

Cox 回帰における Bayes 推定と PHREG プロシジャ

黒田 晋吾

武田薬品工業株式会社 医薬開発本部 日本開発センター
クリニカルデータサイエンス部 統計グループ

Bayesian inference in the Cox's proportional hazards model and PHREG procedure

Shingo Kuroda

Takeda Pharmaceutical Company, Ltd.

概要 : SAS9.2 では, PHREG プロシジャにおいて BAYES ステートメントを記述することにより, Cox 回帰における回帰パラメータの Bayes 推定値を算出することが可能となった. 本稿では, SAS9.2 で用いられている手法である, Cox 回帰における回帰パラメータの Bayes 推定値を算出する際に尤度の代わりに部分尤度を用いる手法について, その妥当性を証明する. また, Rejection Sampling と Gibbs Sampling を用いて Cox 回帰における回帰パラメータの事後分布から乱数生成を行い, SGPLOT プロシジャによる事後分布の密度推定, 及び平均値や中央値等の Bayes 推定値の算出が可能であることを示す.

キーワード : 部分尤度, Bayes の定理, Rejection Sampling, Gibbs Sampling, PHREG プロシジャ, BAYES ステートメント

1 はじめに

生存時間解析においてよく用いられる手法の一つである Cox 回帰では, 部分尤度を利用して回帰パラメータの推定値を求めることにより解析を行う. また Bayes 流の解析では, パラメータの事前分布と尤度の積が事後分布の定数倍となることを利用して解析を行う方法がある. SAS9.2 では, PHREG プロシジャにおいて BAYES ステートメントを記述することにより, Cox 回帰における回帰パラメータの Bayes 推定値を算出することが可能となったが, その際に SAS の内部では, 尤度ではなく部分尤度を用いることによって事後分布を算出し, この事後分布を基に Bayes 推定値を求めている. 本稿では, まず Cox 回帰において回帰パラメータの Bayes 推定値を算出する際に, 回帰パラメータの事前分布と部分尤度の積が回帰パラメータの事後分布の定数倍となることを示し, Bayes 流の解析においても部分尤度を用いることの妥当性を証明する. また, 回帰パラメータの事後分布は複雑な形をしており, 解析的に回帰パラメータの平均値や中央値等を求めることは難しいため, 次に Rejection Sampling と Gibbs Sampling について解説を行い, これらの Sampling 手法を用いて回帰パラメータの事後分布からの乱数を生成し, SGPLOT プロシジャによる事後分布の密度推定, 及び事後平均値, 事後中央値等の Bayes 推定値の算出が可能であることを示す.

2 定義及び仮定

T_1, T_2, \dots, T_n をあるイベント (例えば死亡) が発生するまでの観測時間を表す確率変数とし, $T_i (i = 1, \dots, n)$

はそれぞれ独立に同一の確率分布 $F(t; \theta)$ に従うとする。このとき、

$$\begin{aligned} h(t; \theta) &= \lim_{h \downarrow 0} \frac{P[t \leq T_i \leq t+h \mid T_i \geq t]}{h} \\ &= \frac{f(t; \theta)}{S(t; \theta)} \end{aligned}$$

を T_i のハザード関数という。ただし、 $f(t; \theta)$ は T_i の確率密度関数とし、 $S(t; \theta)$ は

$$S(t; \theta) = P(T_i \geq t)$$

で定義される T_i の生存関数とする。また、累積ハザード関数 $H(t)$ を

$$H(t) = \int_0^t h(t; \theta) dt$$

で定義する。

また、 C_1, C_2, \dots, C_n を打ち切りが発生するまでの観測時間を表す確率変数とし、 C_i ($i = 1, \dots, C_n$) は分布関数 $G(t; \phi)$ 、確率密度関数 $g(t; \phi)$ をもつ同一の分布にそれぞれ独立に従うとする。さらに、

$$U_i = \min(T_i, C_i) \quad (i = 1, \dots, n)$$

とし、 $u_i = t_i$ であれば $\delta_i = 1$ 、 $u_i = c_i$ であれば $\delta_i = 0$ として、 δ_i ($i = 1, \dots, n$) を定義する。ただし、 t_i, c_i, u_i はそれぞれ確率変数 T_i, C_i, U_i の実現値とする。

観測時間及びイベント発生（打ち切り）の有無のデータ (u_i, δ_i) ($i = 1, \dots, n$) が得られたとき、 $\mathbf{u}' = (u_1, \dots, u_n)$ 、 $\boldsymbol{\delta}' = (\delta_1, \dots, \delta_n)$ として、ランダム打ち切りの仮定の下、すなわち θ と ϕ が無関係であり、さらに各 $i = 1, \dots, n$ について、 T_i と C_i が独立であるという仮定の下で、尤度関数 $L(\theta, \phi \mid \mathbf{u}, \boldsymbol{\delta})$ を計算すると、

$$\begin{aligned} L(\theta, \phi \mid \mathbf{u}, \boldsymbol{\delta}) &= \prod_{i=1}^n \{f(u_i; \theta)P(\delta_i = 1 \mid u_i, \phi)\}^{\delta_i} \{g(u_i; \phi)P(\delta_i = 0 \mid u_i, \phi)\}^{1-\delta_i} \\ &= \prod_{i=1}^n \{f(u_i; \theta)(1 - G(u_i, \phi))\}^{\delta_i} \{g(u_i; \phi)S(u_i; \theta)\}^{1-\delta_i} \\ &= \prod_{i=1}^n f(u_i; \theta)^{\delta_i} S(u_i; \theta)^{1-\delta_i} \prod_{i=1}^n (1 - G(u_i, \phi))^{\delta_i} g(u_i; \phi)^{1-\delta_i} \\ &= L(\theta \mid \mathbf{u}, \boldsymbol{\delta})L(\phi \mid \mathbf{u}, \boldsymbol{\delta}) \\ &\propto L(\theta \mid \mathbf{u}, \boldsymbol{\delta}) \end{aligned}$$

となる。ただし、

$$\begin{aligned} L(\theta \mid \mathbf{u}, \boldsymbol{\delta}) &= \prod_{i=1}^n f(u_i; \theta)^{\delta_i} S(u_i; \theta)^{1-\delta_i} \\ L(\phi \mid \mathbf{u}, \boldsymbol{\delta}) &= \prod_{i=1}^n (1 - G(u_i, \phi))^{\delta_i} g(u_i; \phi)^{1-\delta_i} \end{aligned}$$

とする。以上から、ランダム打ち切りの仮定の下でパラメータ θ を推定する場合、 $L(\theta \mid \mathbf{u}, \boldsymbol{\delta})$ を最大にする θ を見つけることにより θ の最尤推定量が得られる。以下、本稿ではランダム打ち切りを仮定する。

3 Cox の比例ハザードモデル

ハザード関数 $h(t; \mathbf{x}, \boldsymbol{\beta})$ に比例ハザード性が成り立っているとする。すなわち, $h(t; \mathbf{x}, \boldsymbol{\beta})$ に対し,

$$h(t; \mathbf{x}, \boldsymbol{\beta}) = h_0(t)h(\mathbf{x}, \boldsymbol{\beta})$$

が成り立っているとする。ただし, \mathbf{x} を共変量, $\boldsymbol{\beta}$ をパラメータとし, $h(\mathbf{x}, \boldsymbol{\beta})$ は t によらない関数で, $h_0(t)$ は $\mathbf{x}, \boldsymbol{\beta}$ によらない関数とする。さらに特別な場合として, 比例ハザード性の仮定の下で, $\mathbf{x}' = (x_1, \dots, x_p)$, $\boldsymbol{\beta}' = (\beta_1, \dots, \beta_p)$ とし,

$$h(\mathbf{x}, \boldsymbol{\beta}) = \exp(\mathbf{x}'\boldsymbol{\beta})$$

としたとき, すなわち,

$$\begin{aligned} h(t; \mathbf{x}, \boldsymbol{\beta}) &= h_0(t) \exp(\mathbf{x}'\boldsymbol{\beta}) \\ &= h_0(t) \exp(\beta_1 x_1 + \dots + \beta_p x_p) \end{aligned}$$

というモデルを考えたとき, これを Cox の比例ハザードモデルという。

次に, Cox の比例ハザードモデルの下でデータ $(u_i, \delta_i, \mathbf{x}_i)$ ($i = 1, \dots, n$) が得られたときの尤度関数を考える。 $\mathbf{u}' = (u_1, \dots, u_n)$, $\boldsymbol{\delta}' = (\delta_1, \dots, \delta_n)$, $X = (\mathbf{x}_1, \dots, \mathbf{x}_n)'$, $\mathbf{x}'_i = (x_{i1}, \dots, x_{ip})$ とし,

$$S(t; \mathbf{x}, \boldsymbol{\beta}) = \exp \left[- \int_0^t h(u; \mathbf{x}, \boldsymbol{\beta}) du \right]$$

であることに注意して,

$$\begin{aligned} L(\boldsymbol{\beta}, h_0(t) \mid \mathbf{u}, \boldsymbol{\delta}, X) &= \prod_{i=1}^n f(u_i; \boldsymbol{\beta}, \mathbf{x}_i)^{\delta_i} S(u_i; \boldsymbol{\beta}, \mathbf{x}_i)^{1-\delta_i} \\ &= \prod_{i=1}^n h(u_i; \boldsymbol{\beta}, \mathbf{x}_i)^{\delta_i} S(u_i; \boldsymbol{\beta}, \mathbf{x}_i) \\ &= \prod_{i=1}^n [h_0(u_i) \exp(\mathbf{x}'_i \boldsymbol{\beta})]^{\delta_i} \left[\exp \left\{ - \int_0^{u_i} h_0(t) \exp(\mathbf{x}'_i \boldsymbol{\beta}) dt \right\} \right] \\ &= \prod_{i=1}^n [h_0(u_i) \exp(\mathbf{x}'_i \boldsymbol{\beta})]^{\delta_i} \prod_{i=1}^n \exp \left[- \int_0^{\infty} Y_i(t) h_0(t) \exp(\mathbf{x}'_i \boldsymbol{\beta}) dt \right] \\ &= \prod_{i=1}^n \left[\frac{\exp(\mathbf{x}'_i \boldsymbol{\beta})}{B(u_i)} \right]^{\delta_i} \prod_{i=1}^n [h_0(u_i) B(u_i)]^{\delta_i} \exp \left[- \int_0^{\infty} B(t) h_0(t) dt \right] \\ &= \tilde{L}_1(\boldsymbol{\beta} \mid \mathbf{u}, \boldsymbol{\delta}, X) \tilde{L}_2(\boldsymbol{\beta}, h_0(t) \mid \mathbf{u}, \boldsymbol{\delta}, X) \end{aligned}$$

