

Stat 544 – Spring 2005

Homework assignment 5

Due on Monday, April 4 in TA's office by 5:00 pm

Problem 1

It is often of interest to identify influential observations in a statistical analysis since conclusions may be sensitive to assumptions about individual cases; in extreme situations, conclusions depend entirely on single observations. The most widely Bayesian case-influence statistics used are the Kullback divergence, the L_1 norm and the χ^2 norm.

(a) Background

Let $p(\theta|y)$ be the posterior distribution of the parameters θ given data y , a vector of observations independent given θ and with y_i observed at covariates x_i . Case-influence assessment compares $p(\theta|y)$ to the case-deleted posterior $p(\theta|y_{(i)})$, where $y_{(i)}$ denotes the observation vector omitting the i th case. The most popular case-influence statistics for Bayesian inference are of the form:

$$D_\theta(g, i) = \int g\left(\frac{p(\theta|y_{(i)})}{p(\theta|y)}\right) p(\theta|y) d\theta \quad (1)$$

where $g(a)$ is a convex function with $g(1) = 0$. The Kullback divergence measures K_1 and K_2 are member of this family with $g_1(a) = a \log(a)$ and $g_2(a) = -\log(a)$. The L_1 norm has $g_3(a) = 0.5|a - 1|$, and the χ^2 divergence has $g_4(a) = (a - 1)^2$.

The influence of case deletion on a posterior distribution can be assessed using the perturbation function $h_i(\theta)$:

$$\frac{p(\theta|y_{(i)})}{p(\theta|y)} = \frac{\text{CPO}_i}{f(y_i|\theta, x_i)} \equiv h_i(\theta)$$

where $\text{CPO}_i = f(y_i|y_{(i)}) = \int f(y_i|\theta, x_i) p(\theta|y_{(i)}) d\theta$, is the predictive density of y_i given $y_{(i)}$, and $f(y_i|\theta, x_i)$ is the sampling density for case i . Estimates of the CPO_i may be approximated using a single posterior sample of size T of θ from $p(\theta|y)$ by

$$\text{CPO}_i^{-1} \approx \frac{1}{T} \sum_{t=1}^T [f(y_i|\theta^t, x_i)]^{-1} \quad (2)$$

where θ^t is the drawn value of θ at iteration t ; that is, by the harmonic mean of the likelihoods of case i .

By using equation 2, equation 1 can be approximated given another sample θ^l , $l = 1, \dots, L$ of θ from $p(\theta|y)$ by

$$\hat{D}(g, i) \approx \frac{1}{L} \sum_{l=1}^L g(h_i(\theta^l))$$

(b) *Problem*

Consider the simple linear regression model:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \quad \epsilon \sim N(0, \sigma^2 \mathbf{I})$$

and the following dataset

i	1	2	3	4	5	6	7	8	9	10	11
x_i	15	26	10	9	15	20	18	11	8	20	7
y_i	95	71	83	91	102	87	93	100	104	94	113

i	12	13	14	15	16	17	18	19	20	21
x_i	9	10	11	11	10	12	42	17	11	10
y_i	96	83	84	102	100	105	57	121	86	100

Assume the following prior distributions:

- $\beta_i \sim N(0, [0.0000001]^{-1})$ $i = 1, 2$
- $\sigma^2 \sim \text{Inv-Gamma}(1, 0.001)$.

For each observation:

- 1) Calculate CPO_i
- 2) Calculate both Kullback divergences.
- 3) Calculate the L_1 norm divergence.
- 4) Calculate the χ^2 norm divergence.
- 5) Interpret the results (i.e. which observations are outliers, and which are influential cases.)

Hint:

This problem can be carried out using WinBugs. You should write one program to do part 1, and another to do parts 2-4.

- *In your first program, fit the model and for each iteration t and for each of the 21 observations i compute $k_i(\theta)$*

$$k_i(\theta) = \frac{1}{f(y_i | \theta^t, x_i)}$$

You may use 6.28 as an approximation for 2π .

