# STAT3016/4116/7016
## Introduction to Bayesian Data Analysis

RSFAS, College of Business and Economics, ANU

Latent variable methods for ordinal data

# Multinomial Regression - Ordered Data

Example: Suppose we are interested in modelling the relationship between educational attainment and the number of children of individuals in a population. Additionally we believe an individual's educational attainment may be influenced by the education level of their parent.

$$Pr(DEG_i = j) = \beta_1 + \beta_2 CHILD_i + \beta_3 PDEG_i + \beta_4 CHILD_i \times PDEG_i$$

Suppose DEG is coded as 1: no degreee; 2: high school degree; 3: grad diploma; 4: bachelor degree; 5: graduate degree.
There is a natural ordering to the levels of DEG which we want to allow for in our model. (note DEG is ordinal but not numeric)

# Bayesian Probit Model for Ordered Data

Idea: We can think of ordinal non-numeric variables as arising from some underlying numeric process.

$$\epsilon_i \stackrel{\text{iid}}{\sim} N(0,1)$$

$$Z_i = x_i^T \boldsymbol{\beta} + \epsilon_i$$

$$Y_i = g(Z_i)$$

$z_i$ is a latent variable; the function $g$ is taken to be non-decreasing. (Note: we don't need any intercept term or additional scale parameter $\sigma^2$ because such information can be represented by $g$).

Let the sample space of $Y$ taken on J discrete values, then

$$y = g(z) = \begin{cases} = 1 & \text{if } -\infty = g_0 < z < g_1 \\ = 2 & \text{if } g_1 < z < g_2 \\ \vdots & \vdots \\ = J & \text{if } g_{J-1} < z < g_J = \infty \end{cases}$$

# Bayesian Probit Model for Ordered Data

$g_1, ..., g_{J-1}$ are like "thresholds", so that moving $z$ past a threshold moves $y$ into the next highest category.

The unknown parameters in the model are $\{\boldsymbol{\beta}, g_1, ..., g_{J-1}, Z_1, ...., Z_n\}$.

If we specify normal prior distributions, then the joint posterior distribution $p(\{\boldsymbol{\beta}, g_1, ..., g_{J-1}, Z_1, ...., Z_n\} \,|\mathbf{Y})$ can be approximated using a Gibbs sampler.

# Bayesian Probit Model for Ordered Data

**Full conditional distribution of $\boldsymbol{\beta}$.**

$$p(\boldsymbol{\beta}) \propto p(\boldsymbol{\beta})p(\mathbf{z}|\boldsymbol{\beta})$$

As per ordinary regression, a MVN prior for $\boldsymbol{\beta}$ gives a MVN posterior distribution for $\boldsymbol{\beta}$. For example, let $\boldsymbol{\beta} \sim MVN(\mathbf{0}, n(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1})$. Then

$$Var[\beta|\mathbf{z}] = \frac{n}{n+1}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}$$

$$E[\beta|\mathbf{z}] = \frac{n}{n+1}(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{z}$$

# Bayesian Probit Model for Ordered Data

**Full conditional distribution of Z.**

$$p(z_i|\boldsymbol{\beta}, \mathbf{y}, \mathbf{g}) \propto \mathrm{dnorm}(\mathbf{x_i}^T\boldsymbol{\beta}, 1) \times \delta_{(a,b)}(z_i)$$

This is the density of a constrained normal distribution. To sample a value z from a normal $(\mu, \sigma^2)$ constrained on the interval $(a, b)$, (where $a = g_{y_i - 1}$ and $b = g_{y_i}$ for observation $Y_i = y_i$) we perform the following two steps

1. sample $u \sim \mathrm{uniform}(\Phi[(a-\mu)/\sigma], \Phi[(b-\mu)/\sigma])$
2. set $z = \mu + \sigma\Phi^{-1}(u)$

where $\Phi$ and $\Phi^{-1}$ are the cdf and inverse-cdf of the standard normal distribution. (nb: make sure your code can handle cases $g_0 = -\infty$ and $g_J = \infty$).

