STAT302 Methods of Data Analysis 1
Module 9: Model Selection

Austin Brown

University of Toronto, Department of Statistical Sciences

November 14, 2024

Practice problems

For practice problems this week, I recommend to implement what we learn today on your project and other real datasets.

Lecture 1

Learning objectives

- Akaike's Information Criterion
- Bayesian Information Criterion
- Implementation in R

Lecture 1: Goodness of fit versus model complexity

Model criterion/scoring

Compare models according to some criterion/score. For example, we may want to approximate the population model or we may want to find a model that predicts the best.

So far, we could use

- RSS
- R^2
- R^2_{adj}

Question: What is the problem with using the RSS? R^2?

Goodness versus complexity

The RSS is a measure of "goodness" of the fit. Larger models have smaller RSS but increased model complexity. We will learn some new criterion of the following form:

$$\{\text{Goodness of fit to the data}\} + \{\text{model complexity penalty}\}$$

Maximum likelihood

The likelihood of the regression model is the probability density of the model with the data fixed and treated as a function of the parameters β, σ^2:

$$\begin{aligned} L(\beta, \sigma^2) &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2\right) \\ &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}RSS(\beta)\right) \end{aligned}$$

Minimizing the residual sum of squares as we shall see can be seen as maximizing the likelihood.

Maximize the likelihood

Maximize the likelihood

Since the log is monotone, we can minimize the negative log-likelihood. $\ell(\boldsymbol{\beta}, \sigma^2) = -\log L(\boldsymbol{\beta}, \sigma^2)$. This is

$$\ell(\boldsymbol{\beta}, \sigma^2) = \frac{n}{2}\log(2\pi) + \frac{n}{2}\log(\sigma^2) + \frac{1}{2\sigma^2}RSS(\beta)$$

Minimize $\boldsymbol{\beta}$:

$$\ell(\hat{\boldsymbol{\beta}}, \sigma^2) = \frac{n}{2}\log(2\pi) + \frac{n}{2}\log(\sigma^2) + \frac{1}{2\sigma^2}RSS(\hat{\boldsymbol{\beta}})$$

Maximize the likelihood

Minimizing σ^2 yields the minimum

$$\frac{d}{d\sigma^2}\ell(\hat{\boldsymbol{\beta}}, \sigma^2) = 0$$

$$\hat{\sigma}^2_{MLE} = \frac{1}{n} RSS(\hat{\boldsymbol{\beta}}).$$

We will skip checking the second derivatives.

Maximize the likelihood

We have maximized the likelihood

$$\max_{\beta,\sigma^2} L(\beta, \sigma^2) = L(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2_{MLE})$$

Compute

$$\ell(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2_{MLE}) = \frac{n}{2}\log(2\pi) + \frac{n}{2}\log(\hat{\sigma}^2_{MLE}) + \frac{1}{2\hat{\sigma}^2_{MLE}} RSS(\hat{\boldsymbol{\beta}})$$

AIC preliminary definition including the constants

Akaike's Information Criterion (AIC) [Akaike, 1973] can be defined

$$\begin{aligned} & 2\ell(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2_{MLE}) + 2(p+2) \\ & = n \log(RSS/n) + n \log(2\pi) + n + 2(p+2). \end{aligned}$$

However, the constants are redundant.

Compute in R:

```
AIC(fit, k = 2)
```

AIC

Removing the redundant terms, we define Akaike's Information Criterion (AIC) [Akaike, 1973]

$$AIC = n \log(RSS/n) + 2(p+1)$$

- The second term penalizes the size (complexity) of the model
- Based on large sample approximation with n being very large
- AIC will generally not select the population model for large samples n, but instead a larger model.
- Roughly, this is motivated for prediction.

AIC in R

It defaults to AIC but you can specify $k = 2$ for AIC. R computes the AIC with

$$AIC = n \log(RSS/n) + 2(p + 1).$$

```
extractAIC(fit, k = 2)
```

Looks like a typo on [Sheather, 2009, page 231].

BIC

Ignoring the constant terms, we define the Bayesian Information Criterion (BIC) [Schwarz, 1978]

$$BIC = n \log(RSS/n) + \log(n)(p+1)$$

- Some definitions use $p+1, p+2$ instead of p.
- Based on Bayesian statistics and large sample approximations
- Assigns a better score to smaller models compared to AIC when $\log(n) > 2$
- Motivated by selecting the population linear regression model. However, smaller models can predict well.

