Week 9: Model Selection – A Purer Bayesian Approach

by Michael Escobar

March 11, 2018

1 Introduction

Last week we looked at various diagnostic methods. In doing that we sometimes found models which did not seem to fit the data as well as other models. When this happens, we would be inclined to prefer the models which seemed to fit the data well. Now this does make sense and does seem to be sound advice. However, some of you might think that you accidentally entered a frequentist class by mistake. Out went all the talk about beliefs and posterior probabilities and in went a bunch of statistics which started to look an awful lot like frequentist hypothesis testing. There were even a few of the statistics which were called p-values and tail area probabilities. What is an honest Bayesian to do? Can we do this like good Bayesians? The answer is the familiar answer for applied Bayesians. Yes, it can be done, but it is a lot harder to do.

2 Posterior Model probabilities

Suppose there are two different models that we might think are good models for the process under study. Then the basic Bayesian inference which we first discussed when talking about model parameters could be used here. That is, first without considering the data think about the belief that one has that say model 1 is the true model as opposed to model 2. This is the prior belief. Then, taking account of the data, what is our updated belief that model 1 is the better model than model 2.

To set notation, consider two model which we will call M1 and M2. These might or might not be nested models. As an example of a nested model, suppose M1 is the model for a simple regression line, say $Y_i = \alpha_1 + \alpha_2 X_i + \epsilon_i$. As a nested model, then M2 might be the the same model but α_2 is equal to zero. As an example of a non-nested model, then assume that M1 is the same simple linear regression line, but now M2 is a regression line where Y_i is a function of a different covariate, say Z_i, so we now model M2 is $Y_i = \beta_1 + \beta_2 Z_i + \eta_i$.

In the usual formulas for the prior, posterior, and the likelihood it is assumed that each of these functions are for a given model. Therefore, each is implicitly conditioned on knowledge of a particular model. Now, we make this assumption explicitly in our notation. Therefore, before we would write the formula for Bayes theorem as:

$$P(\theta|Y) = \frac{P(Y|\theta)P(\theta)}{P(Y)}.$$

Now, we write this formula explicitly including the fact that the above formula's are for a particular model, which we will call M_j. This gives us the following formula:

$$P(\theta|Y, M) = \frac{P(Y|\theta, M)P(\theta|M)}{P(Y|M)}.$$

So, for example, prior probability of the parameter θ is written as $P(\theta|M_j)$ because this distribution is knowledge which is part of the model M_j. A more important value in the above equation is $P(Y|M_j)$. This is the normalizing constant and before we did not worry about this value very much. However, now we see that this value the probability of the data for the given model. Therefore, this value plays the role of the likelihood function when wants to make inferences about model selection. So, for inference between models, we have to look at this value more.

When making inferences about the parameter θ, the term $P(Y|M_j)$ is the normalizing constant. That is, the posterior density function is proportional to the prior density function times the likelihood function. The shape of the posterior distribution is the same as the shape of the function which is the product of

the prior distribution and the likelihood function, but the area under the curve would be different. Since the posterior density function is a density function, then the area under the curve must be one. The value $P(Y|M_j)$ is divided into the product of the prior times the likelihood to make the area under the curve one. So, if we calculate the area under the curve of this product, then we have the value of $P(Y|M-j)$. That is, using the integral operator, which means that we would be summing up this value, we get:

$$P(Y|M_j) = \int P(Y|\theta, M_j)P(\theta|M_j)d\theta.$$

Unfortunately, finding the area under the curve turns out to be a difficult task. In one of the sections that follows some of the different methods of estimating this value are discussed. For the moment, let us assume that we can calculate the value $P(Y|M_j)$ and see how to make inferences about model selection.

The value of $P(Y|M_j)$ is the probability of seeing the data Y when the data is from model M_j. This is a type of likelihood function. Assume that models M_1 and M_2 are the only two models which we believe might be correct. Let $P(M_1)$ be the prior belief that model M_1 is the correct model. Then, using Bayes' Theorem, we can calculate the posterior belief that model M_1 is the correct model after we have considered the data. That is:

$$P(M_1|Y) = \frac{P(Y|M_1)P(M_1)}{\sum_{j=1}^{2} P(Y|M_j)P(M_j)}.$$

Also, we can express this in term of odds. The posterior odds for M_1 versus M_2 is $P(M_1|Y)/P(M_2|Y)$. The odds gives the number of times more we believe that model M_1 is better than M_2. If these odds are less then one, then the odds favour model M_2 over M_1. Using the above formula, we see that the posterior odds equals the following:

$$\frac{P(M_1|Y)}{P(M_2|Y)} = \frac{P(Y|M_1)}{P(Y|M_2)}\frac{P(M_1)}{P(M_2)}.$$

The ratio $P(Y|M_1)/P(Y|M_2)$ is called the Bayes factor. It is the improvement of our belief in model M_1 over M_2 over our prior odds. Some refer[1] to this as the increased weight of evidence that the data supports model M_1 over M_2.

The Bayes Factor has often been used as a way to quantify the evidence of one scientific theory/hypothesis over another. As a calibration of the Bayes factor for the evidence of one model versus another, Jeffreys (1961) (also see the paper by Kass and Raftery) suggested the following[2]:

$\log_{10}(B)$	B	Evidence against H_0
0 to 1/2	1 to 3.2	Not worth more than a bare mention
1/2 to 1	3.2 to 10	Substantial
1 to 2	10 to 100	Strong
> 2	> 100	Decisive

In order to make the calibrated scale similar to the usual deviance statistic and the likelihood ratio statistics, Kass and Raftery (1995) suggested the following calibration:

$2\log_e(B)$	B	Evidence against H_0
0 to 2	1 to 3	Not worth more than a bare mention
2 to 6	3 to 20	Positive
6 to 10	20 to 50	Strong
> 10	> 500	Decisive

[1] An important reference which overviews Bayes Factors is Kass, RE and Raftery, AE, (1995) "Bayes Factors", *Journal of the American Statistical Association, 90,* 773-795. You can access this paper over the web from a UofT server (or proxy server) at www.jstor.org

[2] Tables from Kass and Raftery, 1995.

