BIOS 635: Model Selection

Kevin Donovan

3/2/2021

Model Building

- When building a statistical model, may have to choose one setup vs others
  - *Ex. tuning parameters, **which features to use**, etc.*
  - *More complex model $\nrightarrow$ ``best”*
  - *Need metrics to help us select final model*
  - *Discussed using prediction error, but may want to consider other factors*
    - Ex. model complexity/interpretability
- Here, we focus on **regression models**

Model selection in regression

- **Setup**: response Y, features $X_1, \ldots, X_p$
- **Model**:

$$Y = \beta_0 + \beta_1 X_1 + \ldots + X\beta_p X_p + \epsilon$$

- May want to ID **subset of predictors** which are *most relevant* to prediction
  - *Makes model simpler, may $\downarrow$ overfitting/variance and $\uparrow$ interpretability*
  - *Denote this subset by $S \subset \{1, \ldots, p\}$ with model*

$$Y = \beta_0 + \sum_{j \in S} \beta_j X_j + \epsilon$$

- May want data-driven way of selecting subset

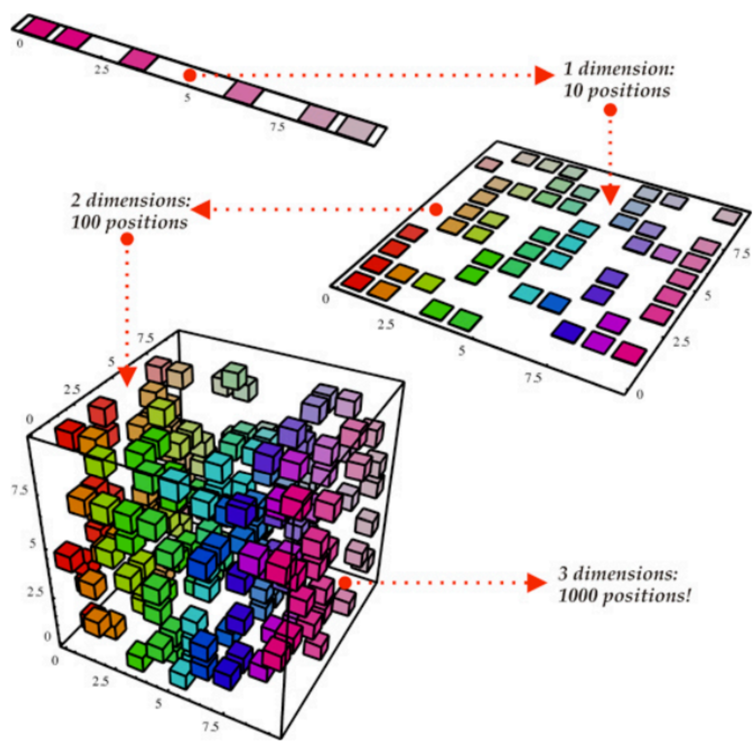
Model selection methods

1. Subset selection
    - *ID subset of predictors through some iterative procedure based on chosen metric*
2. Shrinkage
    - *Fit model with all p predictors, but include penalty term to least squares process*
    - *Penalty shrinks small magnitude estimates to 0*
    - *Also called regularization*

Model selection methods

3. Dimension reduction

- *Project set of p predictors into M dimensional space, $M < p$*
- *Use predictors in new space in regression model*
- *Space often = M linear combinations of p predictors*

