無情報事前分布によるLomax分布のパラメータ推定

理学部 応用数学科
学籍番号 1418112 氏名 三國 憲太郎

目 次

1　はじめに

Lomax 分布は Pareto Type II 分布とも呼ばれ, 経済や保険などに用いられることが多い. データが裾の重い分布に従うと考えられるときは指数分布の代わりに Lomax 分布を使うことが望ましい.

本論文では [1] を元に実際にシミュレーションを実行し, その方法やパフォーマンスについて検討する。ベイズ推定を行う上で, 事後分布が確率の公理を満たさないこと (improper となること) は望ましくない. ここでは 2 つの無情報事前分布を考え, それぞれに対する事後分布が確率の公理を満たすかどうか (proper となるかどうか) を確認する. 推定のパフォーマンスを確認するために最尤法と比較したシミュレーションを行う. MRE(平均相対誤差) と MSE(平均二乗誤差) により推定の精度を評価する.

2　ベイズ理論

2.1　ベイズ推定

ベイズ推定は推定するパラメータを確率変数として考える手法である. パラメータに関する事前の知識から事前分布を決め, 観測されたデータを用いて事後分布を計算する. 求めるパラメータを θ, 事前分布を $p(\theta)$ とする. $\boldsymbol{x}$ を観測データとすると事後分布 $p(\theta|\boldsymbol{x})$ はベイズの定理を用いて

$$p(\theta|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\theta)p(\theta)}{p(\boldsymbol{x})}$$

と表される. 通常, 事後分布は複雑な形になりそのまま扱うことが困難であることが多いため, 事後分布に従う乱数を生成して評価する. 本論文ではメトロポリス法 (2.2 節参照) を用いて事後分布に従う乱数を生成する.

事後分布が

$$\int_{-\infty}^{\infty} p(\theta|\boldsymbol{x})d\theta \neq 1$$

となり, 確率の公理を満たさなくなる場合がある. このような事後分布を improper な事後分布という. 事後分布が確率の公理を満たさない状態は望ましくない. これについて 4 章で詳しく扱う.

事前分布はパラメータに関する事前情報から決めるが, 事前情報がない場合には無情報事前分布を用いる. 本論文では無情報事前分布の一つである Jeffreys の事前分布 (4) と, パラメータ間の事前独立性を仮定した Jeffreys の事前分布 (5) を考える.

2.2　メトロポリス法

この章の参考文献は [2] である. ベイズ統計では事後分布が複雑な形になることが多く, そのままの形では不便であることが多い. そのため事後分布に従う乱数を MCMC 法によって生成し, それをもとに事後分布の評価を行う. メトロポリス法は MCMC 法の一つであり, 以下のアルゴリズムで乱数を生成する. 求めたい分布を $f(\theta)$ とする.

1. 初期値 θ_0 と提案分布を $q(\theta|\theta')$ を決める. $q(\theta|\theta')$ は乱数の候補を生成する確率分布であり, 特にメトロポリス法における提案分布は $q(\theta|\theta') = q(\theta'|\theta)$ を仮定した手法である.

2. $n+1$ 番目の乱数の候補を $q(\theta^*|\theta_n)$ から生成し, 採択率 $r = \dfrac{f(\theta^*)}{f(\theta_n)}$ を計算する.

3. $r \geq 1$ のとき θ^* は $f(\theta)$ の乱数として採択され, $\theta_{n+1} = \theta^*$ となる.

   $r < 1$ のとき確率 r で θ^* を採択する. θ^* が棄却された時, 乱数は更新されずに $\theta_{n+1} = \theta_n$ となる.

3 モデルの定義

Lomax 分布の確率密度関数は

$$f(x|\beta,\alpha) = \frac{\alpha}{\beta}\left(1+\frac{x}{\beta}\right)^{-(\alpha+1)} \qquad (x \geq 0) \tag{1}$$

ただし $\beta > 0,\ \alpha > 0$ とする.

$$\mathrm{E}(X) = \frac{\beta}{\alpha-1} \quad (\alpha > 1),$$

$$\mathrm{Var}(X) = \frac{\alpha\beta^2}{(\alpha-1)^2(\alpha-2)} \quad (\alpha > 2).$$

以下の二つの確率変数を考えると, Lomax 分布の階層的表現が得られる.

$$X|\beta,\lambda \sim \text{Exponential}\left(\frac{\lambda}{\beta}\right), \quad \lambda|\alpha \sim \text{Gamma}(\alpha,1).$$

実際に X,λ の結合確率密度関数は

$$f(x|\beta,\lambda)f(\lambda|\alpha) = \frac{1}{\beta\Gamma(\alpha)}\lambda^{\alpha}\exp\left\{-\lambda\left(1+\frac{x}{\beta}\right)\right\}$$

であり, λ で積分すると

$$\begin{aligned} f(x|\beta,\alpha) &= \frac{1}{\beta\Gamma(\alpha)}\int_0^{\infty}\lambda^{\alpha}\exp\left\{-\lambda\left(1+\frac{x}{\beta}\right)\right\}d\lambda \\ &= \frac{1}{\beta\Gamma(\alpha)}\Gamma(\alpha+1)\left(1+\frac{x}{\beta}\right)^{-(\alpha+1)} \\ &= \frac{\alpha}{\beta}\left(1+\frac{x}{\beta}\right)^{-(\alpha+1)}. \end{aligned}$$

よって, $X \sim \text{Lomax}(\beta,\alpha)$ となる. $\boldsymbol{X} = (X_1,\ldots,X_n)$ を (1) からのランダム標本, $\boldsymbol{\lambda} = (\lambda_1,\ldots,\lambda_n)$ とする. λ の条件付き分布は階層的表現を用いると以下のように表される.

$$\begin{aligned} f(\boldsymbol{\lambda}|\boldsymbol{x},\beta,\alpha) &= \frac{f(\boldsymbol{x}|\boldsymbol{\lambda},\beta,\alpha)f(\boldsymbol{\lambda},\beta,\alpha)}{f(\boldsymbol{x},\beta,\alpha)} \\ &= \frac{f(\boldsymbol{x}|\beta,\boldsymbol{\lambda})f(\boldsymbol{\lambda}|\beta,\alpha)\pi(\beta,\alpha)}{f(\boldsymbol{x}|\beta,\alpha)\pi(\beta,\alpha)} \\ &\propto f(\boldsymbol{x}|\beta,\boldsymbol{\lambda})f(\boldsymbol{\lambda}|\alpha) \\ &\propto \prod_{i=1}^{n}\lambda_i\exp\left\{\frac{-\lambda_i x_i}{\beta}\right\}\prod_{i=1}^{n}\lambda_i^{\alpha-1}\exp\{-\lambda_i\} \\ &= \prod_{i=1}^{n}\lambda_i^{\alpha}\exp\left\{-\lambda_i\left(1+\frac{x_i}{\beta}\right)\right\}. \end{aligned}$$

よって以下が従う.

$$\lambda_i|x_i,\beta,\alpha \sim \text{Gamma}\left(\alpha+1, 1+\frac{x_i}{\beta}\right).$$

