LECTURE 12: BAYESIAN MODEL FITTING

CLASS INTRO

INTRO QUESTIONS

- MCMC methods take samples from the joint posterior distribution for the parameters, say $\boldsymbol{\theta} = \{\beta, \sigma^2, \tau^2, \phi\}$. Discuss how this relates to integration *and* how the collection of samples of $\boldsymbol{\theta}$ can be used to make predictions at other spatial locations.
- For Today:
  - Fitting Bayesian Models

LIKELIHOOD BASED MODEL FITTING

VARIOGRAM BASED MODEL FITTING

- Up until now, we have used a least-squares approach with the variogram to estimate the covariance parameters.
- How would you do this if there were covariates that could be used to explain the process?
- Use the residuals from a linear model.

SOFTWARE DEMOS

- We are going to look at three options for fitting Bayesian spatial models.

1. `krige.bayes()` in `geoR`,
2. `spLM()` in `spBayes`, and
3. using JAGS.

- All of the frameworks are acceptable, but have different strengths and weaknesses.

JAGS

- **JAGS manual**
- JAGS (Just Another Gibbs Sampler) can be called from R using the following framework:

1. Definition of the model (in BUGS)
2. Compile Model
3. Draw Samples

JAGS DEMO - BASIC REGRESSION

- First we will use JAGS to fit a linear regression model. Use the code in the next slide to do this and answer the following questions.

1. What is the sampling model in this case? Note that dnorm takes the precision (= 1 / variance) as an argument.
2. What priors are specified in this model?
3. How do the results match your expectation and that of `lm()`?

CODE

```
# Simulate Data
set.seed(02142019)
alpha.true <- 0
beta.true <- 1
sigma.true <- 2
num.pts <- 100

x <- runif(num.pts,0,10)
y <- rnorm(num.pts, mean = rep(alpha.true, num.pts) + x*beta.true, sd = sigma.true)
data.frame(x=x, y=y) %>% ggplot(aes(x=x,y=y)) + geom_point() + geom_smooth(method='l

# Specify data for JAGS
data.in <- list(x = x, y = y, N = num.pts)

#Define Model
model_string <- "model{
  # Likelihood
  for(i in 1:N){
    y[i]   ~ dnorm(mu[i], sigmasq.inv)
    mu[i] <- alpha + beta * x[i]
  }

  # Priors
  sigma <- 1 / sqrt(sigmasq.inv)
  alpha ~ dnorm(0, 1.0E-6)
```

JAGS DEMO - SPATIAL REGRESSION

- Next we consider a more complicated regression model using JAGS, that with the exponential covariance function.

1. What is `data.spatial` and how is it used in JAGS?
2. What priors are used in this case? Do they seem reasonable?
3. How do your results compare with your expectation?

CODE

```
# simulate data
set.seed(02142019)
num.pts <- 75
sigmasq.true <- 3
tausq.true <- .50
phi.true <- 2
x1 <- runif(num.pts, max = 10)
x2 <- runif(num.pts, max = 10)
d <- dist(cbind(x1,x2), upper=T, diag = T) %>% as.matrix()
Omega <- sigmasq.true * exp(-d * phi.true) + tausq.true * diag(num.pts)
alpha.true <- 0
beta.true <- 1
x.reg <- rnorm(num.pts)
y = rmnorm(1, x.reg*beta.true, Omega)
GP.dat <- data.frame(x1 = x1, x2 = x2, y = y)
GP.dat %>% ggplot(aes(x=x1, y = x2, z=y)) + geom_point(aes(color=y)) +   scale_colou


# Specify data for JAGS
data.spatial <- list(x.reg = x.reg, y = y, N = num.pts, d = d)

#Define Model
model.spatial <- "model{
# data | process
  for(i in 1:N){
```

`KRIGE.BAYES()` DEMO

- For this demonstration we will explore the `krige.bayes()` function in R using a modified script from the function description. With this exploration, answer the following questions.

1. What does the `grf()` function do?
2. Explain the parameters in the `prior.control()` section.
3. Describe the output from `hist(ex.bayes)`.
4. What are the four figures generated from the `image()` function?

CODE

```
set.seed(02132019)
# generating a simulated data-set
ex.data <- grf(100, cov.pars=c(10, .15), cov.model="exponential", nugget = 1)
#
data.frame(x1 = ex.data$coords[,'x'], x2 = ex.data$coords[,'y'], y = ex.data$data) %


# defining the grid of prediction locations:
ex.grid <- as.matrix(expand.grid(seq(0,1,l=15), seq(0,1,l=15)))
#
# computing posterior and predictive distributions
# (warning: the next command can be time demanding)
ex.bayes <- krige.bayes(ex.data, loc=ex.grid,
                        model = model.control(cov.m="exponential"),
                        prior = prior.control(beta.prior = 'flat',
                                              sigmasq.prior = 'reciprocal',
                                              phi.discrete=seq(0, 0.7, l=25),
                                              phi.prior="uniform",
                                              tausq.rel.discrete = seq(0, 1, l=25),
                                              tausq.rel.prior = 'uniform'))

# Plot histograms with samples from the posterior
par(mfrow=c(4,1))
hist(ex.bayes)
par(mfrow=c(1,1))
```

`SPLM()` DEMO

- Another option for fitting Bayesian spatial models is the `spLM()` function in the `spBayes` package. Using the code on the next slide, answer the following questions.

1. What is `w`?
2. What does the `tuning` argument in `spLM()` control?
3. What does the following code return `summary(m.1$p.beta.recover.samples)$quantiles`?
4. Describe the final figure generated by this code.

CODE

```
set.seed(02142019)
rmvn <- function(n, mu=0, V = matrix(1)){
  p <- length(mu)
  if(any(is.na(match(dim(V),p))))
    stop("Dimension problem!")
  D <- chol(V)
  t(matrix(rnorm(n*p), ncol=p)%*%D + rep(mu,rep(n,p)))
}

n <- 100
coords <- cbind(runif(n,0,1), runif(n,0,1))
X <- as.matrix(cbind(1, rnorm(n)))

B <- as.matrix(c(1,5))
p <- length(B)

sigma.sq <- 2
tau.sq <- 0.1
phi <- 3/0.5

D <- as.matrix(dist(coords))
R <- exp(-phi*D)
w <- rmvn(1, rep(0,n), sigma.sq*R)
y <- rnorm(n, X%*%B + w, sqrt(tau.sq))
```

OTHER MODELING OPTIONS

- JAGS is another option for fitting general Bayesian models.
- Additionally, these models can also be implemented from scratch.