BIC in R

Instead of $k = 2$, use $k = \log(n)$. R computes the BIC with

$$BIC = n \log(RSS/n) + \log(n)(p + 1)$$

```
extractAIC(fit, k = log(n))
```

Which criterion to use?

What criterion to use? Make a choice based on your problem and justify this choice.

Let's focus on the goal of AIC/BIC versus the technical details for this course. Both AIC/BIC are approximations based on large samples.

If you are more interested in prediction, AIC may be better suited for your problem. It may select extra variables that help the model predict better.

If you want a smaller model because your goal is to select the population model, BIC may be better suited for your problem.

Some comments

- If the population model is one of the candidates, then BIC will likely select the population model for very large n, but this is not necessarily the case for AIC.
- AIC and BIC are based on large samples, and so may be inaccurate in their selections. In particular, selecting a model with either can introduce bias into your LSE.
- BIC penalizes the size of the model more and so will generally choose a smaller model.

Discussion

- What criterion to use AIC/BIC?
- Does AIC/BIC require checking the model assumptions?
- Does AIC/BIC account for multicollinearity or problematic observations?

Example by hand and in R

Lets compute the AIC and BIC from the summary for the NYC data and check this against how we do this with R.

```
nyc_restaurant_data = read.csv("https://gattonweb.uky.edu/sheather/book//docs/
     datasets/nyc.csv", header = T)
fit = lm(Price ~ East + Food + Decor, data = nyc_restaurant_data)
summmary(fit)
```

Lecture 2

Learning objectives

- All subset selection
- Forward/backward selection
- Implementations in R

Lecture 2: All subset selection

All subset selection

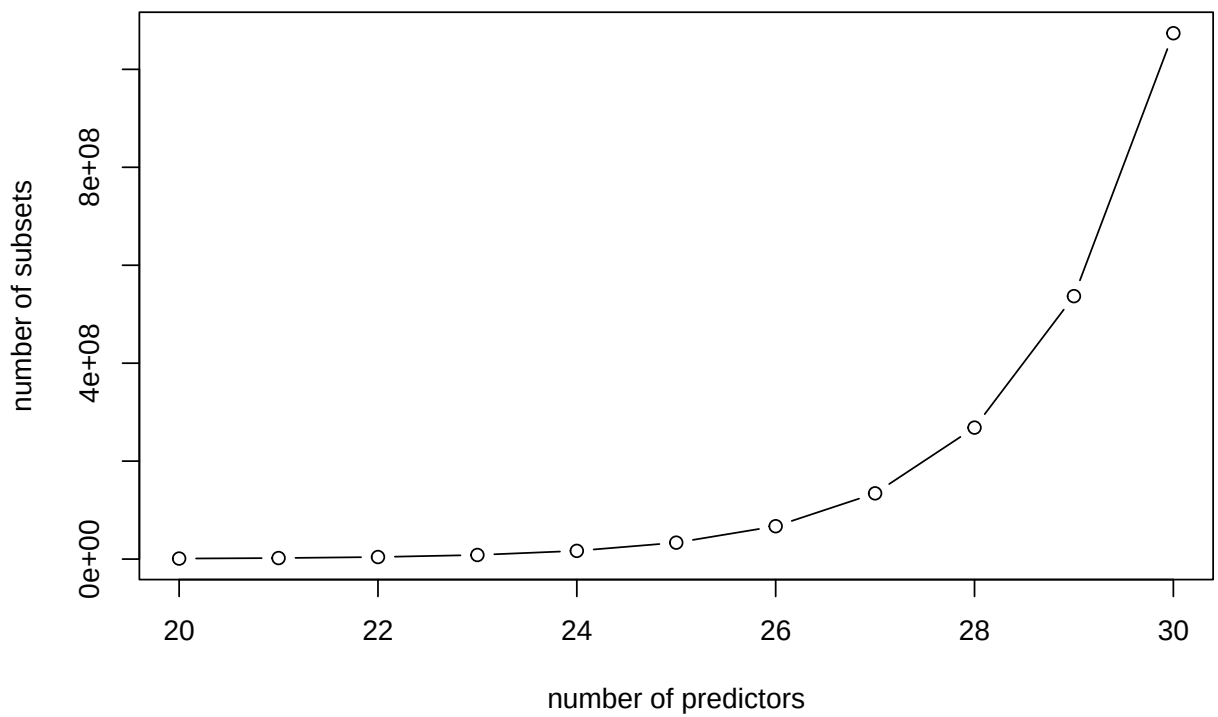
Given a set of predictors $x_1, \ldots, x_p$, look through all possible submodels. So search through all possible subsets of $\{1, \ldots, p\}$.
For example, $\{1\}, \{1, p\}$, etc.
Each subset corresponds to a model. For example,
$\{k, \ldots, K\} \subseteq \{1, \ldots, p\}$ corresponds to the model

$$Y = \beta_0 + \beta_k x_k + \cdots + \beta_K x_K + e.$$

Note: Generally, individual dummy variables are not dropped for categorical variables.