また, 階層的表現を用いてパラメータ α の条件付き分布を変形すると

$$\begin{aligned} f(\alpha|\boldsymbol{x},\boldsymbol{\lambda},\beta) &= \frac{f(\boldsymbol{\lambda}|\alpha,\beta,\boldsymbol{x})f(\alpha,\beta,\boldsymbol{x})}{f(\boldsymbol{x},\boldsymbol{\lambda},\beta)} \\ &= \frac{f(\boldsymbol{\lambda}|\alpha)f(\boldsymbol{x}|\alpha,\beta)\pi(\beta,\alpha)}{f(\boldsymbol{x},\boldsymbol{\lambda},\beta)} \\ &= \frac{f(\boldsymbol{\lambda}|\alpha)f(\boldsymbol{x}|\beta)\pi(\beta,\alpha)}{f(\boldsymbol{x},\boldsymbol{\lambda},\beta)} \\ &\propto f(\boldsymbol{\lambda}|\alpha)\pi(\beta,\alpha). \end{aligned}$$

よって

$$f(\alpha|\boldsymbol{x},\boldsymbol{\lambda},\beta) \propto f(\boldsymbol{\lambda}|\alpha)\pi(\beta,\alpha) \propto [\Gamma(\alpha)]^{-n}\left(\prod_{i=1}^{n}\lambda_i\right)^{\alpha-1}\pi(\beta,\alpha). \tag{2}$$

α の事後分布は $\boldsymbol{x}$ の関数には比例しない形になっていることに注意する. 同様に β の条件付き分布も階層的表現を用いて変形すると

$$\begin{aligned} f(\beta|\boldsymbol{x},\boldsymbol{\lambda},\alpha) &= \frac{f(\boldsymbol{x}|\beta,\boldsymbol{\lambda},\alpha)f(\beta,\boldsymbol{\lambda},\alpha)}{f(\boldsymbol{x},\boldsymbol{\lambda},\alpha)} \\ &= \frac{f(\boldsymbol{x}|\beta,\boldsymbol{\lambda})f(\boldsymbol{\lambda}|\beta,\alpha)\pi(\beta,\alpha)}{f(\boldsymbol{x},\boldsymbol{\lambda},\alpha)} \\ &= \frac{f(\boldsymbol{x}|\beta,\boldsymbol{\lambda})f(\boldsymbol{\lambda}|\alpha)\pi(\beta,\alpha)}{f(\boldsymbol{x},\boldsymbol{\lambda},\alpha)} \\ &\propto f(\boldsymbol{x}|\beta,\boldsymbol{\lambda})\pi(\beta,\alpha). \end{aligned}$$

よって

$$f(\beta|\boldsymbol{x},\boldsymbol{\lambda},\alpha) \propto f(\boldsymbol{x}|\beta,\boldsymbol{\lambda})\pi(\beta,\alpha) \propto \beta^{-n}\exp\left\{-\frac{1}{\beta}\sum_{i=1}^{n}\lambda_i x_i\right\}\pi(\beta,\alpha). \tag{3}$$

4 無情報事前分布

4.1 2 種類の無情報事前分布

Jeffreys の事前分布は以下のように定義される.

$$\pi_J(\beta,\alpha) \propto |I(\beta,\alpha)|^{1/2}$$

ただし

$$I(\beta,\alpha) = n\begin{bmatrix} \dfrac{\alpha}{\beta^2(\alpha+2)} & -\dfrac{1}{\beta(\alpha+1)} \\ -\dfrac{1}{\beta(\alpha+1)} & \dfrac{1}{\alpha^2} \end{bmatrix}.$$

よって

$$\pi_J(\beta,\alpha) \propto \frac{1}{\beta(\alpha+1)\alpha^{1/2}(\alpha+2)^{1/2}} \quad (\beta,\alpha>0). \tag{4}$$

パラメータ間の事前独立性を仮定した場合 Jeffreys の事前分布は

$$\pi_{IJ}(\beta,\alpha) \propto \pi(\beta)\pi(\alpha) = \frac{1}{\beta\alpha} \quad (\beta,\alpha>0). \tag{5}$$

4.2 Jeffreys の事前分布を用いた場合

定理 1. (4) の Jeffreys の事前分布のもとで, 事後分布は proper となる.

Proof. Jeffreys の事前分布のもとで β,α の事後分布は

$$\pi(\beta,\alpha|\boldsymbol{x}) \propto \frac{\alpha^{n-1/2}\beta^{-(n+1)}}{(\alpha+1)(\alpha+2)^{1/2}} \prod_{i=1}^{n}\left(1+\frac{x_i}{\beta}\right)^{-(\alpha+1)}.$$

この式の積分が有限であることを示す.

まず

$$\int_0^\infty \beta^{-(n+1)} \prod_{i=1}^{n}\left(1+\frac{x_i}{\beta}\right)^{-(\alpha+1)} d\beta$$

を考える. $y=\min\{x_1,\ldots,x_n\}$ とすると

$$\left(1+\frac{x_i}{\beta}\right)^{\alpha+1} \geq \left(1+\frac{y}{\beta}\right)^{\alpha+1}.$$

$\alpha\geq 0,\ i=1,\ldots,n$ に対し

$$\prod_{i=1}^{n}\left(1+\frac{x_i}{\beta}\right)^{-(\alpha+1)} \leq \left(1+\frac{y}{\beta}\right)^{-n(\alpha+1)}.$$

よって

$$\int_0^\infty \beta^{-(n+1)} \prod_{i=1}^{n}\left(1+\frac{x_i}{\beta}\right)^{-(\alpha+1)} d\beta \leq \int_0^\infty \beta^{-(n+1)}\left(1+\frac{y}{\beta}\right)^{-n(\alpha+1)} d\beta.$$

変数変換 $u=\dfrac{y}{\beta},\ du=-\dfrac{y}{\beta^2}d\beta$ により

$$\begin{aligned}
&\int_0^\infty \beta^{-(n+1)}\left(1+\frac{y}{\beta}\right)^{-n(\alpha+1)} d\beta \\
=\ & \frac{1}{y^n}\int_0^\infty \frac{u^{n-1}}{(1+u)^{n\alpha+n}} du \\
=\ & \frac{1}{y^n} B(n,n\alpha) \\
=\ & \frac{1}{y^n}\frac{\Gamma(n)\Gamma(n\alpha)}{\Gamma(n\alpha+n)} \\
=\ & \frac{(n-1)!}{y^n}\frac{1}{\prod_{j=0}^{n-1}(n\alpha+j)}.
\end{aligned}$$

よって

$$\begin{aligned}
&\int_0^\infty \int_0^\infty \frac{\alpha^{n-1/2}\beta^{-(n+1)}}{(\alpha+1)(\alpha+2)^{1/2}} \prod_{i=1}^n \left(1+\frac{x_i}{\beta}\right)^{-(\alpha+1)} d\beta d\alpha \\
\leq\ & \frac{(n-1)!}{y^n} \int_0^\infty \frac{\alpha^{n-1/2}}{(\alpha+1)(\alpha+2)^{1/2}} \frac{1}{\prod_{j=0}^{n-1}(n\alpha+j)} d\alpha \\
=\ & \frac{(n-1)!}{y^n} \left(\int_0^1 f(\alpha)d\alpha + \int_1^\infty f(\alpha)d\alpha \right),
\end{aligned}$$