3 Important Links

The new reversible jump patch has been added to WinBugs. Please go to the following link to obtain it:
`http://www.winbugs-development.org.uk/rjmcmc.html`

4 Comments on Model Selection in different disciplines

Different disciplines appear to have different ideas on the notions/philosophies on model selection.

Author's note: The following are some observations from someone who has 25 years experience analyzing data and interacting with these different groups. I am going to simply present these methods. Different disciplines have different impressions of the validity of these methods. As of this writing, I have not fully come to an opinion as to what is the "right way". Still, I believe it is important to present these ideas to the reader.

(Note: in the below, the variables are assumed to be measured for each of the i observation, but the index i is dropped to simplify the notation. Also, it will be common to include a y-intercept parameter, β_0. Again, to simply the discussion, sometimes the β_0 parameter is dropped from the notation but it is still included in the model.)

- Classical statistical studies: There is long tradition of looking at models like the following. First, generate a series of X-variables. Say, generate $x_1, x_2, \ldots, x_p$. Usually these are generated from normal distributions. They may or may not be correlated. Then, one specifies the "true model" and one generates the dependent variables, where only the first q variables are used to define y. So, let $y = \sum_{j=0}^{q} \beta_j x_j$. Note: the sum is only for the first q of the p independent variables. Then, one test the different model selection criteria. So, the selection method is given y and $x_1, \ldots, x_p$ and the method will pick a subset of the p variables. The different methods are compared to see which method is best at picking up the correct subset of x variables. Possible problem: the whole event is rather artificial. The x's that are measured might be surrogates for the actually generators of y. Usually, there is a lot of correlation in the x's due to several of the x's being indirect measure of the underlying "factors" that are effecting the y outcome.

- Causal modeling of observational variables. There are different, related methods. One idea is that one should try and make the analysis of observational data look like a clinical trial/experimental design. This is can be done by some type of matching observations. One way is to use something know as propensity scores to control for the differences between the observations who are assigned different "treatments" (that is, the variable of interest). Also, instead of matching methods, one can do various types of adjustments based on these types of variables. Much of this work is by Donald Rubin and his many disciples. A second, related method is to look at models which constructs the causal path of the different variables. These techniques consider variables which are mediating, moditiers, or confounders. How to choose these variables or how to do deal with a collection of highly correlated variables might cause issues. One of the main figures in this work is Judea Pearl. (Note: I'm not an expert on these techniques... I do want to learn more about these techniques.)

- Epidemiologies: I believe that the current method is to use causal modelling techniques. (Again, I'm am not an expert in these techniques, but you should know people are thinking about these things.) I believe that idea is to carefully think about the question of interest. So, one maybe interested in some outcome variable, call it Y, and how this is affected by another variable, call it X. Then, one needs to look that other variables which might affect this relationship and worry about variables which are modifying variables or variables which are "on the causal path". Variable reduction is usually an issue, since in most studies, there are several variables which are closely related which have been measured. This does tend to be a bit of an art form and a science and they are trying to do the right thing. There is generally a really disdain automatic methods. Problem: If one looks at the CV's of some of these people. They might have 30 papers for one data set. The

role of which is the "variable of interest" and which are the "control variables" will rotate through the 30 papers. So, perhaps these pure notion of just "testing the variable of interest" is not really true. Problem 2: by rotating through 30 different papers, it is not clear that any one is trying to understand the web of relationships between the different x variables and the relation of this web to the outcome.

- (Some) Economist[3] but not all economist: I am not sure if I am simply misunderstanding what is going on or if this really is "doctrine" for a school of economic data analysis. I have been told that what they do is just declare the model of interest. They will say that they are interest in looking at the effect of model x_1 on y and there model will include the variables, say, $x2, x3,$ and x_4. I will be told that this is based on theory and expert opinion. So, they will fit the model:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4.$$

  Then, they will stop and be done. They won't spend anytime "looking around". Problem: This method does not explore the interrelationship between all the different variables in the model. Perhaps this prevents "data dredging". However, it seems to me that one could/should check what is going on between the variables in the data set.

5 Some Bayesian methods for Model Selection

- Direct calculation of the Bayes Factor. Problem: this calculation can be extremely difficult. Many methods which are based on MCMC might be unstable for your particular problem. Some of the methods used are called "bridge samplers" and "path samplers"(???)
- Spike/slab methods: can get posterior probability of model/models and the Bayes Factors. Problem: need to be careful of priors or things won't converge.
- Reversable jump metropolis methods. An advanced sampling technique. There is an add-on to WinBugs which could be used.
- pseudo-Bayes factors. Uses the "leave-one-out" calculation of the likelihood. Just an approximation to the Bayes factor.
- Stochastic Search methods. When the model space is extremely large, assume that a "spike/slab" method would not be effective in moving effectively over the sample space. So, don't assume that he probability of visiting a model is an accurate estimate of the posterior probability of that model. Instead, use the MCMC method to (hopefully) find good potential models. Then, one judges between models by using some other way of judging how good the model is. So, then one might look at the deviance of the model or perhaps calculate the DIC for some of the good looking models.

6 Example 1: Sales data

The following section looks at an example of sales data.[4] The basic problem as presented is that the maker of asphalt roofing shingles is interested in the relationship between sales for a particular year and factors that obviously influence sales: promotional accounts, number of active accounts, number of competing

[3]Not sure of the exact story here. Some economist in health care and other places claim this. However, economist who I know from their statistical work don't seem to follow this pattern. I'm still trying to sort this out...

[4]The example is from Myers (a regression book... full reference needed, page 174). He cites that the data is originally from Neter, Wasserman, and Kutner, *Applied Linear Regression models*, (Homewood, Illinois: Richard D. Irwin, Inc), p439.

brands, and district potential for the sales district. In the below discussion, these four variables are simply called x1, x2, x3, and x4. [5]

In this section, a frequentist analysis is shown. Then, a Bayesian method is demonstrated. The goal here is to find a subset of the four x variables to describe the analysis.