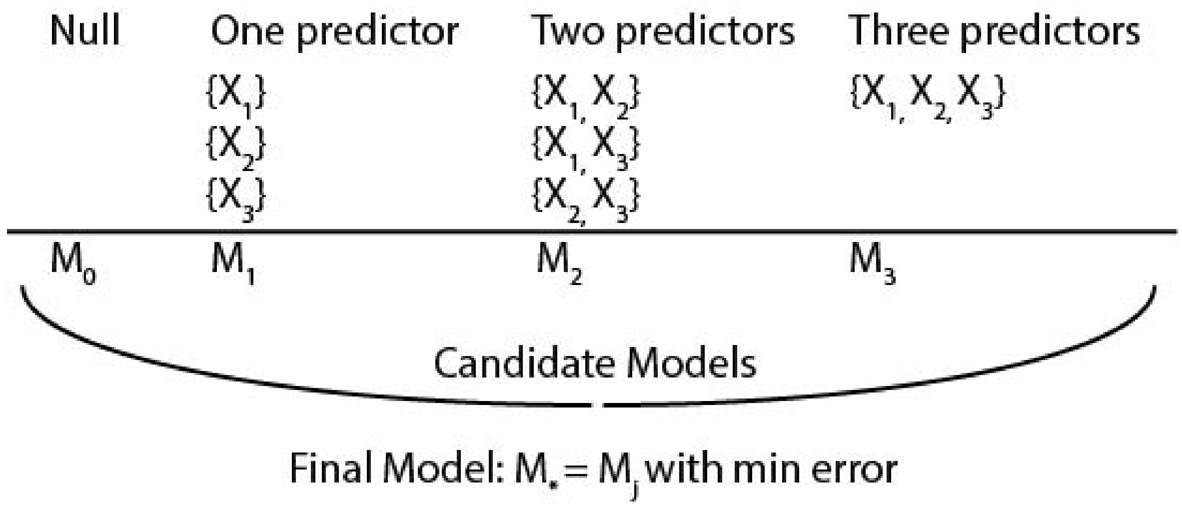


Subset selection

- Let M_0 denote the *null model* where $\hat{Y} = \bar{Y}$ (sample mean)
- Then, create candidate models $M_1, \ldots, M_p$:
  - *For $j = 1, \ldots, p$, select M_j=best subset of predictors of size j*
  - *Best based on chosen metric such as MSE, R^2, RSS, etc.*
- Select *best* model outcome of candidate models $M_0, M_1, \ldots, M_p$
  - *Again, based on chosen metric (which also penalizes overfitting)*
  - *Ex. cross-validated prediction error or corrected training set metric*

Subset selection

- Ex. Y and predictor set X_1, X_2, X_3



- Limitations
  - *Cannot be computed with* ***very large*** p *(*$p > n$*)*
  - *Huge search space* $\rightarrow$ *high chance of selecting model which overfits*
  - *Also* $\rightarrow$ *hard to confidently tell if model is ``best" beyond random chance*
  - *Not compuationally efficient as* p *increases*

Forward stepwise selection

- Algorithm:

    1. Start with null model M_0, no predictors

    2. For $k = 0, \ldots, p$:

    - Consider $p - k$ models which add one predictor to M_k
    - Choose *best* among these models
    - Set this as model $k + 1$, move to $k + 1$ as starting point
    - Results in candidate models $M_0, \ldots, M_p$

    3. Select best from set of candidate models

Forward stepwise selection

- Computationally less intensive then best subset
  - *Much less models fit and examined*
- Not guaranteed to find best possible model (some combos not tried)
- Cannot be run when $p > n$

Backward stepwise selection

- Algorithm:

  1. Start with full M_p, all predictors

  2. For $k = p, p-1, \ldots, 0$:

    - Consider k models which contains all but one of the predictors in M_k
    - Choose *best* among these models
    - Set this as model $k+1$, move to $k+1$ as starting point
    - Results in candidate models $M_0, \ldots, M_p$

  3. Select best from set of candidate models

Backward stepwise selection

- Computationally less intensive then best subset
    - *Much less models fit and examined*
- Not guaranteed to find best possible model (some combos not tried)
- Cannot be run when $p > n$

Stepwise selection visuals

Forward Selection

Step	Model	
0	Null	M_0
1	$\{X_1\}$ $\{X_2\}$ $\{X_3\}$	M_1
2	$\{X_1, X_2\}$ $\{X_1, X_3\}$	M_2
3	$\{X_1, X_2, X_3\}$	M_3

Final Model:
M_j: j min error

Backward Selection

Step	Model	
0	$\{X_1, X_2, X_3\}$	M_3
1	$\{X_1, X_2\}$ $\{X_1, X_3\}$	M_2
2	$\{X_1\}$ $\{X_2\}$ $\{X_3\}$	M_1
3	Null	M_0

