Data Augmentation with Polya-Gamma Latent Variables for Logistic Models

STAT 501: Statistical Literature and Problems

Shen Qu

Question

- Binary response regression models
- Bayesian methods
- Data augmentation
- A simple, exact algorithm

A Data-Augmentation schemes

Observed $Y_1, .., Y_n \sim Bern(p_i)$,i=1,..,n.
Covariates $X = X_1, ...X_p$
Desired $\beta = \beta_0, \beta_1, ..., \beta_p$
The link: $Pr(Y_i = 1|\beta) = H(\mathbf{x}_i^T \boldsymbol{\beta})$ or $p_i = H(z_i)$
The latent variable $\mathbf{Z} = \mathbf{X}\beta + \varepsilon$

$$y_i|z_i = \begin{cases} 1 & \text{if } z_i > 0 \\ 0 & \text{o.w.} \end{cases}$$

- In case of Probit models, $\boldsymbol{\varepsilon} \sim N_p(\mathbf{0}, \mathbf{I})$

A Gibb's sampler:

$$(1)\mathbf{z}^*|\mathbf{y}, \beta \sim N_{tr}(\mathbf{x}_i^T\beta, 1) \text{ truncated by 0 at} \begin{cases} \text{left} & y_i = 1 \\ \text{right} & y_i = 0 \end{cases}$$

$$(2)\beta|\mathbf{y}, \mathbf{z}^* \sim N_p\left(\underbrace{(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{z}^*}_{m_\beta}, \underbrace{(\mathbf{x}^T\mathbf{x})^{-1}}_{v_\beta}\right)$$

Repeat (1) and (2) long enough.

(Albert and Chib, 1993)

Because $\pi(\boldsymbol{\beta}, \boldsymbol{Z}|\boldsymbol{y}) = C\pi(\boldsymbol{\beta})\prod_{i=1}^m \left[\mathbf{1}_{Z_i>0}\mathbf{1}_{y_i=1} + \mathbf{1}_{Z_i\leq 0}\mathbf{1}_{y_i=0}\right]\phi(Z_i)$

$\pi(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{Z}) = C\pi(\boldsymbol{\beta})\prod_{i=1}^m \phi(Z_i; \mathbf{x}_i^T\boldsymbol{\beta}, 1)$

- In case of Logit models, $\varepsilon \sim Logistic_p(\mathbf{0}, \mathbf{I})$

(1) $\quad \boldsymbol{\omega_i^*}|\mathbf{y}, \beta \sim PG(n_i, |\mathbf{x}_i^T \beta|)$

(2) $\quad \beta|\mathbf{y}, \boldsymbol{\omega}^* \sim N_p\left(\mathbf{m}_\omega, \mathbf{v}_\omega\right)$

where

$\mathbf{m}_\omega = \mathbf{v}_\omega(\mathbf{x}'(\mathbf{y} - \mathbf{\frac{1}{2}}) + \mathbf{B}^{-\mathbf{1}}\mathbf{b})$,

$\mathbf{v}_\omega = (\mathbf{x}'\boldsymbol{\Omega}\mathbf{x} + \mathbf{B}^{-\mathbf{1}})^{-\mathbf{1}}$

$\boldsymbol{\Omega} = diag_n(\omega_{\mathbf{i}})$,

The prior $\beta \sim N_p(\mathbf{b}, \mathbf{B})$

Repeat (1) and (2) long enough.

(Polson etal, 2013)

How does DA algorithm work?

Generating the missing data

A well-behaved Markov chain Monte Carlo (MCMC)

A underlying variable Z simulated from the proper distribution

Motivation

Assume pdf $f_X(x) : \mathbb{R}^p \to [0, \infty)$, and estimate $E[g(x)]$.
When $E_{f_X}[g(x)] = \int_{\mathbb{R}^p} g(x) f_X(x) dx$ is hard to numerical integral or analytical approximate,

Monte Carlo Sampling

Regardless of the distribution, if we have
$g(X_1), g(X_2), \ldots, g(X_m) \overset{iid}{\sim} f_X(x)$
then $\frac{1}{m} \sum_{i=1}^{m} g(X_i)$ is a good estimator for $E(g)$.
Consistency: If $E[|g|] < \infty$, then $\frac{1}{n} \sum_{i=1}^{n} g(X_i) \xrightarrow{a.s.} E[g]$
Unbiasedness: $E[\frac{1}{S} \sum Y_k] = E(Y)$