6.1 A Typical Frequentist Analysis

The following is some SAS code to analysis this data:

```
x 'cd c:\mike workstation\bayescourse5\modelselect';

data sales;
  input x1 x2 x3 x4 y;
  cards;
5.5 31 10 8 79.3
2.5 55 8 6 200.1
8 67 12 9 163.2
3 50 7 16 200.1
3 38 8 15 146
2.9 71 12 17 177.7
8 30 12 8 30.9
9 56 5 10 291.9
4 42 8 4 160
6.5 73 5 16 339.4
5.5 60 11 7 159.6
5 44 12 12 86.3
6 50 6 6 237.5
5 39 10 4 107.2
3.5 55 10 4 155
;

proc means data=sales;
  var x1-x4 y;
run;

proc reg data=sales;
  model y= x1-x4/selection=rsquare cp jp bic;
run;
```

Some preliminary comments on the above program:

- The line that starts with the word "data" begins a series of commands which input the data in to the statistical program SAS. The first four columns of numbers are the X-variables. The last column is the y variable.
- The line that starts with the words "proc means" is the beginning of a series of commands which calculates some basic summary statistics for each of the variables.

The last three lines of code provides information on the analysis which is used to find good subsets of x variables.

[5] Hmmm, well, in real applications with real collaborators, labeling your variables with obscure names like this is to be strongly discouraged.

- The "proc reg" procedure in SAS is one of the procedures that does linear regression. The statement: "model y=x1-x4/selection = rsquare cp jp bic;"
- The "model y=x1-x4" signifies that the full model is the variable "y" as the dependent varaible and that the variables x1, x2, x3, and x4 are the four dependent variables.
- The commands: "/selection=rsquare cp jp bic" tell SAS that we want to look at all the possible subsets (with some tuning commands on that which I won't get into). There is a known algebraic trick that allows one to find the best m subsets of each model size. This algorithm is used here. The series of letters: "cp jp bic" tell SAS to compare each of these models using different criteria of "best". The term "cp" refers to Mallows Cp and the term "bic" refers to the BIC statistics which refers to the Bayesian Information Criteria.

The first output from SAS is the output of the Proc Means commands. The following is this output:

The SAS System

The MEANS Procedure

Variable	N	Mean	Std Dev	Minimum	Maximum
x1	15	5.1600000	2.0576686	2.5000000	9.0000000
x2	15	50.7333333	13.5302131	30.0000000	73.0000000
x3	15	9.0666667	2.5485757	5.0000000	12.0000000
x4	15	9.4666667	4.6578147	4.0000000	17.0000000
y	15	168.9466667	79.9763882	30.9000000	339.4000000

This results just gives us the simple summary statistics for the different variables in the data sets. Next, the output from the Proc Reg is below:

The REG Procedure
Model: MODEL1
Dependent Variable: y

R-Square Selection Method

Number in Model	R-Square	C(p)	BIC	J(p)	Variables in Model
1	0.6373	1228.460	115.3028	2831.8411	x3
1	0.5010	1693.971	120.0596	3895.4123	x2
1	0.0928	3088.650	128.9941	7081.8859	x4
1	0.0120	3364.884	130.2715	7713.0084	x1
2	0.9940	11.4013	59.8751	53.4662	x2 x3
2	0.6805	1082.670	113.4388	2860.9714	x3 x4
2	0.6417	1215.316	115.1441	3208.6002	x1 x3
2	0.5090	1668.699	119.8367	4396.7923	x1 x2
2	0.5042	1685.024	119.9811	4439.5778	x2 x4
2	0.1151	3014.641	128.6326	7924.1413	x1 x4

```
       3      0.9970     3.4075      56.6484      31.4078   x1 x2 x3
       3      0.9940    13.3770      61.3357      61.4940   x2 x3 x4
       3      0.6896  1053.643      113.0386    3200.8178   x1 x3 x4
       3      0.5140  1653.770      119.7036    5011.8868   x1 x2 x4
-------------------------------------------------------------------------------
       4      0.9971     5.0000      59.4086      34.9430   x1 x2 x3 x4
```

The first column specifies how many variables are in the model. The table is order by first showing all the models with 1 variable, then with 2, etc. Within the models of the same size, the models are order by how well they fit. For models of the same size, the measures of fit are all ordered the same. This is because, the different criteria are all based on the sum of 1) some function of the residual sum of squares and 2) some penalty for the number of parameters. So, if they all the models have the same number of parameters, then the order is based on the residual sum of squares[6]

The middle four columns are the values of the R-square, Cp, BIC, and the Jp statistics. The last three are there because they were requested in the SAS commands. There are many different statistics that could have been requested. For R-square, bigger is better. However, for the R-square statistics, there is no penalty for the number of parameters in the model, so the full model with all the variables will always have the largest R-square. For the other three statistics–Cp, BIC, and JP– the smaller the number the better. So, the goal is to find a collection of models which have the smallest values of these statistics Note: it is better to find the collection of good models then to simply find the "best model". If there are several models which are close, then you should identify them.

The last column of numbers tell us what variables are in the model for that row in the table.

In looking at the table, we see that there are perhaps three or four models that seem the best. Looking at the BIC statistics, we see the following four potential models:

# in model	BIC	Variables
2	59.8751	x2 x3
3	56.64.84	x1 x2 x3
3	61.3357	x2 x3 x4
4	59.4086	x1 x2 x3 x4

For the BIC, note that:

$$\text{(Difference in BIC)} \approx -2 \times \log(\text{Bayes Factor}).$$

So, when comparing the difference in the BIC statistics, one can use the same scale as one would use for the Bayes factor. So, a difference of between 2 to 6 would be considered "positive" evidence of difference and a difference between 6 and 10 would be considered strong. Therefore, perhaps one might consider the model (x1, x2, x3) to be the best model and that there is some positive evidence that it is better than the other model. The models (x2, x3) and (x1, x2, x3, x4) are about the same and these two models are not much different than model (x2, x3, x4).

