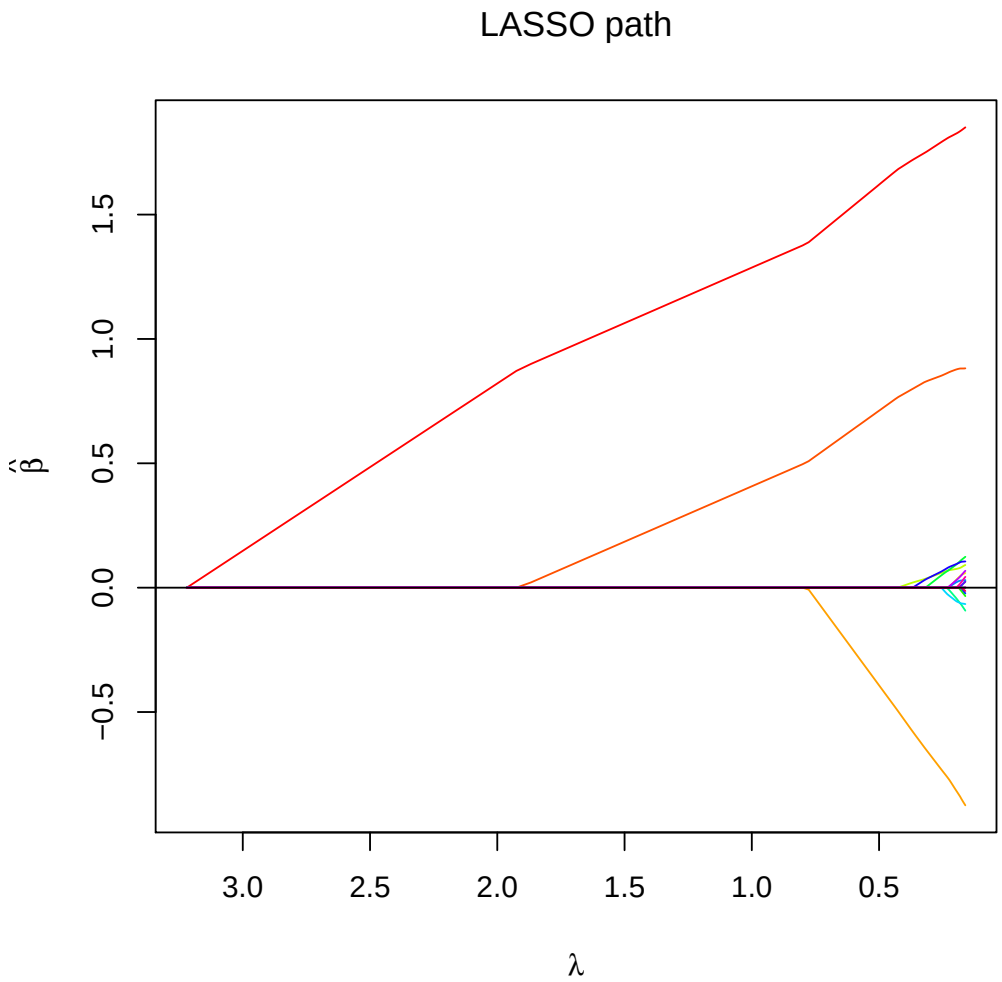
- First generate z_{ij}'s and w independently from $N(0, \tau^2)$. Then compute

$$x_{ij} = z_{ij} + w, 1 \le j \le 4$$
$$x_{i5} = z_{i5} + 2w, x_{i6} = z_{i6} + w$$

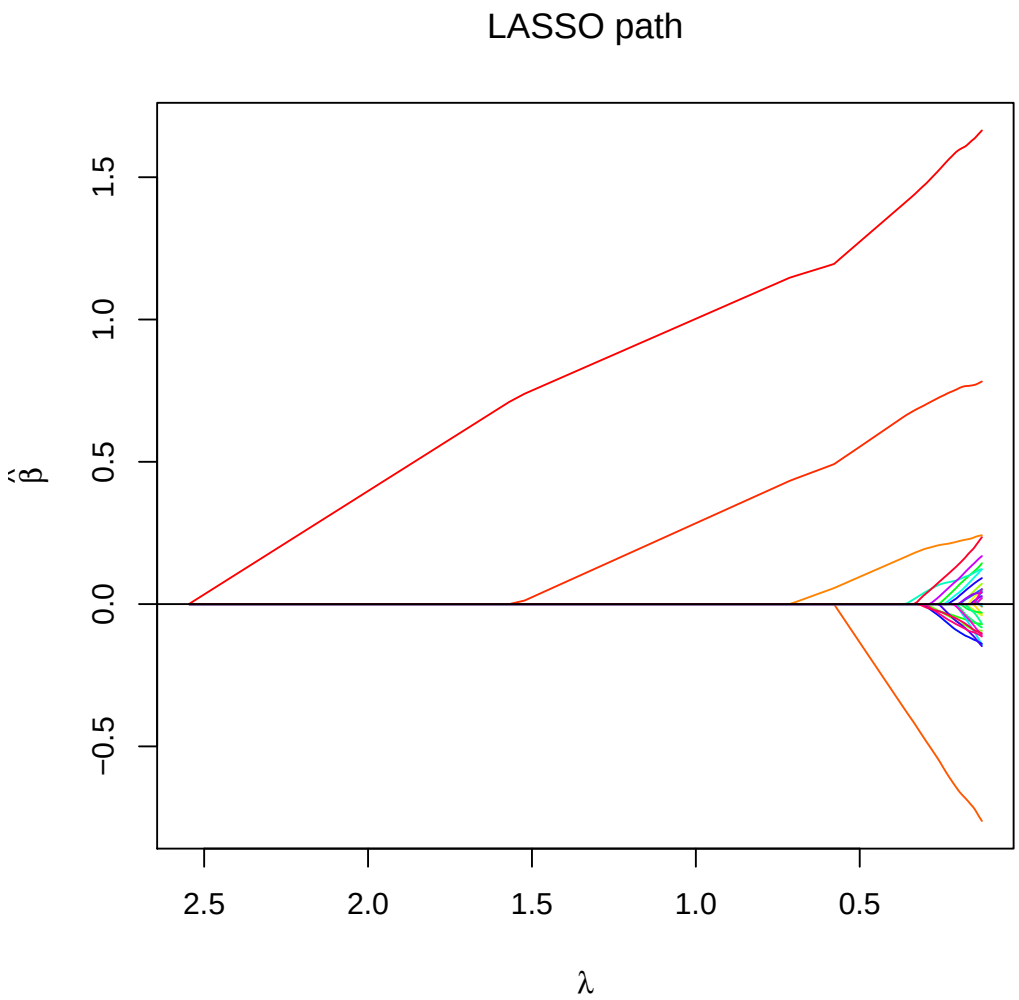
- We set $n = 100$, $p = 100$, $\sigma = 1.5$, $\tau = 1$ and

