Bayesian linear regression (cont.)

Dr. Jarad Niemi

STAT 544 - Iowa State University

April 20, 2017

Outline

- Subjective Bayesian regression
    - Ridge regression
    - Zellner's g-prior
    - Bayes' Factors for model comparison
- Regression with a known covariance matrix
    - Known covariance matrix
    - Covariance matrix known up to a proportionality constant
    - MCMC for parameterized covariance matrix
        - Time series
        - Spatial analysis

Subjective Bayesian regression

Suppose

$$y \sim N(X\beta, \sigma^2 \mathrm{I})$$

and we use a prior for β of the form

$$\beta|\sigma^2 \sim N(b, \sigma^2 B)$$

A few special cases are

- $b = 0$
- B is diagonal
- $B = g\mathrm{I}$
- $B = g(X'X)^{-1}$

Ridge regression

Let

$$y = X\beta + e, \quad E[e] = 0, \quad Var[e] = \sigma^2 \mathrm{I}$$

then ridge regression seeks to minimize

$$(y - X\beta)'(y - X\beta) + g\beta'\beta$$

where g is a penalty for $\beta'\beta$ getting too large.

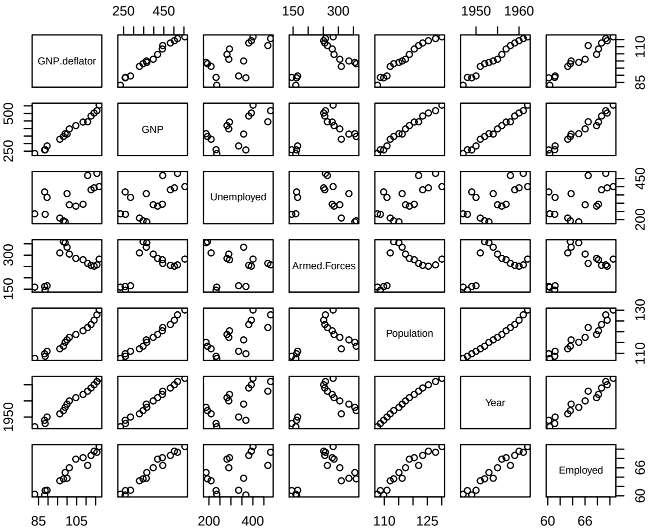
This minimization looks like -2 times the log posterior for a Bayesian regression analysis when using independent normal priors centered at zero with a common variance (c_0) for β:

$$-2\sigma^2 \log p(\beta, \sigma|y) = C + (y - X\beta)'(y - X\beta) + \frac{\sigma^2}{c_0}\beta'\beta$$

where $g = \sigma^2/c_0$. Thus the ridge regression estimate is equivalent to a MAP estimate when

$$y \sim N(X\beta, \sigma^2 \mathrm{I}) \quad \beta \sim N(0, c_0 \mathrm{I}).$$

Longley data set



Default Bayesian regression (unscaled)

```
summary(lm(GNP.deflator~., longley))



Call:
lm(formula = GNP.deflator ~ ., data = longley)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.0086 -0.5147  0.1127  0.4227  1.5503 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)  
(Intercept)  2946.85636 5647.97658   0.522   0.6144  
GNP             0.26353    0.10815   2.437   0.0376 *
Unemployed      0.03648    0.03024   1.206   0.2585  
Armed.Forces    0.01116    0.01545   0.722   0.4885  
Population     -1.73703    0.67382  -2.578   0.0298 *
Year           -1.41880    2.94460  -0.482   0.6414  
Employed        0.23129    1.30394   0.177   0.8631  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.195 on 9 degrees of freedom
Multiple R-squared:  0.9926, Adjusted R-squared:  0.9877 
F-statistic: 202.5 on 6 and 9 DF,  p-value: 4.426e-09
```

Default Bayesian regression (scaled)

