Model Selection for Regression

Building a Regression Model Start to Finish

- Data collection
- Data preparation
- Preliminary model investigation
- Reduction of explanatory variables
- Model selection
- Model validation

Building a Regression Model Start to Finish

- Data collection
- Data preparation
- Preliminary model investigation
- Reduction of explanatory variables
- Model selection
- Model validation

Data collected through controlled experiments & observational studies or mixure; 6 to 15 obs per potential predictor recommended. (not focus in this class)

Data prep - clean the data, typo's, format, storage, warehousing, security, handling (not focus in this class - covered in Data Wharehousing, Big Data Computing courses)

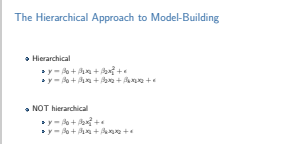
Look at scatterplots, brainstorm, what is the response var? what about potential predictors (explanatory vars)? create a pool of possibilities, talk to the experts, think outside the box, functional forms, interactions

More plots, measure collinearity - need to drop or combine variables (in a meaningful way), automated search routines

The Hierarchical Approach to Model-Building

- Hierarchical
  - $y = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \epsilon$
  - $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_k x_1 x_2 + \epsilon$

- NOT hierarchical
  - $y = \beta_0 + \beta_2 x_1^2 + \epsilon$
  - $y = \beta_0 + \beta_1 x_1 + \beta_k x_1 x_2 + \epsilon$



When selecting an appropriate linear model, it makes sense to use a hierarchical approach. That is, if an independent variable is involved in a higher order term (like a quadratic or interaction term) then the lower order terms involving that variable should remain in the model, even if their coefficients aren't significantly different from 0.

Bottom panel note: The term "hierarchical" can also refer to data that is nested or clustered due to the sampling scheme, which is a different matter.

Model Selection Criterion

- R^2 - higher is better
- R^2_{adj} - higher is better
- c_p - closer to the number of parameters (p) is better
- PRESS - lower is better
- AIC - lower is better
- BIC - lower is better

Model Selection Criterion

- R^2 - higher is better
- R^2_{adj} - higher is better
- C_p - closer to the number of parameters (p) is better
- PRESS - lower is better
- AIC - lower is better
- BIC - lower is better

In model-building, the idea is to find the simplest model that explains the most variation in the response variable. This is known as the principle of parsimony. Overfitting simply complicates the use and interpretations of the model.

Measures like adjusted R-square, Cp, AIC and BIC take into account the number of parameters in a model and provide a penalty for each additional one. AIC and BIC are not mentioned in the Ott textbook, but they are major players in model building.

What's AIC?

Akaike Information Criterion

$$AIC = n \cdot \ln SSE - n \cdot \ln n + 2p$$

(or some variation on this formula)

AIC is a measure of information loss and is particularly useful for comparing models - lower is better.

What's AIC?

Akaike Information Criterion

$$AIC = n \cdot \ln SSE - n \cdot \ln n + 2p$$

(or some variation on this formula)

AIC is a measure of information loss and is particularly useful for comparing models - lower is better.

You didn't see this on our textbook, but its too good to leave out.

In the formula, n is the sample size, SSE is the sum of squares for error, and p is the number of parameters.

BIC or Bayesian Information Criterion, also called SBC for Schwartz' Bayesian Criterion is much like it. Smaller is better. They have their own Wikipedia pages; look them up if you'd like to know more about them!

Collinearity/Multicollinearity

- When two or more predictor variables are highly correlated in a linear regression model.
- Indicated by large values of the Variance Inflation Factor (VIF), such as 10 or more

Collinearity/Multicollinearity

- When two or more predictor variables are highly correlated in a linear regression model.
- Indicated by large values of the Variance Inflation Factor (VIF), such as 10 or more

Collinearity is likely to be common in the life of a data scientist because the data a data scientist will generally be working with is not from a designed experiment and often contains so many variables that high correlations among predictors are inevitable.

Highly correlated variables basically are providing overlapping explanations of the same variation in the response variable.

The Effects of Collinearity

- Large standard error for estimated regression coefficients
- Higher p-values for tests of individual coefficients
- Wider confidence intervals for coefficients

The Effects of Collinearity

- Large standard error for estimated regression coefficients
- Higher p-values for tests of individual coefficients
- Wider confidence intervals for coefficients

The effect of collinearity are primarily seen in the form of inflated standard error for the estimated regression coefficients, which in turn produces larger p-values for tests of individual coefficients and wider confidence intervals for coefficients.

A signature clue is when the F test for the full model has a very low p-value, but no individual coefficient has a small enough p-value to indicate that it differs from 0.