ただし

$$f(\alpha) = \frac{\alpha^{n-1/2}}{(\alpha+1)(\alpha+2)^{1/2}} \frac{1}{\prod_{j=0}^{n-1}(n\alpha+j)}.$$

ここで $j \geq 1,\ \alpha > 0$ に対し $(n\alpha+j) \geq 1$ なので

$$\frac{1}{\prod_{j=0}^{n-1}(n\alpha+j)} \leq \frac{1}{n\alpha}.$$

また $\dfrac{1}{\alpha+1} \leq 1,\ \dfrac{1}{(\alpha+2)^{1/2}} < \dfrac{1}{\sqrt{2}}$ より

$$\begin{aligned}
\int_0^1 f(\alpha)d\alpha &\leq \frac{1}{n\sqrt{2}} \int_0^1 \alpha^{n-3/2} d\alpha \\
&= \frac{1}{n(n-1/2)\sqrt{2}}.
\end{aligned}$$

一方 $(n\alpha+j) \geq n\alpha$ なので

$$\frac{1}{\prod_{j=1}^{n-1}(n\alpha+j)} \leq \frac{1}{n^n}\frac{1}{\alpha^n}.$$

また $\dfrac{1}{\alpha+1} \leq \dfrac{1}{\alpha},\ \dfrac{1}{(\alpha+2)^{1/2}} \leq \dfrac{1}{\alpha^{1/2}}$ より

$$\begin{aligned}
\int_1^\infty f(\alpha)d\alpha &\leq \frac{1}{n^n} \int_1^\infty \alpha^{-2} d\alpha \\
&= \frac{1}{n^n}.
\end{aligned}$$

以上より

$$\begin{aligned}
&\frac{(n-1)!}{y^n} \left(\int_0^1 f(\alpha)d\alpha + \int_1^\infty f(\alpha)d\alpha \right) \\
\leq\ & \frac{(n-1)!}{y^n} \left(\frac{1}{n(n-1/2)\sqrt{2}} + \frac{1}{n^n} \right) \\
<\ & \infty.
\end{aligned}$$

従って Jeffreys の事前分布を用いると事後分布は proper となる. □

4.3 パラメータ間の事前独立性を仮定した Jeffreys の事前分布を用いた場合

定理 2. (5) のパラメータ間の事前独立性を仮定した Jeffreys の事前分布のもとで, 事後分布は improper となる.

Proof. (5) の事前分布のもとで, β, α の結合事後分布は

$$\pi(\beta, \alpha|\boldsymbol{x}) \propto \alpha^{n-1}\beta^{-(n+1)} \prod_{i=1}^{n} \left(1 + \frac{x_i}{\beta}\right)^{-(\alpha+1)}.$$

この式の積分が有限でないことを示す.

$w = \max\{x_1, \ldots, x_n\}$ とすると

$$\int_0^\infty \beta^{-(n+1)} \prod_{i=1}^{n} \left(1 + \frac{x_i}{\beta}\right)^{-n(\alpha+1)} d\beta \quad \geq \quad \int_0^\infty \beta^{-(n+1)} \left(1 + \frac{w}{\beta}\right)^{-n(\alpha+1)} d\beta.$$

変数変換 $v = \dfrac{w}{\beta},\ dv = -\dfrac{w}{\beta^2}d\beta$ により

$$\begin{aligned} & \int_0^\infty \beta^{-(n+1)} \left(1 + \frac{w}{\beta}\right)^{-n(\alpha+1)} d\beta \\ = \quad & \frac{1}{w^n} \int_0^\infty \frac{v^{n-1}}{(1+v)^{n\alpha+n}} dv \\ = \quad & \frac{1}{w^n} \frac{\Gamma(n)\Gamma(n\alpha)}{\Gamma(n\alpha+n)} \\ = \quad & \frac{(n-1)!}{w^n} \frac{1}{\prod_{j=0}^{n-1}(n\alpha+j)}. \end{aligned}$$

これを用いて

$$\begin{aligned} & \int_0^\infty \int_0^\infty \alpha^{n-1}\beta^{-(n+1)} \prod_{i=1}^{n} \left(1 + \frac{x_i}{\beta}\right)^{-(\alpha+1)} d\beta d\alpha \\ \geq \quad & \frac{(n-1)!}{w^n} \int_0^\infty \frac{\alpha^{n-1}}{\prod_{j=0}^{n-1}(n\alpha+j)} d\alpha \\ \geq \quad & \frac{(n-1)!}{w^n} \int_1^\infty \frac{\alpha^{n-1}}{\prod_{j=0}^{n-1}(n\alpha+j)} d\alpha. \end{aligned}$$

ここで, $\alpha \geq 1,\ j < n$ に対し $n\alpha + j < n\alpha + n = n(\alpha+1) \leq 2n\alpha$ となることから

$$\frac{1}{\prod_{j=0}^{n-1}(n\alpha+j)} \geq \frac{1}{(2n)^n} \frac{1}{\alpha^n}.$$

よって

$$\frac{(n-1)!}{w^n} \int_1^\infty \frac{\alpha^{n-1}}{\prod_{j=0}^{n-1}(n\alpha+j)} d\alpha \geq \frac{(n-1)!}{(2wn)^n} \int_1^\infty \alpha^{-1} d\alpha = \infty.$$

従ってパラメータ間の事前独立性を仮定した Jeffreys の事前分布を用いると事後分布は improper となる. □

5　シミュレーション

5.1　方法

[1] を参考にシミュレーションを行う. 定理 1, 定理 2 より Lomax 分布において通常の Jeffreys の事前分布を用いた場合の事後分布は proper となり, パラメータ間の事前独立性を仮定した Jeffreys の事前分布を用いた場合の事後分布は improper となることがわかる. よって通常の Jeffreys の事前分布を用いてシミュレーションを行う.

α の階層的表現 (2) に Jeffreys の事前分布 (4) を代入し, 整理すると以下のようになる.

$$f(\alpha|\boldsymbol{x},\boldsymbol{\lambda},\beta) \propto \frac{1}{(\alpha+1)\alpha^{1/2}(\alpha+2)^{1/2}[\Gamma(\alpha)]^n}\left(\prod_{i=1}^{n}\lambda_i\right)^{\alpha-1}.$$

この分布に従う乱数はメトロポリス法によって生成し, 事後期待値を計算する. β の階層的表現 (3) にも同様に Jeffreys の事前分布 (4) を代入すると以下のようになる.

$$f(\beta|\boldsymbol{x},\boldsymbol{\lambda},\alpha) \propto \beta^{-(n+1)}\exp\left\{-\frac{1}{\beta}\sum_{i=1}^{n}\lambda_i x_i\right\},$$

つまり,

$$\beta|\boldsymbol{x},\boldsymbol{\lambda},\alpha \sim \mathrm{IG}\left(n,\sum_{i=1}^{n}\lambda_i x_i\right).$$

これらを用いてパラメータの事後期待値を計算して推定の精度を確かめる. 推定精度の評価には平均相対誤差 (MRE) と平均二乗誤差 (MSE) を用いる.

$$\mathrm{MRE}=\frac{1}{N}\sum_{j=1}^{N}\frac{\hat{\theta}^{(j)}}{\theta},\quad \mathrm{MSE}=\frac{1}{N}\sum_{j=1}^{N}(\hat{\theta}^{(j)}-\theta)^2.$$

真のパラメータを $\alpha=1.5,\beta=2$ としサイズ $n=(100,110,\ldots,250)$ のサンプルをそれぞれ 1000 個発生させてシミュレーションを行う. MCMC サンプルの自己相関関数は図 1 のようになった. これを考慮し, 間引き数 15 として MCMC サンプルを間引く. 最初に発生させる MCMC サンプルは 15500 個, バーンイン期間を 500 とし, 最終的には 1000 個のサンプルから事後期待値を計算する.