となる。ただし,

$$\begin{aligned} \tilde{L}_1(\boldsymbol{\beta} \mid \mathbf{u}, \boldsymbol{\delta}, X) &= \prod_{i=1}^n \left[\frac{\exp(\mathbf{x}'_i \boldsymbol{\beta})}{B(u_i)} \right]^{\delta_i} \\ \tilde{L}_2(\boldsymbol{\beta}, h_0(t) \mid \mathbf{u}, \boldsymbol{\delta}, X) &= \prod_{i=1}^n [h_0(u_i) B(u_i)]^{\delta_i} \exp \left[- \int_0^{\infty} B(t) h_0(t) dt \right] \\ B(u) &= \sum_{l=1}^n Y_l(u) \exp(\mathbf{x}'_l \boldsymbol{\beta}) \end{aligned}$$

$$Y_i(t) = \begin{cases} 1 & (t \leq u_i), \\ 0 & (t > u_i) \end{cases}$$

とする。ここで、 $\tilde{L}_1(\beta | \mathbf{u}, \delta, X)$ は部分尤度と呼ばれ、

$$\begin{aligned} \tilde{L}_1(\beta | \mathbf{u}, \delta, X) &= \prod_{i=1}^n \left[\frac{\exp(\mathbf{x}_i' \beta)}{B(u_i)} \right]^{\delta_i} \\ &= \prod_{i=1}^n \left[\frac{\exp(\mathbf{x}_i' \beta)}{\sum_{l=1}^n Y_l(u_i) \exp(\mathbf{x}_l' \beta)} \right]^{\delta_i} \\ &= \prod_{i=1}^n \left[\frac{\exp(\mathbf{x}_i' \beta)}{\sum_{l \in R(u_i)} \exp(\mathbf{x}_l' \beta)} \right]^{\delta_i} \end{aligned}$$

と書くことができる $h_0(t)$ によらない β のみの関数である。ただし、 $R(u_i)$ は時間 u_i におけるリスク集合、すなわち、

$$R(u_i) = \{l \mid u_l \geq u_i\}$$

である。通常 $\tilde{L}_1(\beta | \mathbf{u}, \delta, X)$ を最大にする $\beta = \hat{\beta}$ を Newton-Raphson 法等を用いて求め、 $\hat{\beta}$ が最尤推定量と同等の性質を持つとして、Cox の比例ハザードモデルにおける β の推定及び検定を行う。

例 1 観察時間 u_i 、打ち切りの指標 δ_i 、及び共変量 x_i ($i = 1, \dots, 5$) をそれぞれ、 $(11, 1, 0)$, $(12, 1, 1)$, $(13, 0, 0)$, $(16, 1, 1)$, $(21, 1, 1)$ としたときの部分尤度を求める。このとき、各 u_i におけるリスク集合は

$$R(11) = \{1, 2, 3, 4, 5\}, R(12) = \{1, 2, 3, 5\}, R(16) = \{1, 3\}, R(21) = \{3\}$$

であるから、部分尤度 $\tilde{L}_1(\beta | \mathbf{u}, \delta, X)$ は、

$$\tilde{L}_1(\beta | \mathbf{u}, \delta, X) = \left(\frac{1}{3 \exp(\beta) + 2} \right) \left(\frac{\exp(\beta)}{3 \exp(\beta) + 1} \right) \left(\frac{1}{2} \right)$$

となり、部分尤度を最大にする β は

$$\hat{\beta} = \frac{1}{2} \log \left(\frac{2}{9} \right) \approx -0.752$$

となる。

β の最尤推定量を算出する際には、全尤度 $L(\beta, h_0(t) | \mathbf{u}, \delta, X)$ を最大にする β を求めることが必要とされる。上記では部分尤度 $\tilde{L}_1(\beta | \mathbf{u}, \delta, X)$ を最大にする $\beta = \hat{\beta}$ を最尤推定量としているが、 $\tilde{L}_2(\beta, h_0(t) | \mathbf{u}, \delta, X)$ にも β が含まれているため、実際は $\hat{\beta}$ が β の最尤推定量となることは明らかではない。しかし、次の定理により、部分尤度を用いる妥当性が示される。

定理 (Klein et al. (2003)) 部分尤度 $\tilde{L}_1(\beta | \mathbf{u}, \delta, X)$ を最大にする $\beta = \hat{\beta}$ は最尤推定量である。

証明 イベント発生時間を (t_1, \dots, t_D) 、打ち切り時間を (t_{D+1}, \dots, t_n) とし、全尤度

$$\begin{aligned} L(\beta, h_0(t) | \mathbf{u}, \delta, X) &= \prod_{i=1}^n [h_0(u_i) \exp(\mathbf{x}_i' \beta)]^{\delta_i} [\exp \{-H_0(u_i) \exp(\mathbf{x}_i' \beta)\}] \\ &= \prod_{i=1}^D [h_0(t_i) \exp(\mathbf{x}_i' \beta)] \prod_{i=1}^n [\exp \{-H_0(t_i) \exp(\mathbf{x}_i' \beta)\}] \end{aligned}$$

を β を固定して $h_0(t)$ のみの関数と見るとき, $t \notin (t_1, \dots, t_D)$ において $L(\beta, h_0(t) \mid \mathbf{u}, \delta, X)$ は $h_0(t) = 0$ で最大となる. よって, $t \in (t_1, \dots, t_D)$ に対し, $h_0(t_i) = h_{0i}$ ($i = 1, \dots, D$) とすれば, $H_0(t_j) = \sum_{t_i \leq t_j} h_{0i}$ と書けるから,

$$\begin{aligned} L(\beta, h_0(t) \mid \mathbf{u}, \delta, X) &= \prod_{i=1}^D [h_{0i} \exp(\mathbf{x}_i' \beta)] \prod_{j=1}^n \left[\exp \left\{ - \sum_{t_i \leq t_j} h_{0i} \exp(\mathbf{x}_j' \beta) \right\} \right] \\ &= \prod_{i=1}^D [h_{0i} \exp(\mathbf{x}_i' \beta)] \left[\exp \left\{ - \sum_{i=1}^D h_{0i} \sum_{j \in R(t_i)} \exp(\mathbf{x}_j' \beta) \right\} \right] \\ &= \prod_{i=1}^D [h_{0i} \exp(\mathbf{x}_i' \beta)] \left[\exp \left\{ -h_{0i} \sum_{j \in R(t_i)} \exp(\mathbf{x}_j' \beta) \right\} \right] \end{aligned}$$

となる.

$$\frac{\partial \log L(\beta, h_0(t) \mid \mathbf{u}, \delta, X)}{\partial h_{0i}} = \frac{1}{h_{0i}} - \sum_{j \in R(t_i)} \exp(\mathbf{x}_j' \beta) = 0$$

を解くと,

$$h_{0i} = \frac{1}{\sum_{j \in R(t_i)} \exp(\mathbf{x}_j' \beta)}$$

となるので, 上記の式を $L(\beta, \hat{h}_0(t) \mid \mathbf{u}, \delta, X)$ に代入すると,

$$\begin{aligned} L(\beta, \hat{h}_0(t) \mid \mathbf{u}, \delta, X) &= \prod_{i=1}^D [\hat{h}_{0i} \exp(\mathbf{x}_i' \beta)] \left[\exp \left\{ -\hat{h}_{0i} \sum_{j \in R(t_i)} \exp(\mathbf{x}_j' \beta) \right\} \right] \\ &= \prod_{i=1}^D \left[\frac{\exp(\mathbf{x}_i' \beta)}{\sum_{j \in R(t_i)} \exp(\mathbf{x}_j' \beta)} \right] \left[\exp \left\{ \frac{-1}{\sum_{j \in R(t_i)} \exp(\mathbf{x}_j' \beta)} \sum_{j \in R(t_i)} \exp(\mathbf{x}_j' \beta) \right\} \right] \\ &\propto \prod_{i=1}^D \frac{\exp(\mathbf{x}_i' \beta)}{\sum_{j \in R(t_i)} \exp(\mathbf{x}_j' \beta)} \end{aligned}$$

となるから, 部分尤度を最大にする $\beta = \hat{\beta}$ は β の最尤推定量である. また,

$$\hat{h}_{0i} = \frac{1}{\sum_{j \in R(t_i)} \exp(\mathbf{x}_j' \hat{\beta})}$$

となる.