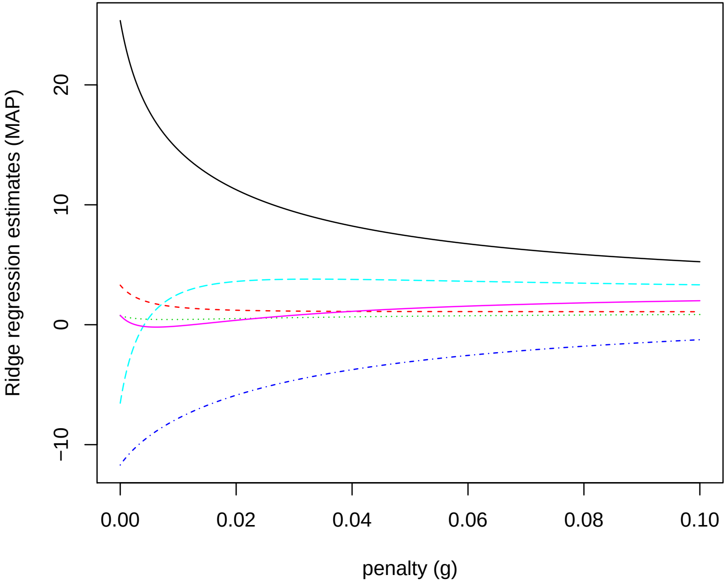
```
y = longley$GNP.deflator
X = scale(longley[,-1])
summary(lm(y~X))
```

```
Call:
lm(formula = y ~ X)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.0086 -0.5147  0.1127  0.4227  1.5503 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)   101.6813     0.2987 340.465   <2e-16 ***
XGNP           26.1933    10.7497   2.437   0.0376 *  
XUnemployed     3.4092     2.8263   1.206   0.2585    
XArmed.Forces   0.7767     1.0754   0.722   0.4885    
XPopulation   -12.0830     4.6871  -2.578   0.0298 *  
XYear          -6.7548    14.0191  -0.482   0.6414    
XEmployed       0.8123     4.5794   0.177   0.8631    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.195 on 9 degrees of freedom
Multiple R-squared:  0.9926, Adjusted R-squared:  0.9877 
F-statistic: 202.5 on 6 and 9 DF,  p-value: 4.426e-09
```

Ridge regression in MASS package

```
library(MASS)
gs = seq(from = 0, to = 0.1, by = 0.0001)
m = lm.ridge(GNP.deflator ~ ., longley, lambda = gs)

# Choose the ridge penalty
select(m)


modified HKB estimator is 0.006836982
modified L-W estimator is 0.05267247
smallest value of GCV  at 0.0057


# Estimates
est = data.frame(lambda = gs, t(m$coef))
est[round(est$lambda,4) %in% c(.0068,.053,.0057),]


       lambda       GNP Unemployed Armed.Forces Population     Year   Employed
0.0057 0.0057 17.219755   1.785199    0.4453260  -9.047254 1.021387 -0.1955648
0.0068 0.0068 16.411861   1.675572    0.4369163  -8.692626 1.548683 -0.1947731
0.0530 0.0530  7.172874   1.096683    0.7190487  -2.911938 3.683572  1.4239190
```

Ridge regression in MASS package



Zellner's g-prior

Suppose

$$y \sim N(X\beta, \sigma^2 \mathrm{I})$$

and you use Zellner's g-prior

$$\beta \sim N(b_0, g\sigma^2(X'X)^{-1}).$$

The posterior is then

$$\begin{aligned}\beta|\sigma^2, y &\sim N\left(\frac{g}{g+1}\left(\frac{b_0}{g} + \hat{\beta}\right), \frac{\sigma^2 g}{g+1}(X'X)^{-1}\right) \\ \sigma^2|y &\sim \text{Inv-}\chi^2\left(n, \frac{1}{n}\left[(n-k)s^2 + \frac{1}{g+1}(\hat{\beta} - b_0)X'X(\hat{\beta} - b_0)\right]\right)\end{aligned}$$

with

$$\begin{aligned}E[\beta|y] &= \frac{1}{g+1}b_0 + \frac{g}{g+1}\hat{\beta} \\ E[\sigma^2|y] &= \frac{(n-k)s^2 + \frac{1}{g+1}(\hat{\beta}-b_0)X'X(\hat{\beta}-b_0)}{n-2}\end{aligned}$$

Setting g

In Zellner's g-prior,

$$\beta \sim N(b_0, g\sigma^2(X'X)^{-1}),$$

we need to determine how to set g.