Monte Carlo Markov chain (MCMC)

When it is impossible to simulate from $f_X(x)$, require the
conditions:

- Constructing a Markov chain, one iteration includes:

1. Draw $Y \sim f_{Y|X}(\text{ů}|x)$.
2. Draw $X_{i+1} \sim f_{X|Y}(\text{ů}|y)$.

Repeat to simulate $f_X(x)$

Tanner and Wong (1987), Swendsen and Wang (1987)

Conditions and Properties

Harris ergodic, which satisfies three properties: irreducible, aperiodic, and recurrent.

A sufficient condition for Harris ergodicity is

$$\mathcal{K} : k(x'|x) > 0 \quad \forall x', x \in \mathbf{X}$$

Definition

A Markov chain, $X = \{X_i\}_{i=0}^{\infty}$,with state space X. If the current state of the chain is $X = x$, then the density of the next state, X', is $k(x'|x)$. The Markov transition density (Mtd) is

$$k(x'|x) = \int_Y f_{X|Y}(x'|y) f_{Y|X}(y|x) dy$$

Check $k(x'|x)$ is a pdf:

$$\begin{aligned}\int_X k(x'|x)dx' &= \int_X \left[\int_Y f_{X|Y}(x'|y)f_{Y|X}(y|x)dy\right] dx' \\ &= \int_Y f_{Y|X}(y|x) \left[\int_X f_{X|Y}(x'|y)dx'\right] dy \\ &= \int_Y f_{Y|X}(y|x)dy = 1\end{aligned}$$

Invariant (stationarity)

f_X is an invariant density for K when

$$f_X(x') = \int_X k(x'|x)f_X(x)dx$$

Then the Markov chain is time homogeneous and the "recurrent" property holds. ???

Detailed balance (updat when server resume)

Symmetric $k(x'|x) = k(x|x')$

$$k(x', x) = k(x'|x)f_X(x) = \int \frac{f(x',y)f(y,x)}{}$$

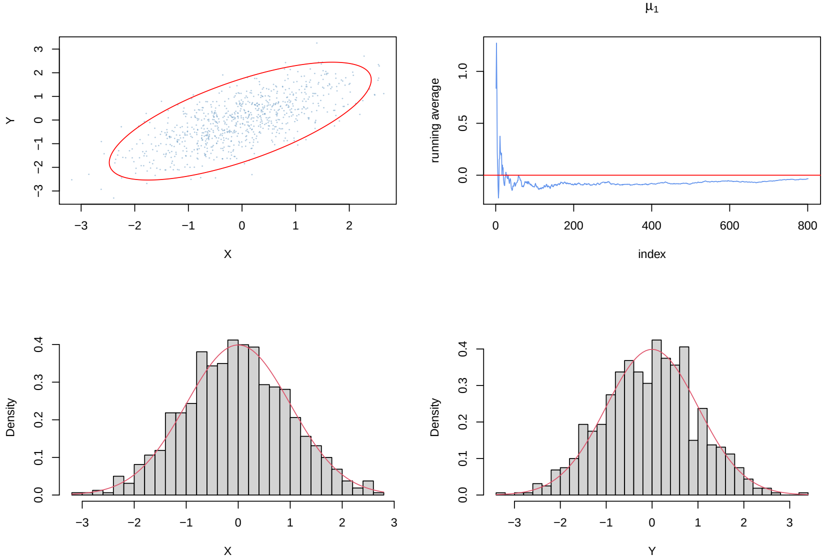
Examples

Ex1: Bivariate Normal Density

$(X \sim N(0,1);\ Y \sim N(0,1))$
$[X, Y \sim N_2(0, 1, \frac{1}{\sqrt{2}});\ f_X(x) = \int_{\mathbb{R}^q} f(x,y)dy]$