# Bayesian Probit Model for Ordered Data

**Full conditional distribution of $\mathbf{g}$.**

- Let $p(\mathbf{g})$ be the prior on $g$.
- What are the restrictions on $g_k$ given $\mathbf{Y}, \mathbf{Z}$?
- We can show that if $p(\mathbf{g})$ is proportional to the product $\prod_{k=1}^{k-1} \mathrm{dnorm}(\mu_k, \sigma_k^2)$ but constrained so that $g_1 < ...., < g_{K-1}$, then the full conditional density of $g_k$ is normal$(\mu_k, \sigma_k^2)$ constrained to the interval $(a_k, b_k)$, where $a_k = \max\{z_i : y_i = k\}$ and $b_k = \min\{z_i : y_i = k+1\}$

# Bayesian Probit Model for Ordered Data - Example

```
X<-cbind(ychild,ypdeg,ychild*ypdeg)
y<-ydegr
#remove missing values
keep<- (1:length(y))[ !is.na( apply( cbind(X,y),1,mean) ) ]
X<-X[keep,] ; y<-y[keep]
ranks<-match(y,sort(unique(y))) ; uranks<-sort(unique(ranks))
n<-dim(X)[1] ; p<-dim(X)[2]
iXX<-solve(t(X)%*%X)  ; V<-iXX*(n/(n+1)) ; cholV<-chol(V)

###starting values
set.seed(1)
beta<-rep(0,p)
z<-qnorm(rank(y,ties.method="random")/(n+1))
g<-rep(NA,length(uranks)-1)
K<-length(uranks)
BETA<-matrix(NA,1000,p) ; Z<-matrix(NA,1000,n) ; ac<-0
mu<-rep(0,K-1) ; sigma<-rep(100,K-1)
S<-25000
```

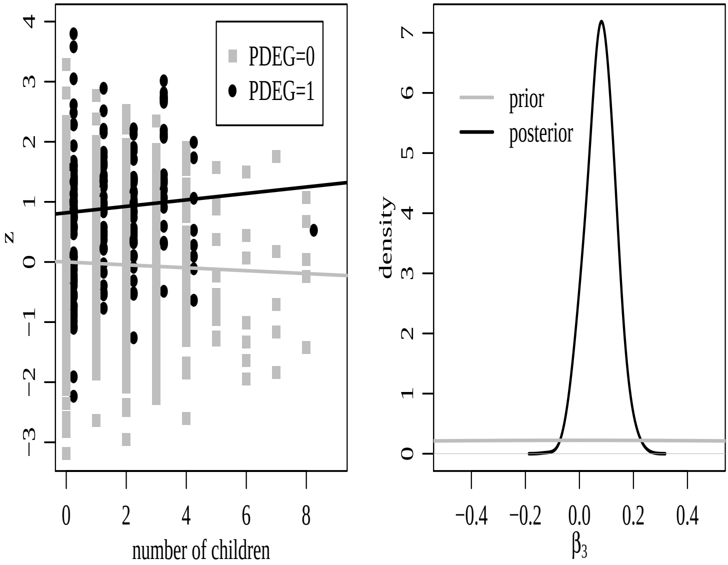
# Bayesian Probit Model for Ordered Data - Example

```
for(s in 1:S)
{

  #update g
  for(k in 1:(K-1))
  {
  a<-max(z[y==k])
  b<-min(z[y==k+1])
  u<-runif(1, pnorm( (a-mu[k])/sigma[k] ),
              pnorm( (b-mu[k])/sigma[k] ) )
  g[k]<- mu[k] + sigma[k]*qnorm(u)
  }

  #update beta
  E<- V%*%( t(X)%*%z )
  beta<- cholV%*%rnorm(p) + E
  #update z
  ez<-X%*%beta
  a<-c(-Inf,g)[ match( y-1, 0:K) ]
  b<-c(g,Inf)[y]
  u<-runif(n, pnorm(a-ez),pnorm(b-ez) )
  z<- ez + qnorm(u)
```

# Bayesian Probit Model for Ordered Data - Example

```
    #help mixing
    c<-rnorm(1,0,n^(-1/3))
    zp<-z+c ; gp<-g+c
    lhr<-  sum(dnorm(zp,ez,1,log=T) - dnorm(z,ez,1,log=T) ) +
           sum(dnorm(gp,mu,sigma,log=T) - dnorm(g,mu,sigma,log=T) )
    if(log(runif(1))<lhr) { z<-zp ; g<-gp ; ac<-ac+1 }
    if(s%%(S/1000)==0)
    {
      cat(s/S,ac/s,"\n")
      BETA[s/(S/1000),]<-  beta
      Z[s/(S/1000),]<- z
    }
  }
> beta.pm<-apply(BETA,2,mean)
> beta.pm
[1] -0.02414342  0.81800047  0.07785450
> apply(BETA,2,function(x) quantile(x,prob=c(.025,.5,.975)))
             [,1]      [,2]        [,3]
2.5%  -0.08838210 0.5810692 -0.02613790
50%   -0.02500787 0.8161399  0.07932693
97.5%  0.03827794 1.0509594  0.17843513
```