*Get the means of your 21 k 's. These means are CPO_i^{-1} defined in equation (2). You will need to compute their inverses outside WinBugs¹. These inverses will be used as data in your second program. A really useful function inside R is **dput**. It will write an R object to look like "WinBugs data" format; you have to edit out some of the output though. Run this program to see what dput does:*

¹If you wish to get fancy, you may want to use bugs.R inside R (see appendix C of the textbook). bugs.R is loaded on all the 322 Snedecor machines and that will allow you to do all calculations from within R

```
x<-1:10
x
dput(x, "")
```

- In your second program, with the CPO's included in your data declaration, you have to compute the different norms. For example to calculate χ^2 divergence, you would compute:

$$\hat{D}(g, i) = \frac{1}{L} \sum_{l=1}^L \left(\frac{CPO_i}{f(y_i | \theta^l, x_i)} - 1 \right)^2$$

Problem 2

Selecting a suitable model from a large class of plausible models is an important problem in statistics. A classic example is the selection of variables in linear regression analysis. One approach to selecting variables is predictive model assessment. This method consists on sampling "new data" Z from a model fitted to all cases, and seeing how consistent the new data are with the observations. Thus, comparisons might be made by sampling replicates $Z_i^{(t)}$ for each observation at iteration t from a normal model with mean $\mu_i^{(t)}$, and variance $(\sigma^2)^{(t)}$ and then comparing those values with the corresponding y_i .

One comparison criterion is to calculate for each model in consideration the following quantity:

$$C^2 = \sum_{i=1}^n [\{E(Z_i) - y_i\}^2 + \text{Var}(Z_i)] \quad (3)$$

Better models will have smaller values of C^2 , or of its square root C . Thus, if for example we have two models: model 1 and model 2, model 1 will be better than model 2 if

$$C_{(1)}^2 - C_{(2)}^2 = \sum_{i=1}^n [\{E(Z_{1i}) - y_i\}^2 - \{E(Z_{2i}) - y_i\}^2] < 0 \quad (4)$$

where $C_{(j)}^2$ and Z_{ji} represent the criterion and the replicates obtained under model j ($j = 1, 2$) respectively. Alternatively, model 1 is better than model 2 if $C_{(1)} - C_{(2)} < 0$ where $C_{(j)}$ stands for $\sqrt{C_{(j)}^2}$.

Our dataset² refers to the heat evolved in calories per gram of cement, a metric outcome, for $n = 13$ cases. The outcome is related to four predictors describing the composition of the cement ingredients.

²Hald, A. (1952) *Statistical Theory with Engineering Applications*. New York: Wiley

x_1	x_2	x_3	x_4	y
7	26	6	60	78.5
1	29	15	52	74.3
11	56	8	20	104.3
11	31	8	47	87.6
7	52	6	33	95.9
11	55	9	22	109.2
3	71	17	6	102.7
1	31	22	44	72.5
2	54	18	22	93.1
21	47	4	26	115.9
1	40	23	34	83.8
11	66	9	12	113.3
10	68	8	12	109.4

(a) Using the criterion described in equation 4 evaluate the following models:

$$\text{Model 1: } y_i = \alpha_0 + \alpha_1 x_{1i} + \alpha_2 x_{2i} + \alpha_3 x_{4i} + \varepsilon_i$$

$$\text{Model 2: } y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \varepsilon_i$$

$$\text{Model 3: } y_i = \gamma_0 + \gamma_1 x_{1i} + \gamma_2 x_{3i} + \gamma_3 x_{4i} + \varepsilon_i$$

$$\text{Model 4: } y_i = \delta_0 + \delta_1 x_{1i} + \delta_2 x_{2i} + \delta_3 x_{3i} + \delta_4 x_{4i} + \varepsilon_i$$