All subset selection

There are 2^p total possible models to compare. Why? There are 2 choices to include/exclude x_1, 2 choices to include/exclude x_2, and so on for p predictors. If we don't count the where we start at the full model, then $2^p - 1$. Even with modern compute, this may be an impossible task if p is large.



All subset selection in R

Leaps will return the best models of a particular size according to the RSS. Then you may plot the BIC for each model. The BIC reported is the change in BIC and not the actual BIC.

```
library(leaps)
best_fit = regsubsets(Price ~ East * Food * Decor * Service, data = df,
                      nvmax=NULL,
                      nbest = 1)
plot(best_fit, scale = "bic")
coef(best_fit, 1)
```

Discussion

- What are the benefits/drawbacks of all subsets selection?
- What if I have 100 predictors how many models will all subset selection need to search?
- Does R check the assumptions when calculating the scores for all subsets?

Example on NYC data

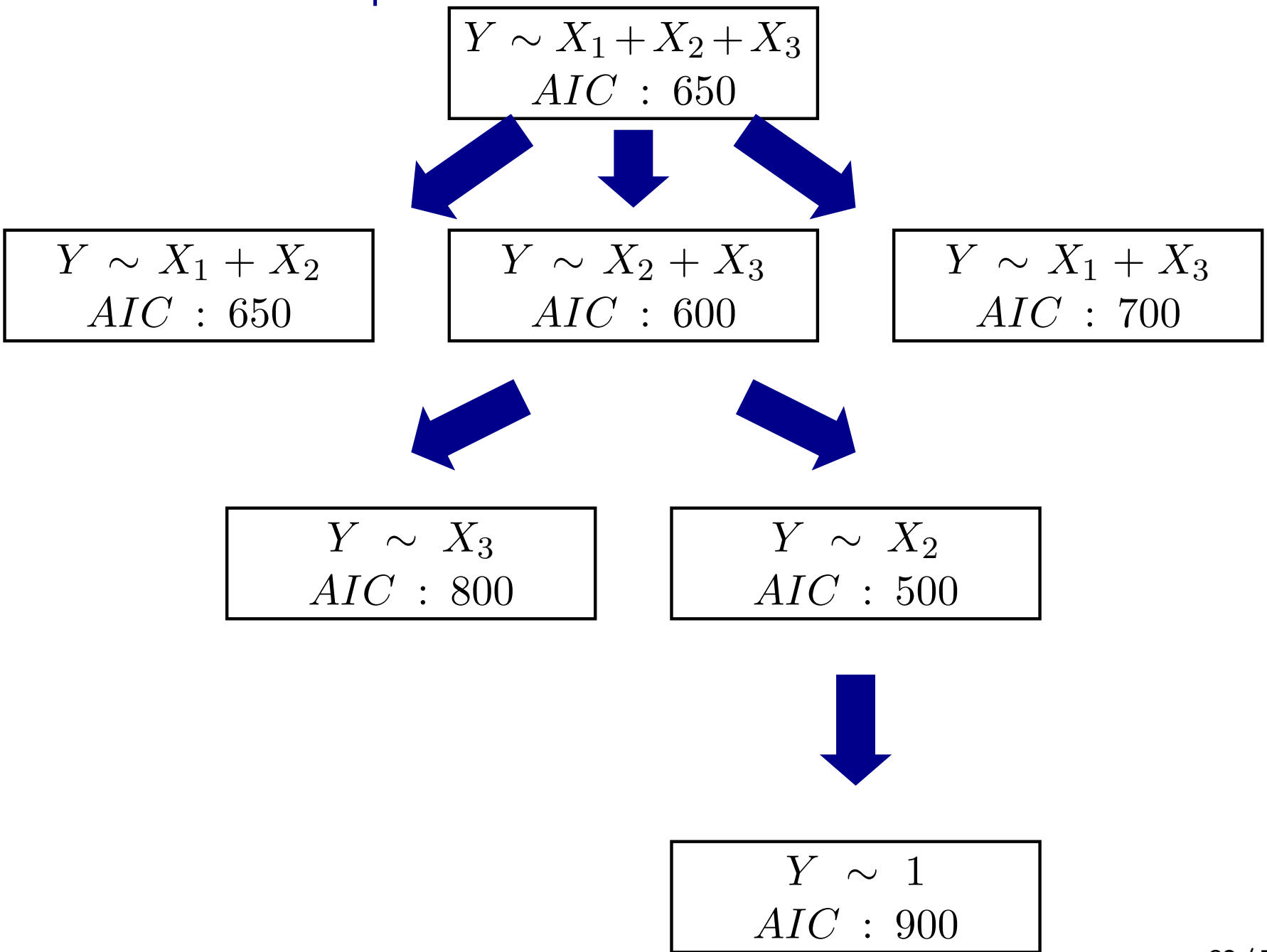
Let's look at all the interactions possible for the NYC dataset and use BIC to select a model.