4 Cox 回帰における Bayes 推定

Cox の比例ハザードモデル

$$\begin{aligned} h(t; \mathbf{x}, \beta) &= h_0(t) \exp(\mathbf{x}' \beta) \\ &= h_0(t) \exp(\beta_1 x_1 + \dots + \beta_p x_p) \end{aligned}$$

におけるパラメータ β をベイズ流のアプローチで推定することを考える. ただし, ここでは $h_0(t)$ の推定は行わないこととする. データ $(u_i, \delta_i, \mathbf{x}_i)$ ($i = 1, \dots, n$) が得られた時の Cox の比例ハザードモデルにおける

尤度は

$$L(\boldsymbol{\beta}, H_0(t) \mid \mathbf{u}, \boldsymbol{\delta}, X) = \prod_{i=1}^n [h_0(u_i) \exp(\mathbf{x}_i' \boldsymbol{\beta})]^{\delta_i} [\exp\{-H_0(u_i) \exp(\mathbf{x}_i' \boldsymbol{\beta})\}]$$

であるから, $\boldsymbol{\beta}, H_0(t)$ の事前分布 $\pi(\boldsymbol{\beta}, H_0(t))$ について

$$\pi(\boldsymbol{\beta}, H_0(t)) = \pi(\boldsymbol{\beta})\pi(H_0(t))$$

を仮定するとき, $\boldsymbol{\beta}, H_0(t)$ の事後分布は

$$p(H_0(t), \boldsymbol{\beta} \mid \mathbf{u}, \boldsymbol{\delta}, X) \propto L(\boldsymbol{\beta}, H_0(t) \mid \mathbf{u}, \boldsymbol{\delta}, X)\pi(\boldsymbol{\beta})\pi(H_0(t))$$

となる. よって, $\boldsymbol{\beta}$ の周辺事後分布は

$$p(\boldsymbol{\beta} \mid \mathbf{u}, \boldsymbol{\delta}, X) \propto \int p(H_0(t), \boldsymbol{\beta} \mid \mathbf{u}, \boldsymbol{\delta}, X) dH_0(t)$$

によって求めることができる. ここで, $H_0(t)$ の事前過程をガンマ過程とすると, ある条件の下で

$$\int p(H_0(t), \boldsymbol{\beta} \mid \mathbf{u}, \boldsymbol{\delta}, X) dH_0(t) = \tilde{L}_1(\boldsymbol{\beta} \mid \mathbf{u}, \boldsymbol{\delta}, X)\pi(\boldsymbol{\beta})$$

となることを Sinha et al. (2003) に従って示そう.

定義 確率過程 $\{Z(t) \mid t \geq 0\}$ がガンマ過程であるとは

(i) $Z(0) = 0$

(ii) 任意の $t_1 < t_2 < \dots < t_n$ に対し, $Z(t_2) - Z(t_1), Z(t_3) - Z(t_2), \dots, Z(t_n) - Z(t_{n-1})$ はたがいに独立

(iii) 左連続な増大関数 $\alpha(t)$ ($\alpha(0) = 0$) と $c > 0$ が存在して, 任意の $t > s$ について,

$$Z(t) - Z(s) \sim Ga(\alpha(t) - \alpha(s), c)$$

を満たすことである. ただし, $Ga(a, b)$ はパラメータ (a, b) のガンマ分布とする. また, このとき $Z(t) \sim \mathcal{GP}(\alpha(t), c)$ と書くこととする.

$H_0(t) \sim \mathcal{GP}(cH^*(t), c)$ とする. ただし, $H^*(t)$ は増大関数で $H^*(0) = 0, c > 0$ とし, $H^*(t), c$ はともに既知とする. イベント又は打ち切りの発生時間を $u_{(1)} < u_{(2)} < \dots < u_{(n)}$ とし, $u_{(0)} = 0$ として,

$$h_j = H_0(u_{(j)}) - H_0(u_{(j-1)})$$

と定義すると,

$$h_j \sim Ga(ch_{0j}, c)$$

となる. ただし, $h_{0j} = H^*(u_{(j)}) - H^*(u_{(j-1)})$ である. ここで, $A_j = \sum_{l \in R(u_{(j)})} \exp(\mathbf{x}_l \boldsymbol{\beta})$ とすると,

$$\begin{aligned} \prod_{i=1}^n P(T_i > u_i \mid X, \boldsymbol{\beta}, H_0(t)) &= \prod_{i=1}^n \exp(-\exp(\mathbf{x}_i \boldsymbol{\beta}) H_0(u_i)) \\ &= \exp\left(-\sum_{i=1}^n \exp(\mathbf{x}_i \boldsymbol{\beta}) H_0(u_i)\right) \end{aligned}$$

$$\begin{aligned}
&= \exp \left(- \sum_{j=1}^n h_j \sum_{l \in R(u_{(j)})} \exp(\mathbf{x}_l \boldsymbol{\beta}) \right) \\
&= \exp \left(- \sum_{j=1}^n h_j A_j \right) \\
&= \prod_{j=1}^n \exp(-h_j A_j)
\end{aligned}$$

であるから, $A_{n+1} = 0$ として,

$$\begin{aligned}
E \left(\prod_{i=1}^n P(T_i > t_i \mid X, \boldsymbol{\beta}, H_0(t)) \right) &= E \left(\prod_{j=1}^n \exp(-h_j A_j) \right) \\
&= \prod_{j=1}^n \int_0^\infty \exp(-h_j A_j) \frac{c}{\Gamma(ch_{0j})} (ch_j)^{ch_{0j}-1} \exp(-ch_j) dh_j \\
&= \prod_{j=1}^n \left(\frac{c}{c + A_j} \right)^{ch_{0j}} \\
&= \prod_{j=1}^n \exp \left(\log \left(\frac{c}{c + A_j} \right)^{ch_{0j}} \right) \\
&= \prod_{j=1}^n \exp \left(cH^*(t_{(j)}) \log \left(\frac{c + A_{j+1}}{c + A_j} \right) \right) \\
&= \prod_{j=1}^n \exp \left(cH^*(t_{(j)}) \log \left(1 - \frac{\exp(\mathbf{x}_j \boldsymbol{\beta})}{c + A_j} \right) \right)
\end{aligned}$$

となり,

$$\begin{aligned}
L(\boldsymbol{\beta} \mid \mathbf{t}, \boldsymbol{\delta}, X) &= \int_0^\infty L(\boldsymbol{\beta}, H_0(t) \mid \mathbf{t}, \boldsymbol{\delta}, X) \pi(H_0(t)) dH_0(t) \\
&= \prod_{i=1}^n \exp \left(cH^*(t_i) \log \left(1 - \frac{\exp(\mathbf{x}_i \boldsymbol{\beta})}{c + A_j} \right) \right) \left(-ch^*(t_i) \log \left(1 - \frac{\exp(\mathbf{x}_i \boldsymbol{\beta})}{c + A_j} \right) \right)^{\delta_i}
\end{aligned}$$

とかける. $d := \sum_{i=1}^n \delta_i$, $h^* := \prod_{i=1}^n (h^*(t_i))^{\delta_i}$ とすると,

$$\begin{aligned}
\lim_{c \downarrow 0} \frac{L(\boldsymbol{\beta} \mid \mathbf{t}, \boldsymbol{\delta}, X)}{(-c)^d h^* (\log c)^{\delta_n}} &= \lim_{c \downarrow 0} \prod_{i=1}^n \exp \left(cH^*(t_i) \log \left(1 - \frac{\exp(\mathbf{x}_i \boldsymbol{\beta})}{c + A_i} \right) \right) \left(\log \left(1 - \frac{\exp(\mathbf{x}_i \boldsymbol{\beta})}{c + A_i} \right) \right)^{\delta_i} \\
&= \prod_{i=1}^{n-1} \left(\frac{\exp(\mathbf{x}_i \boldsymbol{\beta})}{A_i} \right)^{\delta_i} \lim_{c \downarrow 0} \left(\frac{1}{\log c} \log \left(1 - \frac{\exp(\mathbf{x}_n \boldsymbol{\beta})}{c + A_n} \right) \right)^{\delta_n} \\
&= \prod_{i=1}^n \left(\frac{\exp(\mathbf{x}_i \boldsymbol{\beta})}{A_i} \right)^{\delta_i}
\end{aligned}$$

となり, これは Cox の部分尤度に等しい.

例 1(続) 観察時間 U_i , 打ち切りの指標 δ_i , 及び共変量 x_i ($i = 1, \dots, 5$) をそれぞれ $(11, 1, 0)$, $(12, 1, 1)$, $(13, 0, 0)$, $(16, 1, 1)$, $(21, 1, 1)$ としたときの部分尤度 $\tilde{L}_1(\boldsymbol{\beta} \mid \mathbf{u}, \boldsymbol{\delta}, X)$ は,

$$\tilde{L}_1(\boldsymbol{\beta} \mid \mathbf{u}, \boldsymbol{\delta}, X) = \left(\frac{1}{3 \exp(\boldsymbol{\beta}) + 2} \right) \left(\frac{\exp(\boldsymbol{\beta})}{3 \exp(\boldsymbol{\beta}) + 1} \right) \left(\frac{1}{2} \right)$$

であった。 β の事前分布を $\beta \sim N(0, 1)$ とすると、 β の事後分布 $p(\beta | \mathbf{u}, \delta, X)$ は、

$$p(\beta | \mathbf{u}, \delta, X) \propto \left(\frac{1}{3 \exp(\beta) + 2} \right) \left(\frac{\exp(\beta)}{3 \exp(\beta) + 1} \right) \left(\frac{1}{2} \right) \exp \left(-\frac{\beta^2}{2} \right)$$

となる。 また、 β の事前分布を $\pi(\beta) \propto 1$ ($-\infty < \beta < \infty$) とすると、 β の事後分布 $p(\beta | \mathbf{u}, \delta, X)$ は、

$$p(\beta | \mathbf{u}, \delta, X) \propto \left(\frac{1}{3 \exp(\beta) + 2} \right) \left(\frac{\exp(\beta)}{3 \exp(\beta) + 1} \right) \left(\frac{1}{2} \right)$$

となる。 よって、 このようにして得られた β の事後分布 $p(\beta | \mathbf{u}, \delta, X)$ を基に β の推測を行えばよい。

5 Rejection Sampling と Gibbs Sampling

ベイズ手法を用いてパラメータの推定を行う際に、尤度とパラメータの事前分布から得られた事後分布を基に推定を行うが、一般にパラメータの事後分布の密度関数の正確な形を表現することは難しく、また、平均や分散等を求める際に事後分布の密度関数を用いた積分計算を解析的に行うことも難しいことが多い。そこで、そのような場合には事後分布からの乱数を発生させ、その乱数を基に事後平均や事後中央値を求めることとなる。本稿では乱数生成の手法として、Rejection Sampling、及び Gibbs Sampling を用いる。