Here are some thoughts:

- $g = 1$ puts equal weight to prior and likelihood
- $g = n$ means prior has the equivalent weight of 1 observation
- $g \to \infty$ recovers a uniform prior
- Empirical Bayes estimate of g, $\hat{g}_{EG} = \text{argmax}_g p(y|g)$ where

$$p(y|g) = \frac{\Gamma\left(\frac{n-1}{2}\right)}{\pi^{(n+1)/2} n^{1/2}} ||y - \overline{y}||^{-(n-1)} \frac{(1+g)^{(n-1-k)/2}}{\left(1 + g(1+R^2)\right)^{(n-1)/2}}$$

  where R^2 is the usual coefficient of determination.
- Put a prior on g and perform a fully Bayesian analysis.

Zellner's g-prior in R

```
library(BMS)
m = zlm(GNP.deflator~., longley, g='UIP') # g=n
summary(m)


Coefficients
                  Exp.Val.     St.Dev.
(Intercept)  2779.49311839          NA
GNP             0.24802564 0.26104901
Unemployed      0.03433686 0.07300367
Armed.Forces    0.01050452 0.03730077
Population     -1.63485161 1.62641807
Year           -1.33533979 7.10751875
Employed        0.21768268 3.14738044

 Log Marginal Likelihood:
-44.07653
 g-Prior: UIP
Shrinkage Factor: 0.941
```

Bayes Factors for regression model comparison

Consider two models with design matrices $X^1$and X^2 (not including an intercept) and corresponding dimensions (n, p_1) and (n, p_2). Zellner's g-prior provides a relatively simple way to construct default priors for model comparison. Formally, we compare

$$\begin{array}{rl} y & \sim N(\alpha 1_n + X^1\beta^1, \sigma^2 \mathrm{I}) \\ \beta & \sim N(b_1, g_1\sigma^2[(X^1)'(X^1)]^{-1}) \\ p(\alpha, \sigma^2) & \propto 1/\sigma^2 \end{array}$$

and

$$\begin{array}{rl} y & \sim N(\alpha 1_n + X^2\beta^2, \sigma^2 \mathrm{I}) \\ \beta & \sim N(b_2, g_2\sigma^2[(X^2)'(X^2)]^{-1}) \\ p(\alpha, \sigma^2) & \propto 1/\sigma^2 \end{array}$$

Bayes Factors for regression model comparison

The Bayes Factor for comparing these two models is

$$B_{12}(y) = \frac{(g_1+1)^{-p_1/2}\left[(n-p_1-1)s_1^2 + \left(\hat\beta_1 - b_1\right)'(X^1)'X^1\left(\hat\beta_1 - b_1\right)/(g_1+1)\right]^{-(n-1)/2}}{(g_2+1)^{-p_2/2}\left[(n-p_2-1)s_2^2 + \left(\hat\beta_2 - b_2\right)'(X^2)'X^2\left(\hat\beta_2 - b_2\right)/(g_2+1)\right]^{-(n-1)/2}}$$

Now, we can set $g_1 = g_2$ and calculate Bayes Factors.

```
library(bayess)
m = BayesReg(longley$GNP.deflator, longley[,-1], g = nrow(longley))



          PostMean PostStError Log10bf EvidAgaH0
Intercept 101.6813      0.7431
x1         23.8697     25.1230 -0.3966
x2          3.1068      6.6053 -0.5603
x3          0.7078      2.5134 -0.5954
x4        -11.0111     10.9543 -0.3714
x5         -6.1556     32.7640 -0.6064
x6          0.7402     10.7025  -0.614


Posterior Mean of Sigma2: 8.8342
Posterior StError of Sigma2: 13.0037
```

Known covariance matrix

Suppose $y \sim N(X\beta, S)$ where S is a known covariance matrix and assume $p(\beta) \propto 1$.