Lecture 2: Backward selection

Backward selection process

1. Start with the full (most complex) model and compute AIC/BIC
2. Fit all submodels with 1 predictor dropped from the full model and compute AIC/BIC
3. Set the submodel with the best score/criterion to the full model
4. Repeat step 1 or stop according to the stopping condtion. Generally this is if the submodel has larger score than the full model.

Backward selection process visualized



Backward selection complexity

If backward selection does not stop early, then it computes a total of

$$p + p - 1 + \cdots + 1 = \sum_{k=1}^{p} k = p(p+1)/2.$$

total models. This is order p^2 which is polynomial in the number of predictors.

Technically it is $1 + p(p+1)/2$, but we do not count the starting point. This is generally achievable with modern compute even if p is very large compared to all subsets which is order 2^p.

Backward selection complexity

Question: Suppose the list of candidate models contains a good approximation to the population model. Can backward selection miss selecting this model? Why or why not?

Backward selection in R

R stops backward selection once the AIC/BIC of the best submodel is larger than the full model.

```
# Step AIC
reduced_fit = step(fit, direction = "backward", k = 2)

# Step BIC
reduced_fit = step(fit, direction = "backward", k = log(n))
```

Backward selection in R

There are some implementation details in R. R considers even less models. R will first try to drop the highest interaction and polynomial terms only and then other terms. R will also drop a categorical variable (drop all indicators) or keep it (keep all indicators).

Backward selection: Example on NYC data

Let's look at all the interactions possible for the NYC dataset and use BIC to select a model using backward selection.