5.2　結果

比較対象の最尤推定での結果も合わせて MRE, MSE の様子を調べた結果, 図 2 のようになった. 最尤推定よりもベイズ推定の方が精度が優れているという結果となった.

6　パラメータ λ_i ついて

6.1　5 章の問題点

5 章のシミュレーションには問題点がある. パラメータ λ_i $(i=1,2,\ldots,n)$ は 3 章にあるように Gamma$(\alpha,1)$ に従う確率変数であるが, α は未知であるためこの確率変数の実現値を得ることができない. そこで, 2 つの提案を考えシミュレーションを行った.

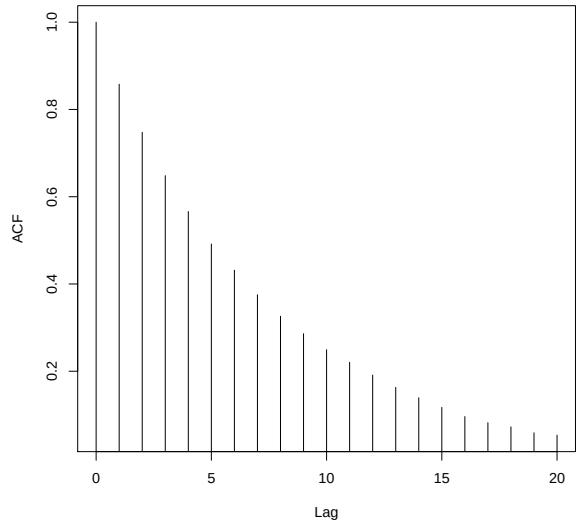

図 1: MCMC サンプルの自己相関関数

6.2 提案 1

まずは λ_i を推定に用いずにパラメータの事後期待値を求めることを考える.
Jeffreys の事前分布を用いたとき, β, α の同時事後分布は

$$\pi(\beta, \alpha|\boldsymbol{x}) \propto \frac{\alpha^{n-1/2}\beta^{-(n+1)}}{(\alpha+1)(\alpha+2)^{1/2}} \prod_{i=1}^{n} \left(1 + \frac{x_i}{\beta}\right)^{-(\alpha+1)}$$

であった. これを用いて 2 変数のメトロポリス法により事後分布を求める. それぞれの MCMC サンプルの自己相関関数は図 3 のようになった. これを考慮し, 間引き数を 30 として MCMC サンプルを間引く. これに伴って最初に発生させる MCMC サンプルの数を増やし 31000 個とする. バーンイン期間を 1000 とし, 最終的に 1000 個のサンプルで事後分布を計算する. 結果は図 4 のようになった. λ_i を用いずにメトロポリス法によって事後期待値を計算した場合, 最尤推定と比べて精度が低下した.

6.3 提案 2

次に, λ_i を推定し, その値を用いて β, α を推定する方法を考える.
サンプル $\boldsymbol{x}$ が与えられた時の λ_i の条件付き分布は

$$\lambda_i|x_i, \beta, \alpha \sim \text{Gamma}\left(\alpha + 1, 1 + \frac{x_i}{\beta}\right)$$

であった. これを最尤推定値 $\hat{\beta}$, $\hat{\alpha}$ を用いて

$$\lambda_i|x_i, \hat{\beta}, \hat{\alpha} \sim \text{Gamma}\left(\hat{\alpha} + 1, 1 + \frac{x_i}{\hat{\beta}}\right)$$

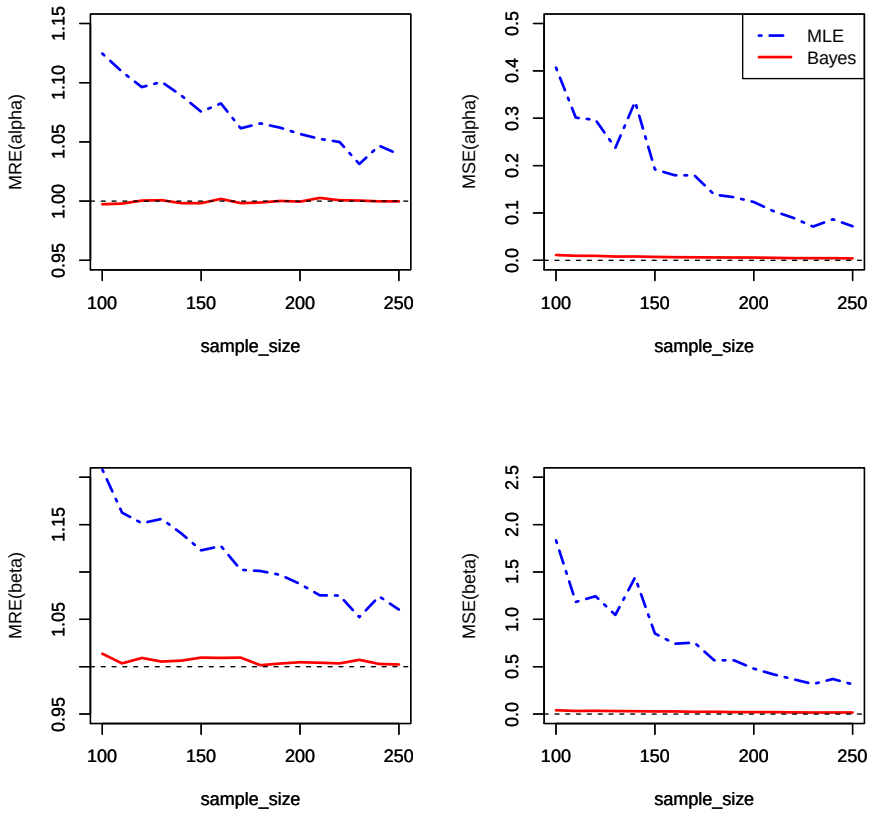

図 2: シミュレーションの結果

として各 λ_i を推定する. そこからさらに 5 章の方法により β, α を推定し, 最尤推定値と精度を比較する. MCMC サンプルの自己相関関数は図 5 のようになった. 自己相関関数は 5 章の方法と似た様子であるため, 同様に間引き数を 15 とし, 最初に発生させる MCMC サンプルの数を 15500 個, バーンイン期間を 500 とした. 結果は図 6 のようになった. こちらの場合も最尤推定よりも優れた精度にはならなかった.

