Department of Mathematics and Statistics
WinBUGS and OpenBUGS applications
Exam 17.5.2011

(5 questions, 30 points total. You may write in English/Finnish)

1. Explain the meaning of the following terms (1 point each):
(A) HPD interval, (B) inverse cdf method, (C) censored observation,
(D) zero inflated Poisson model, (E) exchangeability, (F) unidentifiable parameters.

(A) Highest Posterior Density interval is shortest possible interval that contains a given probability (e.g. 95%) of a posterior distribution. (In multimodal densities it can be a set of disjoint intervals).
(B) If the cumulative probability function $y = F(x)$ can be inverted analytically, $x = F^{-1}(y)$, then we can draw a value y from uniform density $U(0,1)$ and calculate $x = F^{-1}(y)$. The resulting random values X will have cumulative probability F. Proof: $P(F^{-1}(Y) < x) = P(Y < F(x)) = F(x)$.
(C) An observation that is only known to be either larger or smaller than some value, or within some interval. Hence the likelihood contribution will be of the form $P(X > a \mid \theta)$, or $P(X < b \mid \theta)$, or $P(a < X < b \mid \theta)$, instead of $P(X = x \mid \theta)$.
(D) To model count data with an excess amount of zeros, a zero inflated Poisson model may be used. This is a mixture distribution with a probability α of point mass at zero, and with probability $1 - \alpha$ the values have Poisson distribution. Effectively: $Y \mid U \sim \text{Poisson}(\lambda(1-U))$ $, U \sim \text{Bernoulli}(\alpha)$
(E) A sequence of variables is said to be exchangeable, if the permutation of the variable indices does not affect the probability statement. e.g. finite exchangeability is: $P(x_1, \ldots, x_n) = P(x_{r_1}, \ldots, x_{r_n})$ for all permutations r.
(F) Unidentifiable parameters occur when the likelihood function takes the same value for different parameter values: $\pi(X \mid \psi) = \pi(X \mid \psi')$ for some $\psi \neq \psi'$. In a Bayesian analysis, identifiability could be obtained by suitably informative priors, though, but the parameters would not be identifiable from the data alone.

2. (3+3 points):
Explain bayesian predictive model fit diagnostics. Give an example.
Bayesian predictive model fit diagnostics is based on the posterior predictive distribution: $\pi(x_{\text{pred}} \mid x_{\text{obs}}) = \int \pi(x_{\text{pred}} \mid \theta)\pi(\theta \mid x_{\text{obs}})\mathbf{d}\theta$. Various discrepancy functions can then be defined, calculated for the actual observations x_{obs} and the predicted x_{obs}, and the functions may also depend on the unknown parameters θ. Posterior probabilities for these discrepancies can be used for model fit assessment. Also, cross validation technique may be used so that the above probabilities are conditional on some subset of observations, e.g. leaving one out, and then computing the prediction for that. The prediction could then be compared to the actual value, but the prediction was not based on knowing that value. However, in large data sets, this is practically the same as using the whole data set for computing prediction. For example, with simple normal model with unknown mean (flat prior) and assumed variance, we could study the predictive distribution of the smallest data point. This is easily obtained from the simulations by generating the whole data set many times (iterations) and each time (iteration) recording the smallest of the n generated points. Assume the original data has 10 observations, with mean $\bar{x}$. The smallest of these is x_{min}, and:

$$\pi(\mu \mid X, \sigma) = \text{N}(\bar{x}, \sigma^2/n) = \text{N}(\bar{x}, \sigma^2/10)$$

$$\pi(X^* \mid X, \sigma) = \mathrm{N}(\bar{x}, \sigma^2 + \sigma^2/n) = \mathrm{N}(\bar{x}, \sigma^2 11/10).$$

In this case, these are easy to simulate with R by simply drawing from normal densities. The posterior predictive distribution of the smallest value among 10, assuming different σ values, is simulated as

```
sigma<-0.5
for(i in 1:10000){xmin[i]<-min(rnorm(10,mean(x),sigma*sqrt(11/10)))}
```

3. (6 points):
Explain the model behind the following BUGS implementation:

```
model{
for(i in 1:15){ # group
for(j in 1:26){ # individual
second[i,j] ~ dbern(p[i,j])
p[i,j] <- equals(first[i,j],0)*P[1,2]+equals(first[i,j],1)*P[2,2]
first[i,j] ~ dbern(q[i])
}
logit(q[i]) <- equals(organic[i],1)*qq[1]
 + equals(small[i],1)*qq[2] + equals(big[i],1)*qq[3] +e[i]
 e[i] ~ dnorm(0,tau)
}
for(k in 1:3){
qq[k] ~ dnorm(0,0.001);
Q[k]<-exp(qq[k])/(1+exp(qq[k]))
}
P[1,1] <- 1-P[1,2]; P[2,1]<-P[2,2]
P[1,2] ~ dunif(0,1); P[2,2] ~ dunif(0,1)
tau ~ dgamma(0.01,0.01)
}
list(qq=c(0,0,0),tau=1) # inits
list(organic=c(1,0,0,1,0,0,1,0,0,1,0,0,1,0,0),
small=c(0,1,0,0,1,0,0,1,0,0,1,0,0,1,0),
big=c(0,0,1,0,0,1,0,0,1,0,0,1,0,0,1),
first=structure(.Data=c(1,1,1,0,1,1,0,0,0,
1,0,0,0,1,1,0,1,1,1,1,1,0,1,0,0,NA,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,NA,.......rest is not shown),.Dim=c(15,26)),
second=structure(.Data=c(0,0,0,1,1,1,1,0,0,0,0,1,NA,0,1,1,0,1,0,0,
0,0,0,0,1,NA,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,NA,
NA,1,NA,0,NA,1,NA,1,0,0,.......rest is not shown),.Dim=c(15,26)) )
```

The model is Bernoulli Markov chain where the second observation depends on the first observation. In other words, there is a transition probability p_{ij} for individual j in group i. The model for the first value is Bernoulli with logit-link function, depending on the type of the group i and group specific random effect e. Some of the data points are missing and they will be predicted from the posterior distribution.