# Bayesian Probit Model for Ordered Data - Example



# Bayesian Probit Model for Ordered Data - Example

Use `MCMCoprobit` in the library `MCMCpack`

```
>deg.mcmc<-MCMCoprobit(y~X,mcmc=25000)
>summary(deg.mcmc)
......
2. Quantiles for each variable:

                2.5%      25%      50%       75%   97.5%
(Intercept)  1.04077  1.13067  1.17543  1.221405 1.30887
Xychild     -0.07214 -0.03983 -0.02275 -0.005566 0.02632
Xypdeg       0.58479  0.73780  0.81858  0.898584 1.04800
X           -0.05028  0.03102  0.07473  0.118567 0.20080
gamma2       1.54569  1.62999  1.66792  1.706733 1.77033
gamma3       1.74816  1.83527  1.87333  1.912921 1.97828
```

(nb: see other useful packages in `MCMCpack` , eg `MCMCregress` ; `MCMClogit` ; `MCMCpoisson`; `MCMCmnl` )

# Multinomial Regression for unordered data

- $\mathbf{y}_i \sim \mathrm{Multinomial}(n_i; \theta_{i1}, ...., \theta_{iJ})$ (the response can take on one of $J > 2$ unordered categories)
- Specifically, we want to model the probability of being in level $j$ as a function of some other covariates $\mathbf{x}_i$

Logit link:

$$\theta_{ij} = \frac{\exp(\mathbf{x_i}\boldsymbol{\beta}_j)}{\sum_{j=1}^{J} \exp(\mathbf{x_i}\boldsymbol{\beta}_j)}$$

$\boldsymbol{\beta}_J = 0$ for identifiability (arbitrary base line category $J$)

How do we interpret the $\boldsymbol{\beta}_j$??

Assume multivariate normal prior on $\boldsymbol{\beta}$.

# Multinomial Regression for unordered data

Example: The `hsb` data set was collected as a subset of the High School and Beyond study conducted by the National Education Longitudinal Studies program of the National Centre for Education Statistics. The variables are gender, race, socioeconomic status, school type, chosen high school program type, scores on reading, writing, maths, science and social studies. We want to determine which factors are related to the choice of the type of program - academic, vocational or general - that the students pursue in high school. The response is multinomial with three levels.

# HSB Example

```
hsb[1:10,]
 id gender         race    ses schtyp     prog read write math science socst
 70   male        white    low public  general   57    52   41      47    57
121 female        white middle public vocation   68    59   53      63    61
 86   male        white   high public  general   44    33   54      58    31
141   male        white   high public vocation   63    44   47      53    56
172   male        white middle public academic   47    52   57      53    61
113   male        white middle public academic   44    52   51      63    61
 50   male african-amer middle public  general   50    59   42      53    61
 11   male     hispanic middle public academic   34    46   45      39    36
 84   male        white middle public  general   63    57   54      58    51
 48   male african-amer middle public academic   57    55   52      50    51
```

# Multinomial Regression for unordered data - HSB example

```
#Maximum likelihood estimation
library(nnet)
library(MCMCpack)
library(mvtnorm)
mmod<-multinom(prog~g.m+r.aa+r.hisp+r.asian+ses.low+ses.high+
            schtyp.priv+read+write+math+science+socst,Hess=TRUE)
summary(mmod)
...
Coefficients:
         (Intercept)     g.m   r.aa r.hisp r.asian ses.low ses.high schtyp.priv
general         5.22 -0.0926 -0.297 -0.929    1.06   0.396   -0.703      -0.585
vocation       11.05 -0.3210 -0.336 -0.536   -1.04  -1.134   -1.182      -2.055

Std. Errors:
         (Intercept)   g.m  r.aa r.hisp r.asian ses.low ses.high schtyp.priv
general         1.80 0.455 0.735  0.733   0.827   0.535    0.505       0.564 0.
vocation        2.03 0.502 0.748  0.664   1.326   0.592    0.570       0.835 0.
```

