Model Selection in Linear Regression

Basic Ideas

"Model Selection" in linear regression attempts to suggest the best model for a given purpose.

Recall that the two main purposes of linear regression models are:

- **Estimate the effect of one or more covariates** while adjusting for the possible confounding effects of other variables.
- **Prediction** of the outcome for the next set of similar subjects

Also, we must again recall the famous quite from George Box:

"All models are wrong, but some are useful."

The above quote is important here for two reasons:

- If we are looking to estimate effects, it reminds us that we do not need to find a single "final model", and draw all conclusions from that model alone. Since no single model is correct, we are free to estimate a large number of different models, and draw overall conclusions about the effect of a variable, confounding, and so on, from that model.

  So, in some sense, model selection is not relevant to estimating effects, where we will almost always want to look at more that one model, sometimes perhaps as many as 20 or 30 models will be helpful in drawing overall conclusions.

  We have already seen examples of this when we investigated confounding. Nevertheless, it turns out that a Bayesian program for model selection will, because of the way its output is formatted, be very useful for investigating effects. Thus, in learning a program for model selection, we will simultaneously see a way to easily draw overall conclusions for parameter estimation in the face of confounding.

- If we are looking at prediction, then it may make sense to ask "what is the best model that gives 'optimal' predictions for future subjects."

  Here again, we will see that finding a single model is often sub-optimal, and in fact, taking a Bayesian average over many models produces better predictions than any single model.

  There is no widely accepted model building strategy. We will look at some of the most common methods.

Model Selection for Future Predictions

Problem: We wish to predict Y using potential predictor variables $X_1, X_2, \ldots, X_p$.

Challenge: Which subset of the $X_1, X_2, \ldots, X_p$ potential predictors should be included in the model for "best" predictions?

Frequentist Approaches:

- Backwards selection
- Forwards selection
- Backwards and Forwards selection
- All subsets selection
- AIC criterion
- Mallows C_p criterion
- R^2 criterion
- Adjusted R^2 criterion
- PRESS$_p$ criterion

Bayesian Approaches:

- Bayes Factors
- BIC criterion
- DIC criterion

We will now take each of these and define them. Following that, we will compare how they each actually work in practice, including programming them in R.

Backwards Selection: First run a model with all covariates, $X_1, X_2, \ldots, X_p$ included in the model. Then, check to see which of the covariates has the largest p-value, and eliminate it from the model, leaving $p - 1$ indepedent variables left in the model. Repeat this procedure with those that are left, continually dropping variables until some stopping criterion is met. A typical criterion is that all p-values are above some threshold, like $p > 0.1$.

Forwards Selection: First run a model with no covariates, included in the model, i.e., intercept only. Then, run p separate models, one for each of the possible independent variables, keeping track of the p-values each time. At the next step, consider a model with a single variable in it, the one with the lowest p values at the first step. Repeat this procedure, so that at the second step, consider all models that have two parameters in it, the one selected at the first step, and all others, one at a time, and create the second model as the one where the second value has the smallest p value, and so on. Continue to add variables until some stopping criterion is met. A typical criterion is that all p-values left at some stage are above some threshold, like $p > 0.15$, so no more new parameters are added.

Backwards and Forwards Selection: At each stage, consider both dropping and/or adding variables, checking some criterion (e.g., based again on some p-value thresholds). A combination of the above two strategies.

All subsets regression: Alternative to backwards/forwards procedures, a generic term which describes the idea of calculating some "fit" criterion over all possible models. We will see some of these below. In general, if there are p potential predictor variables, there will be 2^p possible models. For example, if here are five possible X variables, there will be $2^5 = 32$ models, and so on.

AIC criterion: Stands for Akaikies Information Criterion. For each model, calculate:

$$AIC = n\ln(SSE) - n\ln(n) + 2p$$

where SSE is the usual residual sum of squares from that model, p is the number of parameters in the current model, and n is the sample size. After doing this for all possible models, the "best" model is the one with the smallest AIC.

Note that the AIC is formed from three terms: The first is a measure of fit, since $n\ln(SSE)$ is essentially the sum of squared residuals. The second term, $n\ln(n)$ is a constant, and really plays no role in selecting the model. The third term, $2p$ is a "penalty" term for adding more terms to the model. This is because the first term *always* decreases as more terms are added into the model, so this is needed for "balance".