Rejection Sampling

乱数を生成したい分布 F の密度関数を $f(x)$ とし、定数 c と乱数の生成が容易である分布 G の密度関数 $g(x)$ を

$$f(x) \leq cg(x), \forall x \in \mathbf{R}$$

が成り立つように選んだとき、次の手順 (1) ~ (3) によって生成される乱数は分布 F に従う。これを Rejection Sampling と呼ぶ。

1. 一様分布 $U(0, 1)$ からの乱数 U を生成する。
2. 分布 G からの乱数 X を生成する。
3. $h(x) := f(x)/(cg(x))$ に対し、 $U \leq h(X)$ であれば X を受容し、 $U > h(X)$ であれば X を棄却する。

上記の乱数 X は、

$$\begin{aligned} P(X \leq x | U \leq h(X)) &= \frac{P(X \leq x, U \leq h(X))}{P(U \leq h(X))} \\ &= \frac{\int_{-\infty}^x \int_0^{h(t)} g(t) du dt}{\int_{-\infty}^{\infty} \int_0^{h(t)} g(t) du dt} \\ &= \frac{\int_{-\infty}^x h(t) g(t) dt}{\int_{-\infty}^{\infty} h(t) g(t) dt} \\ &= F(x) \end{aligned}$$

より、分布 F に従う。また、以上のことから、密度関数の定数倍の形がわかっている場合、すなわち、分布 F の密度関数 $f(x)$ に対して、

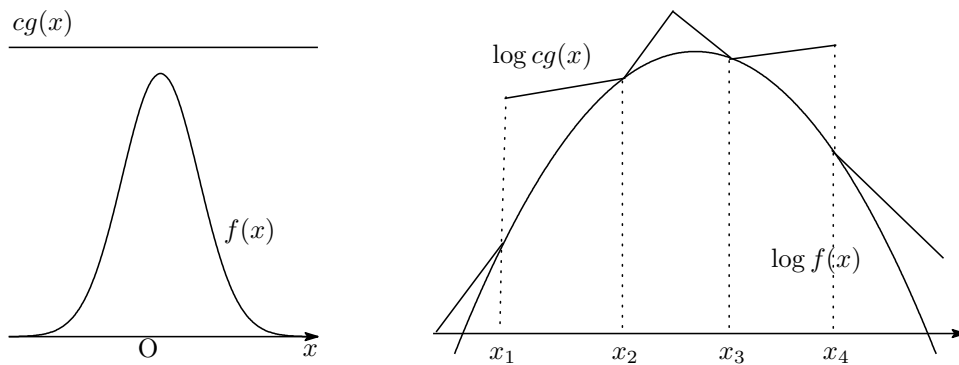
$$f^*(x) = kf(x), k \in \mathbf{R}$$

がわかっている場合であれば、Rejection Sampling により分布 F からの乱数を生成することができる。
 さらに、乱数を生成したい分布の密度関数 $f(x)$ に対し、 $\log f(x)$ が凹関数のときには下記の方法を用いるこ
 とによって乱数の生成が容易でかつ、 $f(x) \leq cg(x)$ を満たす密度関数 $g(x)$ を構成することができる。

1. n 個の点 $x_1 < x_2 < \dots < x_n$ を選ぶ。
2. $l(x) = \log f(x)$ について、 $(x_i, l(x_i))$ と $(x_{i+1}, l(x_{i+1}))$ を通る直線 $m_i(x)$ ($i = 1, \dots, n-1$) を考える。
3. $\log cg(x)$ を

$$\log cg(x) = \begin{cases} m_1(x) & (x < x_1), \\ m_2(x) & (x_1 < x < x_2), \\ \min(m_{i-1}(x), m_{i+1}(x)) & (x_i < x < x_{i+1} \ (i = 2, \dots, n-2)), \\ m_{n-2}(x) & (x_{n-1} < x < x_n), \\ m_{n-1}(x) & (x_n < x), \end{cases}$$

とする。



$f(x)$ を覆う $cg(x)$ の例。左図は一樣分布の密度関数の定数倍。右図は $\log f(x)$ が凹関数であることを利用して $\log cg(x)$ を構成し、その後に逆対数変換を行うことで $cg(x)$ が定まる。

例 2 Rejection Sampling を用いて、切断正規分布

$$f(x) = \begin{cases} \frac{1}{a} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) & (-5 \leq x \leq 5), \\ 0 & (x < -5, x > 5) \end{cases}$$

からの乱数生成を行う。ただし、

$$a = \int_{-5}^5 \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx$$

とする。

Rejection Sampling により乱数の生成を行う場合には、密度関数の定数倍の部分、つまり今の例の場合では $1/(a\sqrt{2\pi})$ を無視してもよいので、

$$f_1(x) = \begin{cases} \exp\left(-\frac{x^2}{2}\right) & (-5 \leq x \leq 5), \\ 0 & (x < -5, x > 5) \end{cases}$$

として, $f_1(x)$ を用いて乱数の生成を行う. まず, 一様分布 $U(-5, 5)$ の密度関数を $g(x)$ とし,

$$cg(x) = 1$$

を満たすように c を選ぶことによって,

$$f_1(x) \leq cg(x) \quad (-5 \leq x \leq 5)$$

とすることができる. 以上より, $f_1(x)$ を覆う関数 $cg(x)$ が定まったので, Rejection Sampling の手順 1. ~ 3. を繰り返すことにより密度関数 $f(x)$ をもつ分布からの乱数を生成することが可能となる.

SAS を用いて乱数を生成するプログラムの例は下記のようになる.

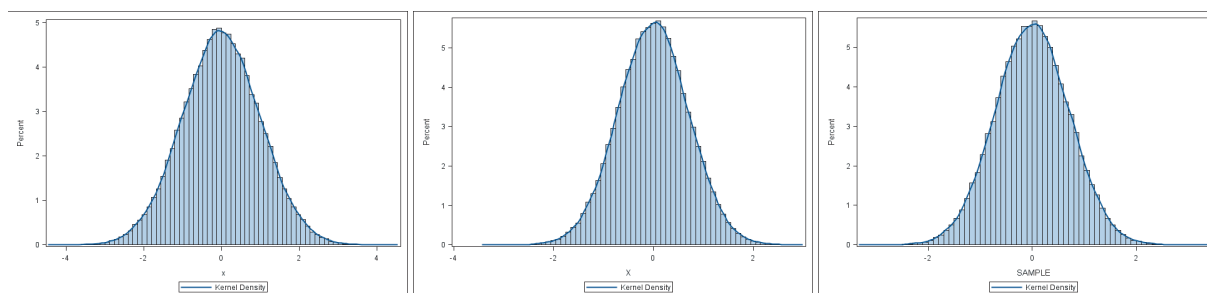
```
data TRNNOR ;
  I=1 ;
  N=100000 ;
  do while ( I<=N ) ;
    U=ranuni(1111) ;
    V=10*(ranuni(1111)-0.5) ;
    H=exp(-V**2)/1 ;

    if ( U<=H ) then do ;
      X=V ;
      I=I+1 ;
      output ;
    end ;
  end ;
run ;
```

*----- 発生させる乱数の数 ;
 *----- 受容と棄却の判定用の一様乱数 ;
 *----- $c(g(x))$ からの乱数 ;
 *----- 受容と棄却の判定 ;

また, $\log f_1(x) = -x^2$ は凹関数であるため, すでに紹介したように, c の値を指定せずに $f_1(x)$ を覆う関数 $cg(x)$ を構成することができる. SAS を用いて $f_1(x)$ を覆う関数 $cg(x)$ を構成し, さらに確率密度関数 $f(x)$ を持つ分布からの乱数を生成するプログラムを参考 1 として記載した.

上記の例において Rejection Sampling を使用し, $cg(x)$ を定数とした場合, $\log f(x)$ が凹関数となる性質を利用して $cg(x)$ を構成した場合のそれぞれにおいて乱数を生成し (それぞれ 100000 個), 生成した乱数のヒストグラムを描くと下図のようになる.



左から, $N(0, 1)$ からの乱数のヒストグラム, $g(x)$ に一様分布の密度関数を使用した Rejection Sampling による乱数のヒストグラム, 凹関数の性質を利用して $cg(x)$ を構成した Rejection Sampling による乱数のヒストグラム.

さらに, $\log f(x)$ が凹関数となる性質を利用して $cg(x)$ を構成し, 乱数を生成する方法において, 乱数生成の効率をより上昇させる方法として, Adaptive Rejection Sampling と呼ばれる, 最初に選んだ n 個の x 軸上

の初期点 $x_1 < x_2 < \dots < x_n$ に点列を追加していく方法もある。詳しくは井桁 (2010), 大森 (2001), Gilks et al. (1995) を参照。

Gibbs Sampling

確率変数 $\mathbf{X} = (X_1, X_2, \dots, X_n)$ について, $\mathbf{X}(-i) = (X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n)$ で条件付けた下での X_i の条件付き密度を $f(x_i | \mathbf{x}(-i))$ とし, 各 $f(x_i | \mathbf{x}(-i))$ は既知であるとする。このとき, 以下の方法により, X_i の周辺分布からの乱数を生成することができる。

1. 初期値 $x_2^{(0)}, \dots, x_n^{(0)}$ を決める。
2. $x_1^{(1)}, \dots, x_n^{(1)}$ を以下により生成する。
 - $f(x_1 | x_2^{(0)}, \dots, x_n^{(0)})$ から乱数を生成し, $x_1^{(1)}$ とする。
 - $f(x_2 | x_1^{(1)}, x_3^{(0)}, \dots, x_n^{(0)})$ から乱数を生成し, $x_2^{(1)}$ とする。
 - \vdots
 - $f(x_n | x_1^{(1)}, \dots, x_{n-1}^{(1)})$ から乱数を生成し, $x_n^{(1)}$ とする。
3. 2 を繰り返す。

上記の手法は Gibbs Sampling と呼ばれる。Gibbs Sampling により生成される乱数列 $X_i^{(1)}, \dots, X_i^{(n)}, \dots$ はその条件付き密度関数が

$$f(x_i^{(k)} | x_i^{(k-1)}, x_i^{(k-2)}, \dots, x_i^{(1)}) = f(x_i^{(k)} | x_i^{(k-1)}) := g(x | t) \quad (\forall k > 1)$$