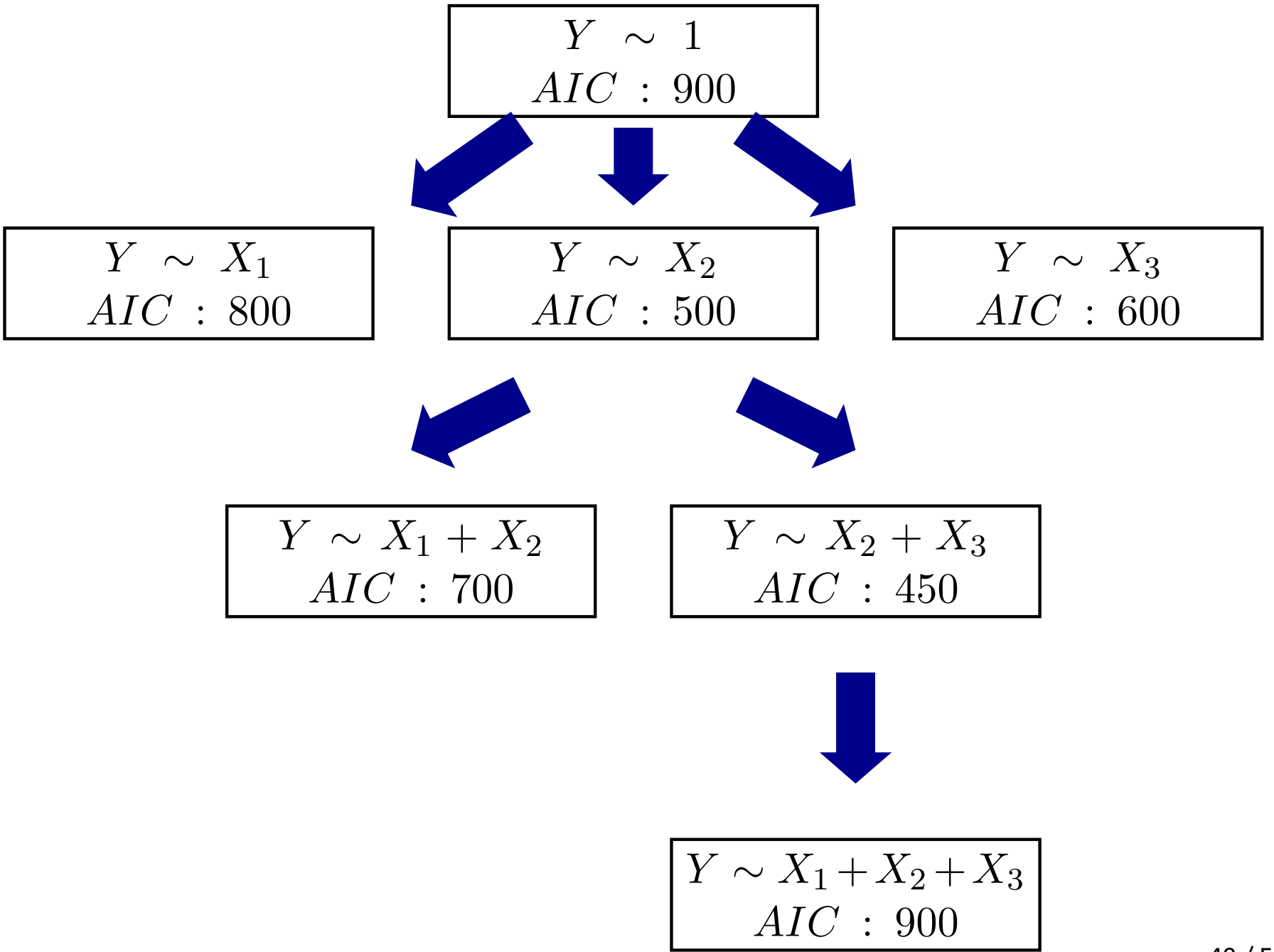
```
fit = lm(Price ~ East * Food * Decor * Service, data = nyc_restaurant_data)

# Check the full model assumptions
par(mfrow = c(2, 2))
plot(fit, which = c(1, 2, 3, 4))

stepfit = step(fit, direction="backward", k = log(n))
summary(stepfit)

# Check the final model assumptions
par(mfrow = c(2, 2))
plot(stepfit, which = c(1, 2, 3, 4))
```

Lecture 2: Forward selection

Forward selection

1. Start with the simple (least complex) model and compute AIC/BIC
2. Fit all models with 1 predictor added to the simpler model and compute AIC/BIC
3. Set the new simpler model with the best score/criterion to the full model
4. Repeat step 1 or stop according to the stopping condition. Generally this is if the new model has larger score than the simpler model.

Forward selection



Forward selection complexity

If forward selection does not stop early, then it searches

$$p + p - 1 + \cdots + 1 = \sum_{k=1}^{p} k = p(p+1)/2.$$

total models. Technically it is $1 + p(p+1)/2$, but we do not count the starting point.

Forward selection in R

For R, you will need to set the scope of the models, so set the lower to the formula for the null model and the upper to the formula for the full model. R stops once the best new model has larger AIC/BIC then the simpler model.

```
# AIC
stepfit = step(fit_null,
               scope=list(lower = y ~ 1,
                          upper = y ~ x_1 * x_2),
               direction="forward",
               k = 2)

# BIC
stepfit = step(fit_null,
               scope=list(lower = y ~ 1,
                          upper = y ~ x_1 * x_2),
               direction="forward",
               k = log(n))
```

Foward selection: Example on NYC data

Let's look at all the interactions possible for the NYC dataset and use BIC to select a model with forward selection.

```
fit_full = lm(Price ~ East * Food * Decor * Service, data = nyc_restaurant_data)
fit_null = lm(Price ~ 1, data = nyc_restaurant_data)

# Check the full model assumptions
par(mfrow = c(2, 2))
plot(fit_full, which = c(1, 2, 3, 4))


# Fit the selected model using BIC
stepfit = step(fit_null,
               scope=list(lower = Price ~ 1,
                          upper = Price ~ East * Food * Decor * Service),
               direction="forward",
               k = log(n))
summary(stepfit)

# Check the final model assumptions
par(mfrow = c(2, 2))
plot(stepfit, which = c(1, 2, 3, 4))
```

Pros/Cons

- What are the benefits/drawbacks of forward/backward selection?
- Does R check the assumptions when calculating the scores for backward/forward selection?