$$(\beta_1, \beta_2, \beta_3) = (2, 1, -1)$$
$$\beta_j = 0, j \ge 4.$$

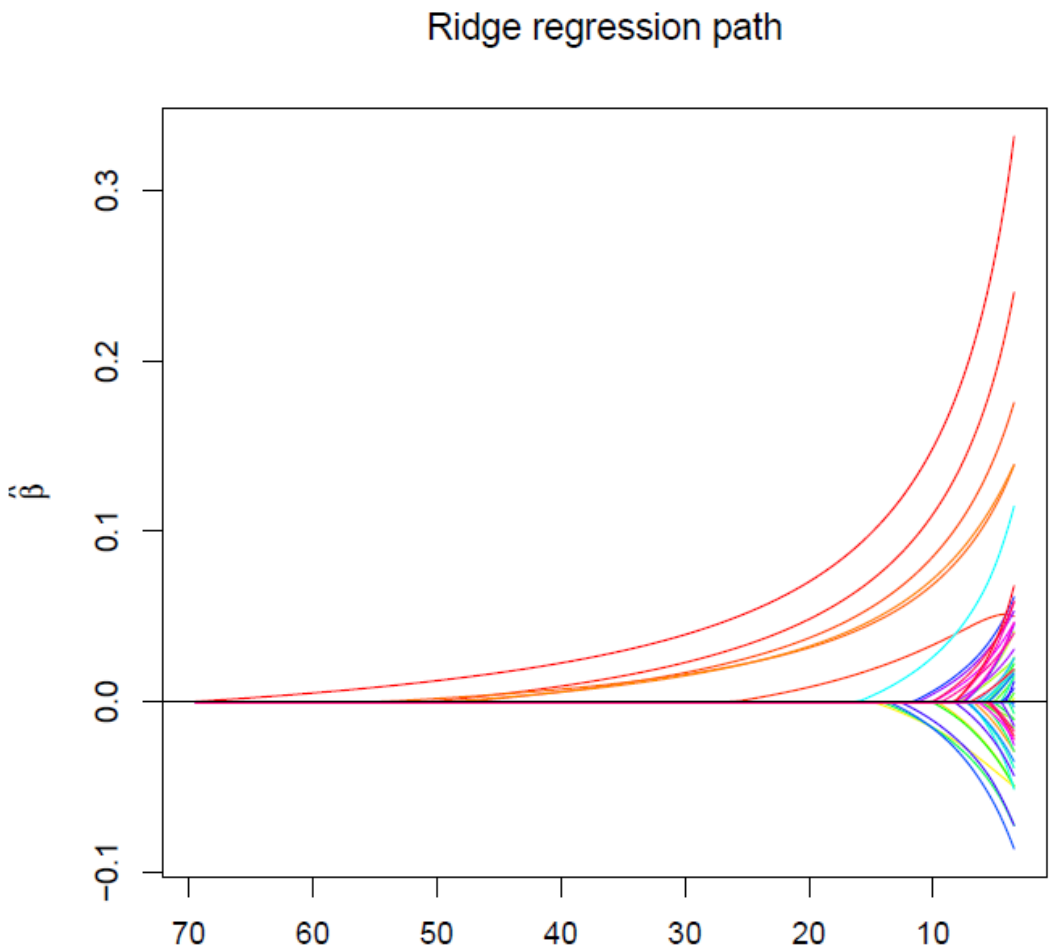
Comparing Lasso and Ridge: Example



Comparing Lasso and Ridge: Example



Comparing Lasso and Ridge: Example

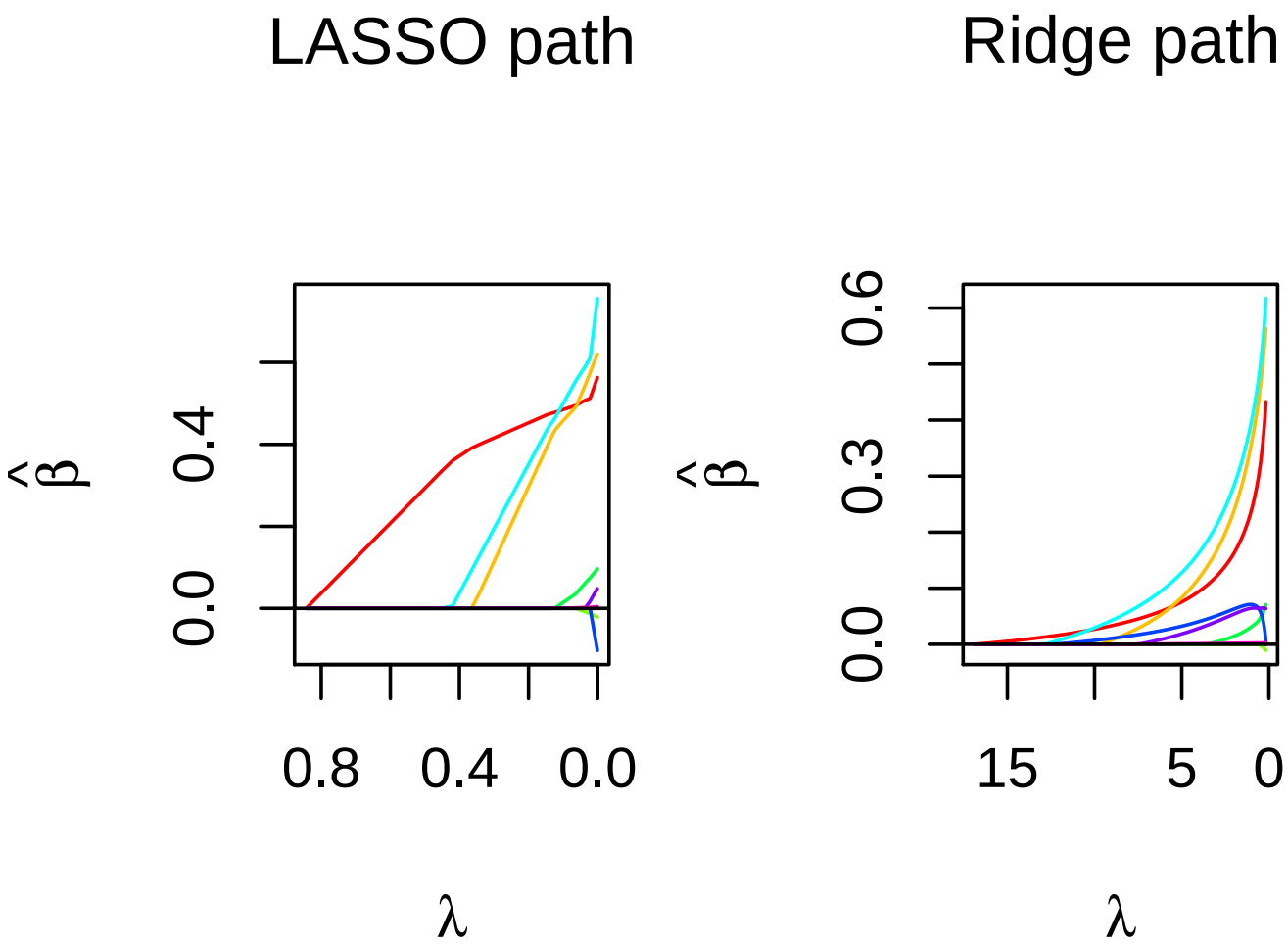


Comparing Lasso and Ridge: Example

Prostate dataset:

- lweight: Log prostate weight
- age: The man's age
- lbph: Log of the amount of benign hyperplasia
- svi: Seminal vesicle invasion: 1 = yes, 0 =No
- lcp: Log of capular penetration
- gleason: Gleason score
- pgg45: Percent of Gleason scores 4 or 5
- lpsa: Log PSA