を満たすことより推移核 $g(x | t)$ をもつマルコフ連鎖であり, さらに, X_i の周辺分布 $f_{X_i}(x)$ について,

$$\begin{aligned} f_{X_i}(x_i) &= \int \dots \int f(x_1, \dots, x_n) dx_1 \dots dx_{i-1} dx_{i+1} \dots dx_n \\ &= \int \dots \int f(x_i | \mathbf{x}(-i)) f(\mathbf{x}(-i)) dx_1 \dots dx_{i-1} dx_{i+1} \dots dx_n \\ &= \int \dots \int f(x_i | \mathbf{x}(-i)) \int f(x_1, \dots, x_n) dx_i dx_1 \dots dx_{i-1} dx_{i+1} \dots dx_n \\ &= \int \dots \int f(x_i | \mathbf{x}(-i)) f(\mathbf{x}(-i) | t) f_{X_i}(t) dt dx_1 \dots dx_{i-1} dx_{i+1} \dots dx_n \\ &= \int \left[\int \dots \int f(x_i | \mathbf{x}(-i)) f(\mathbf{x}(-i) | t) dx_1 \dots dx_{i-1} dx_{i+1} \dots dx_n \right] f_{X_i}(t) dt \\ &= \int g(x | t) f_{X_i}(t) dt \end{aligned}$$

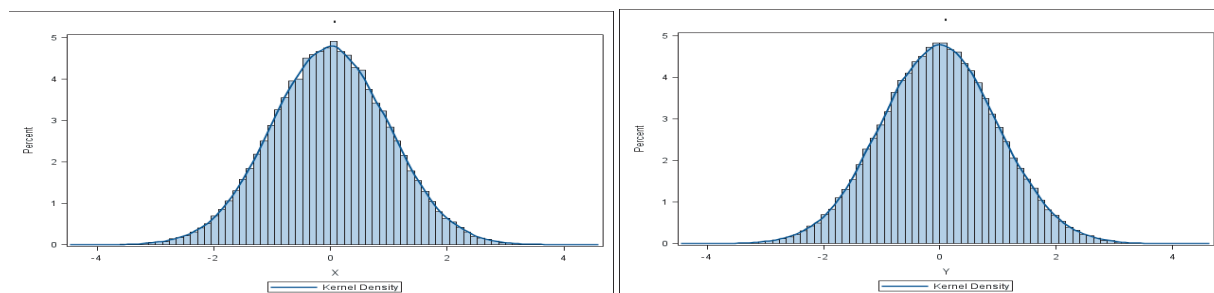
より $f_{X_i}(x)$ が定常分布となっていることがわかる。また, このとき, Gibbs Sampling により生成される乱数列 $X_i^{(1)}, \dots, X_i^{(n)}, \dots$ は定常分布 $f_{X_i}(x)$ に収束することが知られている (Chan(1993)). 以上のことから, 十分大きな n に対し, Gibbs Sampling により生成された乱数 $X_i^{(n)}$ は密度関数 $f_{X_i}(x)$ をもつ分布からの乱数と見なすことができる。

例 3 Gibbs Sampling を用いて, 2 変量正規分布からの乱数 X, Y を生成する。ただし, $E(X) = E(Y) = 0$, $V(X) = V(Y) = 1$, $\rho(X, Y) = 1/2$ とする。 $X = x$ を与えたときの Y の条件付分布は $N(x/2, (1 - (1/4)))$ であり, $Y = y$ を与えたときの X の条件付分布は $N(y/2, (1 - (1/4)))$ であるから, $f(x | y)$, $f(y | x)$ を算出することが可能である。また, Gibbs Sampling では $f(x | y)$, $f(y | x)$ からの乱数の生成を行うこととなる

が、今の例では $f(x|y)$, $f(y|x)$ は正規分布の密度関数となるので、乱数の生成は容易である。SAS を用いて Gibbs Sampling を行うプログラムは次のようになる。

```
DATA mnor_G ;
  keep X Y ;
  X=0 ; Y=0 ;          *----- 初期値 ;
  n=100000 ;           *----- 標本の大きさ ;
  seed=1111 ;          *----- シード値 ;
  do i=1 to n ;
    if i=1 then do ;
      X=Y/2+sqrt(3/4)*rannor(seed) ;
      Y=X/2+sqrt(3/4)*rannor(seed) ;
      output ;
      Z=Y ;
    end ;
    else do ;
      X=Z/2+sqrt(3/4)*rannor(seed) ;
      Y=X/2+sqrt(3/4)*rannor(seed) ;
      output ;
      Z=Y ;
    end ;
  end ;
run ;
```

上記のプログラムによって Gibbs Sampling を行い、 X , Y についてそれぞれ 100000 個の乱数を生成した結果、 X と Y の標本相関係数は 0.497 であった。また、 X , Y それぞれのヒストグラムは下図のようになった。



左図は、 X のヒストグラム。右図は Y のヒストグラム。

また、上記の例では各条件付き分布 $f(x_i | x(-i))$ からの乱数生成が容易であるが、 $f(x_i | x(-i))$ からの乱数生成が容易でない場合や $f(x_i | x(-i))$ の定数倍の形しかわからない場合には、各条件付き密度関数の対数をとった $\log f(x_i | x(-i))$ が凹関数であれば、参考 1 のマクロ %RS を応用することにより Gibbs Sampling を行うことができる（例 4 参照）。

上記で解説した Rejection Sampling と Gibbs Sampling を用いて、Cox 回帰におけるパラメータの Bayes 推定値を算出する。

例 1(続) 観察時間 U_i , 打ち切りの指標 δ_i , 及び共変量 x_i ($i = 1, \dots, 5$) をそれぞれ $(11, 1, 0)$, $(12, 1, 1)$, $(13, 0, 0)$, $(16, 1, 1)$, $(21, 1, 1)$ とし、 β の事前分布を $\beta \sim N(0, 1)$ としたとき、 β の事後分布 $p(\beta | \mathbf{u}, \boldsymbol{\delta}, X)$ は、

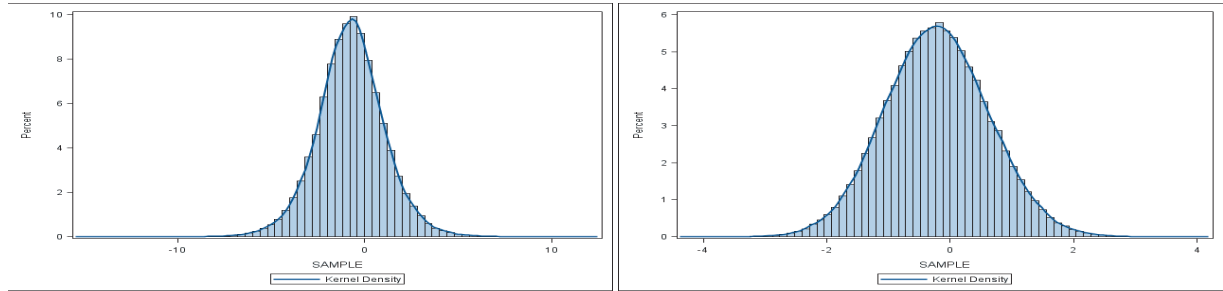
$$p(\beta | \mathbf{u}, \boldsymbol{\delta}, X) \propto \left(\frac{1}{3 \exp(\beta) + 2} \right) \left(\frac{\exp(\beta)}{3 \exp(\beta) + 1} \right) \left(\frac{1}{2} \right) \exp \left(-\frac{\beta^2}{2} \right)$$

となり, β の事前分布を $\pi(\beta) \propto 1$ ($-\infty < \beta < \infty$) としたとき, β の事後分布 $p(\beta | \mathbf{u}, \boldsymbol{\delta}, X)$ は,

$$p(\beta | \mathbf{u}, \boldsymbol{\delta}, X) \propto \left(\frac{1}{3 \exp(\beta) + 2} \right) \left(\frac{\exp(\beta)}{3 \exp(\beta) + 1} \right) \left(\frac{1}{2} \right)$$

となった. それぞれの β の事後分布から Rejection Sampling により乱数を生成させ, 平均値, 中央値, 標準偏差, 90%CI を求めると下記のようになる. また, 生成した乱数のヒストグラムを描くと下図のようになる.

事前分布	平均値	中央値	標準偏差	90%CI
一様分布	-0.74	-0.74	1.814	(-3.68, 2.21)
平均 0, 分散 1 の正規分布	-0.22	-0.22	0.842	(-1.59, 1.18)



左図は, β の事前分布を一様分布とした場合の事後分布からのヒストグラム. 右図は β の事前分布を平均 0, 分散 1 の正規分布とした場合の事後分布からのヒストグラム.