What Collinearity Does NOT Affect

- F-statistics & p-values for the full model or subsets of coefficients
- R^2
- R^2_{adj}
- AIC
- Predicted values
- Standard errors of predicted values (these can be slightly affected)

What Collinearity Does NOT Affect

- F-statistics & p-values for the full model or subsets of coefficients
- R^2
- R^2_{adj}
- AIC
- Predicted values
- Standard errors of predicted values (these can be slightly affected)

Much work can still be done with a regression model even in the presence of collinearity because there are a number of measures that are unaffected by it. Particularly if your aim is to use the model to estimate values the response variable and make predictions.

Can anything be done to alleviate collinearity?

- Ignore it if it doesn't affect what you are doing with the regression model
- Combine correlated variables in a meaningful way to make a single variable
- Omit the predictor with the highest VIF
- Centering often removes collinearity for quadratic, cubic and interaction terms (centering is to subtract the mean from each data value for a given variable)
- Employ factor analysis to reduce the number of predictors (may be difficult to interpret results)

Automated Search Procedures Like “step”" and “regsubsets”

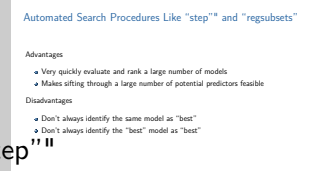
Advantages

- Very quickly evaluate and rank a large number of models
- Makes sifting through a large number of potential predictors feasible

Disadvantages

- Don’t always identify the same model as “best”
- Don’t always identify the “best” model as “best”

Automated Search Procedures Like "step" and "regsubsets"



Be sure to spend some time with the swirl lesson called Regression Model Selection and also investigating step and regsubsets in R on your own. Automated model search procedures like these have some powerful advantages when they are used wisely.

That is, they can fit and rank a large number of regression models in a matter of seconds. They also make it feasible to consider a large number of potential predictor variables.

These automated methods don't replace the human element. You have to consider what criteria is being used to rank the models. Different methods sometimes produce different models as being the best - which is in quotes here, because sometimes it isn't well-defined what is meant by "best" since there are various criteria

Logistic Regression Model Selection using AIC with "step"

Suppose the variable POND_AREA in the farmpond data set was suspected of having an effect on the species richness being at least 4. Consider fitting the logistic regression model containing FISH, TOTNITR, and POND_AREA and all possible interactions among them.

```
rich.out.full <- glm(RICH~FISH*TOTNITR*POND_AREA,data=farmpond,
                     family="binomial")
```

Model selection using AIC with “step”

The R function **step** is used as follows to evaluate

```
rich.out.full <- glm(RICH~FISH*TOTNITR*POND_AREA,data=farmpond,
                     family="binomial")
step(rich.out.full)
```

The function "step" is used exactly the same way with logistic regression as with multiple linear regression. The stepping can be performed forward, backward, or the default of both.

First step from the output of “step”

```
## Start:  AIC=30.9
## RICH ~ FISH * TOTNITR * POND_AREA
##
##                             Df Deviance    AIC
## - FISH:TOTNITR:POND_AREA  1   14.897 28.897
## <none>                        14.897 30.897
```

First step from the output of "step"

```
## Start:  AIC=30.9
## RICH ~ FISH * TOTNITR * POND_AREA
##
##                          Df Deviance    AIC
## - FISH:TOTNITR:POND_AREA  1   14.897 28.897
## <none>                        14.897 30.897
```

Depending on the size of the full model, the step procedure can produce a great deal of output. Only the first block is shown in this slide.

With the default setting of both directions, the step procedure begins by fitting the full model and compares the AIC of that model to the model with the 3-way interaction term FISH-by-TOTNITR-by-Pond_AREA removed. Since the AIC is lower without it, it is removed from the model and the procedure continues.

Last step from the output of “step”

```
## Step:  AIC=23.92
## RICH ~ FISH + TOTNITR + FISH:TOTNITR
##
##                 Df Deviance    AIC
## <none>               15.924 23.924
## - FISH:TOTNITR  1   25.591 31.591
```

Last step from the output of "step"

```
## Step:  AIC=23.92
## RICH ~ FISH + TOTNITR + FISH:TOTNITR
##
##               Df Deviance    AIC
## <none>             15.924 23.924
## - FISH:TOTNITR  1   25.591 31.591
```

After several iterations of dropping higher order terms and and working down to possibly dropping first-order terms, the procedure stops when no options produce a lower AIC.

The final model given by this automated process is shown here to be the one that includes the main effect due to FISH and TOTNITR and the interaction of them. Notice that the removal of the interaction between FISH and TOTNITR causes the AIC to increase, and so it is retained.

POND AREA was totally eliminated from the model in a previous step.