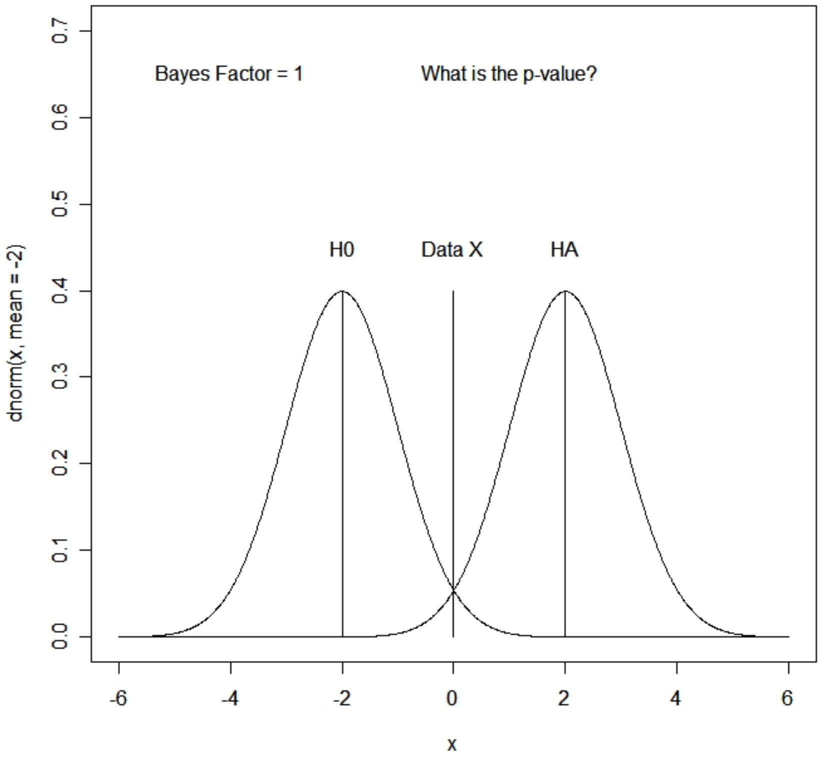
Mallows C_p criterion: Attempts to measure bias from a regression model. See textbook for full description, we omit details here.

R^2 criterion: Choose the model with largest R^2. In general, this model will simply be the largest model, so not a very useful criterion. Can be helpful in choosing among models with the same numbers of included parameters. Not further discussed here.

Adjusted R^2 criterion: As above, but Adjusted R^2 penalizes for numbers of parameters, so largest model not necessarily always best. Generally selects models that are too large, because "penalty" is too small. Not further discussed here.

PRESS$_p$ criterion: PRESS stands for Prediction sum of squares. Similar to SSE, but now calculates residual errors from a model that *leaves out* each variable, in turn. Not penalized for having more parameters, but largest model *not* always chosen. Still, other criteria tend to have better properties, so not discussed further here.

Bayes Factors: Consider the following picture:



The Bayes Factor = 1 on the previous page becasue it is defined as:

$$BF = \frac{Pr\{\text{data}|\text{model}_1\}}{Pr\{\text{data}|\text{model}_2\}}$$

If model$_i$ is not fully determined because of unknown parameters, then the Bayes Factor is still defined as

$$BF = \frac{Pr\{\text{data}|\text{model}_1\}}{Pr\{\text{data}|\text{model}_2\}}$$

but now where $Pr\{\text{data}|\text{model}_i\}$ is defined as

$$Pr\{\text{data}|\text{model}_i\} = \int (\text{likelihood} \times \text{prior})\, d\theta_i$$

where θ_i represents the vector of all unknown parameters for model i. What the above integral really means is that when there are unknown parameters in the terms in the definition of the Bayes Factor, we integrate them out (i.e., like an average) over the prior distribution for these parameters.

Problem: These can be hard to calculate, as they can involve high dimensional integrals. So, we can approximate Bayes Factors by the BIC, see below.

BIC criterion: Stands for Bayesian Information Criterion, sometimes called the SBC for Schwartz Bayesian Criterion. For each model, calculate:

$$BIC = n \ln(SSE) - n \ln(n) + \ln(n)p$$

where SSE is the usual residual sum of squares from that model, p is the number of parameters in the current model, and n is the sample size. After doing this for all possible models, the "best" model is the one with the smallest BIC.

Note the similarity between AIC and BIC, only the last term changes. We will compare the properties of these two criteria in detail later.

Details are omitted here (see article by Raftery), but it can be shown that the BIC is related to an approximate Bayes factor, from a model with low information prior distributions (equal to one prior observation centered at the null value of zero for each coefficient).

In some ways the best criterion to use for predictions, in large part because it leads to model averaging, see Raftery for details. We will extensively use Raftery's program called bic.glm for the rest of the year, extremely useful for model selection for prediction *and* for estimating effects.

DIC criterion: Stands for "Deviance Information Criterion". Similar to the BIC, but designed for hierarchical models, estimates an "effective number of parameters" for hierarchical models. Beyond the scope of this course, but beware of its existence if you need to select a hierarchical model.