例 4 観察時間 U_i , 打ち切りの指標 δ_i , 性別 x_{1i} (1 : 男性, 0 : 女性), BMI x_{2i} ($i = 1, \dots, 5$) をそれぞれ (98, 1, 0, 21.5), (104, 1, 0, 36.7), (1278, 1, 1, 23.6), (1889, 0, 1, 23.3), (2031, 1, 0, 32.4), (2114, 0, 1, 25.8), (2595, 0, 0, 30.1) とし,

$$h(t, \mathbf{x}, \boldsymbol{\beta}) = h_0(t) \exp(\beta_1 x_1 + \beta_2 x_2)$$

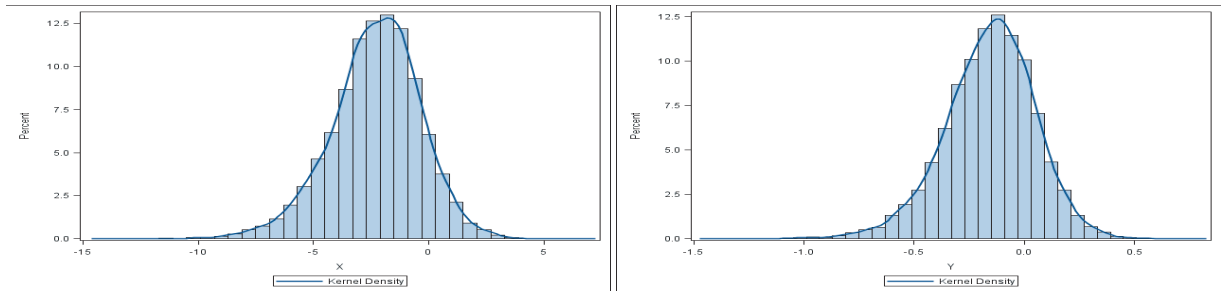
というモデルに基づき, β_1, β_2 の Bayes 推定値を算出することを考える. また, $\pi(\beta_1, \beta_2) \propto 1$ ($-\infty < \beta_1, \beta_2 < \infty$) とする. このとき, β_1, β_2 の事後密度は部分尤度の定数倍となるので,

$$\begin{aligned}
 & p(\beta_1, \beta_2 | \mathbf{u}, \boldsymbol{\delta}, X) \\
 & \propto \frac{\exp(21.5\beta_2)}{\exp(21.5\beta_2) + \exp(36.7\beta_2) + \exp(\beta_1 + 23.6\beta_2) + \exp(\beta_1 + 23.3\beta_2) + \exp(32.4\beta_2) + \exp(\beta_1 + 25.8\beta_2) + \exp(30.1\beta_2)} \\
 & \quad \times \frac{\exp(36.7\beta_2)}{\exp(36.7\beta_2) + \exp(\beta_1 + 23.6\beta_2) + \exp(\beta_1 + 23.3\beta_2) + \exp(32.4\beta_2) + \exp(\beta_1 + 25.8\beta_2) + \exp(30.1\beta_2)} \\
 & \quad \times \frac{\exp(\beta_1 + 23.6\beta_2)}{\exp(\beta_1 + 23.6\beta_2) + \exp(\beta_1 + 23.3\beta_2) + \exp(32.4\beta_2) + \exp(\beta_1 + 25.8\beta_2) + \exp(30.1\beta_2)} \\
 & \quad \times \frac{\exp(32.4\beta_2)}{\exp(32.4\beta_2) + \exp(\beta_1 + 25.8\beta_2) + \exp(30.1\beta_2)}
 \end{aligned}$$

となる. ここで, Gibbs Sampling によって β_1, β_2 のそれぞれの周辺分布からの乱数を生成したいが, 今の例では Gibbs Sampling の手順における条件付き密度からの乱数生成が容易でないため, 条件付き密度からの乱数生成の際に $\log p(\beta_1, \beta_2 | \mathbf{u}, \boldsymbol{\delta}, X)$ が凹関数である性質を用いた Rejection Sampling を使用する. 上記の手順を実行するプログラムを参考 2 として記載した.

以上により, β_1, β_2 の平均値, 中央値, 標準偏差, 90%CI を求めると下記のようになる. また, 生成した乱数のヒストグラムを描くと下図のようになる.

	平均値	中央値	標準偏差	90%CI
β_1	-2.31	-2.19	1.961	(-5.71,0.71)
β_2	-0.16	-0.15	0.207	(-0.52,0.16)



左図は、 β_1 のヒストグラム。右図は β_2 のヒストグラム。

6 PHREG プロシジャにおける BAYES ステートメント

SAS9.2 からは PHREG プロシジャで BAYES ステートメントを使用することにより、CoX 回帰におけるパラメータの Bayes 推定を行うことができる。大まかなアルゴリズムとしては、Adaptive Rejection Sampling や Gibbs Sampling を使用したものであり、これらの解説については前節までで行った (SAS(2006))。以下に Bayes ステートメントにおける主なオプションを紹介する。

(1) COEFF 又は COEFFPRIOR オプション

「COEFF=UNIFORM」又は「COEFF=NORMAL」の指定により、Cox 回帰におけるパラメータの事前分布を一様分布とするか正規分布とするかを選択できる。COEFF オプションを指定しない場合、パラメータの事前分布は一様分布となる。また、「COEFF=NORMAL」を指定した場合、各パラメータの事前分布の平均値、分散及び共分散を指定することもできる。デフォルトは「COEFF=UNIFORM」である。

(2) INITIAL オプション

「INITIAL=SAS データセット名」の指定により、Gibbs Sampling を行う際の初期値を指定することができる。

(3) NBI オプション

「NBI=n」(n は自然数) の指定により、Burn-in までの繰り返し数 n (最初の n 個の乱数を捨てること) を指定できる。デフォルトは「NBI=2000」である。

(4) NMC オプション

「NMC=n」(n は自然数) の指定により、Burn-in 後の繰り返し数 n (モンテカルロ法を使う際の乱数の数 n) を指定できる。デフォルトは「NMC=10000」である。

(5) SEED オプション

「SEED=n」(n は整数) の指定により、乱数生成の際のシード値を指定できる。

(6) SUM 又は SUMMARY オプション

「SUM=(出力したい要約統計量)」の指定により、指定した要約統計量を出力することができる。指定可能な要約統計量は、DESC 又は DESCRIPTIVE で平均、標準偏差、四分位点の出力、INTERVAL (ALPHA= α) で $100(1-\alpha)\%$ 確信区間、HPD 区間の出力、CORR で分散共分散行列の出力を行う。デフォルトは「SUM=(DESCRIPTIVE INTERVAL)」(α は 0.05) であり、「SUM=ALL」により、すべての要約統計量を出力することが可能である。

(7) THIN 又は THINNING オプション

「THIN=n」(n は自然数) の指定により, 1 番目, n+1 番目, 2n+1 番目, … の乱数を用いて推測を行うように指定できる. デフォルトは「THIN=1」である.

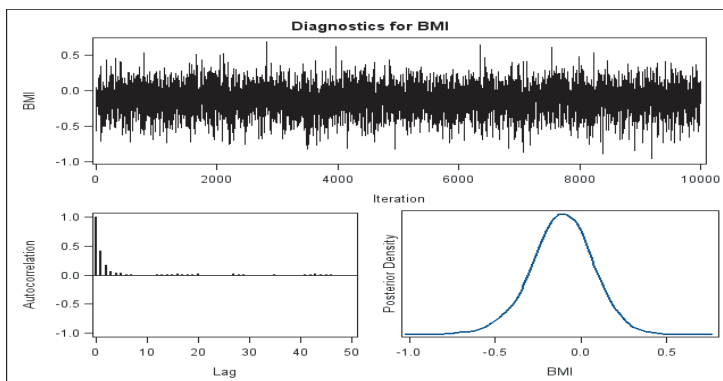
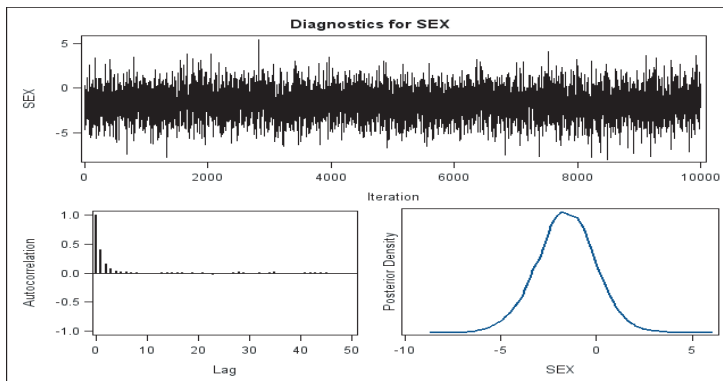
例 4 のデータを解析する際のプログラム例及びその出力の一部は以下のようになる.

```
data DATA ;
  input TIME CENSOR SEX BMI ;
  datalines ;
    98 1 0 21.5
    1889 0 1 23.3
    2595 0 0 30.1
    104 1 0 36.7
    2114 0 1 25.8
    1278 1 1 23.6
    2031 1 0 32.4
  ;
run ;

data INITIAL ;
  input SEX BMI ;
  datalines ;
    0 0
  ;
run ;

data NORMALPRIOR ;
  input _TYPE_ $ SEX BMI ;
  datalines ;
    MEAN 0 0
    COV 10 3
    COV 3 5
  ;
run ;

ods graphics on ;
proc phreg data=DATA alpha=0.1 ;
  model TIME*CENSOR(0) = SEX BMI ;
  bayes coeff=NORMAL(input=NORMALPRIOR)
  initial=INITIAL
nbi=0 nmc=10000 seed=1111
  sum=ALL thin=1 ;
run ;
ods graphics off ;
```



PHREG プロシジャ
Bayesian Analysis
モデルの詳細

Burn-In Size				0
MC Sample Size				10000
Thinning				1
全体	イベント	打ち切り		パーセント 打ち切り
7	4	3		42.86
回帰係数の正規事前分布				
		共分散行列		
パラメータ	平均	SEX	BMI	
SEX	0	10	3	
BMI	0	3	5	
Initial Values of the Chain				
連鎖	シード	SEX	BMI	
1	1111	0	0	

パラメータ	N	事後要約			パーセント点		
		平均	標準偏差	25%	50%	75%	
SEX	10000	-1.5978	1.5493	-2.5855	-1.5807	-0.5844	
BMI	10000	-0.1108	0.1849	-0.2261	-0.1058	0.0106	

パラメータ	アルファ	事後間隔		HPD 区間	
		両側対称区間			
SEX	0.100	-4.2000	0.8843	-4.1522	0.9206
BMI	0.100	-0.4258	0.1814	-0.4141	0.1911

Posterior Covariance Matrix		
Parameter	SEX	BMI
SEX	2.4001	0.1828
BMI	0.1828	0.0342

Posterior Correlation Matrix		
Parameter	SEX	BMI
SEX	1.0000	0.6382
BMI	0.6382	1.0000

7 おわりに

本稿において, Cox 回帰におけるパラメータを Bayes 流に推定する際に, 尤度の代わりに部分尤度を用いる妥当性を Sinha ら (2003) に従って示した. また, Rejection Sampling 及び Gibbs Sampling を紹介し, これらの乱数生成手法を用いることによって, Cox 回帰におけるパラメータの事後分布からの乱数を生成可能なことを示した.