4. (6 points):
Write a BUGS model code for the following posterior distribution:

$$\pi(\mu, \sigma, \alpha_1, \ldots, \alpha_D \mid X_1, \ldots, X_D) \propto \prod_{d=1}^{D} \pi(X_d \mid N, \mu, \alpha_d)\pi(\alpha_d \mid \sigma)\pi(\sigma)\pi(\mu)$$

where X_d is number of positive results from N tests done on day d ($d = 1, \ldots, D = 20$), and each of the N tests is composed of m individual samples pooled together. Hence: $X_d \sim \text{Bin}(N, 1-(1-p_d)^m)$. Prevalence of individual samples is p_d with some average level and day-to-day variation described by random effect: $\text{logit}(p_d) = \mu + \alpha_d$, $\alpha_d \sim \text{N}(0, \sigma^2)$, and uninformative hyper priors for μ and σ. Include code for reporting posterior results for $p_0 = \text{logit}^{-1}(\mu)$ and $q_0 = 1-(1-p_0)^m$, and X^* for a predicted new day. N and m are assumed fixed.

```
model{
# Model with heterogeneity between days
# (day-to-day variation of prevalence )
# Each pool (of e.g. animals) in day d has
# probability p[d] for an animal in the pool j of day d
# to be positive, and logit(p[j,d]) = mu + alpha[d]
# Assume there are subN subsamples in a pool, and N pools per day, D days.
# The outcome of each pooled test for each day is the outcome variable
# (N*D observations, each 0 or 1).
mu ~  dnorm(0,0.0001);
tau <- pow(sigma,-2); sigma ~dunif(0,1000)
# day-to-day variation = 1/tau
p0pool <- 1-pow(1-p0,subN)
# average pool level prevalence (to be estimated as result)
logit(p0) <- mu
# average animal level prevalence (to be estimated as result)
for(d in 1:D){ # total of D days
X[d] ~ dbin(ppool[d],N)
# observed number of positive pools
ppoolhat[d] <- X[d]/N # for plotting results
phat[d] <- 1-pow(1-ppoolhat[d],1/subN) # for plotting results
ind[d] <- d # for plotting results
ppool[d] <- 1-pow(1-p[d],subN)
logit(p[d]) <- mu + alpha[d] # animal prevalence for day d
alpha[d] ~ dnorm(0,tau);
}
# prediction:
alphastar ~ dnorm(0,tau)
logit(pstar) <- mu + alphastar
ppoolstar <- 1-pow(1-pstar,subN)
xstar ~ dbin(ppoolstar,N)
}
list(D=10,N=5,subN=3,X=c(5,5,4,1,0,1,1,4,3,0))
```

```
#inits: list(mu=0,sigma=1)
```

5. (6 points):
Find, explain and correct where possible the errors in the following BUGS model code.

```
model{
for(i in 1:3){
x[i,1:5] ~ dmultin(N,p[1:5])
}
xpred[1:5] ~ dmultin(M,p[1:5])
M <- step(q-0.5)*50 + (1-step(q-0.5))*10
q ~ dunif(0,1)
p ~ ddirh(a[])
}}
list(x=structure(.Data=c(3,2,8,1,1,4,1,3,2,5,3,1,7,2,2),.Dim=(5,3)),
a=c(1,1,1,1,1),N=15)
```

Some syntax errors are: `dmultin` should be `dmulti`, and `N` should appear after `p[1:5]`. `ddirh` should be `ddirch` in WinBUGS (`ddirich` in OpenBUGS). `p` should be `p[1:5]` in the prior definition. Misplaced `}`. Data list should have `.Dim=c(3,5)`. The main problem is that the M in the multinomial model is not allowed to be random in WinBUGS. If it was, a more complicated sampling would be required (block sampling, to ensure sum of the x always equals M). However, since this model appears to be only trying to generate predictions `xpred` under randomly chosen M (not trying to compute posterior under random M), it is possible to define the predicted variable as a deterministic node as follows.

```
model{
for(i in 1:3){
x[i,1:5] ~ dmulti(p[1:5],N)
}
 xpred1[1:5] ~ dmulti(p[1:5],M[1])
 xpred2[1:5] ~ dmulti(p[1:5],M[2])
 for(i in 1:5){xpred[i]<-xpred1[i]*step(q-0.5)+xpred2[i]*(1-step(q-0.5))}
q ~ dunif(0,1)
p[1:5] ~ ddirch(a[])
}
list(x=structure(.Data=c(3,2,8,1,1,
                         4,1,3,2,5,
                         3,1,7,2,2),.Dim=c(3,5)),
a=c(1,1,1,1,1),N=15,M=c(50,10))
```