1. Draw $(Y|X = x) \sim N(\frac{x}{\sqrt{2}}, \frac{1}{2})$.
2. Draw $(X|Y = y) \sim N(\frac{y}{\sqrt{2}}, \frac{1}{2})$.

```
g.bvn<-function (n, mu1, s1, mu2, s2, rho,step=20){
  x <- 0; y <- 0;
  mat      <- matrix(ncol=2,nrow = n)
  mat[1, ] <- c(x, y)
  for (i in 2:n) {
    x <- rnorm(1, mu1+(s1/s2)*rho*(y-mu2), sqrt((1-rho^2)*s
    y <- rnorm(1, mu2+(s2/s1)*rho*(x-mu1), sqrt((1-rho^2)*s
    mat[i, ] <- c(x, y)}
  colnames(mat) <- c("X","Y");
  thinned=seq(round(n*0.2),n,step)
  mat[thinned,]
```

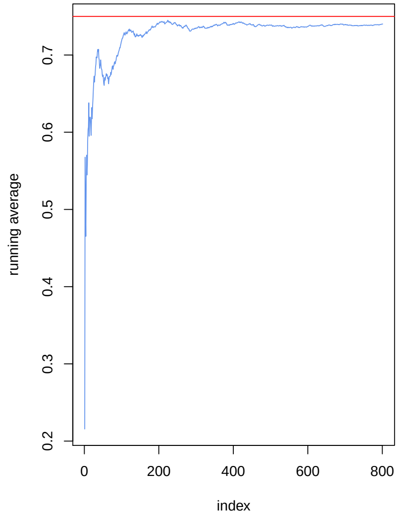


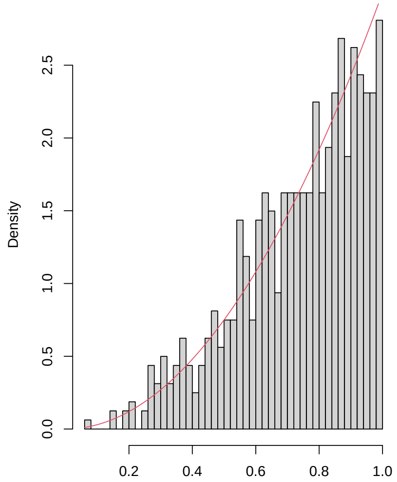
Ex2: Simple Slice Sampler (Neal, 2003)

$(f_X(x) = 3x^2 I_{(0,1)}(x);\ E[X] = \int_0^1 x f_X(x)dx = 0.75)$
$[f(x,y) = 3xI(0 < y < x < 1);\ f_X(x) = \int_0^x f(x,y)dy]$
$f_Y(y) = \int_y^1 f(x,y)dx = \frac{3}{2}(1 - y^2)$

1. Draw $(Y|X=x) \sim Unif(0,x)$ and $U \sim Unif(0,1)$.
2. Update $(X|Y=y,U=u) = \sqrt{u(1-y^2)+y^2}$.

```r
g.ex2<-function (n,step=20){
  x <- .5
  y <- .5
  X <- NA
  for (i in 1:n) {
    y <- runif(1,0,x)
    u <- runif(1,0,1)
 X[i] <- x <- sqrt(u*(1-y^2)+y^2)
      }
 thinned=seq(round(n*0.2),n,step)
 X[thinned]
}
```