SAS9.2 では, 本稿で紹介した方法を PHREG プロシジャの Bayes ステートメントを用いることによって実行可能となっており, Bayes 推定を行う際には非常に便利なプロシジャとなっている. PHREG プロシジャでは現在のところ, 事前分布として一様分布, 正規分布のみ指定可能となっているため, 一様分布や正規分布以外の分布を事前分布として指定する場合には, Sinha ら (2003) により, 部分尤度を求めたうえでパラメータの事後分布を算出し, 事後分布からの乱数を生成することが必要となるが, 事前分布の密度関数 $f(x)$ の対数をとった $\log f(x)$ が凹関数の場合には, 本稿で紹介した Rejection Sampling のプログラム %RS と Gibbs Sampling を組み合わせた手法を応用することにより, パラメータの事後分布からの乱数生成を行うことが可能となる.

謝辞

本稿を作成するにあたり, 熱心なるご指導をいただいた方々に心より御礼申し上げます. また, ともに論文作成に励んだ井桁正堯さんに感謝の意を伝えたいと思います.

連絡先

e-mail : Kuroda_Shingo@takeda.co.jp

参考 1 Rejection Sampling のプログラム例

$\log f(x)$ が凹関数であることを利用して $f(x)$ を覆う関数 $cg(x)$ を構成し, Rejection Sampling により密度関数 $f(x)$ をもつ分布からの乱数を生成する SAS プログラムの例は下記のようなになる.

```
proc fcmp outlib=work.myfunc.xxx ;
  function f(x) ;
    f=-x**2 ;
    return (f) ;
  endsub ;
run ;
```



```

options cmplib=work.myfunc ;

%macro RS(LIST,SIMSIZE,SEED=1111) ;
/*
  LIST    x 軸上の初期点
  SIMSIZE 乱数の生成数
  SEED    乱数のシード
*/
  data TRNNOR_2(keep=SAMPLE) ;
*----- 初期値の読み込み ;
  array c[*] c1-c999 ;
  %let I=1 ;
  %do %until( %qscan(&LIST,&I," ")=%str() ) ;
    c[&I]=%qscan(&LIST,&I," ") ;
    %let I = %eval(&I+1) ;
  %end ;
*----- x 軸上の点の数の計算 ;
  PNT=%eval(&I-1) ;
*----- 配列の定義 ;
  array a[*] a1-a&I ; *----- 直線の傾きの配列 ;
  array b[*] b1-b&I ; *----- 直線の切片の配列 ;
  array d[*] d1-d&I ; *----- 2 直線の交点の x 座標の配列 ;
  array S[2,&I] S1_1-S1_&I S2_1-S2_&I ; *----- 区間の面積の配列 ;
*----- その他の初期値の設定 ;
  k =1 ; *----- 生成数 ;
  ACC=0 ; *----- 0:棄却, 1:受容 ;

  do while ( k<=&SIMSIZE ) ;
*----- (c[i], f(c[i])) と (c[i+1], f(c[i+1])) を結ぶ直線の傾きと切片の計算 ;
  do i=1 to PNT-1 ;
    a[i]=(f(c[i+1])-f(c[i]))/(c[i+1]-c[i]) ;
    b[i]=f(c[i+1])-a[i]*c[i+1] ;
  end ;
*----- 直線の交点の計算 ;
  do i=3 to PNT-1 ;
    d[i]=(b[i-2]-b[i])/(a[i]-a[i-2]) ;
  end ;
*----- 各区間の曲線下面積の計算 ;
  do i=1 to PNT+1 ;
    if i=1 then do ;
      S[1,i]=0 ;
      S[2,i]=exp(a[i]*c[i]+b[i])/a[i] ;
    end ;
    else if i=2 then do ;
      S[1,i]=0 ;
      S[2,i]=(exp(a[i]*c[i]+b[i])-exp(a[i]*c[i-1]+b[i]))/a[i] ;
    end ;
    else if i=PNT then do ;
      S[1,i]=(exp(a[i-2]*c[i]+b[i-2])-exp(a[i-2]*c[i-1]+b[i-2]))/a[i-2] ;
      S[2,i]=0 ;
    end ;
    else if i=PNT+1 then do ;
      S[1,PNT+1]=-exp(a[i-2]*c[i-1]+b[i-2])/a[i-2] ;
      S[2,PNT+1]=0 ;
    end ;
    else do ;
      S[1,i]=(exp(a[i-2]*d[i]+b[i-2])-exp(a[i-2]*c[i-1]+b[i-2]))/a[i-2] ;
      S[2,i]=(exp(a[i]*c[i]+b[i])-exp(a[i]*d[i]+b[i]))/a[i] ;
    end ;
  end ;
*----- 全面積の和の計算 ;
  SS = sum(of S[*]) ;

```

```

*----- 乱数生成を行う区間を面積比に対応させて選択 ;
u1 =ranuni(&SEED.) ;
CS =0 ;
BLOCK=0 ;

do j=1 to PNT+1 ;
do i=1 to 2 ;
CS=CS+S[i,j]/SS ;
if CS>u1 and BLOCK=0 then do ;
FLGi =i ;
FLGj =j ;
BLOCK=1 ;
end ;
end ;
end ;
*----- 選択した区間から乱数を生成し、受容と棄却の判定 ;
u2=ranuni(&SEED.) ; *----- 逆関数法用の一様乱数 ;
u3=ranuni(&SEED.) ; *----- 判定用の一様乱数 ;

if FLGi=1 then do ;
if FLGj=PNT+1 then do ;
SAMPLE=log(1-u2)/a[FLGj-2]+c[FLGj-1] ;
H =exp(f(SAMPLE))/exp(a[PNT-1]*SAMPLE+b[PNT-1]) ;
if u3<=H then ACC=1 ;
else ACC=0 ;
end ;
else do ;
if FLGj=PNT then SAMPLE=log(u2*exp(a[FLGj-2]*c[FLGj])+(1-u2)*exp(a[FLGj-2]*c[FLGj-1]))
/a[FLGj-2] ;
else SAMPLE=log(u2*exp(a[FLGj-2]*d[FLGj])+(1-u2)*exp(a[FLGj-2]*c[FLGj-1]))
/a[FLGj-2] ;

H=exp(f(SAMPLE))/exp(a[FLGj-2]*SAMPLE+b[FLGj-2]) ;
if u3<=H then ACC=1 ;
else ACC=0 ;
end ;
end ;

if FLGi=2 then do ;
if FLGj=1 then do ;
SAMPLE=log(u2)/a[FLGj]+c[FLGj] ;
H =exp(f(SAMPLE))/exp(a[FLGj]*SAMPLE+b[FLGj]) ;
if u3<=H then ACC=1 ;
else ACC=0 ;
end ;
else do ;
if FLGj=2 then SAMPLE=log(u2*exp(a[FLGj]*c[FLGj])+(1-u2)*exp(a[FLGj]*c[FLGj-1]))
/a[FLGj] ;
else SAMPLE=log(u2*exp(a[FLGj]*c[FLGj])+(1-u2)*exp(a[FLGj]*d[FLGj]))
/a[FLGj] ;

H=exp(f(SAMPLE))/exp(a[FLGj]*SAMPLE+b[FLGj]) ;
if u3<=H then ACC=1 ;
else ACC=0 ;
end ;
end ;

if ACC=1 then do ;
if SAMPLE<-5 or SAMPLE>5 then ACC=0 ; *----- 範囲外の乱数を棄却 ;
else do ;
k=k+1 ;
output ;
end ;
end ;

```

```

end ;
run ;
%mend ;

%RS(LIST=-10.0 -9.5 -9.0 -8.5 -8.0 -7.5 -7.0 -6.5 -6.0 -5.5 -5.0
-4.5 -4.0 -3.5 -3.0 -2.5 -2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0
2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0 9.5 10.0,
SIMSIZE=100000) ;

```

上記のマクロ %RS を実行することにより、正規分布からの 100000 個の乱数を生成することができる。

参考 2 例 4 のプログラム例 (Gibbs Sampling と Rejection Sampling の組み合わせ)

```

proc fcmp outlib=work.myfunc.xxx ;
function f(x,y) ;
f= log( (exp(21.5*y)/(exp(21.5*y)+exp(36.7*y)+exp(x+23.6*y)+exp(x+23.3*y)
+exp(32.4*y)+exp(x+25.8*y)+exp(30.1*y)))
*(exp(36.7*y)/(exp(36.7*y)+exp(x+23.6*y)+exp(x+23.3*y)+exp(32.4*y)+exp(x+25.8*y)+exp(30.1*y)))
*(exp(x+23.6*y)/(exp(x+23.6*y)+exp(x+23.3*y)+exp(32.4*y)+exp(x+25.8*y)+exp(30.1*y)))
*(exp(32.4*y)/(exp(32.4*y)+exp(x+25.8*y)+exp(30.1*y)))) ;
return (f) ;
endsub ;
run ;
options cmplib=work.myfunc ;

%macro GS(LIST, Y, SIMSIZE, SEED=1111) ;
/*
LIST    x 軸上の初期点
Y       Y の初期値
SIMSIZE 乱数の生成数
SEED    乱数のシード
*/
data GS_PHREG(keep=X Y) ;
*----- 初期値の読み込み ;
array c[*] c1-c999 ;
%let I = 1 ;
%do %until( %qscan(&LIST, &I, " ")=%str() ) ;
c[&I.] = %qscan(&LIST.,&I., " ") ;
%let I = %eval(&I.+1) ;
%end ;
*----- x 軸上の点の数の計算 ;
PNT = %eval(&I.-1) ;
*----- 配列の定義 ;
array a[*] a1-a&I. ; *----- 直線の傾きの配列 ;
array b[*] b1-b&I. ; *----- 直線の切片の配列 ;
array d[*] d1-d&I. ; *----- 2 直線の交点の x 座標の配列 ;
array LF[*] LF1-LF&I. ;
array S[2,&I.] S1_1-S1_&I. S2_1-S2_&I. ; *----- 区間の面積の配列 ;
*----- その他の初期値の設定 ;
k =1 ; *----- 生成数 ;
ACC =0 ; *----- 0:棄却, 1:受容 ;
COUNT =0 ; *----- 0:X の乱数を生成中, 1:Y の乱数を生成中 ;
Y =&Y. ; *----- Y の初期値 ;
*----- ギブスサンプリング ;
do while (k <= &SIMSIZE) ;
*----- 乱数の棄却, 採択, 出力に関する処理 ;
if (ACC ne COUNT) then do ; *----- Y の乱数を生成 ;
if (ACC=1) then do ; *----- SAMPLE を採択する場合 ;
X=SAMPLE ; COUNT=1 ;
end ;
do i=1 to PNT ; LF[i] = f(X,c[i]) ; end ;