Beside using these basic "cutpoints" to look at the model, I would suggest that one still keep in mind the different models. Basically, from this table, we can see that the models with x2 and x3 are the most important models. Considering this as a "base" model, I would suggest that the result suggest that the adding either x1 or x4 to the model (x2, x3) appears to be recommended.

6.2 Bayes Model Selection with a Spike/Slab

There are several papers written on the idea of a "spike/slab" model. Some important references include: Mitchell and Beauchamp 1988 JASA, Carlin and Chib 19xx, George and McCulloch 1993 JASA (note:

[6]The residual sum of squares is used for linear models. For general linear models like logistic regression or Poisson regression, the deviance takes the place of the residual sum of squares.

I know that these are incomplete reference... you can find them easily enough if you want...) I want to concentrate on the method as proposed by kuo and Mallick (19xx??) which is has been sent with this document. First, I'll review the basic strategy of the spike/slab method and then review the main ideas proposed by Mallick and Kuo. For the remainder of the idea, please refer to the paper by Kuo and Mallick.

Before describing the spike/slab strategy, I first describe the basic, full model. For the full model, there are several independent (x) variables. Assume that there are p independent variables, $x_1, \ldots, x_p$. These are used to predict a dependent variable, y. So, the basic linear model is:

$$y = \sum_{j=0}^{p} \beta_j x_j,$$

where x_0 is assume here to be just the constant 1.

In this setting, model selection means that we wish to find a subset of the x variables which fit the y values as well as the full model. One way to do this is to add the parameters $\delta_1, \ldots, \delta_p$ to the model in the following equation:

$$y = \sum_{j=0}^{p} \beta_j \delta_j x_j,$$

where δ_j takes on the value 1 or 0. Note that when δ_j equals zero, then the variable x_j is effectively removed by the model. Therefore, one can define a subset as the vector $\delta = (\delta_1, \ldots, \delta_p)$. So, the vector δ has length p and is a string of 0's and 1's. In the Bayesian setting, the individual δ_j's have a prior distribution and the posterior distribution of δ_j gives the posterior probability that x_j is in the model. Also, since the vector δ define a subset of variables (which are the x_j's which correspond to non zero δ_j's). Also, the posterior probability of a particular vector δ is the posterior probability of the model which corresponds to 1's in that particular δ vector. For example if p=4, then the posterior probability of, say, the vector $\delta = (1, 1, 0, 0)$ corresponds to the posterior probability of the model (x_1, x_2).

(Aside: this is just describing a simple design matrix of the x's. When the design matrix includes interaction terms or includes terms for a polynomial fit, then one might need to be more clever in how the δ_j are set up. Some of this is discussed in the Kuo and Mallick paper.)

Another feature of this model is that the coefficient of a particular variable x_j is $\beta_j \delta_j$. If we assume that β_j has a continuous distribution (for both the prior and posterior), then when δ_j equals one the posterior distribution of the coefficient of x_j is the continuous distribution of β_j. This is the "slab". When δ_j equals zero, then the distribution of the coefficient of x_j is equal to zero. This is the spike.

It is possible that one might have a problem with this model if one uses too disperse a prior on the beta parameter. To see why this might be, imagine how the MCMC algorithm might work. To simply this thought experiment, consider the simplest case where p=1 and we are simply looking to see if we want the model with β_1 versus the model without β_1. Now, suppose we first look to sample δ_1 conditionally on knowing β_0 and β_1. Basically, the algorithm will assign δ_1 equal to zero or one depending on how close y is to either (β_0) or $(\beta_0 + \beta_1 x_1)$. That is, the amount that the value of β_1 improves the fit of $(\beta_0 + \beta_1 x_1)$ to y compared to fitting β_0 alone, then the greater the chance that δ_1 will be set to 1. For this to happen, the parameter β_1 will be in some small range of values. Suppose for the moment that the algorithm assign 0 to δ_1. When this happens, the parameter β_1 is not connected to either y or x_1. So, the sampler will then be sampling β_1 without the data. This means that the new β_1 is then sampled from the prior distribution. If the prior distribution is too disperse, then the sampled values of β_1 will be all over the place (with a disperse prior) and there will be little chance that β_1 will be assigned a value which will make $(\beta_0 + \beta_1 x_1)$ a better fit than β_0.

For the details of the Kuo and Mallick strategy, please see the paper. Here are some of the primary features of there suggested method:

1. The model is fit with a basic linear model with normal prior on the β_j's. The precision of the normal prior on the β_j is set to a value between $\frac{1}{16}$ and 4.

2. A Bernoulli prior is used as the priors for the δ_j parameters. If there is no preference for which variable is in the model, then one might use a prior Bernoulli distribution with parameter 0.5.

3. Standardize the x's and y term. That is, subtract off the mean and then divide by the standard deviation of each variable.

The reason for standardizing the parameters, is that the β_j parameters are then related to the (partial correlation? t-statistics??? something like that???) relationship between x_j and y. Since we are really interested in knowing if β_j is about zero or not, then having a prior that is not too disperse. (Well, I admit that there could be more explanation here... again, check Kuo and Mallick...)

6.3 Kuo-Mallick applied to Sales data

The following is the model file to do the Kuo and Mallick method to the Sales data:

```
model{
for( i in 1:15){

sx1[i]<- (x1[i]-5.16)/2.057
sx2[i]<- (x2[i]-50.73)/13.53
sx3[i]<- (x3[i]-9.07)/12.549
sx4[i]<- (x4[i]-9.47)/4.658
sy[i]<-  (y[i]-168.947)/79.976

sy[i]~dnorm(mu[i],tau)
mu[i]<- del[1]*beta[1]*sx1[i] + del[2]*beta[2]*sx2[i]+
   del[3]*beta[3]*sx3[i] + del[4]*beta[4]*sx4[i]

}

for(ix in 1:4){
beta[ix]~dnorm(0,tau0)
}

tau~dgamma(.5,.01)

abeta[1]<-beta[1]*2.057/79.976
abeta[2]<-beta[2]*13.53/79.976
abeta[3]<-beta[3]*12.549/79.976
abeta[4]<-beta[4]*4.658/79.976



for(k in 1:4){del[k]~dbern(pp)}

for(i1 in 1:2){
  for(i2 in 1:2){
   for(i3 in 1:2){
     for(i4 in 1:2){
          mod[i1,i2,i3,i4]<-equals((2-i1),del[1])*equals( (2-i2),del[2])*
                           equals( (2-i3),del[3])*equals( (2-i4),del[4])
}}}}
```

```
}
```