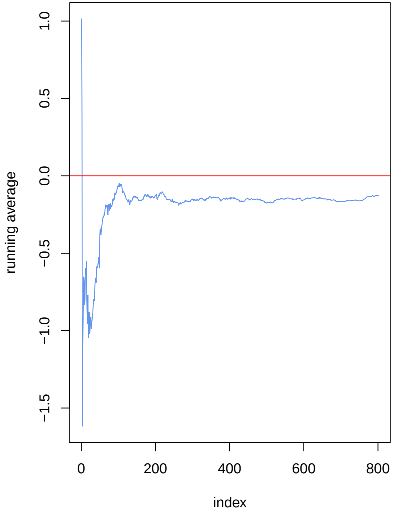


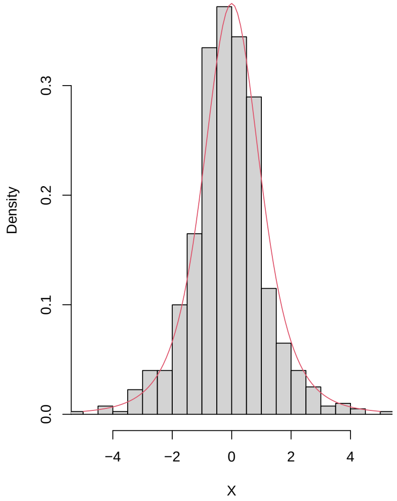
Ex3: t: Normal-Gamma

$(X \sim t_4,\ f_X(x) = \frac{3}{8}(1 + \frac{x^2}{4})^{-\frac{5}{2}})$

$[f(x,y) = \frac{4}{\sqrt{2\pi}} y^{\frac{3}{2}} \exp\{-y(\frac{x^2}{2} + 2)\} I_{(0,\infty)}(y)\ ]$

1. Draw $(Y|X = x) \sim Gamma(\frac{5}{2}, \frac{x^2}{2} + 2)$.
2. Draw $(X|Y = y) \sim N(0, y^{-1})$





Ex4: Location-scale student's t

EM algorithm (Dempster et al., 1977).

$(Z_i \sim t_{\nu=4,\mu,\sigma^2}$,i=1,..,m, $f_Z(z) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\sigma^2}\Gamma(\frac{\nu}{2})}(1 + \frac{(z-\mu)^2}{\nu\sigma^2})^{-\frac{\nu+1}{2}})$

$p(z, y|\mu, \sigma^2) =$

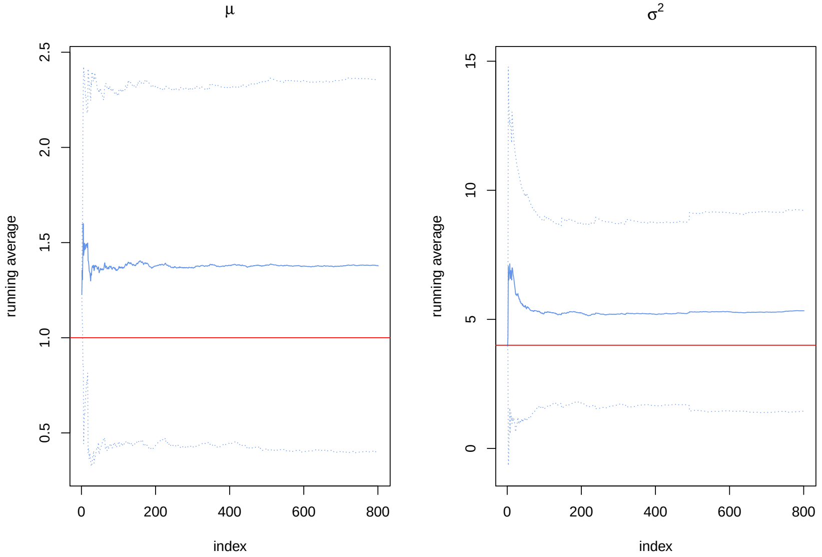
$\sqrt{y_i} \quad \Big[\quad y_i \quad \quad {}^2\Big] \; \left(\frac{\nu}{2}\right)^{\left(\frac{\nu}{2}\right)} \quad \frac{\nu}{2}-1 \quad \Big[\quad \nu \quad \Big]$

$$p((\mu, \sigma^2), y|z) \propto \pi(\mu, \sigma^2) p(z, y|\mu, \sigma^2) = \frac{1}{\sigma^2} p(z, y|\mu, \sigma^2)$$

1. Draw $(Y_i|\mu, \sigma^2, z) \sim Gamma(\frac{\nu+1}{2}, \frac{1}{2}(\frac{(z_i-\mu)^2}{\sigma^2} + \nu))$.
   $\hat{\mu} = \frac{1}{y_.} \sum_{j=1}^{m} z_j y_j$, $\hat{\sigma}^2 = \frac{1}{y_.} \sum_{j=1}^{m} y_j (z_j - \hat{\mu})^2$
2. Draw $(\sigma^2|y, z) \sim IG(\frac{m+1}{2}, \frac{y_. \hat{\sigma}^2}{2})$.
3. Draw $(\mu|\sigma^2, y, z) \sim N(\hat{\mu}, \frac{\sigma^2}{y_.})$.