# Multinomial Regression for unordered data - HSB example

```
#starting values and fit
b2<-summary(mmod)$coefficients[1,]+rnorm(k,0,1)
b3<-summary(mmod)$coefficients[2,]+rnorm(k,0,1)

eta2<-X%*%b2
eta3<-X%*%b3
expeta2<-exp(eta2)
expeta3<-exp(eta3)
p1<-1/(1+expeta2+expeta3)
p2<-p1*expeta2
p3<-p1*expeta3

p<-cbind(p1,p2,p3)
k<-length(b2)
# Priors
mu0<-rep(0,k)
Tau0<-diag(.01,k)
Sigma0<-solve(Tau0)

cov2<-cov3<-diag(0.8,k,k) #proposal covariance
tune<-diag(0.004,k,k)
cov2<-tune%*%solve(Tau0+solve(cov2))%*%tune
cov3<-tune%*%solve(Tau0+solve(cov3))%*%tune
```

# Multinomial Regression for unordered data - HSB example

```
for (i in 1:nsim) {
 # Calculate Likelihood based on old value of b2 and b3
  eta2<-X%*%b2
  eta3<-X%*%b3
  p1<-1/(1+exp(eta2)+exp(eta3))
  p2<-p1*exp(eta2)
  p3<-p1*exp(eta3)
  lold<-sum(log(p1)*(yy==0)+log(p2)*(yy==1)+log(p3)*(yy==2))

 # Draw Candidates
  b2new<-b2 + rmvnorm(1,rep(0,k),cov2)
  b3new<-b3 +rmvnorm(1,rep(0,k),cov3)
  eta2new<-X%*%c(b2new)
  eta3new<-X%*%c(b3new)
  p1new<-1/(1+exp(eta2new)+exp(eta3new))
  p2new<-p1new*exp(eta2new)
  p3new<-p1new*exp(eta3new)
  lnew<-sum(log(p1new)*(yy==0)+log(p2new)*(yy==1)+log(p3new)*(yy==2))
```

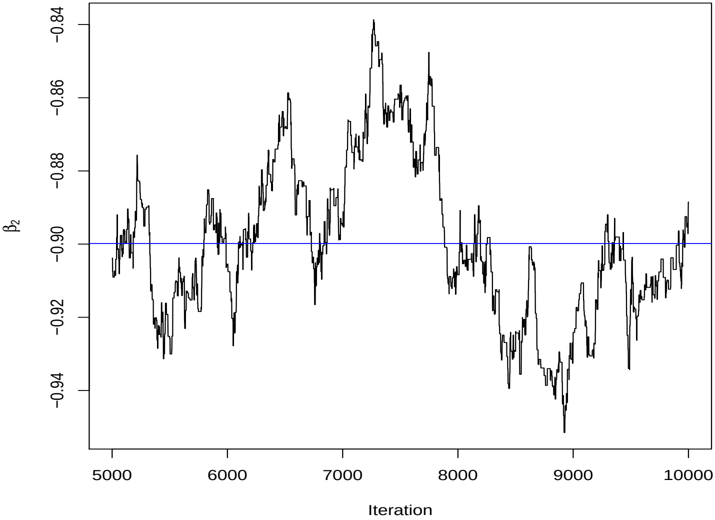
# Multinomial Regression for unordered data - HSB example

```
 # Acceptance prob on log scale
   r<-lnew+dmvnorm(b2new,mu0,Sigma0,log=T)+dmvnorm(b3new,mu0,Sigma0,log=T)-
(lold+dmvnorm(b2,mu0,Sigma0,log=T)+dmvnorm(b3,mu0,Sigma0,log=T))
   if(log(runif(1))<r){
    b2<-c(b2new)
    b3<-c(b3new)
  if (i>5000) A<-A+1
    }

 # Store Results
   Beta2[i,]<-b2
   Beta3[i,]<-b3
  if (i%%100==0) print(c(i,lnew,lold))

} # End MCMC
```

# Multinomial Regression for unordered data - HSB example

```
> beta2.mean
(Intercept)         g.m        r.aa      r.hisp     r.asian     ses.low
     1.8752     -0.8998     -4.1229      1.0159      8.7586      2.2900
> beta3.mean
(Intercept)         g.m        r.aa      r.hisp     r.asian     ses.low
    18.5915     -0.2272     -3.0765      0.4052      4.4434     -0.9036
> A/(nsim-5000)
[1] 0.199
```

# Multinomial Regression for unordered data - HSB example



# Multinomial Regression for unordered data - HSB example

```
mcmc.model<-MCMCmnl(prog~g.m+r.aa+r.hisp+r.asian+ses.low+ses.high+schtyp.priv+
      read+write+math+science+socst,baseline="academic",data=hsb,
         burnin=1000,mcmc=10000, thin=1, tune=1)
>  summary(mcmc.model)

..................
```

# Rank Likelihood and the Gaussian Copula Model

Suppose we have a collection of non-numeric ordinal variables and we want to understand the relationships between all the variables. The multivariate normal model is inappropriate because the variables are not measured on a meaningful numerical scale..