Comparing Lasso and Ridge: Example

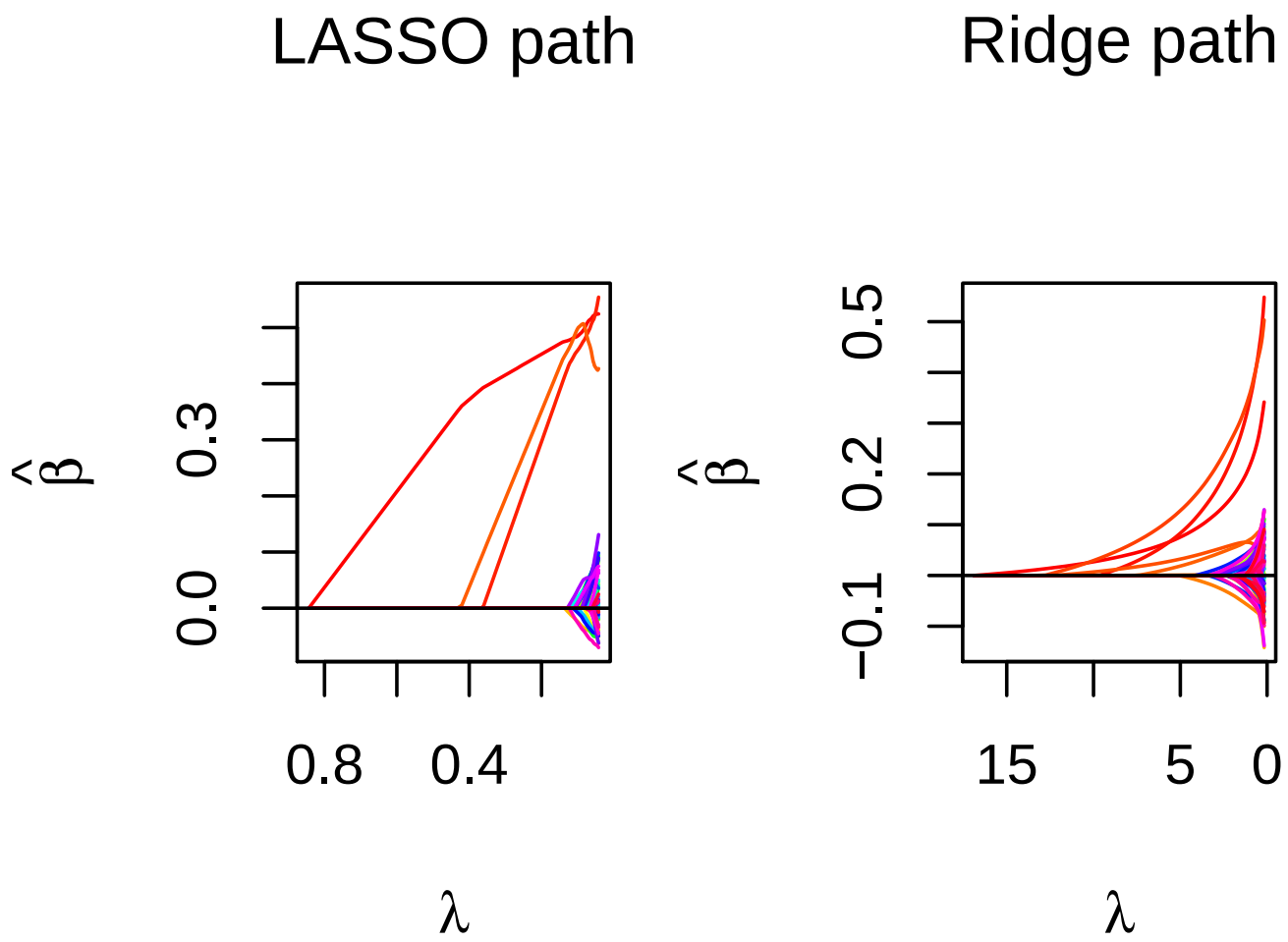


Comparing Lasso and Ridge: Example

Add some artificial noise variables:

```
library(ncvreg)
data(prostate)
X <- as.matrix(prostate[,1:8])
y <- prostate$lpsa
n=length(y)
p=100
X1 <- matrix(rnorm(n*p), nrow=n, ncol=p)
X=cbind(X, X1)
par(mfrow=c(1,2), mar=c(7,4.2,7,1))
fit <- ncvreg(X,y, gamma=1000, alpha=1.0)
plot(fit,main=expression(paste("LASSO path")), cex=0.6)
fit <- ncvreg(X,y, lambda.min=0.01, gamma=1000, alpha=0.05)
plot(fit,main=expression(paste("Ridge path")), cex=0.6)
```

Comparing Lasso and Ridge: Example



Orthogonal Design and Geometry

Orthogonal Design in PLS

Insight about the nature of the penalization methods can be gleaned from the orthogonal design case. When the design matrix multiplied by $n^{-1/2}$ is orthonormal, i.e. $\mathbf{X}^\top \mathbf{X} = n\mathbf{I}_p$, the penalized least squares problem reduces to the minimization of

$$\frac{1}{2n}\|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{LS}\|^2 + \frac{1}{2}\|\widehat{\boldsymbol{\beta}}^{LS} - \boldsymbol{\beta}\|^2 + \sum_{j=1}^{p} p_\lambda(|\beta_j|),$$

where $\widehat{\boldsymbol{\beta}}^{LS} = n^{-1}\mathbf{X}^\top \mathbf{y}$ is the OLS estimate.

Now the optimization problem is *separable* in β_j's. It suffices to consider the univariate PLS problem

$$\hat{\theta}(z) = \arg\min_{\theta \in R}\{\frac{1}{2}(z-\theta)^2 + p_\lambda(|\theta|)\}.$$

Orthogonal Design: PLS

? show that the PLS estimator $\hat{\theta}(z)$ possesses the properties:

- sparsity if $\min_{t\geq 0}\{t + p'_\lambda(t)\} > 0$;
- approximate unbiasedness if $p'_\lambda(t) = 0$ for large t;
- continuity if and only if $\arg\min_{t\geq 0}\{t + p'_\lambda(t)\} = 0$.

- In general for penalty functions, the singularity at the origin(i.e. $p'_\lambda(0+) > 0$) is needed for generating sparsity in variable selection and the concavity is needed to reduce the bias.
- These conditions are applicable for general PLS problems and more.

Orthogonal Design in Lasso

- Under orthogonal design, i.e., $\mathbf{X}^\top \mathbf{X} = n\mathbf{I}_p$, Lasso estimation can be greatly simplified as discussed above. The problem becomes solving

$$\widehat{\beta}_j = \arg\min_{\beta_j} \frac{1}{2}(\widehat{\beta}_j^{LS} - \beta_j)^2 + \lambda \|\beta_j\|_1.$$

- Solution: the Lasso estimator is given by the soft-thresholding operator:

$$\widehat{\beta}_j = \mathcal{S}(\widehat{\beta}_j^{LS}; \lambda),$$

where

$$\mathcal{S}(z;\lambda) = \text{sgn}(z)(|z| - \lambda)_+ = \begin{cases} z - \lambda, & \text{if } z > \lambda \\ 0, & \text{if } |z| \le \lambda \\ z + \lambda, & \text{if } z < -\lambda. \end{cases}$$

$\mathcal{S}(\cdot;\lambda)$ is called the soft-thresholding operator.

Soft-thresholding

- Suppose we wish to recover an unknown function f on [0,1] from noisy data

$$d_i = f(t_i) + \sigma z_i, \qquad i = 0, \cdots, n-1$$

where $t_i = i/n$, $z_i \sim N(0,1)$. The term de-noising is to optimize the mean-squared error $n^{-1}E\|\hat{f} - f\|_2^2$.

- Donoho and Johnstone proposed a soft-thresholding estimator

$$\hat{\beta}_j = \text{sgn}(\hat{\beta}_j^o)(|\hat{\beta}_j^o| - \gamma)^+.$$

- They applied it to the coefficients of a wavelet transform of a function measured with noise, then backtransformed to obtain a smooth estimate of the function.
- They proved many optimality results for the soft-thresholding estimator, one of which shows that asymptotically the estimator comes as close as subset selection to the performance of an ideal subset selector - one that used information about the actual parameters

Hard-thresholding

- The Hard-thresholding operator is defined by

$$\mathcal{H}(z;\lambda) = \text{sgn}(z)(|z| - \lambda)_+ = \begin{cases} z, & \text{if } |z| > \lambda \\ 0, & \text{if } |z| \leq \lambda \end{cases}$$

- It is the solution to

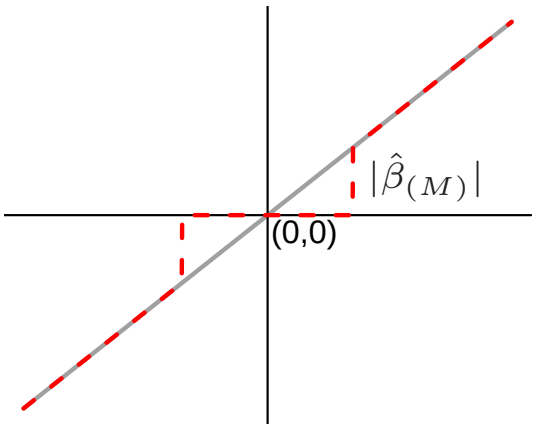
$$\mathcal{H}(z;\lambda) = \arg\min_b \left\{ \frac{1}{2}(z-b)^2 + \lambda^2 - (|b| - \lambda)^2 I(|b| < \lambda) \right\}$$

- This corresponds to the best subset selection. Note that the BSS of size k reduces to choosing the k largest coefficients in absolute value and setting the rest to 0.

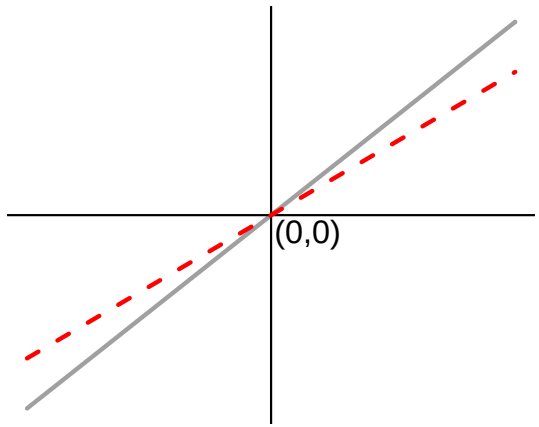
Comparison

Estimator	Formula
Best subset (size M)	$\hat{\beta}_j \cdot I(\lvert\hat{\beta}_j\rvert \geq \lvert\hat{\beta}_{(M)}\rvert)$
Ridge	$\hat{\beta}_j/(1+\lambda)$
Lasso	$\text{sign}(\hat{\beta}_j)(\lvert\hat{\beta}_j\rvert - \lambda)_+$