A more general DA algorithm developed by Meng and van Dyk (1999)

```r
g.tls<-function (n,nu,z,step=20){
  m        <- length(z)
  y        <- rgamma(m,nu/2,nu/2)
  y.       <- sum(y)
  mu       <- sum(z*y)/y.
sigma.sq  <- sum((z-mu)^2*y)/y.
theta     <- matrix(ncol = 2, nrow = n)
theta[1,] <- c(mu,sigma.sq)
  for (i in 1:n) {
        y  <- rgamma(m,(nu+1)/2,rate = ((z-mu)^2/sigma.sq
        y. <- sum(y)
    hat.mu <- sum(z*y)/y.
hat.sigma.sq <- sum((z-hat.mu)^2*y)/y.
  sigma.sq <- 1/rgamma(1, (m+1)/2, rate=(hat.sigma.sq*y./
        mu <- rnorm(1, hat.mu, sqrt(sigma.sq/y.))
  theta[i, ] <- c(mu,sigma.sq)
  }
  thinned=seq(round(n*0.2),n,step)
  theta[thinned,]
}
```



Ex5: Probit Model

Simulate the observed data

```
beta.t <- c(-1,1/2,1/4)
```

$$\mathbf{z}^*|\mathbf{y},\beta \sim N_{tr}(\mathbf{x}_i^T\beta, 1) \text{ truncated by 0 at} \begin{cases} \text{left} & y_i = 1 \\ \text{right} & y_i = 0 \end{cases}$$

```
z.cond <- function(beta){
   ez<-(X%*%beta)
   u<-runif(m,0,1)
   z <- ez + qnorm(ifelse(Y==1,u+(1-u)*pnorm(0,ez,1),
                                  u*pnorm(0,ez,1)))
   return(z)
 }
```

$$\beta|\mathbf{y},\mathbf{z}^* \sim N_p(\underbrace{(\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{z}^*}_{m_\beta}, \underbrace{(\mathbf{x}^T\mathbf{x})^{-1}}_{V_\beta})$$

```
beta.cond = function(z,V,cholV){
   beta <- V%*%(t(X)%*%z)+cholV%*%rnorm(p)
  return(beta)
}
```

- Gibbs' sampler

```
g.probit<-function (N,X,Y,m,p,step=20){
iXX<-chol2inv(chol(t(X)%*%X)); V<-iXX*(m/(m+1)); cholV<-cho

beta.ini <- runif(p,-3,3)
   Z     <- matrix(NA,N,m)
   z.ini <- z.cond(beta.ini)
Beta     <- matrix(NA,nrow=(N+1),ncol=p)
Beta[1,] <- beta.ini <- beta.cond(z.ini,V,cholV)

for(i in 1:N){
   Z[i,]      <- z.cond(Beta[i,])
Beta[(i+1),] <- beta.cond(Z[i,],V,cholV)
}
thinned=seq(round(N*0.2),N,step)
Beta[thinned,]
}
```