$$\text{Model 5: } y_i = \phi_0 + \phi_1 x_{1i} + \phi_2 x_{2i} + \varepsilon_i$$

$$\text{Model 6: } y_i = \varphi_0 + \varphi_1 x_{1i} + \varphi_2 x_{4i} + \varepsilon_i$$

$$\text{Model 7: } y_i = \eta_0 + \eta_1 x_{2i} + \eta_2 x_{3i} + \eta_3 x_{4i} + \varepsilon_i$$

$$\text{Model 8: } y_i = \kappa_0 + \kappa_1 x_{1i} + \kappa_2 x_{3i} + \varepsilon_i$$

where $\varepsilon \sim N(0, \sigma^2 \mathbf{I})$.

Use the following prior distributions:

- $\sigma^2 \sim \text{Inv-gamma}(12.5, 62.5)$.
- For the parameters of each model a $N(0, [0.00001]^{-1})$.

Comments:

- This problem can also be done using WinBugs.
- Two WinBugs functions that may be helpful are the function **step()** and the function **sum()**. The step function takes the following values

$$\text{step}(a) = \begin{cases} 1 & \text{if } a \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

- A WinBugs example program to obtain draws from the predictive posterior distribution in a simple linear regression model (indicate by the **z**'s) is:

```

model{ for (i in 1:N){ y[i] ~ dnorm(mu[i],tau.y)
                    mu[i] = b[1]+b[2]*x[i]
                    z[i] ~ dnorm(mu[i],tau.y) }
  for (i in 1:2){b[i] ~ dnorm(0,0.00001)}
  tau.y ~ ... }

Data
list(y=c(...),N=..)

```

- If $x \sim \text{Gamma}(\alpha, \beta)$, then $x^{-1} \sim \text{Inv-Gamma}(\alpha, \beta)$. Both distributions have the same α and β . Thus, if $\sigma^2 \sim \text{Gamma}(12.5, 62.5)$ then $\sigma^{-2} \sim \text{Inv-Gamma}(12.5, 62.5)$. In the textbook appendix A (page 580) you will find a description of the relationship between those two distributions.
- The z 's are called "new" since they are realizations from the predictive posterior distribution (i.e. from $p(z|y)$) as opposite to your y 's that are realizations from $p(y|\theta)$, where θ stands for all the parameters in your model. Note that your z 's are what in class and in the textbook are referred as \tilde{y} (see for example pages 8 and 137.) For convenience here they have been called z instead of \tilde{y} , since there is no ambiguity given the fact that the wording of problem 2 assigns the name z to observations drawn from the predictive posterior distribution.
- You have $i = 1, \dots, 13$ observations: y_i ; and $j = 1, \dots, 8$ models, each model with a different mean.

In each iteration, for each one of those 13 observations you draw eight z_{ji} 's, one for each one of the eight models.

Then, you have to calculate $(z_{ji} - y_i)^2$, and then you have to sum those values over i for each j . These sums are proportional to C_j^2 . Note that you don't need to consider the part that involves the sum of the variances of the z_{ij} 's since it will cancel out when you subtract the C^2 from one model with the C^2 from another since all models have the same variance. Further, note that since at each iteration you have only one value of z_{ij} you have to use this value in lieu of $E(Z_{ij})$.

- It may also pay off to be a little inefficient while programming, in the sense that you have to define the mean of 8 different models and they have a different number of parameters (some have three, another four, one with five). Thus, you have two choices, one is to define several different loops depending how many parameters your model has. The other is to define an 8x5 matrix where each row will contain the parameters needed to define the mean of each one of the 8 models. This has the advantage that you will need only two double-loops, one for defining the means and another to assign priors. Using this approach you are defining more parameters than you really need, and because of this your code will be inefficient. For this problem however, in terms of machine-time you will expend about an extra nano-second per iteration. Remember, the shorter your program, the easier to find where the errors are.