7 まとめ

[1] のシミュレーションの再現ではベイズ推定の方が精度が良いことが確認できたが, パラメータ λ_i が真のパラメータ α によって定まる分布からの確率変数であり, 推定には用いることができないと考えられるため, 他の手法で推定を行うことができないかを検討した. 今回提案した二つの手法は λ_i を使わない方法と, λ_i を推定してから β, α を推定する方法であるため, 現実的に用いることは可能であるが, 推定の精度は最尤推定より劣っていて, 実用的な手法を得ることはできなかった. 今後は λ_i を別の方法で生成, または λ_i を使わない方法によってより精度の高い推定法を考えることが課題である.

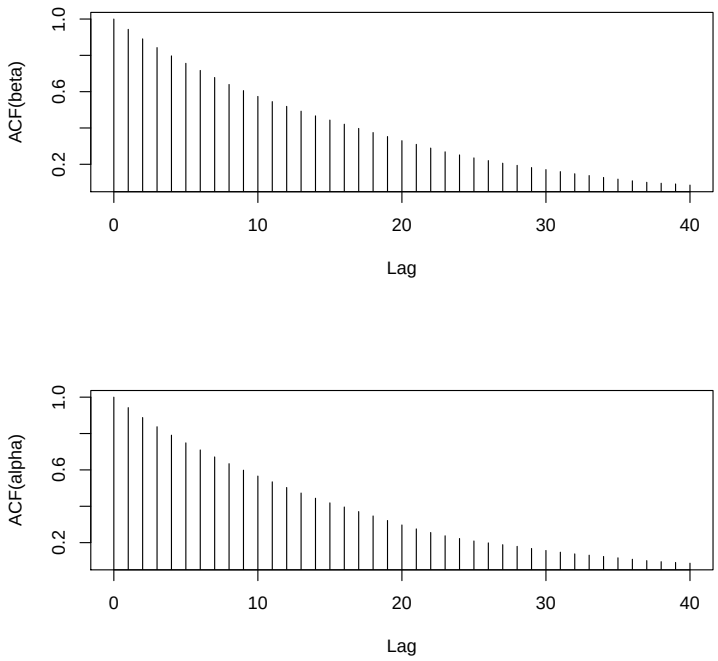

図 3: 提案 1 の MCMC サンプルの自己相関関数

8　シミュレーションに用いた R プログラム

5 章のプログラム

```
library("univariateML")
library("extraDistr")

true_beta = 2
true_alpha = 1.5

Metropolis <- function(current) {
  propose <- abs(rnorm(1,current,1))
  postOdds <- post_alpha(propose) / post_alpha(current)
  if(is.nan(postOdds)) current
  else if(min(postOdds,1) > runif(1,0,1)) propose else current
}

sample_size <- seq(100,250,by = 10)
```

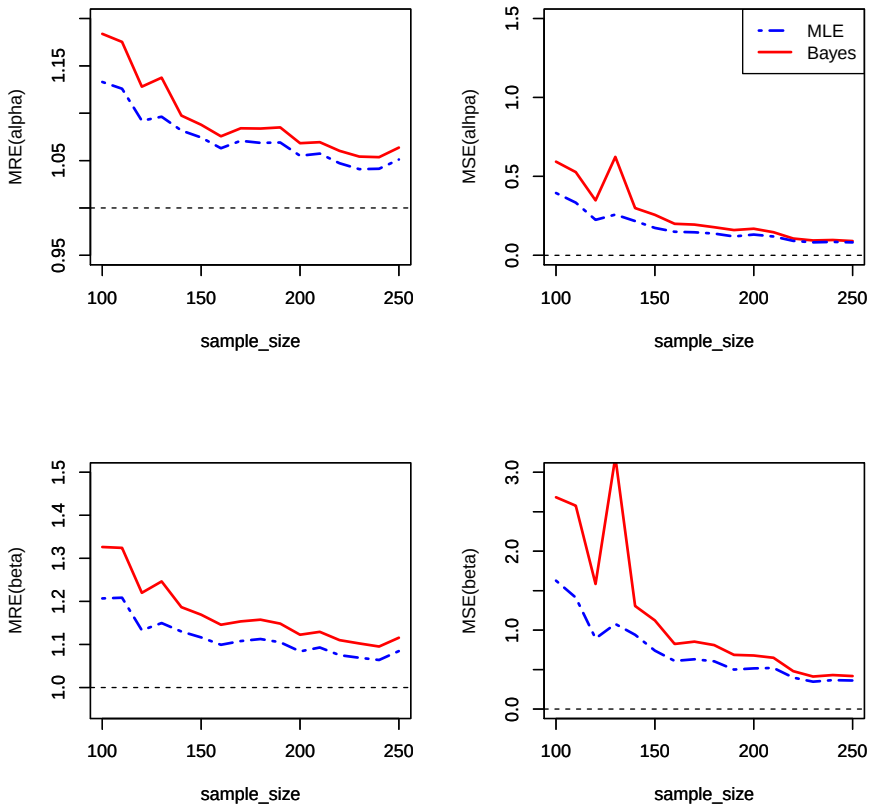

図 4: 提案 1 の推定の結果

```
bayes_mre_alpha_all <- c()
bayes_mse_alpha_all <- c()

bayes_mre_beta_all <- c()
bayes_mse_beta_all <- c()

MLE_mre_alpha_all <- c()
MLE_mse_alpha_all <- c()

MLE_mre_beta_all <- c()
MLE_mse_beta_all <- c()


for (n in sample_size){
```

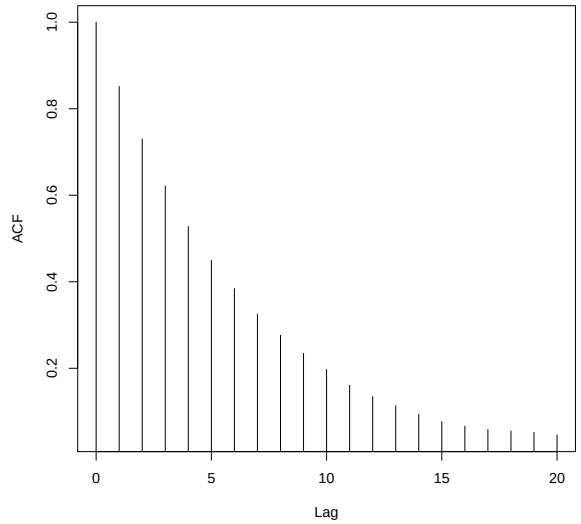

図 5: 提案 2 の MCMC サンプルの自己相関関数

```
N = 1000 #1 種類のサンプル数につき N 回実験して平均を取る

bayes_mres_alpha <- c()
bayes_mses_alpha <- c()

bayes_mres_beta <- c()
bayes_mses_beta <- c()

MLE_mres_alpha <- c()
MLE_mses_alpha <- c()

MLE_mres_beta <- c()
MLE_mses_beta <- c()

for (j in 1:N){
  x_total <- c()
  l_total <- c()

  #Lomax に従う確率変数を n 個生成
  for (k in 1:n){
    l <- rgamma(1,true_alpha,1)
    x <- rexp(1,l/true_beta)
```