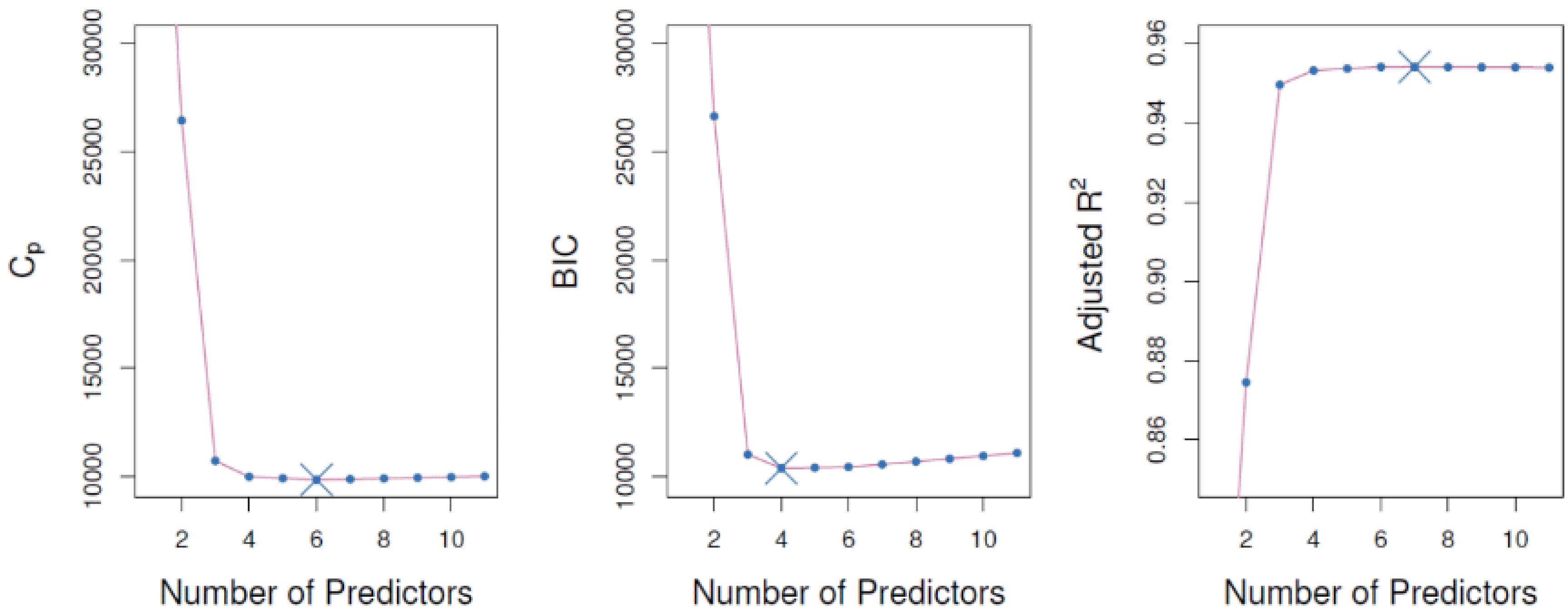
Final Model:
M_j: j min error

Choosing a metric

- Recall in all stepwise methods, candidate models selected using RSS, R^2, etc.
- When candidate models compared, **need to use test set error**
  - *Or some approximation*
  - *Training error would **not** result in optimal model*
- Either 1) *adjust* training error or 2) *directly* estimate testing error

Adjusted metrics

- Calculated from training set **but** penalize model complexity
- Examples: C_p, AIC, BIC, adjusted R^2



Details on metrics

Mallow's C_p:

$$C_p = \frac{1}{n}(\mathrm{RSS} + 2d\hat{\sigma}^2)$$

- d=# of non-zero parameters
- $\hat{\sigma}^2$ = estimate of ϵ variance

AIC:

$$AIC = -2\log(L) + 2*d$$

- d=# of non-zero parameters
- L is maximized likelihood based on model
- With linear model with $\epsilon \sim \mathrm{Normal}(0, \sigma^2)$, AIC=$C_p$

Details on metrics

BIC:

$$BIC = \frac{1}{n}(RSS + \log(n)d * \hat{\sigma}^2)$$

- Uses different penalty then C_p
- Since $\log(n) > 2$ for $n > 7$, BIC penalty generally higher
- $\rightarrow$ smaller model then C_p often chosen

Adjusted R^2:

$$\text{Adjusted } R^2 = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)}$$

where TSS is total sum of squares - Like R^2, but with penalty added for more complex model

Using cross-validation

- **Recall**: selection methods return candidate models M_k for $k = 0, 1, \ldots$
- **Goal**: select *best* model $\leftrightarrow$ select $\hat{k}$, $M_{\hat{k}}$
- To do this, need to compute each model's test set error using CV
- **Better then using adjusted metrics**
  - *Direct estimate of test set*
  - *Doesn't require estimate of error variance* σ^2
  - *More flexible, as doesn't require likelihood,* σ^2 *estimator*
- **But** may be computationally costly