Comparing the various criteria

Comments on the frequentist approaches

- Backward and forward selection procedures will tend to give different models as “best”. The conclusions are dependent on the order in which you try out models.
- P-values may be poorly approximated in small sample sizes, but most software packages ignore this.
- Because so many tests are done, there is a large danger of multiple testing, so unimportant independent variables may enter the model.
- For the same reasons, confidence intervals from models selected by backward/forward procedures tend to be too narrow.
- Backward and forward selection procedures are based on p-values, and so have all of the usual problems associated with them, plus an additional problem: In these methods, the next step depends on what happened in the previous step, but the calculated p-values ignore this, and so are not interpretable in the usual way as “true” p-values.
- Also, p-values assume only two models are being compared, for example we compare

  $$Y = X_1 + \ldots + X_{p-1} + X_p$$

  is compared to

  $$Y = X_1 + \ldots + X_{p-1}$$

  But in reality, there are many other models simultaneously being considered.
- P-values tend to exaggerate the “weight of evidence”, **leading to models that are too large**, in general (we will see an example of this later).
- These methods tend to select a “single best model”, but what if two or more models are equally plausible?
- Prior knowledge about covariates is ignored, wasteful of information when good information exists.
- Models must be “nested”, i.e., cannot test the following two models:

  $$Y = X_1 + X_3 + X_4$$

and

$$Y = X_1 + X_2$$

- Problems with generalizability of the models, because they are in general too large.
- Methods, including the AIC, ignore the sample size, again tending towards models that are too large.

Comments on the Bayesian approaches

- Conclusions independent of the order in which you try out models.
- Models need not be nested to be compared.
- Prior knowledge, when available, can be incorporated in the inferential process.
- Very simple to use, as we will see, there are easy to use R programs that provide extremely useful information in addition to implementing the criteria automatically.
- Bayes Factors and the BIC lead easily to "model averaging", shown to provide "optimal predictions" if two or more models are plausible.
- Because models tend not to be too large, these methods tend to lead to models that better generalize to other populations (i.e., they avoid "over-fitting" the data).
- Sample size is considered in the calculations (compare AIC to BIC, where it is like AIC has a sample size of 2, regardless of real sample size.).

Comparing the AIC to the BIC

AIC	BIC
• AIC tends to have models that are "too big", good for prediction (perhaps), but not so good for understanding whether specific covariates are important or not. • Tends to overfit. • Not consistent: Even with an infinite sample size, will not necessarily converge to correct model, tends to remain too big. • Performs better for complex models "Truth is high dimensional". • Usable for non-nested models. • Equivalent to a stepwise procedure with a critical value of 15.7%. • Not equivalent to a Bayes Factor, so no natural way to get model probabilities, so no model averaging.	• BIC tends to have models that are "too small", but still have optimal predictive properties. Will not exaggerate importance of covariates (sometimes too conservative). • Tends to underfit (but see next point!). • Consistent, asymptotically will return the correct model. • Performs better for simple models, "Truth is low dimensional". • Usable for non-nested models. • Not related to any significant tests. • Asymptotically equivalent to a Bayes Factor, so easily leads to model probabilities, so easy to perform model averaging (done automatically in BIC software for R).

Example of using the R BMA package for Bayesian model selection

```
#  Here is an example of using the BMA model selection program

#  First, we will generate some data:

x <- rnorm(1000, mean=0, sd=4)
z <- rnorm(1000, mean=5, sd=20)
w <- rnorm(1000, mean=-3, sd=4)
y <- rnorm(1000, mean = 3*x + 2*z, sd=25)

#  Y will be our independent variable, and we will create a matrix for
#  the other variables as follows:

data.matrix <- matrix(c(x,w,z), byrow=F, nrow=1000)
```

```
#  Note that both x and z contribute to y, but not w.

#  Now run BMA program, which is downloadable from the R web site in the
#  usual way (i.e., from R program, use the packages menu item, find
#  BMA, then download, and "load package")

#  There are many functions we can use, we will illustrate simple linear
#  regression model selection here using bicreg.

output<- bicreg(data.matrix, y)

output


# Call:
# bicreg(x = data.matrix, y = y)
#
#
# Posterior probabilities(%):
#   X1    X2    X3
# 100.0   6.8 100.0
#
# Coefficient posterior expected values:
# (Intercept)           X1           X2           X3
#    0.56763      3.19356     -0.01756      1.97760

#  We can also get some more info:

 output$postprob
[1] 0.93150804 0.06849196

output$namesx
[1] "X1" "X2" "X3"

output$label
[1] "X1X3"   "X1X2X3"

output$r2
 [1] 72.719 72.765

output$bic
 [1] -1285.164 -1279.944

output$size
```

```
 [1] 2 3

output$probne0
 [1] 100.0   6.8 100.0

output$postmean
 [1]  0.56762931  3.19355682 -0.01756048  1.97759807

output$postsd
 [1] 0.85994375 0.20177235 0.08305759 0.04085244

 output$ols
            Int       X1         X2       X3
[1,]  0.6194303 3.192800  0.0000000 1.977538
[2,] -0.1368776 3.203849 -0.2563875 1.978413

 output$se
          Int        X1        X2         X3
[1,] 0.824321 0.2017452 0.0000000 0.04085239
[2,] 1.011271 0.2018597 0.1987171 0.04084441
```