For this model, note that following:

- The first thing that is done in the "for" loop, is that the variables are standardized. That is the lines that start with the variables like `sx1` and `sx2`, etc. Note, that it would be more efficient to standardized the variables outside of Winbugs and then feed Winbugs the standardize variables. As written here, Winbugs will re-standardize the variables after every iteration which is obviously inefficient.
- Also, note the block of defined variables with names like `abeta[1]`, etc. These variables are the β parameters which multiply the unstandardized variables. That is, these are the real β's for the model. Again, it would be more efficient to do this outside Winbugs instead of creating these variables.
- Also, note the code which defines the variable `mod`. This variable is inside 4 nested "for" loops. This variable is the vector which corresponds to the δ vector discussed in the previous subsection. In this model, `mod` is a a 4 dimensional array. Each dimension corresponds to the inclusion/exclusion of one of the x_j variables in the model. More discussion on how to interpret this is included below.
- In this model, no value is specified for `pp` and `tau0`. These are the parameters for the prior on the δ parameters and the β_j parameters respectively. Therefore, these values are read as data. For the output given below, the parameter `pp` is set to .5 and the parameter `tau0` is set to .0625. The suggestion given by Kuo and Mallick is to set `tau0` to a value between .0625 and 4.

The following is the output from the WinBugs program:

```
Node statistics
 node  mean  sd  MC error 2.5\% median 97.5\% start sample
abeta[1] -2.724E-4 0.09684 6.212E-4 -0.1972 0.001343 0.1964 1 20000
abeta[2] 0.1022 0.02133 4.309E-5 0.09393 0.1021 0.1105 1 20000
abeta[3] -0.5477 0.01948 1.339E-4 -0.5852 -0.5478 -0.5094 1 20000
abeta[4] -3.462E-4 0.2319 0.001649 -0.4541 -6.888E-4 0.4555 1 20000
beta[1] -0.01059 3.765 0.02415 -7.666 0.05221 7.635 1 20000
beta[2] 0.6041 0.1261 2.547E-4 0.5552 0.6038 0.653 1 20000
beta[3] -3.491 0.1241 8.533E-4 -3.73 -3.491 -3.246 1 20000
beta[4] -0.005943 3.981 0.02831 -7.798 -0.01183 7.82 1 20000
del[1] 0.1162 0.3205 0.01139 0.0 0.0 1.0 1 20000
del[2] 0.9992 0.02827 8.024E-4 1.0 1.0 1.0 1 20000
del[3] 1.0 0.0 7.071E-13 1.0 1.0 1.0 1 20000
del[4] 0.007 0.08337 9.785E-4 0.0 0.0 0.0 1 20000
mod[1,1,1,1] 3.5E-4 0.01871 1.632E-4 0.0 0.0 0.0 1 20000
mod[1,1,1,2] 0.1158 0.32 0.01136 0.0 0.0 1.0 1 20000
mod[1,1,2,1] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[1,1,2,2] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[1,2,1,1] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[1,2,1,2] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[1,2,2,1] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[1,2,2,2] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[2,1,1,1] 0.00665 0.08128 9.559E-4 0.0 0.0 0.0 1 20000
mod[2,1,1,2] 0.8763 0.3292 0.01133 0.0 1.0 1.0 1 20000
mod[2,1,2,1] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
```

```
mod[2,1,2,2] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[2,2,1,1] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[2,2,1,2] 8.0E-4 0.02827 8.024E-4 0.0 0.0 0.0 1 20000
mod[2,2,2,1] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[2,2,2,2] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
tau 144.3 61.36 1.064 55.61 134.5 294.1 1 20000
```

In the above output from the `stat` command, let us first look at the `del` variables. The following is a cleaner version of the results:

δ_j	Prob of x_j in a model
del[1]	0.1162
del[2]	0.9992
del[3]	1.0
del[4]	0.007

So, from the above, we see that in the selected models, the variable x_1 appeared in 11% of the models, x_2 appeared in 99.92%, x_3 appeared in 100%, and x_4 appeared in 0.7% of the models.

In order to see the distribution of the models that were selected in the MCMC, see the below table which is a cleaner version of what appears from the `stat` command.

node	Prob	Model
mod[1,1,1,1]	3.5E-4	(x_1, x_2, x_3, x_4)
mod[1,1,1,2]	0.1158	(x_1, x_2, x_3)
mod[1,1,2,1]	0.0	(x_1, x_2, x_4)
mod[1,1,2,2]	0.0	(x_1, x_2)
mod[1,2,1,1]	0.0	(x_1, x_3, x_4)
mod[1,2,1,2]	0.0	(x_1, x_3)
mod[1,2,2,1]	0.0	(x_1, x_4)
mod[1,2,2,2]	0.0	(x_1)
mod[2,1,1,1]	0.00665	(x_2, x_3, x_4)
mod[2,1,1,2]	0.8763	(x_2, x_3)
mod[2,1,2,1]	0.0	(x_2, x_4)
mod[2,1,2,2]	0.0	(x_2)
mod[2,2,1,1]	0.0	(x_3, x_4)
mod[2,2,1,2]	8.0E-4	(x_3)
mod[2,2,2,1]	0.0	(x_4)
mod[2,2,2,2]	0.0	$(\cdot)$

The first thing to note is the structure of the array `mod`. This variable is a 4 dimensional array with 16 different elements. For each iteration of the MCMC, there is a unique combination of values for the vector `del`. For example, if the model that is selected in one step of the MCMC is the model (x_2, x_3), then the values of `del[2]` and `del[3]` are set to one and the value of the other `del`'s are set to zero. Now, for the `mod` variable, only one of the 16 different elements is set to 1 and all the others are set to zero. The way that `mod` is coded here, the first element of the dimension is set to 1 if that corresponding x_j is in the model. So, if the MCMC has selected the model (x_2, x_3), then only `mod[2,1,1,2]` is set to one and all the other elements are set to zero. Therefore, the percentage of times that the model (x_2, x_3) is selected by the MCMC is the average of the node `mod[2,1,1,2]`. Also, the percentage of times that the model is selected is the posterior probability of that model.