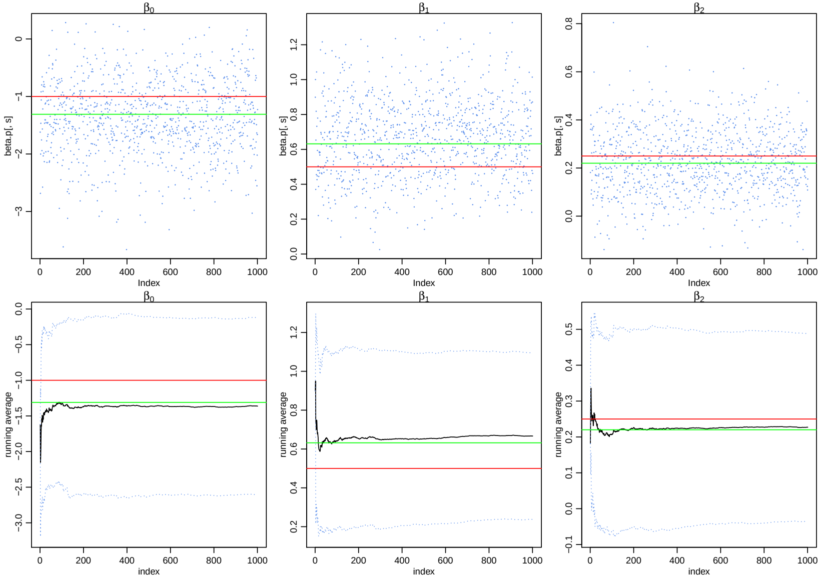
- Gibbs' results

	beta.p.mean	beta.p.median	beta.p.ll	beta.p.ul	beta.p.sd
Beta0	-1.362	-1.381	-2.655	-0.1813	0.6337
Beta1	0.6669	0.6714	0.2524	1.13	0.2191
Beta2	0.227	0.2199	-0.0332	0.483	0.1336

Iteratively Reweighted Least Squares (IRLS) Algorithm

Fisher Scoring??? Newton-Raphson algorithm "gradient descent level II"???

	glm.p	glm.p.sd	2.5 %	97.5 %	glm.p.marg	2.5 %	97.5
Beta0	-1.31	0.5669	-2.503	-0.2209	-0.438	-0.1282	-0.0
Beta1	0.6318	0.2254	0.2095	1.092	0.2112	0.0107	0.05
Beta2	0.22	0.2117	-0.1916	0.6492	0.0735	-0.0098	0.0



Logit Model

$Y_1, .., Y_n \sim Bern(p_i)$,i=1,..,n;

$P_i(Y = 1) = \frac{1}{1+\exp(-\mathbf{x}_i^T \beta)}$;

Derive the full conditional pdf

$\pi(\omega|\beta, y)$

$\pi(\beta|\omega, y)$

(update when the server resume)

$\boldsymbol{\omega}_i^*|\mathbf{y},\beta \sim PG(n_i, |\mathbf{x}_i^T\beta|)$

```
w.cond <- function(beta,m){
   w <- rpg.devroye(num=m, h=1, z=abs(X%*%beta))
   return(w)
 }
```

$\beta|\mathbf{y},\boldsymbol{\omega}^* \sim N_p\left(\mathbf{m}_\omega, \mathbf{v}_\omega\right)$

```
beta.cond = function(y,w,p,varbeta=100){
   matbetapr= diag(rep(1/varbeta,p))
   OMG       = diag(w);
   OMGinv    = diag(1/w)
   eta       = OMGinv%*%(y-0.5)
   varmat    = chol2inv(chol(matbetapr+t(X)%*%OMG%*%X))
   meanvec   = varmat%*%(t(X)%*%OMG%*%eta)
   beta <- mvtnorm::rmvnorm(1, mean = meanvec, sigma = varm
  return(beta)
 }
```

- Gibbs' sampler

```r
g.logit<-function (N,X,Y,step=20){
  p <- dim(X)[2]; m <- dim(X)[1]
  beta.ini <- runif(p,-3,3)
  W        <- matrix(NA,nrow=N,ncol=m)
  w.ini    <- w.cond(beta.ini,m)
  Beta     <- matrix(NA,nrow=(N+1),ncol=p)
  Beta[1,] <- beta.ini <- beta.cond(Y,w.ini,p)
  for(i in 1:N){
    W[i,] = w.cond(Beta[i,],m)
    Beta[(i+1),]=beta.cond(Y,W[i,],p)
  }
  thinned=seq(round(N*0.2),N,step)
  Beta[thinned,]
}
```