Model selection in R

- Let's predict cancer mortalities at the county level
  - *Use AIC with backward selection*

```
cancer_data <- read_csv("../data/cancer_reg.csv") %>%
  select(-avgAnnCount, -avgDeathsPerYear, -incidenceRate, -binnedInc, -Geography) %>%
  select(TARGET_deathRate, medIncome, povertyPercent, MedianAge:BirthRate) %>%
  drop_na()

lm_stepwise <- train(TARGET_deathRate~., data=cancer_data,
                     method="leapBackward",
                     tuneGrid = data.frame(nvmax = 1:(dim(cancer_data)[2]-1)),
                     trControl = trainControl(method = "cv"))
```

```
## Linear Regression with Backwards Selection
##
## 591 samples
##  26 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 532, 532, 534, 532, 531, 533, ...
## Resampling results across tuning parameters:
##
##   nvmax  RMSE      Rsquared   MAE
##    1     24.99498  0.1625159  18.62722
##    2     24.37371  0.2109940  18.35073
##    3     23.71663  0.2551368  17.67099
```

```
##    4     23.86133  0.2488781  17.72325
##    5     23.65605  0.2633266  17.54385
##    6     23.70479  0.2625729  17.46915
##    7     23.67046  0.2664717  17.61590
##    8     23.39496  0.2843112  17.35506
##    9     23.59097  0.2773196  17.40356
##   10     23.64689  0.2740729  17.39205
##   11     23.57063  0.2795802  17.44957
##   12     23.29212  0.2925899  17.11340
##   13     23.13197  0.3007204  16.98297
##   14     22.79576  0.3170371  16.78720
##   15     22.64654  0.3244291  16.66691
##   16     22.59391  0.3265925  16.60857
##   17     22.61048  0.3258817  16.60352
##   18     22.60786  0.3263568  16.57909
##   19     22.56475  0.3287595  16.53458
##   20     22.59275  0.3272826  16.57637
##   21     22.65431  0.3243107  16.64186
##   22     22.65382  0.3244367  16.64295
##   23     22.61065  0.3263925  16.64384
##   24     22.57927  0.3281544  16.62737
##   25     22.57890  0.3281997  16.62567
##   26     22.58602  0.3278674  16.63561
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was nvmax = 19.
```

```
##       RMSE  Rsquared      MAE Resample
## 1  21.85831 0.3677664 17.91847   Fold08
## 2  23.04136 0.3203861 17.79430   Fold02
## 3  23.25166 0.4063912 17.84603   Fold01
## 4  25.40572 0.1286813 17.59783   Fold05
## 5  20.61690 0.4205600 14.21606   Fold10
## 6  28.33937 0.2017885 19.26403   Fold09
## 7  19.84173 0.5196920 14.20194   Fold07
```

```
## 8  20.42463 0.4917483 14.95377   Fold06
## 9  21.54065 0.2366606 15.91054   Fold04
## 10 21.32721 0.1939206 15.64279   Fold03
```

```
##       RMSE   Rsquared        MAE
## 22.5647535  0.3287595 16.5345757
```

```
##            (Intercept)              medIncome          MedianAgeMale
##           2.559636e+02           3.992729e-04          -8.212816e-01
##         PercentMarried           PctNoHS18_24        PctBachDeg18_24
##           2.336800e+00          -3.938348e-01          -9.269618e-01
##           PctHS25_Over      PctBachDeg25_Over     PctEmployed16_Over
##           7.305994e-01          -1.181867e+00          -1.055444e+00
##     PctEmpPrivCoverage      PctPublicCoverage PctPublicCoverageAlone
##           5.363687e-01          -8.191981e-01           1.557620e+00
##               PctBlack           PctOtherRace   PctMarriedHouseholds
##           2.000031e-01          -1.362176e+00          -2.737930e+00
##              BirthRate
##          -9.802407e-01
```