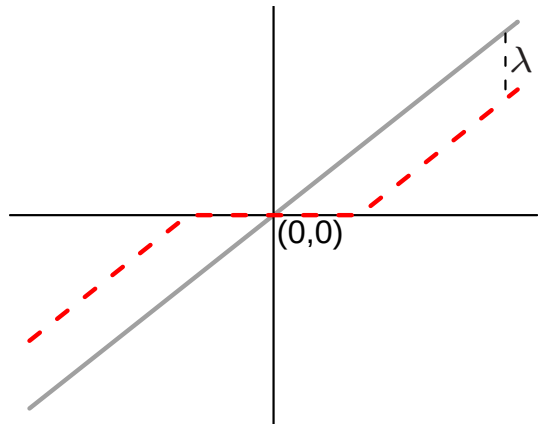
Best Subset



Ridge



Lasso



More on Orthogonal Design

- Ridge regression can be viewed as minimizing

$$\sum_{i=1}^{n}(y_i - \sum_j \beta_j x_{ij})^2 \text{ subject to } \sum_j \beta_j^2 \leq t,$$

  Under orthogonal design, the solution is given by

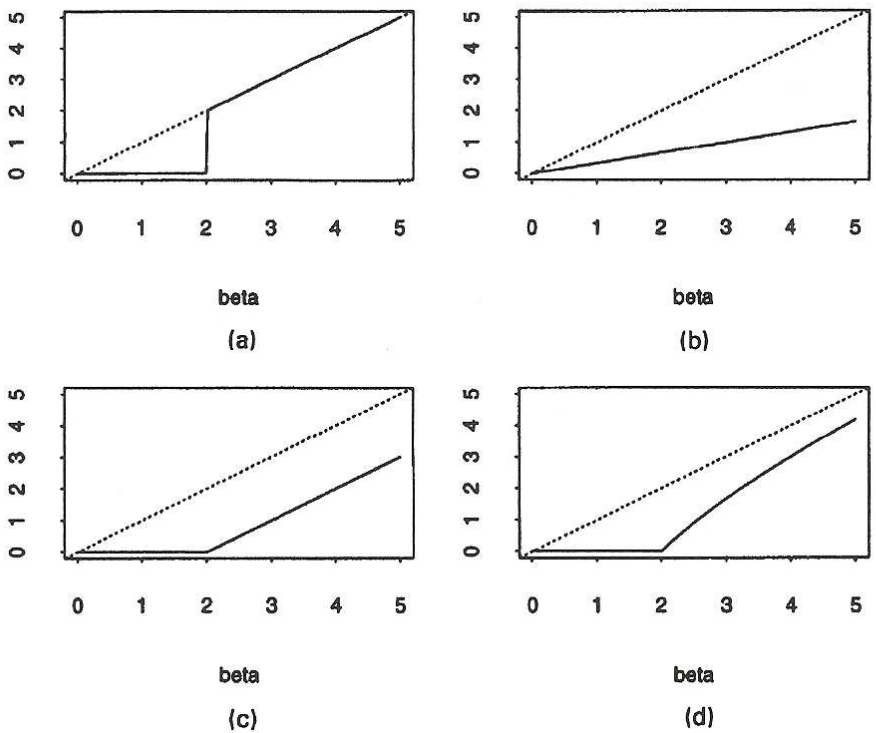
$$\frac{1}{1+\gamma}\hat{\beta}_j^{LS},$$

  where γ is determined by t.

- The non-negative garotte solution is given by

$$(1 - \frac{\gamma}{(\hat{\beta}_j^{LS})^2})_+ \hat{\beta}_j^{LS}.$$

Comparison


0 1 2 3 4 5
0 1 2 3 4 5
beta
(a)
0 1 2 3 4 5
0 1 2 3 4 5
beta
(b)
0 1 2 3 4 5
0 1 2 3 4 5
beta
(c)
0 1 2 3 4 5
0 1 2 3 4 5
beta
(d)

Geometry of Lasso

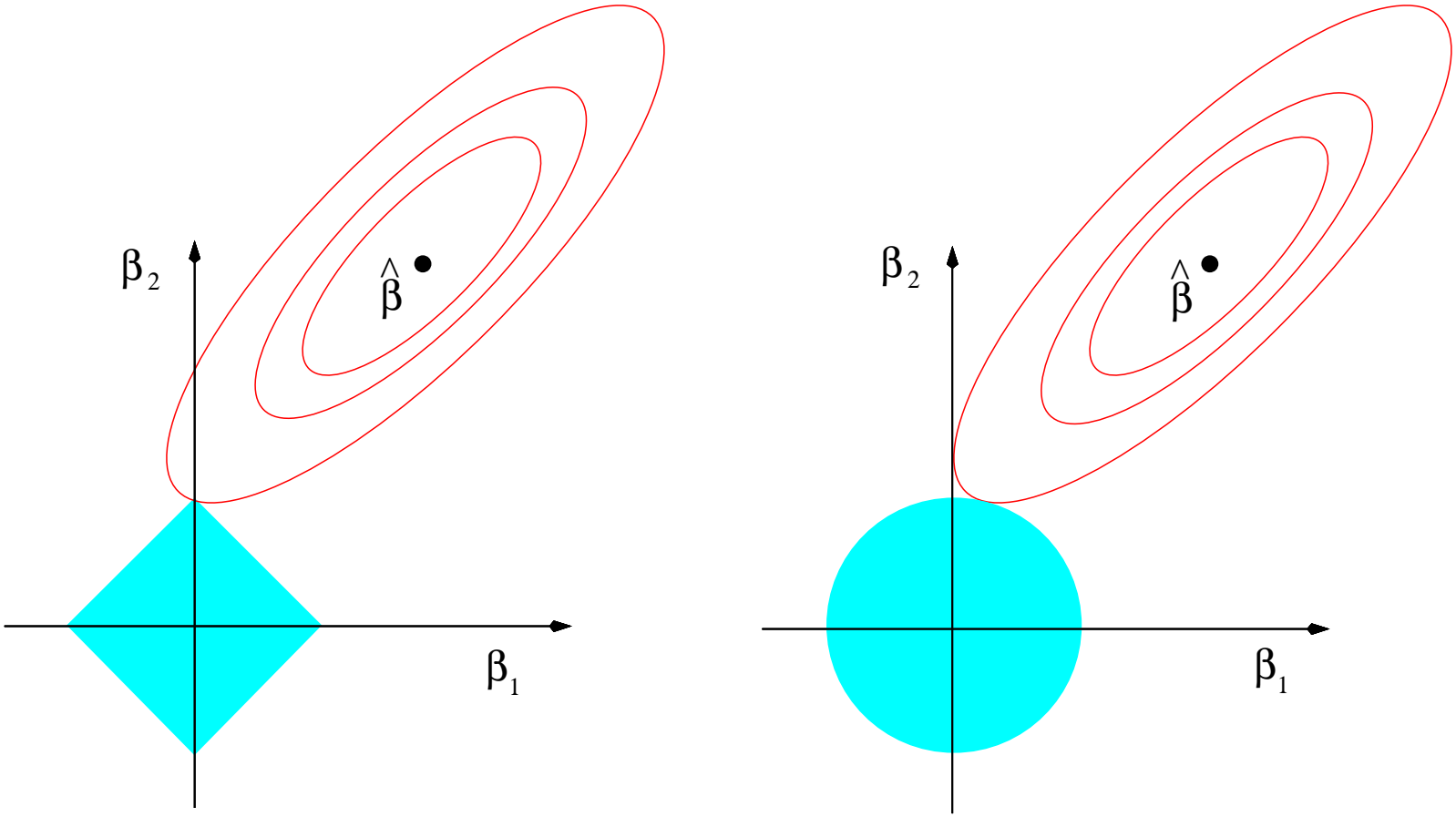
The residual sum of squared error term $\sum_{i=1}^{n}(y_i - \sum_j \beta_j x_{ij})^2$ equals the following quadratic function plus a constant

$$(\beta - \hat{\beta}^{LS})^T \mathbf{X}^T \mathbf{X} (\beta - \hat{\beta}^{LS})$$

- The contour of the above function is elliptical.
- These ellipsoids are centered at the OLS estimates.
- For Lasso, the L_1 constraint region is a ???.
- For ridge, the L_2 constraint region is a ???.