```

```

end ;
else do ;
    *----- X の乱数を生成 ;
    if (ACC=1) then do ;
        *----- SAMPLE を採択する場合 ;
        Y=SAMPLE ; COUNT=0 ; k=k+1 ; output ;
    end ;
    do i=1 to PNT ; LF[i] = f(c[i],Y) ; end ;
end ;
*----- (c[i], f(c[i])) と (c[i+1], f(c[i+1])) を結ぶ直線の傾きと切片の計算 ;
do i=1 to PNT-1 ;
    a[i]=(LF[i+1]-LF[i])/(c[i+1]-c[i]) ;
    b[i]=LF[i+1]-a[i]*c[i+1] ;
end ;
*----- 直線の交点の計算 ;
do i=3 to PNT-1 ;
    d[i]=(b[i-2]-b[i])/(a[i]-a[i-2]) ;
end ;
*----- 各区間の曲線下面積の計算 ;
do i=1 to PNT+1 ;
    if (i <= 2) then do ;
        S[1,i]=0 ;
        S[2,i]=(exp(a[i]*c[i]+b[i])-(i-1)*exp(a[i]*c[1]+b[i]))/a[i] ;
    end ;
    else if (i >= PNT) then do ;
        S[1,i]=((PNT+1-i)*exp(a[i-2]*c[i]+b[i-2])-exp(a[i-2]*c[i-1]+b[i-2]))/a[i-2] ;
        S[2,i]=0 ;
    end ;
    else do ;
        S[1,i]=(exp(a[i-2]*d[i]+b[i-2])-exp(a[i-2]*c[i-1]+b[i-2]))/a[i-2] ;
        S[2,i]=(exp(a[i]*c[i]+b[i])-exp(a[i]*d[i]+b[i]))/a[i] ;
    end ;
end ;
*----- 全面積の和の計算 ;
SS = sum(of S[*]) ;
*----- 乱数生成を行う区間を面積比に対応させて選択 ;
u1 =ranuni(&SEED.) ;
CS =0 ;
BLOCK=0 ;

do j=1 to PNT+1 ;
    do i=1 to 2 ;
        CS = CS + S[i,j]/SS ;
        if CS > u1 and BLOCK=0 then do ;
            FLGi =i ; FLGj =j ; BLOCK=1 ;
        end ;
    end ;
end ;
*----- 選択した区間から乱数を生成し、受容と棄却の判定 ;
u2=ranuni(&SEED.) ;
u3=ranuni(&SEED.) ;
*----- 逆関数法用の一様乱数 ;
*----- 判定用の一様乱数 ;

if FLGi=1 then do ;
    if FLGj=PNT+1 then do ;
        SAMPLE=log(1-u2)/a[FLGj-2]+c[FLGj-1] ;
    if COUNT=1 then H=exp(f(X,SAMPLE))/exp(a[PNT-1]*SAMPLE+b[PNT-1]) ;
        else H=exp(f(SAMPLE,Y))/exp(a[PNT-1]*SAMPLE+b[PNT-1]) ;
        if u3<=H then ACC=1 ;
        else ACC=0 ;
    end ;
    else do ;
        if FLGj=PNT then SAMPLE=log(u2*exp(a[FLGj-2]*c[FLGj])+(1-u2)*exp(a[FLGj-2]*c[FLGj-1]))
            /a[FLGj-2] ;
        else SAMPLE=log(u2*exp(a[FLGj-2]*d[FLGj])+(1-u2)*exp(a[FLGj-2]*c[FLGj-1]))

```

```

/a[FLGj-2] ;
if COUNT=1 then H=exp(f(X,SAMPLE))/exp(a[FLGj-2]*SAMPLE+b[FLGj-2]) ;
else H=exp(f(SAMPLE,Y))/exp(a[FLGj-2]*SAMPLE+b[FLGj-2]) ;
if u3<=H then ACC=1 ;
else ACC=0 ;
end ;
end ;

if FLGi=2 then do ;
if FLGj=1 then do ;
SAMPLE=log(u2)/a[FLGj]+c[FLGj] ;
if COUNT=1 then H=exp(f(X,SAMPLE))/exp(a[FLGj]*SAMPLE+b[FLGj]) ;
else H=exp(f(SAMPLE,Y))/exp(a[FLGj]*SAMPLE+b[FLGj]) ;
if u3<=H then ACC=1 ;
else ACC=0 ;
end ;
else do ;
if FLGj=2 then SAMPLE=log(u2*exp(a[FLGj]*c[FLGj])+(1-u2)*exp(a[FLGj]*c[FLGj-1]))
/a[FLGj] ;
else SAMPLE=log(u2*exp(a[FLGj]*c[FLGj])+(1-u2)*exp(a[FLGj]*d[FLGj]))
/a[FLGj] ;

if COUNT=1 then H=exp(f(X,SAMPLE))/exp(a[FLGj]*SAMPLE+b[FLGj]) ;
else H=exp(f(SAMPLE,Y))/exp(a[FLGj]*SAMPLE+b[FLGj]) ;
if u3<=H then ACC=1 ;
else ACC=0 ;
end ;
end ;

end ;
run ;
%mend ;
%GS(LIST=-10.0 -9.5 -9.0 -8.5 -8.0 -7.5 -7.0 -6.5 -6.0 -5.5 -5.0
-4.5 -4.0 -3.5 -3.0 -2.5 -2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0
2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0 9.5 10.0,
Y=1, SIMSIZE=10000) ;

```

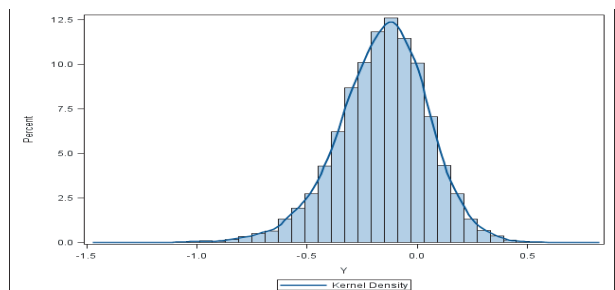
参考3 SGPLOT プロシジャを用いたグラフの作成

SAS9.2 から使用することができる SGPLOT プロシジャを用いると、生成した乱数の収束診断のプロットやヒストグラムの作成、自己相関のグラフの作成等を簡単に行うことができる。例4において SGPLOT プロシジャを使用したプログラム、及び出力される図は以下のようになる。

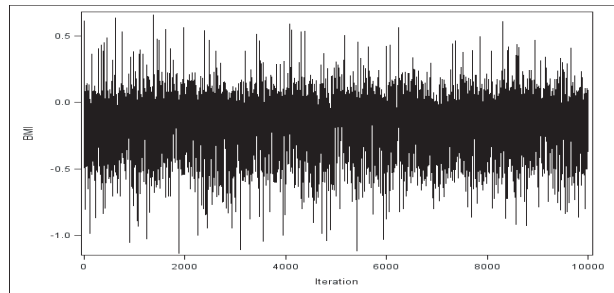
```

ods graphics on / imagefmt=png ;
proc sgplot data=GS_PHREG ;
  histogram BMI ;
  density BMI / type=kernel ;
run ;
ods graphics off ;

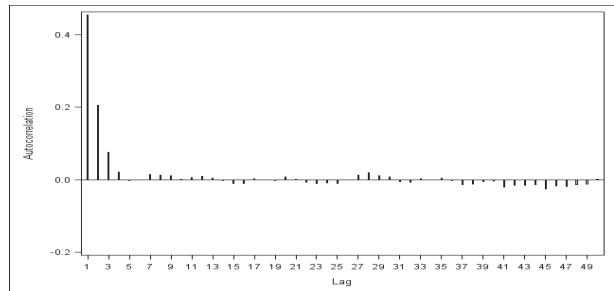
```



```
ods graphics on / imagefmt=png ;
proc sgplot data=GS_PHREG;
  series x=n y=BMI; *--- n は乱数の生成番号
  xaxis label="Iteration" ;
run;
ods graphics off ;
```



```
ods graphics on / imagefmt=png ;
proc sgplot data=CORR;
  vbar n / response=BMI
  barwidth=0.1 ;
  xaxis fitpolicy=thin label="Lag" ;
  yaxis min=-0.2 label="Autocorrelation" ;
run;
ods graphics off ;
```



参考文献

- Cassella, G. and George, E. I. (1992). Explaining the Gibbs Sampler. *The American Statistician*, **46**, 167-174.
- Chan, K. S. (1993). Asymptotic Behavior of the Gibbs Sampler. *Journal of the American Statistical Association*, **88**, 320-326.
- Gilks, W. R. and Tan, K. K. C. (1995). Adaptive Rejection Metropolis Sampling within Gibbs Sampling. *Applied Statistics*, **44**, 455-472.
- Klein, J. P. and Moeschberger, M. L. (2003). *Survival Analysis*, Springer, Berlin.
- Lachin, J. M. (2000). *Biostatistical Methods: The Assessment of Relative Risks*, Wiley, New York.
- SAS/STAT(R) User's Guide. (2006). Preliminary Capabilities for Bayesian Analysis in SAS/STAT(R) Software .
- Sinha, D. Ibrahim, J. G. and Chen, M. H. (2003). A bayesian justification of Cox's partial likelihood. *Biometrika*, **90**, 629-641.
- 井桁正堯 (2010). MCMC プロシジャによるベイズ推定と ARS アルゴリズムの実装.
- 大森裕浩 (2001). マルコフ連鎖モンテカルロ法の最近の展開. *日本統計学会誌*, **31**, 305-344.