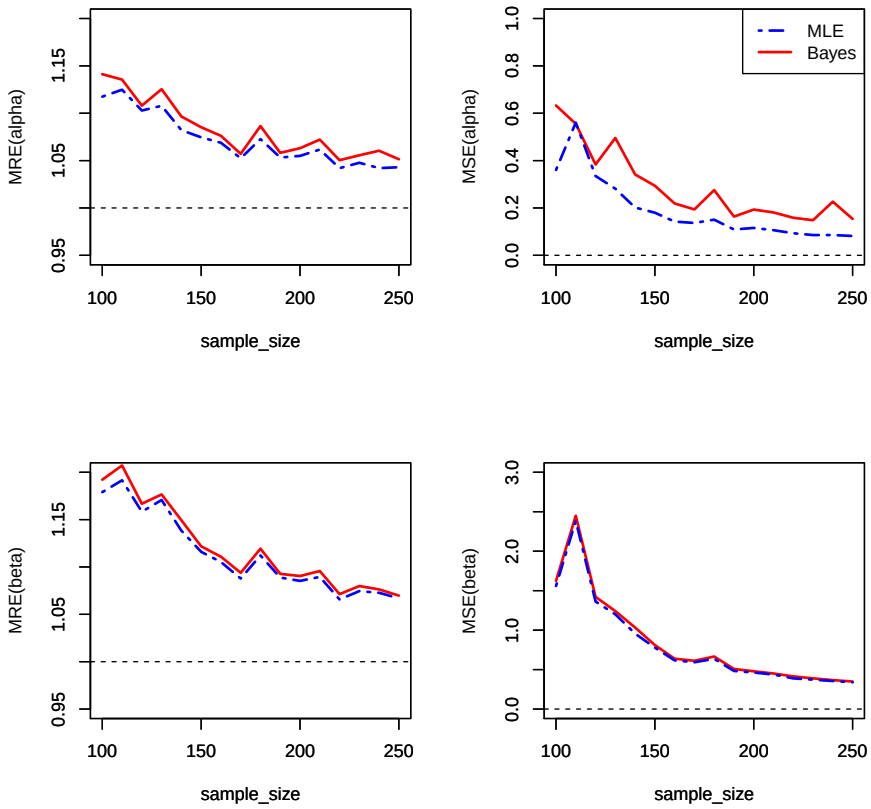

図 6: 提案 2 の推定の結果

```
  x_total <- append(x_total,x)
  l_total <- append(l_total,l)
}

post_alpha <-
  function(alpha)
    (1/((alpha + 1)*alpha^(1/2)*(alpha + 2)^(1/2)*(gamma(alpha))^n))*
    (prod(l_total))^(alpha - 1)


nsteps <- 15500 #mcmc サンプリングの数
alpha_mcmcsample <- rep(0,nsteps) #サンプルを格納するベクトル
alpha_mcmcsample[1] <- 5 #初期値
burn_in <- 500
```

```
for (i in 1:nsteps){
  alpha_mcmcsample[i+1] <- Metropolis(alpha_mcmcsample[i])
}

#burn_in
alpha_mcmc_burned <- alpha_mcmcsample[-(1:burn_in)]
length(alpha_mcmc_burned)

#thin
alpha_mcmc_thinned <- c()
thin = 15
for (i in 1:(nsteps-burn_in)){
  if(i%%thin == 0){
    alpha_mcmc_thinned <- append(alpha_mcmc_thinned, alpha_mcmc_burned[i])
  }
}


bayes_alpha_estimated <- mean(alpha_mcmc_thinned)
bayes_beta_estimated <- sum(x_total*l_total)/(n-1)
MLE_beta <- 1/mllomax(x_total)[1]
MLE_alpha <- mllomax(x_total)[2]


bayes_mres_alpha <-
append(bayes_mres_alpha, bayes_alpha_estimated*(1/true_alpha))
bayes_mses_alpha <-
append(bayes_mses_alpha, (bayes_alpha_estimated - true_alpha)^2)

bayes_mres_beta <-
append(bayes_mres_beta, bayes_beta_estimated*(1/true_beta))
bayes_mses_beta <-
append(bayes_mses_beta, (bayes_beta_estimated - true_beta)^2)

MLE_mres_alpha <-
append(MLE_mres_alpha, MLE_alpha*(1/true_alpha))
MLE_mses_alpha <-
append(MLE_mses_alpha, (MLE_alpha - true_alpha)^2)

MLE_mres_beta <-
```

```
    append(MLE_mres_beta, MLE_beta*(1/true_beta))
    MLE_mses_beta <-
    append(MLE_mses_beta, (MLE_beta - true_beta)^2)

  }
  bayes_mre_alpha_all <-
  append(bayes_mre_alpha_all, (1/N)*sum(bayes_mres_alpha))
  bayes_mse_alpha_all <-
  append(bayes_mse_alpha_all, (1/N)*sum(bayes_mses_alpha))

  bayes_mre_beta_all <-
  append(bayes_mre_beta_all, (1/N)*sum(bayes_mres_beta))
  bayes_mse_beta_all <-
  append(bayes_mse_beta_all, (1/N)*sum(bayes_mses_beta))

  MLE_mre_alpha_all <-
  append(MLE_mre_alpha_all, (1/N)*sum(MLE_mres_alpha))
  MLE_mse_alpha_all <-
  append(MLE_mse_alpha_all, (1/N)*sum(MLE_mses_alpha))

  MLE_mre_beta_all <-
  append(MLE_mre_beta_all,(1/N)*sum(MLE_mres_beta))
  MLE_mse_beta_all <-
  append(MLE_mse_beta_all,(1/N)*sum(MLE_mses_beta))
}

pdf("result.pdf")
par(mfrow=c(2,2))
plot(sample_size, bayes_mre_alpha_all, ylim = c(0.95, 1.15), ylab = "",
type="l", lwd = 2, col = "red")
par(new=T)
plot(sample_size, MLE_mre_alpha_all, ylim = c(0.95, 1.15), ylab = "MRE(alpha)",
lty=6, type="l", lwd = 2, col = "blue")
abline(h=1,lty=2)

plot(sample_size, bayes_mse_alpha_all, ylim = c(0.0, 0.5), ylab = "",
type="l", lwd = 2, col = "red")
par(new=T)
plot(sample_size, MLE_mse_alpha_all, ylim = c(0.0, 0.5), ylab = "MSE(alpha)",
lty=6, type="l", lwd = 2, col = "blue")
abline(h=0,lty=2)
```

```
legend("topright", legend=c("MLE","Bayes"),
lty=c(6,1),lwd = 2, col=c("blue","red"))

plot(sample_size, bayes_mre_beta_all, ylim = c(0.95, 1.2),  ylab = "",
type="l", lwd = 2, col = "red")
par(new=T)
plot(sample_size, MLE_mre_beta_all, ylim = c(0.95, 1.2),  ylab = "MRE(beta)",
lty=6, type="l", lwd = 2, col = "blue")
abline(h=1,lty=2)

plot(sample_size, bayes_mse_beta_all, ylim = c(0.0, 2.5), ylab = "",
type="l", lwd = 2, col = "red")
par(new=T)
plot(sample_size, MLE_mse_beta_all, ylim = c(0.0, 2.5),  ylab = "MSE(beta)",
lty=6, type="l", lwd = 2, col = "blue")
abline(h=0,lty=2)
dev.off()
```