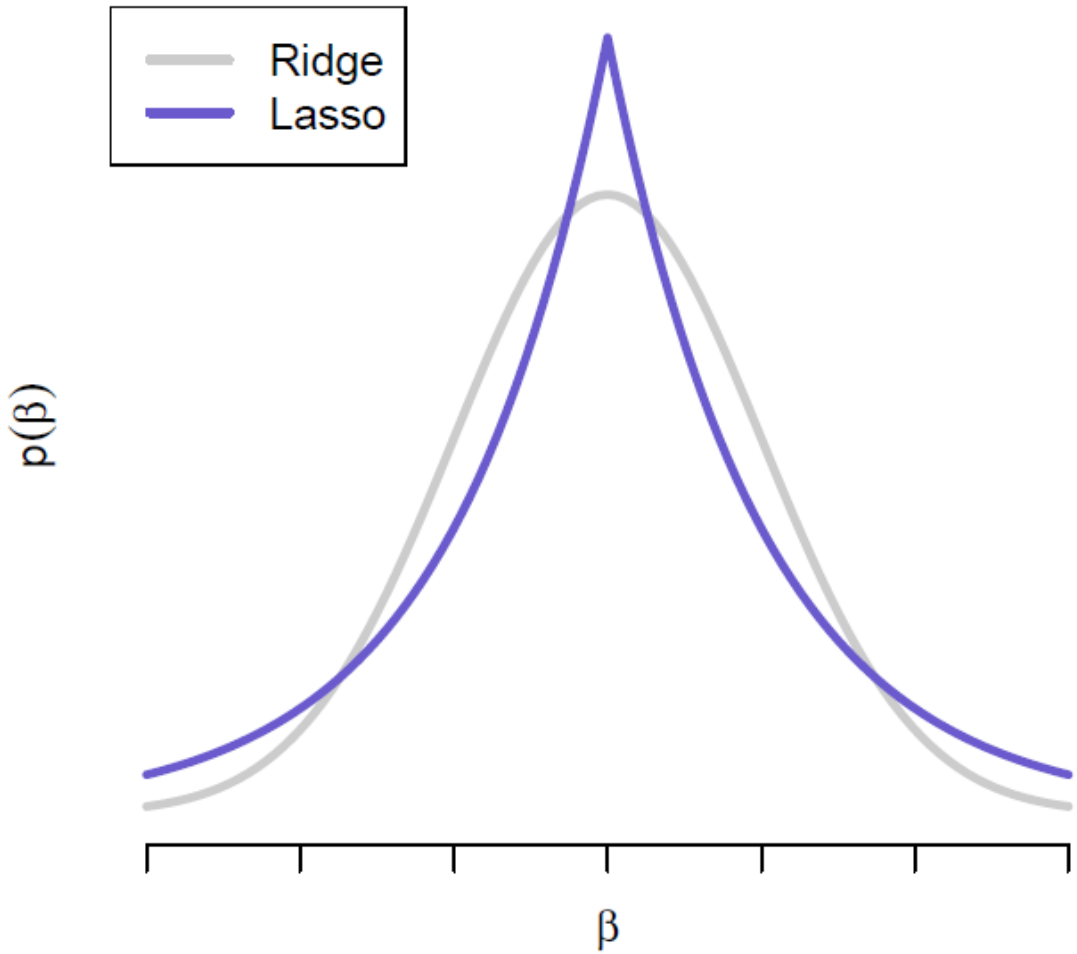
- Can you explain their different behaviors now?

Geometry of Lasso



Geometry of Lasso

- Another way of compare lasso and ridge is from a Bayesian perspective (**?**):



Computation by Coordinate Descent

Computation Algorithms

Lasso is a convex programming problem. Several algorithms have been proposed for computing L_1-penalized estimates.

- Convex optimization algorithms (**?**).
- Least angle regression (LARS) (**?**).
- Coordinate descent (**?**).
- Others.....

Here we first focus on the *coordinate descent algorithm*, which is simple, stable and efficient for a variety of high-dimensional models.

Coordinate Descent Algorithm

- Coordinate descent algorithms optimize a target function with respect to a single parameter at a time, iteratively cycling through all parameters until convergence is reached.
- They are ideal for problems that have a simple closed form solution in a single dimension but lack one in higher dimensions.

Coordinate Descent Algorithm: Derivation

The problem is to minimize L with respect to β_j, given current values for the regression coefficients $\tilde{\beta}_k, k \neq j$. Define

$$L_j(\beta_j; \lambda) = \frac{1}{2n} \sum_{i=1}^{n} \left(y_i - \sum_{k \neq j} x_{ik} \tilde{\beta}_k - x_{ij} \beta_j \right)^2 + \lambda |\beta_j|.$$

Denote $\tilde{y}_{ij} = \sum_{k \neq j} x_{ik} \tilde{\beta}_k, \tilde{r}_{ij} = y_i - \tilde{y}_{ij}$, and $\tilde{z}_j = n^{-1} \sum_{i=1}^{n} x_{ij} \tilde{r}_{ij}$. where $\tilde{r}_{ij}$ are the partial residuals with respect to the j^{th} covariate. Some algebra shows that

$$L_j(\beta_j; \lambda) = \frac{1}{2}(\beta_j - \tilde{z}_j)^2 + \lambda_1 |\beta_j| + \frac{1}{2n} \sum_{i=1}^{n} \tilde{r}_{ij}^2 - \frac{1}{2} \tilde{z}_j^2.$$

Thus, letting $\tilde{\beta}_j$ denote the minimizer of $L_j(\beta_j; \lambda)$,

$$\tilde{\beta}_j = S(\tilde{z}_j; \lambda),$$

where $S(\cdot; \lambda)$ is the soft-threshold operator.

Coordinate Descent Algorithm

For any fixed λ, given the current value $\tilde{\beta}^{(s)}$ in the sth iteration for $s = 0, 1 \ldots$, the algorithm for determining $\hat{\beta}$ is:

(1) Calculate

$$\tilde{z}_j = n^{-1} \sum_{i=1}^{n} x_{ij} r_i + \tilde{\beta}_j^{(s)},$$

where $r_i = y_i - \tilde{y}_i$ is the current residual.

(2) Update $\tilde{\beta}_j^{(s+1)}$ using (1).

(3) Update $r_i \leftarrow r_i - (\tilde{\beta}_j^{(s+1)} - \tilde{\beta}_j^{(s)}) x_{ij}$ for all i.

Coordinate Descent Algorithm

The efficiency of coordinate descent algorithms comes from several sources

- one dimension problem is usually easy to solve and often has explicit solution.
- many matrix operations may be avoided and the updates can be computed very rapidly.
- if the problem is convex, then the algorithm is guaranteed to converge to a global minimum.
- if we are computing a continuous path of solutions, our initial values will never be far from the solution and few iterations will be required.

Pathwise Solution

The coordinate descent algorithm can be used repeatedly to compute $\widehat{\beta}(\lambda)$ on a grid of λ values. Let λ_{max} be the smallest value for which all coefficients are 0 and λ_{min} be the minimum value of λ.

- We can use $\lambda_{\text{max}} = \max_{1 \le j \le p} |\mathbf{x}_j^\top \mathbf{y}|/n$. If the design matrix is full rank, λ_{min} can be 0; otherwise, we use $\lambda_{\text{min}} = \epsilon \lambda_{\text{max}}$ for some small ϵ, e.g., $\epsilon = 1e-4$.
- Let $\lambda_0 > \lambda_1 \cdots \lambda_K$ be a grid of decreasing λ values, where $\lambda_0 = \lambda_{\text{max}}$ and $\lambda_K = \lambda_{\text{min}}$.
- Start at λ_0 for which $\widehat{\beta}$ has the solution 0 (or close to 0), and proceed along the grid using the value of $\widehat{\beta}$ at the previous point of λ in the grid as the initial values for the current point. This is called warm start.

Penalty Parameter Selection

Penalty Parameter Selection: Cross Validation

- Divide the data into V roughly equal parts (5 or 10).
- For each $v = 1, \ldots, V$, fit the model with parameter λ to the other $V - 1$ parts. Denote the resulting estimates by $\hat{\boldsymbol{\beta}}^{-v}(\lambda)$.
- Compute the prediction error (PE) in predicting the vth part:

$$\text{PE}_v(\lambda) = \sum_{i \in v\text{th part}} (y_i - x_i' \hat{\boldsymbol{\beta}}^{-v}(\lambda))^2.$$

- Compute the overall cross-validation error

$$\text{CV}(\lambda) = \frac{1}{V} \sum_{v=1}^{V} \text{PE}_v(\lambda).$$

- Carry out the above steps for many values of λ and choose the value of λ that minimizes $\text{CV}(\lambda)$.

Penalty Parameter Selection: AIC and GCV

The Akaika's information criterion (AIC) and generalized cross validation (GCV).

- An AIC-type criterion for choosing λ is

$$\text{AIC}(\lambda) = \log\{\|\boldsymbol{y} - X\hat{\boldsymbol{\beta}}_n(\lambda)\|^2/n\} + 2df(\lambda)/n,$$

  where for the Lasso estimator, $df(\lambda) = \#$ of nonzero coefficients in the model fitted with λ.

- A generalized cross validation score (Wahba, 1990) is defined as

$$\text{GCV}(\lambda) = \frac{\|\boldsymbol{y} - X\hat{\boldsymbol{\beta}}_n(\lambda)\|^2}{n(1 - df(\lambda)/n)^2}.$$

It can be seen that these two criterions are close to each other when $df(\lambda)$ is relatively small compared to n.

