**TITLE A Bayesian Mixture Model for Changepoint Estimation Using Ordinal Predictors**

Emily Roberts, MS (ORCID 0000-0002-5838-9691) and Lili Zhao, PhD

Department of Biostatistics, University of Michigan

Corresponding author information

Emily Roberts ekrobe@umich.edu Department of Biostatistics, University of Michigan,

1415 Washington Heights, Ann Arbor, MI 48109

**APPENDIX**

**JAGS Code for a continuous outcome:**

model {

for (i in 1:n){ # n subjects

for (k in 1:K){ # K possible changepoints of the ordinal predictor

X.c[i, k] <- equals(Z[k], 1) \* x[i, k] / sd2[k] + equals(Z[k], 0) \* step(x[i, k] - tau[k]) # apply standardization to variables if treated as binary

}

}

for (i in 1:n){ # n subjects

y[i] ~ dnorm(mu[i], w) # outcome y follows normal distribution

mu[i] <- beta0+ inprod(beta[], X.c[i, ]) # multiply coefficients and covariates for mean

}

for (k in 1:K){

tau[k] ~ dcat(p[k, ]) # categorical prior for tau

p[k, 1:L] ~ ddirch(alpha[]) # Dirichlet prior for alpha

Z[k] ~ dbern(pz[k]) # Bernoulli prior for Z

pz[k] ~ dbeta(0.5, 0.5) # Beta prior for p

beta[k] ~ dnorm(0, 0.001) # normal prior for coefficients

}

w ~ dgamma(0.01, 0.01) # gamma prior for error term

beta0 ~ dnorm(0, 0.001) # normal prior for intercept term

}

**JAGS Code for a continuous outcome with Lasso penalty:**

model {

for (i in 1:n){ # n subjects

for (k in 1:K){ # K possible changepoints of the ordinal predictor

X.c[i, k] <- equals(Z[k], 1)\*x[i, k] / sd2[k] + equals(Z[k], 0) \* step(x[i, k] - tau[k]) # apply standardization to variables if treated as binary

}

}

}

for (i in 1:n){ # n subjects

y[i] ~ dnorm(mu[i], w) # outcome y follows normal distribution

mu[i] <- beta0+ inprod(gamma[] \* beta[], X.c[i, ]) # multiply coefficients and covariates for mean

}

tt = lambda\*w # tt is the precision for beta and lambda is set to a constant as a tuning parameter for the Lasso penalty

for (k in 1:K){

tau[k] ~ dcat(p[k, ]) # categorical prior for tau

p[k, 1:L] ~ ddirch(alpha[]) # Dirichlet prior for alpha

Z[k] ~ dbern(pz[k]) # Bernoulli prior for Z

pz[k] ~ dbeta(0.5, 0.5) # Beta prior for p

beta[k] ~ ddexp(0, tt) # beta is double exponential with precision tt

gamma[k] ~ dbern(0.5) # use for inclusion indicator for covariate selection

}

w ~ dgamma(0.01, 0.01)

beta0 ~ dnorm(0, 0.001)

}

**JAGS Code for a continuous outcome with Horseshoe prior penalty:**

model {

for (i in 1:n){ # n subjects

for (k in 1:K){ # K possible changepoints of the ordinal predictor

X.c[i, k] <- equals(Z[k],1) \* x[i, k] / sd2[k] + equals(Z[k], 0) \* step(x[i, k] - tau[k])} # apply standardization to variables if treated as binary

}

for (i in 1:n){ # n subjects

y[i] ~ dnorm(mu[i], w) # outcome y follows normal distribution

mu[i] <- beta0+ inprod(gamma[] \* beta[], X.c[i, ]) # multiply coefficients and covariates for mean

}

for (k in 1:K){

tau[k] ~ dcat(p[k,]) # categorical prior for tau

p[k, 1:L] ~ ddirch(alpha[]) # Dirichlet prior for alpha

Z[k] ~ dbern(pz[k]) # Bernoulli prior for Z

pz[k] ~ dbeta(0.5, 0.5) # Beta prior for p

beta[k] ~ dnorm(0, lambda \* v[k] \* vs)