6.2 のプログラム

```
library("univariateML")
library("extraDistr")

true_beta = 2
true_alpha = 1.5

sample_size <- seq(100,250,by = 10)

bayes_mre_alpha_all <- c()
bayes_mse_alpha_all <- c()

bayes_mre_beta_all <- c()
bayes_mse_beta_all <- c()

MLE_mre_alpha_all <- c()
MLE_mse_alpha_all <- c()

MLE_mre_beta_all <- c()
MLE_mse_beta_all <- c()
```

```
for (n in sample_size){
  #1 種類のサンプル数につき N 回実験して平均を取る
  N = 1000
  bayes_mres_alpha <- c()
  bayes_mses_alpha <- c()

  bayes_mres_beta <- c()
  bayes_mses_beta <- c()

  MLE_mres_alpha <- c()
  MLE_mses_alpha <- c()

  MLE_mres_beta <- c()
  MLE_mses_beta <- c()

  for (j in 1:N){
    x_total <- c()

    #Lomax に従う確率変数を n 個生成
    for (k in 1:n){
      l <- rgamma(1,true_alpha,1)
      x <- rexp(1,l/true_beta)
      x_total <- append(x_total,x)
    }

    post <- #事後分布
      function(beta, alpha)
        ((alpha^(n-1/2))*beta^(-n-1))/((alpha + 1)*(alpha + 2)^(1/2))*
        (prod(1+x_total/beta))^(-(alpha + 1))

    nsteps <- 31000 #mcmc サンプリングの数
    beta_mcmc <- rep(0,nsteps)
    alpha_mcmc <- rep(0,nsteps)
    beta_mcmc[1] <- 5
    alpha_mcmc[1] <- 5
    burn_in <- 1000


    for (i in 1:(nsteps-1)){
      propose_beta <- abs(rnorm(1,beta_mcmc[i],1))
```

```
  propose_alpha <- abs(rnorm(1,alpha_mcmc[i],1))
  postOdds <-
  post(propose_beta, propose_alpha) / post(beta_mcmc[i], alpha_mcmc[i])
  if(is.nan(postOdds)){
    beta_mcmc[i+1] <- beta_mcmc[i]
    alpha_mcmc[i+1] <- alpha_mcmc[i]
  }
  else if(min(postOdds,1) > runif(1,0,1)){
    beta_mcmc[i+1] <- propose_beta
    alpha_mcmc[i+1] <- propose_alpha
  }else {
    beta_mcmc[i+1] <- beta_mcmc[i]
    alpha_mcmc[i+1] <- alpha_mcmc[i]
  }
}


#burn-in
burned_beta <- beta_mcmc[-(1:burn_in)]
burned_alpha <- alpha_mcmc[-(1:burn_in)]
length(burned_beta)
length(burned_alpha)

#thin
thinned_beta <- c()
thinned_alpha <- c()

for (i in 1:(nsteps-burn_in)){
  if(i%%10 == 0){
    thinned_beta <- append(thinned_beta, burned_beta[i])
    thinned_alpha <- append(thinned_alpha, burned_alpha[i])
  }
}

bayes_beta_estimated <- mean(thinned_beta)
bayes_alpha_estimated <- mean(thinned_alpha)
MLE_beta <- 1/mllomax(x_total)[1]
MLE_alpha <- mllomax(x_total)[2]

bayes_mres_alpha <-
append(bayes_mres_alpha, bayes_alpha_estimated*(1/true_alpha))
```

```
    bayes_mses_alpha <-
    append(bayes_mses_alpha, (bayes_alpha_estimated - true_alpha)^2)

    bayes_mres_beta <-
    append(bayes_mres_beta, bayes_beta_estimated*(1/true_beta))
    bayes_mses_beta <-
    append(bayes_mses_beta, (bayes_beta_estimated - true_beta)^2)

    MLE_mres_alpha <-
    append(MLE_mres_alpha, MLE_alpha*(1/true_alpha))
    MLE_mses_alpha <-
    append(MLE_mses_alpha, (MLE_alpha - true_alpha)^2)

    MLE_mres_beta <-
    append(MLE_mres_beta, MLE_beta*(1/true_beta))
    MLE_mses_beta <-
    append(MLE_mses_beta, (MLE_beta - true_beta)^2)

  }
  bayes_mre_alpha_all <-
  append(bayes_mre_alpha_all, (1/N)*sum(bayes_mres_alpha))
  bayes_mse_alpha_all <-
  append(bayes_mse_alpha_all, (1/N)*sum(bayes_mses_alpha))

  bayes_mre_beta_all <-
  append(bayes_mre_beta_all, (1/N)*sum(bayes_mres_beta))
  bayes_mse_beta_all <-
  append(bayes_mse_beta_all, (1/N)*sum(bayes_mses_beta))

  MLE_mre_alpha_all <-
  append(MLE_mre_alpha_all, (1/N)*sum(MLE_mres_alpha))
  MLE_mse_alpha_all <-
  append(MLE_mse_alpha_all, (1/N)*sum(MLE_mses_alpha))

  MLE_mre_beta_all <-
  append(MLE_mre_beta_all,(1/N)*sum(MLE_mres_beta))
  MLE_mse_beta_all <-
  append(MLE_mse_beta_all,(1/N)*sum(MLE_mses_beta))
}

pdf("notlambda.pdf")
```

```
par(mfrow=c(2,2))
plot(sample_size, bayes_mre_alpha_all, ylim = c(0.95, 1.2), ylab = "",
type="l", lwd = 2, col = "red")
par(new=T)
plot(sample_size, MLE_mre_alpha_all, ylim = c(0.95, 1.2), ylab = "MRE(alpha)",
lty=6, type="l", lwd = 2, col = "blue")
abline(h=1, lty=2)

plot(sample_size, bayes_mse_alpha_all,ylim = c(0, 1.5), ylab = "",
type="l", lwd = 2, col = "red")
par(new=T)
plot(sample_size, MLE_mse_alpha_all,ylim = c(0, 1.5), ylab = "MSE(alhpa)",
lty=6, type="l", lwd = 2, col = "blue")
abline(h=0, lty=2)
legend("topright", legend=c("MLE","Bayes"),
lty=c(6,1), lwd = 2, col=c("blue","red"))

plot(sample_size, bayes_mre_beta_all, ylim = c(0.95, 1.5), ylab = "",
type="l", lwd = 2, col = "red")
par(new=T)
plot(sample_size, MLE_mre_beta_all, ylim = c(0.95, 1.5), ylab = "MRE(beta)",
lty=6, type="l", lwd = 2, col = "blue")
abline(h=1, lty=2)

plot(sample_size, bayes_mse_beta_all, ylim = c(0, 3.0), ylab = "",
type="l", lwd = 2, col = "red")
par(new=T)
plot(sample_size, MLE_mse_beta_all, ylim = c(0, 3.0), ylab = "MSE(beta)",
lty=6, type="l", lwd = 2, col = "blue")
abline(h=0, lty=2)
dev.off()
```