From the last table, we see that the model with the highest posterior probability is the model (x_2, x_3) with a posterior probability of 0.8763. This is followed by the model (x_1, x_2, x_3) with a posterior probability of 0.1158 and model (x_2, x_3, x_4) with a posterior probability of 0.0063. Most of the other models

Table 1: Comparing Sales models

	M1:(X1, X2, X3)	M2:(X2, X3)	M3:(X1, X3)	Difference (btw M1 and M2)
Chidev2	11.86	13.31	15.97	
deviance	-39.47	-31.68	29.22	
pseudoBF	-35.34	-28.86	32.98	6.49
DIC2	-32.25	-27.7	32	4.45
(pd)	(7.21)	(3.94)	(3.08)	
BIC	56.65	59.88	115	3.23
2log(BF)	2log(.1158)	2log(87.67)	log(≈ 0)	-4.05

are actually not even selected in the MCMC algorithm.

From the posterior probabilities, we can calculate the Bayes factor between different models. Note, that the prior probability of each δ_j parameter was a Bernoulli distribution with the the probability of being one of 0.5. So, the prior probability of each of the 16 models was equal. So, the prior odds between two different models is one. Therefore, the Bayes factor between two models is the ratio of the posterior probability of the two models. So, as an example, let us find the Bayes factor between the two leading models. That is, let us compare the model (x_2, x_3) versus the model (x_1, x_2, x_3). Therefore, the log(Bayes factor) can be calculated as follows:

$$\begin{aligned} \log(\text{BF}) &= \log\left(\frac{P(\text{model } x_2, x_3|\text{data})}{P(\text{model } x_1, x_2, x_3|\text{data})}\right) \\ &= \log\left(\frac{.8763}{.1158}\right) \\ &= \log(7.567) = 2.024 \end{aligned}$$

So, the log Bayes factor suggest that there is "some positive evidence" that the model (x_2, x_3) is better than the model (x_1, x_2, x_3).

Similarly, one can calculate the log Bayes factor comparing model (x_2, x_3) and model (x_2, x_3, x_4). The log Bayes factor between these two models is 4.88. So, this would be interpreted as "strong" evidence that the model (x_2, x_3) is better then the model (x_2, x_3, x_4).

6.4 Summary of Sales data

In the file "`SalesPsuedoBF DIC.txt`" and the associated output file, the psuedo Bayes factor and the DIC is calculated for the following models of sales: (X1, X2, X3), (X2, X3), and (X1, X3). The results are in Table 1.

Note that M3:(X1, X3) is the worse model. (Aside: it is easier to see what direction the better model is versus the worse model by looking at M3.)

When comparing models M1:(X1, X2, X3) to M2:(X2, X3), it appears that pseudo BF, DIC, and BIC provide similar results for this dataset. The difference is between 3 and 6 which seems to signify that there is positive evidence to prefer M1(X1, X2, X3) to M2(X2, X3). For those three statistics, note that we want the lower number. (Again, M3 is much, much bigger than the statistics for M1 or M2.) For the Bayes Factor, the statistics seems to go the other way.

7 Data example: estimate squid weight

In this example, one is wants to estimate the weight of a squid given different measurements on the squid. The variables that are used for the approximation are rostrum length (ros-len), wing length (wing-len), rostrum to notch (ros-not), notch to wing (not-wing), and the width.

Here is a summary of the results with the files and output that generate this summary following the table.

Number in Model	R-Square	BIC	Variables in Model	X variables	K-M Prob of model
1	0.9455	-8.0766	width	X5	.3704
2	0.9584	-10.5305	not-wing width	X4 X5	.2282
2	0.9575	-10.1776	ros-len width	X1 X5	.2230
3	0.9616	-9.1759	wing-len not-wing width	X2 X4 X5	.0343
3	0.9592	-8.3512	ros-len not-wing width	X1 X4 X5	.0191
3	0.9585	-8.1428	ros-not not-wing width	X3 X4 X5	.0150
3	0.9584	-8.1006	ros-len wing-len width	X1 X2 X5	.0123
3	0.9575	-7.8066	ros-len ros-not width	X1 X3 X5	.0120

Note that there is some agreement between the different analyses. The Bayesian model prefers the models (x_1), (x_4, x_5), and (x_1, x_5). The top models for the BIC are the models (x_4, x_5), (x_1, x_5), and (x_2, x_4, x_5).