Combining both forward/backward selection

```
# AIC
stepfit = step(fit_full, direction="both", k = 2)
```

MASS package version

The MASS package also has stepwise that does the same thing.

```
# MASS package example
library(MASS)

fit_full = lm(Price ~ East * Food * Decor * Service, data = nyc_restaurant_data)

stepAIC(fit_full,
        k = log(n),
        direction = "backward")
```

Lecture 3

Learning objectives

- Implement selection methods in R
- Basic introduction to penalization methods (Will not be on exam)

Limitations of automated methods

- Multicollinearity can introduce issues.
- F-testing, AIC, and BIC will likely disagree on the chosen model.
- Relies on assumptions and does not check assumptions.
- Ignores context and purpose of the model
- Can create bias in the model. For example, backward selection may pick the non-optimal set of predictors.

Activity

Prostate data [Sheather, 2009, page 239]:

The goal is to predict thelog psa score (lpsa) from a number of measurements with the log-cancer volume (lacavol) including log prostate weight (lweight), age, log of benign prostatic hyperplasia (lpbh), seminal vesicle invasion (svi), log of capsular penetration (lcp), Gleason score (gleason), and percent of Gleason scores 4 or 5 (pgg45).

Activity

Let's plot the full model using variables of interest and check the assumptions.

- Find the best model according to AIC.
- Are all the coefficients of the predictors statistically significant? Use significance $\alpha = .05$
- Find the best model according to BIC.
- Are all the coefficients of the predictors statistically significant? Use significance $\alpha = .05$

Lecture 3: Some optional stuff

Ridge or ℓ_2 Penalization

Let us look at some more advanced methods using this idea of goodness of fit + model complexity penalty. Ridge regression solves the following:

$$\hat{\boldsymbol{\beta}}_{Ridge} = \text{argmin}_{\beta} \left[RSS(\beta) + \lambda \sum_{k=1}^{p} |\beta_k|^2 \right]$$

- Requires choosing λ. Larger lambdas penalize more and smaller lamdas penalize less.
- The extra term is often called "weight decay" in the A.I. literature.
- The first term is the usual RSS and acts as a "goodness of fit" while the second term roughly acts to penalize the "model complexity".
- The solution $\hat{\boldsymbol{\beta}}_{Ridge}$ is biased and can be solved explicitly.

LASSO or ℓ_1 Penalization

Let us look at some more advanced methods using this idea of goodness of fit + model complexity penalty. The LASSO [Tibshirani, 1996] solves the following:

$$\hat{\boldsymbol{\beta}}_{LASSO} = \text{argmin}_{\beta} \left[RSS(\beta) + \lambda \sum_{k=1}^{p} |\beta_k| \right]$$

- Requires choosing λ. Larger lambdas penalize more and smaller lamdas penalize less.
- The first term is the usual RSS and acts as a "goodness of fit" while the second term roughly acts to penalize the "model complexity".
- The solution $\hat{\boldsymbol{\beta}}_{LASSO}$ is biased and can be solved on a computer with iterative methods.

Implementation in R

Interestingly, the Lasso can drop variables automatically if lambda is chosen correctly.

```
library(glmnet)

y = nyc_restaurant_data$Price
X = model.matrix(Price ~ East + Food + Decor + Service, data = nyc_restaurant_
    data)
X = X[, -1] # Remove intercept
fit = glmnet(X, y,
             alpha = 1, # specifies the penalty
             lambda = 2 # specifies the lambda
             )
coef(fit)
```

Module takeaways

1. How are the likelihood criteria of goodness computed?
2. What are the differences and similarities between the likelihood criteria of goodness?
3. How is a "best"/"preferred" model selected using likelihood criteria?
4. How does all possible subsets and the automated selection procedures select a "best" model?
5. What are the advantages and disadvantages of using automated and all possible subsets model selection?

References I

H Akaike. Information theory and an extension of maximum likelihood principle. In *Proc. 2nd int. symp. on information theory*, pages 267–281, 1973.

Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461 – 464, 1978.

Simon Sheather. *A modern approach to regression with R*. Springer Science & Business Media, 2009.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.