#half cauchy prior for v[k]

xi[k] ~ dnorm(0, 1)

tau.eta[k] ~ dgamma(0.5, 0.5)

v[k] <- abs(xi[k]) / sqrt(tau.eta[k])

gamma[k] ~ dbern(0.5) # use for inclusion indicator for covariate selection

}

# the following lines denote a half cauchy prior for vs

xis ~ dnorm(0, 1)

tau.etas ~ dgamma(0.5, 0.5) # chi^2 with 1 d.f.

vs <- abs(xis) / sqrt(tau.etas)

w ~ dgamma(0.01, 0.01) # gamma prior for error term

beta0 ~ dnorm(0, 0.001) # normal prior for intercept term

}

**JAGS Code for a binary outcome:**

model {

for (i in 1:n){ # n subjects

for (k in 1:K){ # K possible changepoints of the ordinal predictor

X.c[i,k ] <- equals(Z[k],1) \* x[i, k] / sd2[k] + equals(Z[k], 0) \* step(x[i, k] - tau[k]) # apply standardization to variables if treated as binary

}

}

for (i in 1:n){ # n subjects

y[i] ~ dbin(mu[i], 1) # outcome y follows binary distribution

mu[i] <- 1 / (1 + exp(-z[i])) # mean of logit model based on z below

z[i] <- beta0+ inprod(beta[], X.c[i, ]) # multiply coefficients and covariates for mean

}

for (k in 1:K){ # K possible changepoints of the ordinal predictor

tau[k] ~ dcat(p[k, ]) # categorical prior for tau

p[k, 1:L] ~ ddirch(alpha[]) # Dirichlet prior for alpha

Z[k] ~ dbern(pz[k]) # Bernoulli prior for Z

pz[k] ~ dbeta(0.5, 0.5) # Beta prior for p

beta[k] ~ dnorm(0, 0.001) # normal prior for coefficients

}

w ~ dgamma(0.01, 0.01) # gamma prior for error term

beta0 ~ dnorm(0, 0.001) # normal prior for intercept term

}

**JAGS Code for a survival outcome:**

data{

eps <- 1.0E-10 # tolerance for time steps

for(i in 1:n){ # n subjects

for(j in 1:T){ # T time points

[i,j] <- step(y[i] - t[j] - eps) # counting process jump = 1 if t[j] <= y < t[j + 1]

dN[i, j] <- Y[i, j] \* step(t[j + 1] - y[i] + eps) \* d[i] # counting process increments

delta[i, j] <- Y[i, j]\*(min(y[i], t[j+1]) - t[j]) # censor indicator

}

}

}

model{

for (i in 1:n){ # n subjects

for (k in 1:K){ # K possible changepoints of the ordinal predictor

X.c[i, k] <- equals(Z[k], 1) \* x[i, k] / sd2[k] + equals(Z[k], 0) \* step(x[i, k] - tau[k]) # apply standardization to variables if treated as binary

}

}

for(j in 1:T){ # T time points

for(i in 1:n) { # n subjects

dN[i, j] ~ dpois(Idt[i, j]) # counting process jumps have Poisson prior

Idt[i, j] <- Y[i, j] \* exp(inprod(beta[], X.c[i, ])) \* delta[i, j] \* dL0[j] # intensity process mean

}

dL0[j] ~ dgamma(mu[j], c) # increment of cumulative hazard, c represents confidence in GP prior

mu[j] <- dL0.star[j] \* c # mean of GP prior

}

for (j in 1:T) { # T time points

dL0.star[j] <- r \* (t[j+1] - t[j]) # mean term based on r\*increment t in time

}

for (k in 1:K){

tau[k] ~ dcat(p[k, 1:L[k]]) # categorical prior for tau

p[k, 1:L[k]] ~ ddirch(alpha[1:L[k], k]) # Dirichlet prior for alpha

Z[k] ~ dbern(pz[k]) # Bernoulli prior for Z

pz[k] ~ dbeta(0.5, 0.5) # Beta prior for p

beta[k] ~ dnorm(0, 0.001) # normal prior for coefficients

}

}