7.1 Program files and output for the squid example

Below are the program files and output for the squid example:

```
/*  squid.sas
Bayes course
Michael Escobar
March 20, 2010

All the files associated with the Squid example.

These files are used for example of models selection.
To run these different programs, you will need to cut this file into the
different files.

============================================================

Basic frequentist analysis via SAS:
------------------------------------------------------------
FILE: Squid.sas
Program file to run program in SAS.
------------------------------------------------------------

*/


options ls=79  ps=60;

x 'cd c:\mike workstation\bayescourse5\modelselect';

data squid;
  infile squid;
  input ros_len wing_len ros_not not_wing width wght;
run;
```

```
proc means data=squid;
run;


proc reg data=squid covout outest=xpxi;
  model wght=ros_len wing_len ros_not not_wing width/selection=rsquare best=8 jp cp bic;
run;

/*

Note: here is the data file (squid.dat):

1.31 1.07 0.44 0.75 0.35 1.95
1.55 1.49 0.53 0.90 0.47 2.90
0.99 0.84 0.34 0.57 0.32 0.72
0.99 0.83 0.34 0.54 0.27 0.81
1.05 0.90 0.36 0.64 0.30 1.09
1.09 0.93 0.42 0.61 0.31 1.22
1.08 0.90 0.40 0.51 0.31 1.02
1.27 1.08 0.44 0.77 0.34 1.93
0.99 0.85 0.36 0.56 0.29 0.64
1.34 1.13 0.45 0.77 0.37 2.08
1.30 1.10 0.45 0.76 0.38 1.98
1.33 1.10 0.48 0.77 0.38 1.90
1.86 1.47 0.60 1.01 0.65 8.56
1.58 1.34 0.52 0.95 0.50 4.49
1.97 1.59 0.67 1.20 0.59 8.49
1.80 1.56 0.66 1.02 0.59 6.17
1.75 1.58 0.63 1.09 0.59 7.54
1.72 1.43 0.64 1.02 0.63 6.36
1.68 1.57 0.72 0.96 0.68 7.63
1.75 1.59 0.68 1.08 0.62 7.78
2.19 1.86 0.75 1.24 0.72 10.15
1.73 1.67 0.64 1.14 0.55 6.88

*/

------------------------------------------------------------
FILE: Squid.lst
Program file to run program in SAS.
------------------------------------------------------------
                              The SAS System                                1
                                                  23:35 Saturday, March 11, 2006

                             The MEANS Procedure

Variable     N            Mean         Std Dev         Minimum         Maximum
-------------------------------------------------------------------------------
ros_len     22       1.4690909       0.3568368       0.9900000       2.1900000
wing_len    22       1.2672727       0.3236427       0.8300000       1.8600000
```

ros_not	22	0.5236364	0.1331503	0.3400000	0.7500000
not_wing	22	0.8572727	0.2277463	0.5100000	1.2400000
width	22	0.4640909	0.1494963	0.2700000	0.7200000
wght	22	4.1950000	3.2065762	0.6400000	10.1500000