Model selection in R

- Alternative method

```
lm_stepwise <- train(TARGET_deathRate~., data=cancer_data,
                     method="lmStepAIC",
                     trControl = trainControl(method = "cv"))
```

```
## Linear Regression with Stepwise Selection
##
## 591 samples
##  26 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 531, 531, 532, 531, 532, 532, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   22.37345  0.3385838  16.48653
```

```
##        RMSE  Rsquared      MAE Resample
## 1  21.99004 0.2699433 17.18171   Fold01
## 2  18.60319 0.3798764 14.19004   Fold02
## 3  25.43122 0.1572250 18.86901   Fold03
## 4  20.12955 0.4666180 15.59605   Fold04
## 5  29.95741 0.1592850 20.60013   Fold05
## 6  23.93986 0.2837405 16.84723   Fold06
## 7  18.38660 0.4861199 14.53857   Fold07
## 8  24.27482 0.2313546 18.25960   Fold08
```

```
## 9  22.28671 0.4186312 15.39401   Fold09
## 10 18.73515 0.5330440 13.38893   Fold10
```

```
##       RMSE   Rsquared        MAE
## 22.3734549  0.3385838 16.4865296
```

```
## Stepwise Model Path 
## Analysis of Deviance Table
## 
## Initial Model:
## .outcome ~ medIncome + povertyPercent + MedianAge + MedianAgeMale + 
##     MedianAgeFemale + AvgHouseholdSize + PercentMarried + PctNoHS18_24 + 
##     PctHS18_24 + PctSomeCol18_24 + PctBachDeg18_24 + PctHS25_Over + 
##     PctBachDeg25_Over + PctEmployed16_Over + PctUnemployed16_Over + 
##     PctPrivateCoverage + PctPrivateCoverageAlone + PctEmpPrivCoverage + 
##     PctPublicCoverage + PctPublicCoverageAlone + PctWhite + PctBlack + 
##     PctAsian + PctOtherRace + PctMarriedHouseholds + BirthRate
## 
## Final Model:
## .outcome ~ medIncome + MedianAgeMale + PercentMarried + PctNoHS18_24 + 
##     PctBachDeg18_24 + PctHS25_Over + PctBachDeg25_Over + PctEmployed16_Over + 
##     PctEmpPrivCoverage + PctPublicCoverage + PctPublicCoverageAlone + 
##     PctBlack + PctOtherRace + PctMarriedHouseholds + BirthRate
## 
## 
##                         Step Df    Deviance Resid. Df Resid. Dev      AIC
## 1                                                 564   265992.6 3664.660
## 2           - povertyPercent  1   0.8819439       565   265993.5 3662.662
## 3         - AvgHouseholdSize  1   3.2698486       566   265996.7 3660.669
## 4          - MedianAgeFemale  1  71.7709376       567   266068.5 3658.829
## 5       - PctPrivateCoverage  1 133.2173116       568   266201.7 3657.124
## 6  - PctPrivateCoverageAlone  1  98.6494862       569   266300.4 3655.343
## 7                 - PctAsian  1 267.5576578       570   266567.9 3653.937
## 8               - PctHS18_24  1 418.4249581       571   266986.4 3652.864
## 9          - PctSomeCol18_24  1 572.0872132       572   267558.5 3652.129
```

```
## 10              - PctWhite  1 618.7957889       573   268177.2 3651.494
## 11             - MedianAge  1 695.6916331       574   268872.9 3651.025
## 12   - PctUnemployed16_Over  1 594.3327859       575   269467.3 3650.330
```

```
## 
## Call:
## lm(formula = .outcome ~ medIncome + MedianAgeMale + PercentMarried + 
##     PctNoHS18_24 + PctBachDeg18_24 + PctHS25_Over + PctBachDeg25_Over + 
##     PctEmployed16_Over + PctEmpPrivCoverage + PctPublicCoverage + 
##     PctPublicCoverageAlone + PctBlack + PctOtherRace + PctMarriedHouseholds + 
##     BirthRate, data = dat)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -71.847 -12.018   0.679  11.965 109.524 
## 
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             2.560e+02  2.487e+01  10.291  < 2e-16 ***
## medIncome               3.993e-04  1.745e-04   2.288  0.02253 *  
## MedianAgeMale          -8.213e-01  3.499e-01  -2.347  0.01925 *  
## PercentMarried          2.337e+00  4.155e-01   5.624 2.91e-08 ***
## PctNoHS18_24           -3.938e-01  1.374e-01  -2.867  0.00429 ** 
## PctBachDeg18_24        -9.270e-01  2.893e-01  -3.204  0.00143 ** 
## PctHS25_Over            7.306e-01  2.442e-01   2.992  0.00289 ** 
## PctBachDeg25_Over      -1.182e+00  4.052e-01  -2.917  0.00368 ** 
## PctEmployed16_Over     -1.055e+00  2.412e-01  -4.375 1.44e-05 ***
## PctEmpPrivCoverage      5.364e-01  1.988e-01   2.698  0.00717 ** 
## PctPublicCoverage      -8.192e-01  4.695e-01  -1.745  0.08157 .  
## PctPublicCoverageAlone  1.558e+00  4.795e-01   3.248  0.00123 ** 
## PctBlack                2.000e-01  8.702e-02   2.298  0.02191 *  
## PctOtherRace           -1.362e+00  3.125e-01  -4.359 1.55e-05 ***
## PctMarriedHouseholds   -2.738e+00  3.827e-01  -7.155 2.56e-12 ***
## BirthRate              -9.802e-01  5.018e-01  -1.953  0.05125 .  
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.65 on 575 degrees of freedom
## Multiple R-squared:  0.386,  Adjusted R-squared:   0.37
## F-statistic:  24.1 on 15 and 575 DF,  p-value: < 2.2e-16
```