Let's extend the ordered probit model for univariate ordinal data to a latent multivariate normal model that can handle both numeric and non-numeric ordinal data.

Let $\mathbf{Y}_1, ..., \mathbf{Y}_p$ be i.i.d random samples from a $p$-variate population. The latent normal model is:

$$\mathbf{Z}_1, ..., \mathbf{Z}_p \stackrel{\text{iid}}{\sim} MVN(\mathbf{0}, \Psi)$$

$$Y_{i,j} = g_j(Z_{i,j})$$

$\Psi$ is a correlation matrix with diagonal elements $= 1$ (represent joint dependencies)

$g_1, ..., g_p$ are non-decreasing functions (represent marginal densities)

# Rank Likelihood and the Gaussian Copula Model

We can show that $F_j(y) = \Phi(g_j^{-1}(y))$ and so the marginal distributions of the $Y_j$'s are fully determined by the $g_j$'s and do not depend on the correlation matrix $\Psi$.

This is a **copula model** - a model with separate parameters for the univariate marginal distributions and the multivariate dependencies.

We will be using the multivariate normal copula model.

If we are primarily interested in dependencies among variables, then $g_1, ..., g_p$ are nuisance parameters.

Base inference on the *rank-likelihood*, then we only need a prior on $\Psi$.

# Rank Likelihood and the Gaussian Copula Model

We know that:

$$R(\mathbf{Y}) = \{\mathbf{Z} : z_{i_1,j} < z_{i_2,j} \text{ if } y_{i_1,j} < y_{i_2,j}\}$$

The probability of this event, $Pr(\mathbf{Z} \in R(\mathbf{Y})|\Psi)$ does not depend on $g_1, ..., g_p$. $Pr(\mathbf{Z} \in R(\mathbf{Y})|\Psi)$ is called the *rank-likelihood*. We will make an MCMC approximation to $p(\Psi, \mathbf{Z}|\mathbf{Z} \in R(\mathbf{Y}))$.

The correlation matrix $\Psi$ requires its diagonal elements to be 1 for which we do not have a conjugate class of prior distributions. Lets rewrite our model.

# Rank Likelihood and the Gaussian Copula Model

$$\Sigma \sim \mathrm{inverseWishart}(\nu_0, S_0^{-1})$$

$$\Psi = h(\Sigma) \text{ where } h(\Sigma) = \sigma_{i,j}/\sqrt{\sigma_i^2 \sigma_j^2}$$

$$\mathbf{Z}_1, ..., \mathbf{Z}_n \overset{\text{iid}}{\sim} MVN(\mathbf{0}, \Psi)$$

$$Y_{i,j} = g_j(Z_{i,j})$$

We can show that:

$$\Sigma | \mathbf{Z} \sim \mathrm{InverseWishart}(\nu_0 + n, [S_0 + \mathbf{Z}^T \mathbf{Z}]^{-1})$$

and the posterior distribution of $Z_{ij}$ is a constrained normal where
$E[Z_j | \Sigma, z_{-j}] = \Sigma_{j,-j}(\Sigma_{-j,-j})^{-1} z_{-j}$
$Var[Z_j | \Sigma, z_{-j}] = \Sigma_{j,j} \Sigma_{j,-j}(\Sigma_{-j,-j})^{-1} \Sigma_{-j,j}$ and the constraints are
(a,b) where $a = max\{z_{k,j} : y_{k,j} < y_{i,j}\}$ and
$b = min\{z_{k,j} : y_{i,j} < y_{k,j}\}$

# Rank Likelihood and the Gaussian Copula Model - Example

The package `sbgcop` estimates the parameters of a Gaussian copula. It also provides a semiparametric imputation procedure for missing multivariate data.
Example: social mobility data (from ordinal probit model example). Let's examine the relationship between DEG, CHILD, INC, PDEG, PCHILD, PINC and AGE.

```
X<-cbind(ychild,ypdeg,yincc,ypchild,ydegr,ypincc,yage)
fit<-sbgcop.mcmc(X)
summary(fit)
plot(fit)
```

# Rank Likelihood and the Gaussian Copula Model - Example