Penalty Parameter Selection: BIC

The GCV and AIC are reasonable criteria for tuning. However, they tend to select more variables than the true model contains. Another criterion that is more aggressive in seeking a sparse model is the Bayesian information criterion (BIC):

$$\text{BIC}(\lambda) = \log\{\|\boldsymbol{y} - X\hat{\boldsymbol{\beta}}_n(\lambda)\|_2^2/n\} + \log(n)df(\lambda)/n.$$

The tuning parameter λ_n is selected as the minimizer of $\text{AIC}(\lambda)$, $\text{GCV}(\lambda)$ or $\text{BIC}(\lambda)$.

Least Angle Regression

Least Angle Regression

- Automatic model-building algorithms are familiar, and sometimes notorious, in the linear model literature: Forward Selection, Backward Elimination, Best Subset Selection and many others are used to produce good linear models for predicting a response $\mathbf{y}$ on the basis of some measured covariates $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_p$.
- The Lasso and Forward Stagewise regression will be discussed here.
- They are both motivated by a unified approach called Least Angle Regression("LARS").
  - Unifying explanation
  - Fast implementation
  - Fast way to choose tuning parameter

Forward Stepwise Selection

Forward Selection, or "forward stepwise regression":

- Given a collection of possible predictors, we select the one having largest absolute correlation with the response y, say x_{j_1}, and perform linear regression of y on x_{j_1}. This leaves a residual vector orthogonal to x_{j_1}, now considered to be the response.
- We project the other predictors orthogonally to x_{j_1} and repeat the selection process. After k steps this results in a set of $x_{j_1}, x_{j_2}, \cdots, x_{j_k}$ that are then used in the usual way to construct a k-parameter linear model.
- Forward Selection is an aggressive fitting technique that can be overly greedy, perhaps eliminating at the second step useful predictors that happen to be correlated with x_{j_1}.

Forward Stagewise Selection

Forward Stagewise is a much more cautious version of Forward Selection, which may take many tiny steps as it moves toward a final model.

1. It begins with $\hat{\mu} = \mathbf{X}\widehat{\beta} = 0$ and builds up the regression function in successive small steps. We assume the covariates have been standardized to have mean 0 and unit length and that the response has mean 0.
2. If $\hat{\mu}$ is the current Stagewise estimate, let $c(\hat{\mu})$ be the vector of current correlations $\hat{c} = c(\hat{\mu}) = \mathbf{X}^\top(\mathbf{y} - \hat{\mu})$, so that $\hat{c}_j$ is proportional to the correlation between covariate x_j and the current residual vector.

Forward Stagewise Selection: Continued

3 The next step of the Stagewise algorithm is taken in the direction of the greatest current correlation,

$$\hat{j} = \arg\max_j |\hat{c}_j| \ and \ \hat{\mu} \to \hat{\mu} + \varepsilon \ \text{sgn}(\hat{c}_{\hat{j}}) \ x_{\hat{j}},$$

with ε some small constant.

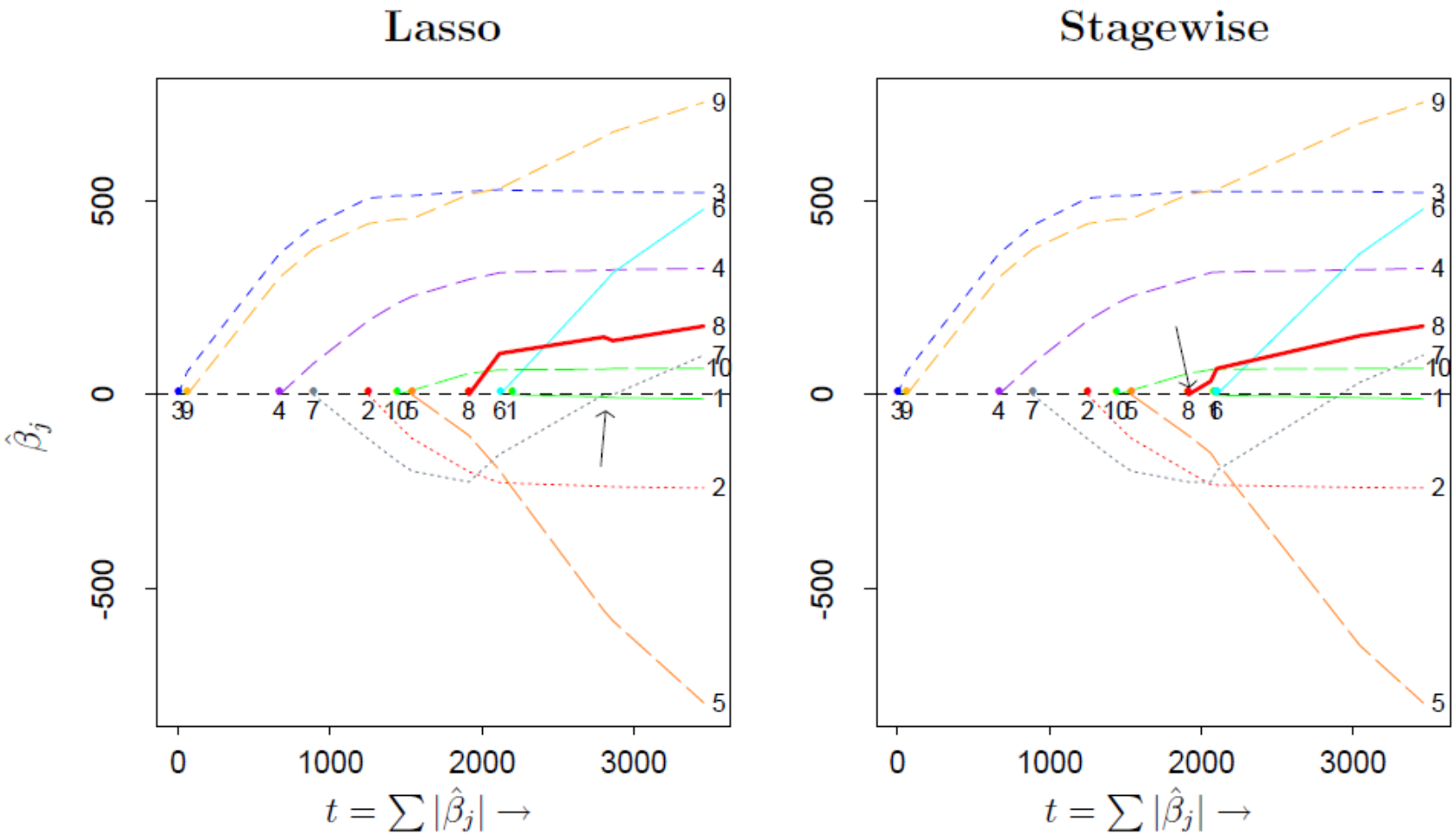
Small is important here: the "big" choice $\varepsilon = |\hat{c}_{\hat{j}}|$ leads to the classic Forward Selection technique, which can be overly greedy, impulsively eliminating covariates which are correlated with $x_{\hat{j}}$.

- Forward Stagewise = Least Squares Boosting.

Comparison between Lasso and Stagewise: Example

- How does these related to Lasso?
- Now let's use the diabetes study to compare Lasso and Stagewise regression.
- The left panel shows all Lasso solutions $\widehat{\boldsymbol{\beta}}(t)$, as t increases from 0, where $\widehat{\boldsymbol{\beta}} = 0$, to $t = 3460.0$, where $\widehat{\boldsymbol{\beta}}$ equals the OLS regression vector, the constraint no longer binding. The Lasso has a parsimony property.
- The right panel shows the coefficient plot for Stagewise. The estimates were built up in a large number of Stagewise steps.
- Surprise???

Comparison between Lasso and Stagewise: Example



Surprising??!! Although their definitions seem to be completely different, the results are nearly identical.

Comparison between Lasso and Stagewise: Example

Are Lasso and infinitesimal forward stagewise identical?

- With orthogonal predictors, yes.
- Otherwise similar

How to explain this?

- Least Angle Regression provides explanation, and fast implementation.

Stepwise, Forward Stagewise, Least Angle

Stepwise regression:

- Pick predictor most correlated with **y**.
- Bring predictor completely into model (full LS fit)

Forward stagewise:

- Pick predictor most correlated with **y**
- Increment coefficient for predictor.

Least Angle Regression:

- Pick predictor most correlated with **y**
- Bring predictor into model only to extent it is better than others.
- Move in least-squares direction until another variable is as correlated.

The LARS Algorithm

Least Angle Regression is a stylized version of the Stagewise procedure. Only p steps are required for the full set of solutions.