6.3 のプログラム

```
library("univariateML")
library("extraDistr")

true_beta = 2
true_alpha = 1.5

Metropolis <- function(current) {
  propose <- abs(rnorm(1,current,1))
```

```
  postOdds <- post_alpha(propose) / post_alpha(current)
  if(is.nan(postOdds)) current
  else if(min(postOdds,1) > runif(1,0,1)) propose
  else current
}

sample_size <- seq(100,250,by = 10)

bayes_mre_alpha_all <- c()
bayes_mse_alpha_all <- c()

bayes_mre_beta_all <- c()
bayes_mse_beta_all <- c()

MLE_mre_alpha_all <- c()
MLE_mse_alpha_all <- c()

MLE_mre_beta_all <- c()
MLE_mse_beta_all <- c()



for (n in sample_size){
  #1 種類のサンプル数につき N 回実験して平均を取る
  N = 1000

  bayes_mres_alpha <- c()
  bayes_mses_alpha <- c()

  bayes_mres_beta <- c()
  bayes_mses_beta <- c()

  MLE_mres_alpha <- c()
  MLE_mses_alpha <- c()

  MLE_mres_beta <- c()
  MLE_mses_beta <- c()

  for (j in 1:N){
    x_total <- c()
    #l_total <- c()
```

```
lambdas <- c()

#Lomax に従う確率変数を n 個生成
for (k in 1:n){
  l <- rgamma(1,true_alpha,1)
  x <- rexp(1,l/true_beta)
  x_total <- append(x_total,x)
}

#最尤推定の結果
MLE_alpha <- mllomax(x_total)[2]
MLE_beta <- 1/mllomax(x_total)[1]

for (i in 1:n){
  lambdas[i] <- rgamma(1, MLE_alpha + 1, 1 + x_total[i]/MLE_beta)
}

#alpha の事後分布
post_alpha <-
  function(alpha)
    (1/((alpha+1)*alpha^(1/2)*(alpha+2)^(1/2)*(gamma(alpha))^n))*
    (prod(lambdas))^(alpha - 1)


nsteps <- 15500 #mcmc サンプリングの数
alpha_mcmcsample <- rep(0,nsteps) #サンプルを格納するベクトル
alpha_mcmcsample[1] <- 5 #初期値
burn_in <- 500

for (i in 1:nsteps){
  alpha_mcmcsample[i+1] <- Metropolis(alpha_mcmcsample[i])
}

#burn-in
alpha_mcmc_burned <- alpha_mcmcsample[-(1:burn_in)]
length(alpha_mcmc_burned)

#thin
alpha_mcmc_thinned <- c()
thin = 15
for (i in 1:(nsteps-burn_in)){
```

```
    if(i%%thin == 0){
      alpha_mcmc_thinned <- append(alpha_mcmc_thinned, alpha_mcmc_burned[i])
    }
  }

  #ベイズ推定値
  bayes_alpha_estimated <- mean(alpha_mcmc_thinned)
  bayes_beta_estimated <- sum(x_total*lambdas)/(n-1)


  bayes_mres_alpha <-
  append(bayes_mres_alpha, bayes_alpha_estimated*(1/true_alpha))
  bayes_mses_alpha <-
  append(bayes_mses_alpha, (bayes_alpha_estimated - true_alpha)^2)

  bayes_mres_beta <-
  append(bayes_mres_beta, bayes_beta_estimated*(1/true_beta))
  bayes_mses_beta <-
  append(bayes_mses_beta, (bayes_beta_estimated - true_beta)^2)

  MLE_mres_alpha <-
  append(MLE_mres_alpha, MLE_alpha*(1/true_alpha))
  MLE_mses_alpha <-
  append(MLE_mses_alpha, (MLE_alpha - true_alpha)^2)

  MLE_mres_beta <-
  append(MLE_mres_beta, MLE_beta*(1/true_beta))
  MLE_mses_beta <-
  append(MLE_mses_beta, (MLE_beta - true_beta)^2)

}

bayes_mre_alpha_all <-
append(bayes_mre_alpha_all, (1/N)*sum(bayes_mres_alpha))
bayes_mse_alpha_all <-
append(bayes_mse_alpha_all, (1/N)*sum(bayes_mses_alpha))

bayes_mre_beta_all <-
append(bayes_mre_beta_all, (1/N)*sum(bayes_mres_beta))
bayes_mse_beta_all <-
append(bayes_mse_beta_all, (1/N)*sum(bayes_mses_beta))
```

```
  MLE_mre_alpha_all <-
  append(MLE_mre_alpha_all, (1/N)*sum(MLE_mres_alpha))
  MLE_mse_alpha_all <-
  append(MLE_mse_alpha_all, (1/N)*sum(MLE_mses_alpha))

  MLE_mre_beta_all <-
  append(MLE_mre_beta_all,(1/N)*sum(MLE_mres_beta))
  MLE_mse_beta_all <-
  append(MLE_mse_beta_all,(1/N)*sum(MLE_mses_beta))
}

pdf("mle-bayes.pdf")
par(mfrow=c(2,2))
plot(sample_size, bayes_mre_alpha_all, ylim = c(0.95, 1.2), ylab = "",
type="l", lwd = 2, col = "red")
par(new=T)
plot(sample_size, MLE_mre_alpha_all, ylim = c(0.95, 1.2), ylab = "MRE(alpha)",
lty=6, type="l", lwd = 2, col = "blue")
abline(h=1,lty=2)

plot(sample_size, bayes_mse_alpha_all, ylim = c(0.0, 1.0), ylab = "",
type="l", lwd = 2, col = "red")
par(new=T)
plot(sample_size, MLE_mse_alpha_all, ylim = c(0.0, 1.0), ylab = "MSE(alpha)",
lty=6, type="l", lwd = 2, col = "blue")
abline(h=0,lty=2)
legend("topright", legend=c("MLE","Bayes"),
lty=c(6,1),  lwd = 2, col=c("blue","red"))

plot(sample_size, bayes_mre_beta_all, ylim = c(0.95, 1.2),  ylab = "",
type="l", lwd = 2, col = "red")
par(new=T)
plot(sample_size, MLE_mre_beta_all, ylim = c(0.95, 1.2),  ylab = "MRE(beta)",
lty=6, type="l", lwd = 2, col = "blue")
abline(h=1,lty=2)

plot(sample_size, bayes_mse_beta_all, ylim = c(0.0, 3.0), ylab = "",
type="l", lwd = 2, col = "red")
par(new=T)
plot(sample_size, MLE_mse_beta_all, ylim = c(0.0, 3.0),  ylab = "MSE(beta)",
```

```
lty=6, type="l", lwd = 2, col = "blue")
abline(h=0,lty=2)
dev.off()
```

参考文献

[1] Paulo H. Ferreira, Eduardo Ramos, Pedro L. Ramos, Jhon F.B. Gonzales, Vera L.D. Tomazella, Ricardo S. Ehlers, Eveliny B. Silva, and Francisco Louzada. Objective bayesian analysis for the lomax distribution. *Statistics and Probability Letters*, 2019.

[2] 涌井良幸. 道具としてのベイズ統計. 日本実業出版社, 2009.