Let L be a Cholesky factor of S, i.e. $LL^\top = S$, then the model can be rewritten as

$$L^{-1}y \sim N(L^{-1}X\beta, \mathrm{I}).$$

The posterior, $p(\beta|y)$, is the same as for ordinary linear regression replacing y with $L^{-1}y$, X with $L^{-1}X$ and σ^2 with 1 where L^{-1} is inverse of L. Thus

$$\begin{array}{rll} \beta|y & \sim N(\hat{\beta}, V_\beta) & \\ V_\beta & = ([L^{-1}X]^\top L^{-1}X)^{-1} & = (X^\top S^{-1}X)^{-1} \\ \hat{\beta} & = ([L^{-1}X]^\top L^{-1}X)^{-1}[L^{-1}X]^\top L^{-1}y & = V_\beta X^\top S^{-1}y \end{array}$$

So rather than computing these, just transform your data using $L^{-1}y$ and $L^{-1}X$ and force $\sigma^2 = 1$.

Autoregressive process of order 1

A mean zero, stationary autoregressive process of order 1 assumes

$$\epsilon_t = r\epsilon_{t-1} + \delta_t$$

with $-1 < r < 1$ and $\delta_t \stackrel{ind}{\sim} N(0, v^2)$.

Suppose

$$y_t = X_t'\beta + \epsilon_t$$

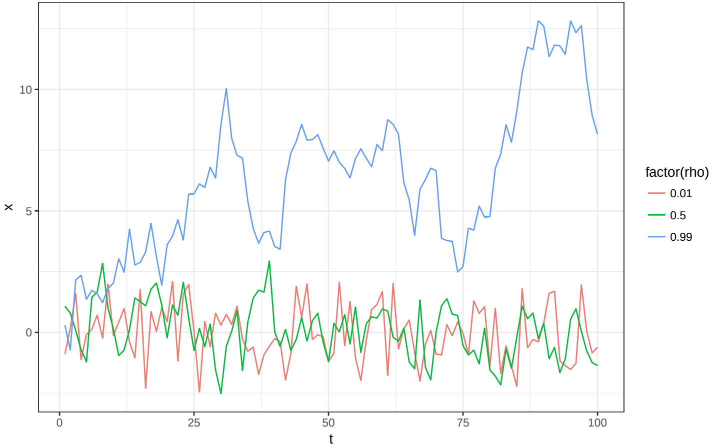
or, equivalently,

$$y \sim N(X\beta, S)$$

where $S = s^2 R$ with

- stationary variance $s^2 = v^2/[1 - r^2]$ and
- correlation matrix R with elements $R_{ij} = r^{|i-j|}$.

Example autoregressive processes



Calculate posterior

```
ar1_covariance = function(n, r, v) {
  V = diag(n)
  v^2/(1-r^2) * r^(abs(row(V)-col(V)))
}

# Covariance
n = 100
S = ar1_covariance(n,.9,2)

# Simulate data
set.seed(1)
library(MASS)
k = 50
X = matrix(rnorm(n*k), n, k)
beta = rnorm(k)
y = mvrnorm(1,X%*%beta, S)

# Estimate beta
Linv = solve(t(chol(S)))
Linvy = Linv%*%y
LinvX = Linv%*%X
m = lm(Linvy ~ 0+LinvX)

# Force sigma=1
Vb = vcov(m)/summary(m)$sigma^2
```

Credible intervals

```
# Credible intervals
sigma = sqrt(diag(Vb))
ci = data.frame(lcl=coefficients(m)-qnorm(.975)*sigma,
                ucl=coefficients(m)+qnorm(.975)*sigma,
                truth=beta)
head(ci,10)
```

```
               lcl        ucl      truth
LinvX1  -2.17120084 -1.1402125 -1.5163733
LinvX2   0.16358494  1.2519609  0.6291412
LinvX3  -1.86349405 -0.9497331 -1.6781940
LinvX4   0.34866009  1.3955061  1.1797811
LinvX5   1.03663767  1.8074717  1.1176545
LinvX6  -1.84593196 -0.7322210 -1.2377359
LinvX7  -1.64201301 -0.8329486 -1.2301645
LinvX8   0.12817405  1.0358989  0.5977909
LinvX9  -0.03442773  0.9363690  0.2988644
LinvX10 -0.30381498  0.7243550 -0.1101394
```

```
all.equal(Vb[1:k^2], solve(t(X)%*%solve(S)%*%X)[1:k^2])
```

```
[1] TRUE
```

```
all.equal(as.numeric(coefficients(m)), as.numeric(Vb%*%t(X)%*%solve(S)%*%y))
```

```
[1] TRUE
```

Variance known up to a proportionality constant

Consider the model

$$y \sim N(X\beta, \sigma^2 S)$$

for a known S with default prior $p(\beta, \sigma^2) \propto 1/\sigma^2$.