1. We start with $\widehat{\beta} = 0$ (same as FS), and find the predictor most correlated with $\mathbf{y}$, say x_{j_1}.
2. We take the largest step possible in the direction of this predictor until some other predictor, say x_{j_2}, has as much correlation with the current residual.
3. At this point LARS parts company with Forward Selection. Instead of continuing along x_{j_1}, LARS proceeds in a direction equiangular between the two predictors until a third variable x_{j_3} earns its way into the "most correlated" set.
4. LARS then proceeds equiangularly between x_{j_1}, x_{j_2}, x_{j_3}, i.e., along the "least angle direction", until a fourth variable enters, etc.

LARS builds up estimates $\hat{\mu} = X\widehat{\beta}$, in successive steps, each step adding one covariate to the model, so that after k steps, k of the $\widehat{\beta}_j$'s are non-zero.

The LARS Algorithm: Two-predictor case

- In a two-predictor case, the current correlations depend only on the projection onto the linear space $L(x)$ spanned by x_1 and x_2.
- The algorithm begins at $\hat{\mu}_0 = 0$, then augments $\hat{\mu}_0$ in the direction of x_1, to $\hat{\mu}_1 = \hat{\mu}_0 + \hat{\gamma}_1 x_1$.
  - Stagewise would choose γ_1 equal to some small value ε, and then repeat the process many times.
  - Classic Forward Selection would take γ_1 large enough to make $\hat{\mu}_1$ equal the projection of $\mathbf{y}$ onto $L(x_1)$.
  - LARS uses an intermediate value of γ_1, the value that makes the residual equally correlated with x_1 and x_2.
- The next LARS estimate is $\hat{\mu}_2 = \hat{\mu}_1 + \gamma_2 u_2$, with γ_2 chosen to make $\hat{\mu}_2 = y_2$ in the case $m = 2$. With $m > 2$ covariates, γ_2 would be smaller, leading to another change of direction.

The LARS Algorithm: Two-predictor case

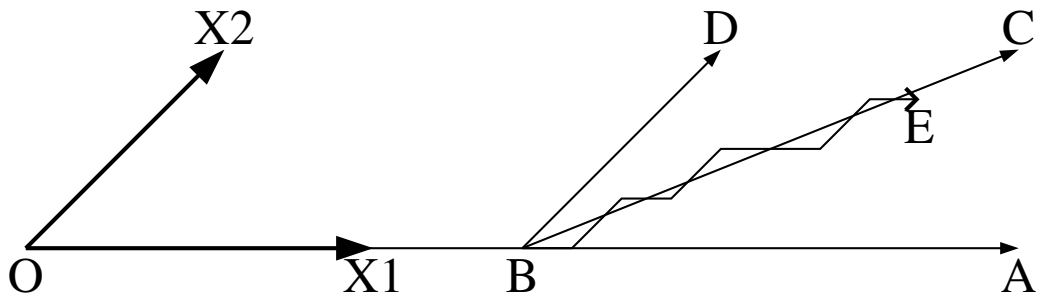


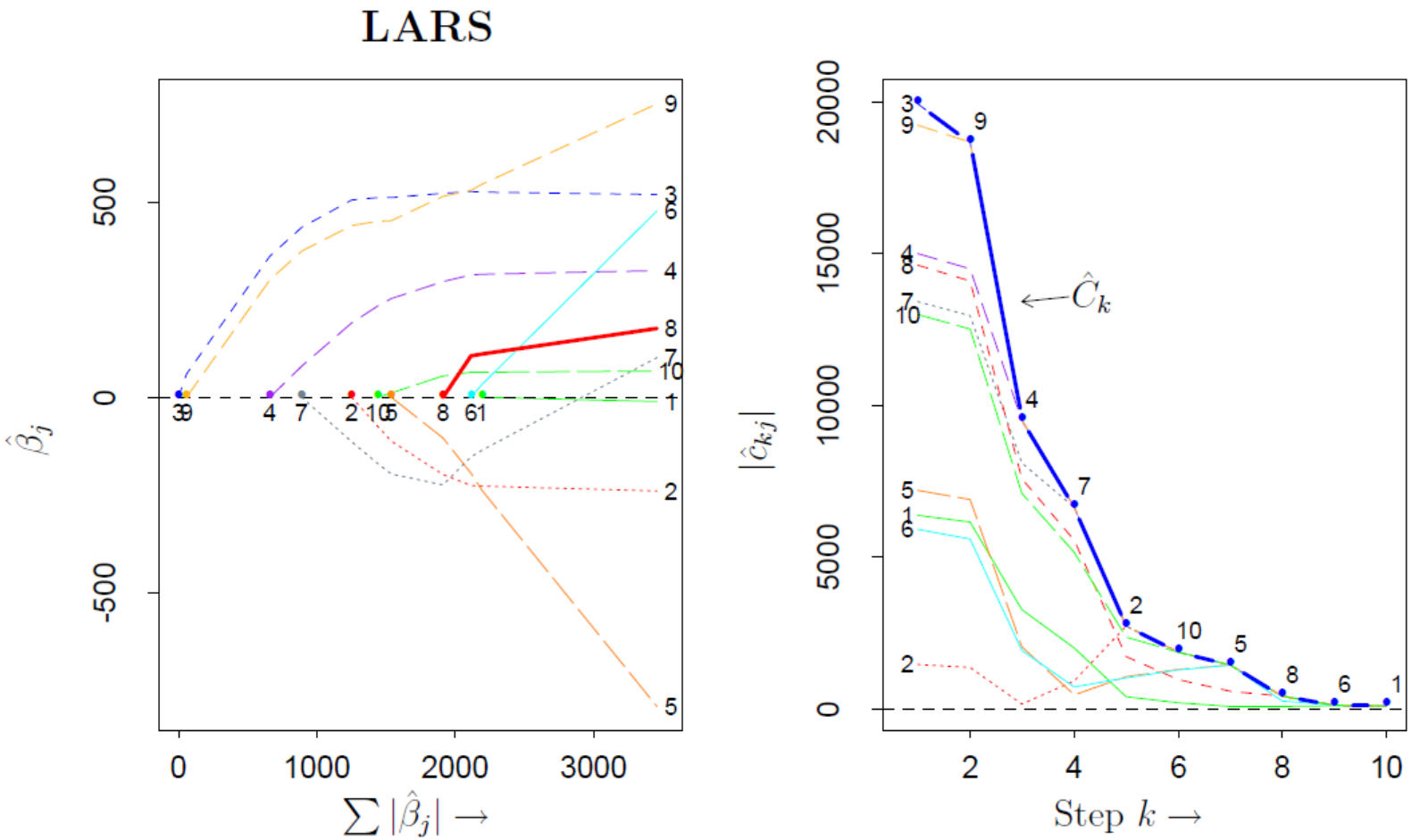
FIG 3. *The LAR algorithm in the case of 2 predictors. O is the prediction based solely on an intercept. $C = \hat{Y} = \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2$ is the ordinary least-squares fit, the projection of Y onto the subspace spanned by X_1 and X_2. A is the forward stepwise fit after one step; the second step proceeds to C. Stagewise takes a number of tiny steps from O to B, then takes steps alternating between the X_1 and X_2 directions, eventually reaching E; if allowed to continue it would reach C. LAR jumps from O to B in one step, where B is the point such that BC bisects the angle ABD. At the second step it jumps to C. LASSO follows a path from O to B, then from B to C. Here LAR agrees with LASSO and stagewise (as the step size $\to 0$ for stagewise). In higher dimensions additional conditions are needed for exact agreement to hold.*

The LARS Algorithm

1. Standardize the predictors to have mean zero and unit norm. Start with the residual $\mathbf{r} = \mathbf{y} - \bar{\mathbf{y}}$ (equivalent to centering) and $\widehat{\boldsymbol{\beta}} = 0$.
2. Find the predictor $\mathbf{x}_j$ most correlated with r.
3. Move $\widehat{\beta}_j$ from 0 towards its least-squares coefficient $< \mathbf{x}_j, \mathbf{r} >$, until some other competitor $\mathbf{x}_k$ has as much correlation with the current residual as does $\mathbf{x}_j$. Can you compute the step size?
4. Move $\widehat{\beta}_j$ and $\widehat{\beta}_k$ in the direction defined by their joint least squares coefficient of the current residual on $(\mathbf{x}_j, \mathbf{x}_k)$ (this keeps the correlations tied and decreasing), until some other competitor $\mathbf{x}_l$ has as much correlation with the current residual.
5. Continue in this way until all p predictors have been entered.