Testing Subsets of Coefficients with "anova"

```
rich.out <- glm(RICH~FISH + TOTNITR,data=farmpond,family="binomial")
rich.out2 <- glm(RICH~FISH*TOTNITR + POND_AREA,data=farmpond,family="b
anova(rich.out,rich.out2,test="Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: RICH ~ FISH + TOTNITR
## Model 2: RICH ~ FISH * TOTNITR + POND_AREA
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        37     25.591
## 2        35     15.437  2   10.154 0.006238 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Testing Subsets of Coefficients with "anova"

```
rich.out <- glm(RICH~FISH + TOTNITR,data=farmpond,family="binomial")
rich.out2 <- glm(RICH~FISH+TOTNITR + POND_AREA,data=farmpond,family="binomial")
anova(rich.out,rich.out2,test="Chi")

## Analysis of Deviance Table
##
## Model 1: RICH ~ FISH + TOTNITR
## Model 2: RICH ~ FISH + TOTNITR + POND_AREA
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        37     25.591
## 2        35     15.437  2   10.154 0.006238 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Two models may be compared directly using the R function "anova". It is important to remember that the smaller model should be nested within the larger model, in keeping with the hierarchical approach to model-building.

You must specify to conduct the test using the Chi-square distribution when testing logistic regression models. The test is based on a comparison of the deviance for each model and the test statistic follows a chi-square distribution - in this case with 2 degrees of freedom, since the full model has 2 additional terms- this is the difference in the df for the full and reduced models. 10.154 is the difference between the residual deviance of each model 25.591 for the reduced model minus 15.437 for the full model.

Logistic Regression Classification Table

```
##       predRICH
##        Sp Rich<4 Sp Rich>=4 Sum
##   0            9          4  13
##   1            4         23  27
##   Sum         13         27  40
```

```
## [1] "Proportion correctly predicted =  0.8"
```

Logistic Regression Classification Table

Logistic Regression Classification Table

```
##      predRICH
##        Sp Rich<4 Sp Rich>=4 Sum
##  0             9          4  13
##  1             4         23  27
##  Sum          13         27  40

## [1] "Proportion correctly predicted =  0.8"
```

One way to evaluate a logistic regression model is by looking at the classification table. The rows are defined by the counts for the observed response variable and the columns are the counts obtained using the predicted probabilities. If the predicted probabiity for a given set of values for the predictor variables exceeds some predefined cutoff probability, which is typically 0.5, then the predicted outcome in this example is that the species richness is at least 4, otherwise it is assigned an outcome of being less than 4.

The cutoff threshhold probability can be whatever makes sense for your given application.

This table is for the model from the farmpond data predicting if species richness is above 4 or not from the presence or absence of

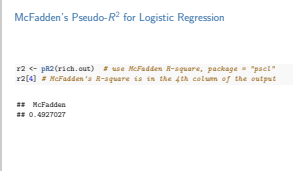
McFadden's Pseudo-R^2 for Logistic Regression

```
r2 <- pR2(rich.out)  # use McFadden R-square, package = "pscl"
r2[4] # McFadden's R-square is in the 4th column of the output
```

```
##  McFadden
## 0.4927027
```

Model Selection for Regression

McFadden's Pseudo-R^2 for Logistic Regression



Mcfadden's is one of a few different pseudo-r-squared values that can be computed to measure the explanatory capabilities of a logistic regression model. It turns out that there isn't really a direct R-squared-type of measure for logistic regression like there is in linear regression, but the basic idea that it will be between 0 and 1 and the larger-the-better still applies.

It is something of a measure of the predictive ability of the model. This was for the model from the farmpond data predicting if species richness is above 4 or not from the presence or absence of fish and the total nitrogen in the pond. It's really somewhat mediocre at 0.49.

Hosmer-Lemeshow GOF Test for Logistic Regression

```r
# Use the package "ResourceSelection"
hoslem.test(farmpond$RICH, fitted(rich.out), g=5)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  farmpond$RICH, fitted(rich.out)
## X-squared = 0.90299, df = 3, p-value = 0.8247
```

Hosmer-Lemeshow GOF Test for Logistic Regression

```
# Use the package "ResourceSelection"
hoslem.test(farmpond$RICH, fitted(rich.out), g=5)

##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  farmpond$RICH, fitted(rich.out)
## X-squared = 0.90299, df = 3, p-value = 0.8247
```

While the classification table and McFadden's R-squared are ways to evaluate the predictive ability of a logistic regression model, the Hosmer-Lemeshow goodness-os-fit test is a way to evaluate the fit of the model, or lack thereof.

In this test, which can basically be thought of as having a null hypothesis stating that the model fits vs an alternative that it doesn't, begins by splitting the sample into a predetermined number of groups and comparing observed to expected outcomes for both number of successes and number of failures in each group.

Commonly, the number of groups chosen is between 5 and 10, with 10 being the default in many software packages. Since the sample size for the farmpond data is only 40, a smaller number of groups