The posterior is

$$\begin{array}{rl} p(\beta, \sigma^2|y) & = p(\beta|\sigma^2, y)p(\sigma^2|y) \\ \beta|\sigma^2, y & \sim N(\hat{\beta}, \sigma^2 V_\beta) \\ \sigma^2|y & \sim \text{Inv-}\chi^2(n-k, s^2) \\ \beta|y & = t_{n-k}(\hat{\beta}, s^2 V_\beta) \\ \\ \hat{\beta} & = (X^\top S^{-1} X)^{-1} X^\top S^{-1} y \\ V_\beta & = (X^\top S^{-1} X)^{-1} \\ s^2 & = \frac{1}{n-k}(L^{-1}y - L^{-1}X\hat{\beta})^\top (L^{-1}y - L^{-1}X\hat{\beta}) \\ & = \frac{1}{n-k}(y - X\hat{\beta})^\top S^{-1}(y - X\hat{\beta}) \end{array}$$

where $LL' = S$.

AR1 process

Consider the model

$$y \sim N(X\beta, \sigma^2 R)$$

where R is the correlation matrix from an AR1 process.

This is exactly what we had before, except we do not assume $\sigma = 1$.

Posterior with unknown σ^2

```
m    = lm(Linvy ~ 0+LinvX)
Vb   = vcov(m)
bhat = coefficients(m)
df   = n-k
s2   = sum(residuals(m)^2)/df

# Credible intervals
cbind(confint(m), Truth=beta)[1:10,]
```

```
              2.5 %     97.5 %      Truth
LinvX1  -2.17051088 -1.1409024 -1.5163733
LinvX2   0.16431330  1.2512325  0.6291412
LinvX3  -1.86288255 -0.9503446 -1.6781940
LinvX4   0.34936066  1.3948056  1.1797811
LinvX5   1.03715353  1.8069558  1.1176545
LinvX6  -1.84518665 -0.7329663 -1.2377359
LinvX7  -1.64147158 -0.8334900 -1.2301645
LinvX8   0.12878152  1.0352915  0.5977909
LinvX9  -0.03377805  0.9357193  0.2988644
LinvX10 -0.30312691  0.7236669 -0.1101394
```

Parameterized covariance matrix

Suppose

$$y \sim N(X\beta, S(\theta))$$

where $S(\theta)$ is now unknown, but can be characterized by a low dimensional θ, e.g.

- Autoregressive process of order 1:

$$S(\theta) = \sigma^2 R(\rho), R_{ij}(\rho) = \rho^{|i-j|}$$

- Gaussian process with exponential covariance function:

$$S(\theta) = \tau^2 R(\rho) + \sigma^2 \mathrm{I}, R_{ij}(\rho) = \exp(-\rho d_{ij})$$

- Conditionally autoregressive (CAR) model:

$$S(\theta) = \sigma^2 (D_w - \rho W)^{-1}$$

MCMC for parameterized covariance matrices

Suppose

$$y \sim N(X\beta, S(\theta))$$

then an MCMC strategy is

1. Sample $\beta|\theta, y$, i.e. regression with a known covariance matrix.
2. Sample $\theta|\beta, y$.

Alternatively, if

$$y \sim N(X\beta, \sigma^2 R(\theta))$$

then an MCMC strategy is

1. Sample $\beta, \sigma^2|\theta, y$, i.e. regression when variance is known up to a proportionality constant..
2. Sample $\theta|\beta, \sigma^2, y$.

Since θ exists in a low dimension, many of the methods we have learned can be used, e.g. ARS, MH, slice sampling, etc.

Summary

- Subjective Bayesian regression
    - Ridge regression
    - Zellner's g-prior
    - Bayes' Factors for model comparison
- Regression with a known covariance matrix
    - Known covariance matrix
    - Covariance matrix known up to a proportionality constant
    - MCMC for parameterized covariance matrix
        - Time series
        - Spatial analysis