The LARS Algorithm: Details

The LARS Algorithm: Details



The LARS Algorithm: Details

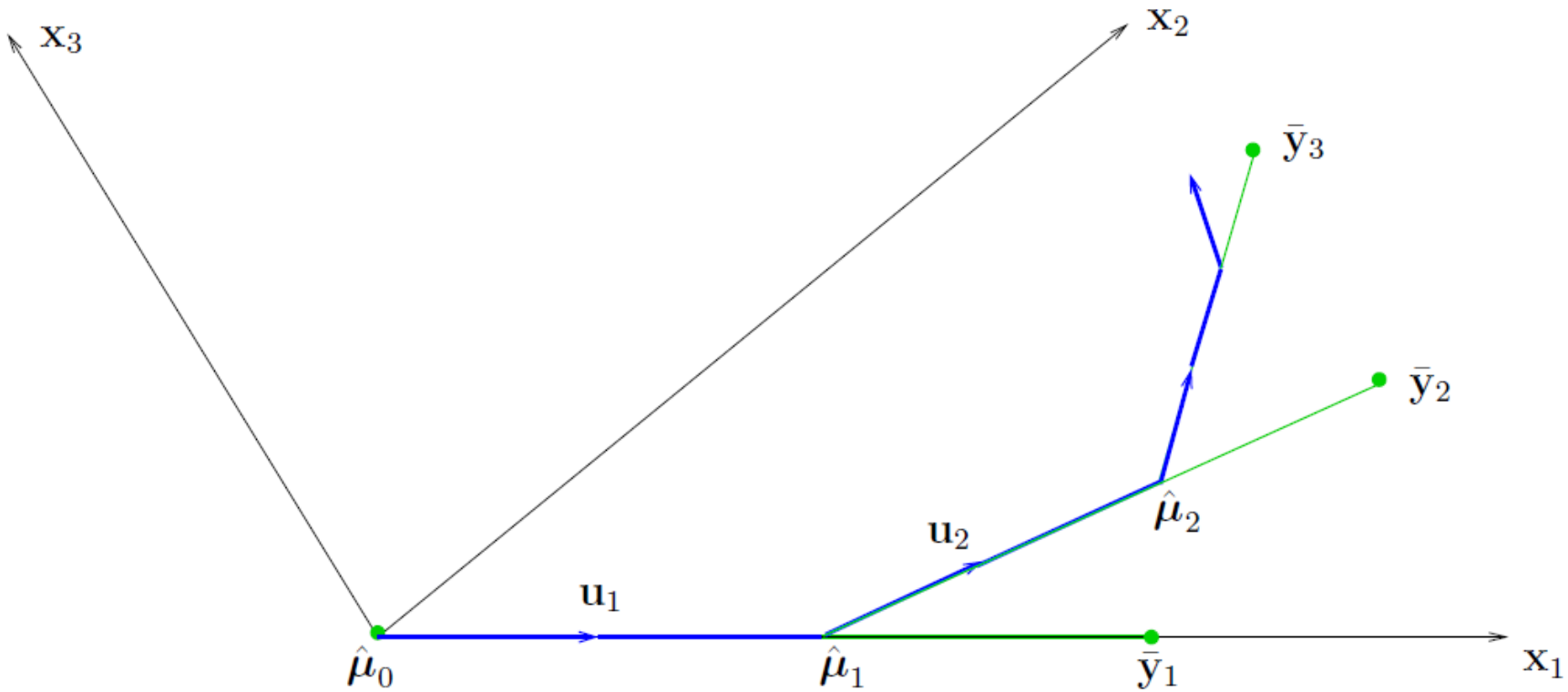
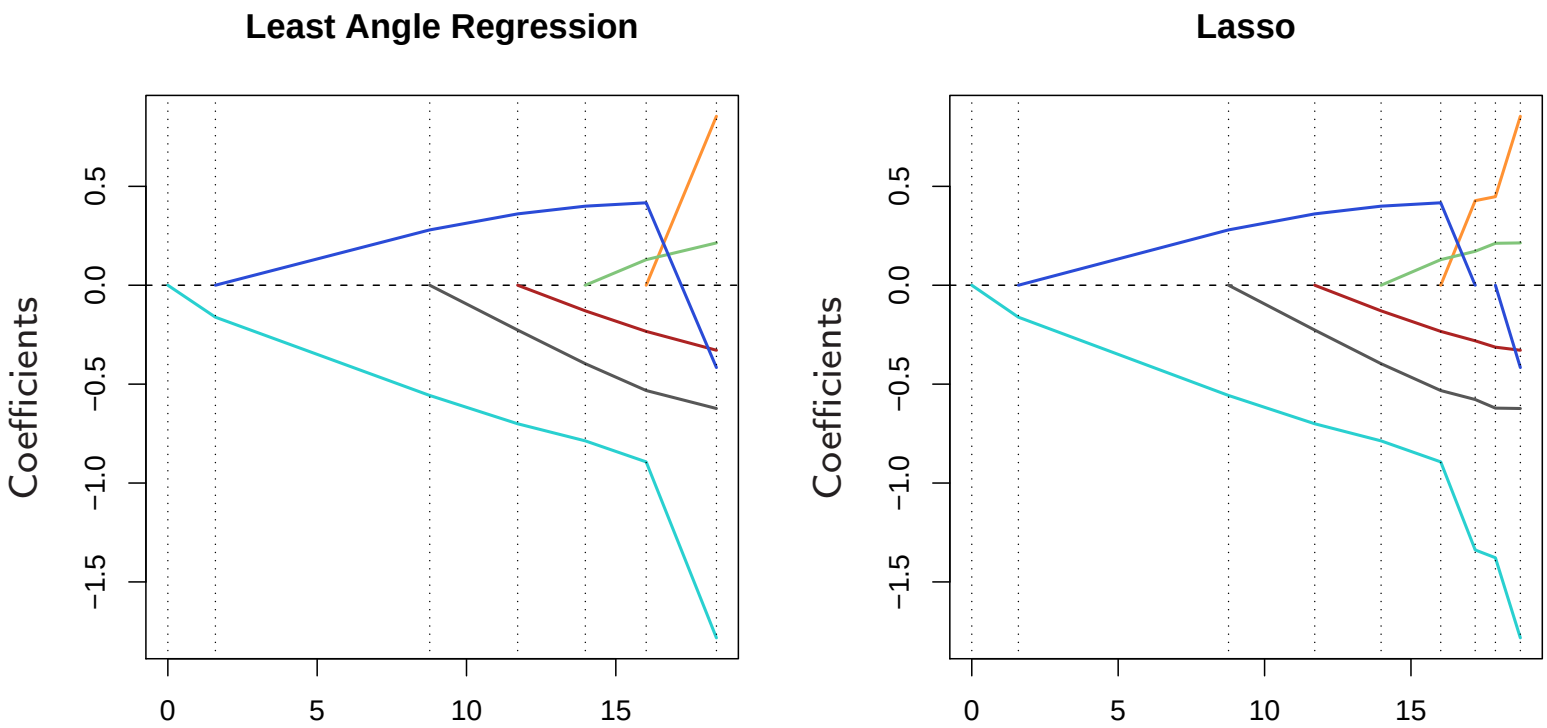


Figure 4. At each stage the LARS estimate $\widehat{\boldsymbol{\mu}}_k$ approaches, but does not reach, the corresponding OLS estimate $\bar{\mathbf{y}}_k$.

Comparison between LARS and Lasso: Example



Connection between LARS and Lasso

For LARS, suppose $\mathcal{A}$ is the active set of variables at some stage in the algorithm, tied in their absolute inner-product with the current residuals $\mathbf{r}$. Then

$$\mathbf{x}_j^\top \mathbf{r} = \gamma * \text{sgn}(\mathbf{x}_j^\top \mathbf{r}), \qquad \forall j \in \mathcal{A}$$
$$|\mathbf{x}_j^\top \mathbf{r}| \leq \gamma, \qquad \forall j \notin \mathcal{A}$$

For Lasso, Let $\mathcal{B}$ be the active set of variables in the solution for a given value of λ. Then

$$\mathbf{x}_j^\top \mathbf{r} = \lambda \cdot \text{sgn}(\beta_j), \qquad \forall j \in \mathcal{B}$$
$$|\mathbf{x}_j^\top \mathbf{r}| \leq \lambda, \qquad \forall j \notin \mathcal{B}$$

Modified LARS Algorithm for Lasso

4a If a non-zero coefficient hits zero, drop its variable from the active set of variables and recompute the current joint least squares direction.

Modified LARS Algorithm for Lasso

The LARS is slightly different than either Lasso or Stagewise. In fact, both Lasso and Stagewise are variants of LARS, or say, modified versions of LARS. For more details, we refer to Theorem 1 and Theorem 2 in **?**.

- Theorem 1. Under the Lasso modification, and assuming the "one at a time" condition, the LARS algorithm yields all Lasso solutions.
- Theorem 2. Under the Stagewise modification, the LARS algorithm yields all Stagewise solutions.