The REG Procedure
Model: MODEL1
Dependent Variable: wght

R-Square Selection Method

Number of Observations Read 22
Number of Observations Used 22

Number in Model	R-Square	C(p)	BIC	J(p)	Variables in Model
1	0.9455	5.7794	-8.0766	0.6418	width
1	0.9223	15.9084	-1.6110	0.9152	ros_len
1	0.9140	19.5335	0.2770	1.0131	ros_not
1	0.8857	31.8621	5.6640	1.3459	not_wing
1	0.8812	33.8244	6.4085	1.3988	wing_len
2	0.9584	2.1456	-10.5305	0.5370	not_wing width
2	0.9575	2.5565	-10.1776	0.5492	ros_len width
2	0.9486	6.4172	-7.1763	0.6635	wing_len width
2	0.9477	6.8317	-6.8821	0.6757	ros_not width
2	0.9386	10.7846	-4.2836	0.7927	ros_len ros_not
2	0.9288	15.0819	-1.7983	0.9199	ros_not not_wing
2	0.9229	17.6524	-0.4436	0.9960	ros_len wing_len
2	0.9226	17.7619	-0.3878	0.9992	ros_len not_wing
3	0.9616	2.7354	-9.1759	0.5437	wing_len not_wing width
3	0.9592	3.8138	-8.3512	0.5788	ros_len not_wing width
3	0.9585	4.0951	-8.1428	0.5879	ros_not not_wing width
3	0.9584	4.1525	-8.1006	0.5898	ros_len wing_len width
3	0.9575	4.5563	-7.8066	0.6029	ros_len ros_not width
3	0.9488	8.3265	-5.2706	0.7254	wing_len ros_not width
3	0.9420	11.3275	-3.4641	0.8229	ros_len wing_len ros_not
3	0.9386	12.7821	-2.6422	0.8701	ros_len ros_not not_wing
4	0.9630	4.1582	-6.9128	0.5772	ros_len wing_len not_wing width
4	0.9619	4.6037	-6.6296	0.5932	wing_len ros_not not_wing width

```
       4     0.9593    5.7556     -5.9166      0.6343  ros_len ros_not not_wing
                                                       width
       4     0.9588    5.9859     -5.7771      0.6425  ros_len wing_len ros_not
                                                       width
       4     0.9432   12.7951     -1.9987      0.8858  ros_len wing_len ros_not
                                                       not_wing
-------------------------------------------------------------------------------
       5     0.9633    6.0000     -4.2645      0.6298  ros_len wing_len ros_not
                                                       not_wing width




============================================================
Winbugs analysis


This information is used in Winbugs for the method suggested by Kuo and Mallick
------------------------------------------------------------
Basic information for Winbugs...
------------------------------------------------------------


============================================================
Winbugs analysis

This first series of files uses Winbugs in a method suggested by Kuo and Mallick
------------------------------------------------------------
FILE: Squid3M.txt

Model file for Winbugs...
------------------------------------------------------------

model{
#  Data from Myers, 1990, Classical and Modern Regression
#   with Applications, PWS-Kent Publishing Company: Boston, pg 99,
# He cites the source as: Rudolf Freund, SAS Tutorial, "Regression with SAS
#  with Emphasis on Proc REG"  paper presented at Eight Annual SAS Users Group
#  International Conference, New Orleans, Louissiana, January 16-19, 1083).
#

for( i in 1:22){

sRosL[i]<-(RosL[i]-1.469)/0.3568
sWingL[i]<-(WingL[i]-1.267)/0.3236
sRos2Not[i]<-(Ros2Not[i]-0.5236)/0.1332
sNot2Wing[i]<-(Not2Wing[i]-0.8573)/0.2277
swidth[i]<-(width[i]-0.464)/0.1495
swgt[i] <- (wgt[i]-4.195)/3.2066
```

```
swgt[i]~dnorm(mu[i],tau)
mu[i]<- del[1]*beta.r*sRosL[i] + del[2]*beta.w*sWingL[i]+
   del[3]*beta.rn*sRos2Not[i] + del[4]*beta.nw*sNot2Wing[i] +
   del[5]*beta.wd*swidth[i]

}

#beta.0~dnorm(0,tau0)
beta.r~dnorm(0,tau0)
beta.w~dnorm(0,tau0)
beta.rn~dnorm(0,tau0)
beta.nw~dnorm(0,tau0)
beta.wd~dnorm(0,tau0)
tau~dgamma(.5,.01)

sbeta.r<-beta.r*0.3568
sbeta.w<-beta.w*0.3236
sbeta.rn<-beta.rn*0.1332
sbeta.nw<-beta.nw*0.2277
sbeta.wd<-beta.wd*0.1495

for(k in 1:5){del[k]~dbern(pp)}

for(i1 in 1:2){
  for(i2 in 1:2){
   for(i3 in 1:2){
     for(i4 in 1:2){
       for(i5 in 1:2){
          mod[i1,i2,i3,i4,i5]<-equals((2-i1),del[1])*equals( (2-i2),del[2])*
                                              equals( (2-i3),del[3])*equals( (2-i4),del[4])*
                                               equals( (2-i5),del[5])
}}}}}

}


------------------------------------------------------------
FILE: Squid3S.txt


------------------------------------------------------------


#  note, here is the data file (with out the "#" of course):
#
#   list(t = c(94.3, 15.7, 62.9, 126, 5.24, 31.4, 1.05, 1.05, 2.1, 10.5),
#   x = c(5,1,5,14, 3,19,1,1, 4,22), N = 10)


# Note, here is the initialization file.  If you don't initialize y.rep, then
#  you will need to gen.inits() command in script file
```

```
#
#  list(alpha=1,beta=1,theta=c(1,1,1,  1,1,  1,1,1,  1,1),
#       x.rep=c(1,1,1,  1,1,  1,1,1,  1,1) )


-----------------------------------------------------------------
FILE: SquidD1.txt

Data file for Winbugs...
(Aside, for Winbugs, needed data names such as RosL[] instead of RosL)
-----------------------------------------------------------------

RosL[] WingL[] Ros2Not[] Not2Wing[] width[] wgt[]
1.31 1.07 0.44 0.75 0.35 1.95
1.55 1.49 0.53 0.90 0.47 2.90
0.99 0.84 0.34 0.57 0.32 0.72
0.99 0.83 0.34 0.54 0.27 0.81
1.05 0.90 0.36 0.64 0.30 1.09
1.09 0.93 0.42 0.61 0.31 1.22
1.08 0.90 0.40 0.51 0.31 1.02
1.27 1.08 0.44 0.77 0.34 1.93
0.99 0.85 0.36 0.56 0.29 0.64
1.34 1.13 0.45 0.77 0.37 2.08
1.30 1.10 0.45 0.76 0.38 1.98
1.33 1.10 0.48 0.77 0.38 1.90
1.86 1.47 0.60 1.01 0.65 8.56
1.58 1.34 0.52 0.95 0.50 4.49
1.97 1.59 0.67 1.20 0.59 8.49
1.80 1.56 0.66 1.02 0.59 6.17
1.75 1.58 0.63 1.09 0.59 7.54
1.72 1.43 0.64 1.02 0.63 6.36
1.68 1.57 0.72 0.96 0.68 7.63
1.75 1.59 0.68 1.08 0.62 7.78
2.19 1.86 0.75 1.24 0.72 10.15
1.73 1.67 0.64 1.14 0.55 6.88
END


-----------------------------------------------------------------
FILE: Squid3D1.txt

Data file for Winbugs...
-----------------------------------------------------------------

list(pp=.5,tau0=.0625)

-----------------------------------------------------------------
FILE: Squid3I1.txt

Parameter initialization file for Winbugs...
-----------------------------------------------------------------
```

```
list(beta.r=1, beta.w=1, beta.rn=1, beta.nw=1, beta.wd=1,tau=.01)

------------------------------------------------------------
FILE: Squid30.txt

Output file for this analysis
------------------------------------------------------------

Node statistics
 node  mean  sd  MC error 2.5% median 97.5% start sample
abeta[1] -2.724E-4 0.09684 6.212E-4 -0.1972 0.001343 0.1964 1 20000
abeta[2] 0.1022 0.02133 4.309E-5 0.09393 0.1021 0.1105 1 20000
abeta[3] -0.5477 0.01948 1.339E-4 -0.5852 -0.5478 -0.5094 1 20000
abeta[4] -3.462E-4 0.2319 0.001649 -0.4541 -6.888E-4 0.4555 1 20000
beta[1] -0.01059 3.765 0.02415 -7.666 0.05221 7.635 1 20000
beta[2] 0.6041 0.1261 2.547E-4 0.5552 0.6038 0.653 1 20000
beta[3] -3.491 0.1241 8.533E-4 -3.73 -3.491 -3.246 1 20000
beta[4] -0.005943 3.981 0.02831 -7.798 -0.01183 7.82 1 20000
del[1] 0.1162 0.3205 0.01139 0.0 0.0 1.0 1 20000
del[2] 0.9992 0.02827 8.024E-4 1.0 1.0 1.0 1 20000
del[3] 1.0 0.0 7.071E-13 1.0 1.0 1.0 1 20000
del[4] 0.007 0.08337 9.785E-4 0.0 0.0 0.0 1 20000
mod[1,1,1,1] 3.5E-4 0.01871 1.632E-4 0.0 0.0 0.0 1 20000
mod[1,1,1,2] 0.1158 0.32 0.01136 0.0 0.0 1.0 1 20000
mod[1,1,2,1] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[1,1,2,2] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[1,2,1,1] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[1,2,1,2] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[1,2,2,1] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[1,2,2,2] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[2,1,1,1] 0.00665 0.08128 9.559E-4 0.0 0.0 0.0 1 20000
mod[2,1,1,2] 0.8763 0.3292 0.01133 0.0 1.0 1.0 1 20000
mod[2,1,2,1] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[2,1,2,2] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[2,2,1,1] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[2,2,1,2] 8.0E-4 0.02827 8.024E-4 0.0 0.0 0.0 1 20000
mod[2,2,2,1] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
mod[2,2,2,2] 0.0 0.0 7.071E-13 0.0 0.0 0.0 1 20000
tau 144.3 61.36 1.064 55.61 134.5 294.1 1 20000
```