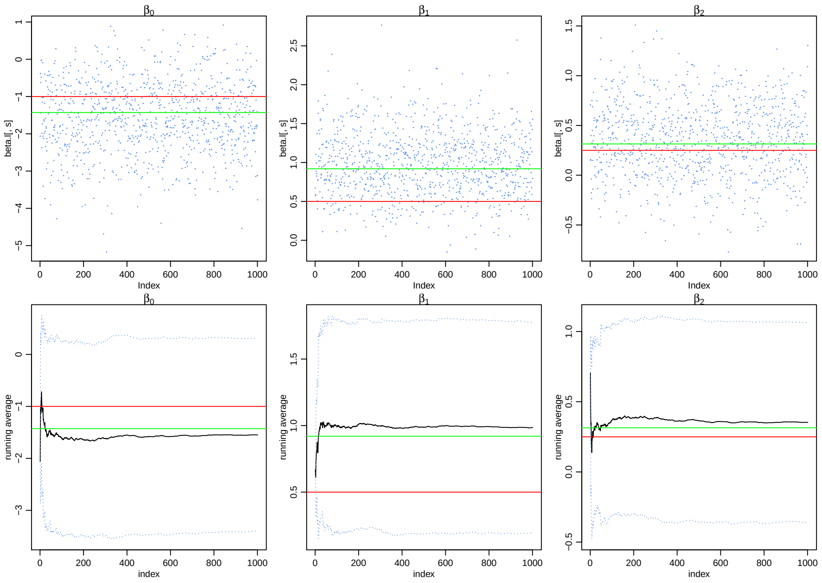
- Gibbs' results

	beta.l.mean	beta.l.median	beta.l.ll	beta.l.ul	beta.l.sd
Beta0	-1.551	-1.567	-3.406	0.2414	0.9464
Beta1	0.9853	0.967	0.2874	1.791	0.4042
Beta2	0.3539	0.3466	-0.3787	1.034	0.364

Iteratively Reweighted Least Squares (IRLS) Algorithm

```
fit.l.glm <- glm(Y~.,X.df,family=binomial(link="logit"))
glm.l<- coef(fit.l.glm)
glm.l.sd <- summary(fit.l.glm)$coef[,2] # sqrt(diag(vcov(f
glm.l.ci <- confint(fit.l.glm)
# glm.l.marg <-mean(dlogis(X %*% glm.l))*glm.l
par.glm.l<- cbind(glm.l,glm.l.sd,glm.l.ci)
rownames(par.glm.l) <- beta.name
pander(round((par.glm.l),4))
```

	glm.l	glm.l.sd	2.5 %	97.5 %
Beta0	-1.428	0.8932	-3.301	0.2633
Beta1	0.92	0.3796	0.231	1.744
Beta2	0.3145	0.3392	-0.3383	1.02



β0
β1
β2
beta.l[, s]
Index
running average
index

Means

	irls.p	gibbs.p	irls.l	gibbs.l
Beta0	-1.31	-1.362	-1.428	-1.551
Beta1	0.6318	0.6669	0.92	0.9853
Beta2	0.22	0.227	0.3145	0.3539

Standard deviations

	irls.p	gibbs.p	irls.l	gibbs.l
Beta0	0.5669	0.6337	0.8932	0.9464
Beta1	0.2254	0.2191	0.3796	0.4042
Beta2	0.2117	0.1336	0.3392	0.364

literature

Jun S. Liu & Ying Nian Wu (1999) Parameter Expansion for Data Augmentation, Journal of the American Statistical Association, 94:448, 1264-1274, DOI: 10.1080/01621459.1999.10473879

- Gelman, A. (2014). Bayesian data analysis (Third edition.). CRC Press.

11.7 Bibliographic note

Tanner and Wong (1987) introduced the idea of iterative simulation to many statisticians, using the special case of 'data augmentation' to emphasize the analogy to the EM algorithm (see Section 13.4).

Auxiliary variables

12.1 Efficient Gibbs samplers

Gibbs sampler computations can often be simplified or convergence accelerated by adding auxiliary variables, for example indicators for mixture distributions, as described in Chapter 22. The idea of adding variables is also called data augmentation and is often a useful conceptual and computational tool, both for the Gibbs sampler and for the EM algorithm (see Section 13.4).

12.7 Bibliographic note

For the relatively simple ways of improving simulation algorithms

Imai, K., and van Dyk, D. A. (2005). A Bayesian analysis of the multinomial probit model using marginal data augmentation. Journal of Econometrics. 124, 311–334. https://doi.org/10.1016/j.jeconom.2004.02.002

Rubin, D. B. (1987b). A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR algorithm. Discussion of Tanner and Wong (1987). Journal of the American Statistical Association 82, 543–546.

Tanner, M. A., and Wong, W. H. (1987). The calculation of posterior distributions by data augmentation (with discussion). Journal of the American Statistical Association 82, 528–550.

van Dyk, D. A., and Meng, X. L. (2001). The art of data augmentation (with discussion). Journal of Computational and Graphical Statistics 10, 1–111.