Model selection

Yue Jiang

Duke University

A disclaimer

The following material was used during a live lecture. Without the accompanying oral comments and discussion, the text is incomplete as a record of the presentation. A full recording may be found via Zoom on the course Sakai site.

An activity

Load the dataset `dat`, which contains 2,000 observations of an outcome variable `y` and 100 predictors labeled `X1` through `X100`. Using your favorite model/variable selection techniques for linear regression, come up with the "best" model (you choose what that means!).

What was the form of the final model you chose? Is there sufficient evidence that at least one of these β terms is non-zero? That is, test the following hypotheses:

- $H_0 : \beta_1 = \beta_2 = \cdots = \beta_{100} = 0$
- H_1 : At least one $\beta_k \neq 0$ for $k = 1, 2, \cdots, 100$.

What might you conclude with these data?

Model selection

Why do we even want to select a model?

We might want to **explain** or **describe** relationships in populations (often by performing statistical inference on a sample from this population). In some cases, this might give us causal or mechanistic understanding of the phenomenon at hand; in others, we might only be able to explore associations.

We alternatively might want to **predict** values of new observations, often by examining how well our model does using our data at hand in terms of performance on holdout data.

Explanation vs. prediction

Is there a relationship between buprenorphine-naloxone and time-to-relapse (while controlling for clinical, behavioral, and demographic factors)?

Given a patient's characteristics (including whether they receive BUP-NX), can we predict their time-to-relapse? Can we come up with a model that does the best job of doing so?

We might use similar (or even the same) tools to both cases, but these are **distinct** questions. With that said, we're often interested in addressing both questions.

See also this Stack Exchange discussion here.

Model selection

Why do we even want to select a model?

"Choosing the best model" is an unfortunate and perhaps misleading use of words. In explaining our data, there might be multiple reasonable models that all do an acceptable job explaining.

We might also consider parsimony. Ease of explanation and conceptualization might more than make up for a marginal increase in some metric.

Cross-validation for prediction

Leave-one-out CV: sequentially leave out each one of our observations and fit model on the remainder to capture prediction accuracy.

- Convenient closed-form results for linear regression, otherwise computationally expensive.
- Not great in practice - only small changes are considered.

k-fold CV: partition data into k mutually exclusive partitions; use each partition as test set for models trained on remainder of data (so each datapoint is used in the test set once).

- Works well in practice, but can be sensitive to random partitions
- Averaging over random partitions can help with robustness.

Stepwise selection

R^2_{adj}, C_p, AIC, BIC, etc. are all used to quantify some notion of prediction error. These quantities are often used as stopping criteria for stepwise regression techniques.

In examining these quantities, they seek to maximize (or minimize) some function of the (log-)likelihood, with perhaps a penalty term based on the number of chosen parameters/variables in the model.

- AIC penalizes linearly in p
- BIC penalizes like $p \log(n)$

Stepwise selection

Greedy algorithms for variable selection.

Backward elimination:

- Evaluate some criterion for a "full" model
- Sequentially delete each variable and evaluate the same criterion
- Choose the model with the best chosen metric, repeating as necessary until further deletion does not improve the model

Forward selection:

- Evaluate some criterion for an intercept-only model
- Sequentially add each variable and evaluate the same criterion
- Choose the model with the best chosen metric, repeating as necessary until further addition does not improve the model

Hybrid forward/backward methods also exist, adding or deleting as necessary.

Stepwise selection

Automatic stepwise methods are very popular, but not great in practice (e.g., horrible in the presence of colinearity, and this isn't even the worst problem with them!).

> *"Personally, I would no more let an automatic routine select my model than I would let some best-fit procedure pack my suitcase."* - Ronan Conroy.

LASSO (and friends)

Instead of ordinary least squares, which minimizes the sum of squared errors over the covariate vector β

$$\min_{\beta} \frac{1}{n} \left|\left|Y - X\beta\right|\right|_2^2,$$

LASSO adds the additional L_1 constraint

$$\left|\left|\beta\right|\right|_1 \leq t,$$

and is a special case of elastic net.

Similar automatic regularization has been proposed for other settings such as for GLMs, GEE, and survival data. Note that LASSO has a nice Bayesian interpretation in terms of a double-exponential prior on regression parameters.

LASSO (and friends)

The penalty that LASSO imposes on regression coefficients takes a specific shape - because it is "sharp" at zero, LASSO can also set some coefficients to be *exactly* zero, thus resulting in variable selection (although it has some issues...).

Changing the form of this penalty results in other techniques, e.g., MCP, SCAD penalties.

Penalized regression techniques are a very interesting field of research and have application beyond variable selection - I encourage you to take further courses on then. Of course, you should always be careful with such automatic regularization procedures for variable selection purposes!

Classification

What about for classification tasks?

- Bagging: Classifiers are created from bootstrap resamples
- Boosting: Similar, but in future iterations of resampling, weighting is used for misclassifications to "focus" on areas where the model performed poorly.

Back to our activity...

```
m1 <- lm(y ~ ., data = dat); round(summary(m1)$coef[-1,], 3)
```

```
##     Estimate Std. Error t value Pr(>|t|)
## X1     0.086      0.031   2.739    0.006
## X2    -0.018      0.032  -0.555    0.579
## X3     0.000      0.032  -0.008    0.993
## X4     0.016      0.032   0.517    0.605
## X5    -0.017      0.032  -0.535    0.593
## X6    -0.031      0.032  -0.995    0.320
## X7    -0.032      0.032  -1.014    0.311
## X8    -0.006      0.030  -0.208    0.835
## X9     0.029      0.032   0.890    0.374
## X10   -0.013      0.031  -0.400    0.689
## X11   -0.064      0.032  -2.029    0.043
## X12    0.015      0.031   0.466    0.641
## X13    0.034      0.031   1.087    0.277
## X14    0.024      0.032   0.747    0.455
## X15    0.004      0.032   0.117    0.907
## X16   -0.046      0.031  -1.488    0.137
## X17   -0.025      0.031  -0.804    0.422
## X18   -0.032      0.032  -1.004    0.316
## X19    0.070      0.031   2.251    0.025
## X20    0.042      0.031   1.358    0.175
```

Back to our activity...

```
m2 <- lm(y ~ X1 + X11 + X19, data = dat)
summary(m2)
```

```
##
## Call:
## lm(formula = y ~ X1 + X11 + X19, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.81047 -0.65261  0.00075  0.65403  3.03093
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01698    0.03122   0.544  0.58648
## X1           0.08442    0.03087   2.735  0.00636
## X11         -0.06220    0.03151  -1.974  0.04866
## X19          0.06947    0.03093   2.246  0.02492
##
## Residual standard error: 0.9851 on 996 degrees of freedom
## Multiple R-squared:  0.01626,    Adjusted R-squared:  0.01329
## F-statistic: 5.486 on 3 and 996 DF,  p-value: 0.000969
```

Back to our activity...

```r
library(MASS)
m3 <- stepAIC(m1, direction = "both", trace = F)
summary(m3)
```

```
##
## Call:
## lm(formula = y ~ X1 + X11 + X16 + X19, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.83423 -0.65687 -0.00603  0.64656  3.06008
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01436    0.03123   0.460  0.64570
## X1           0.08579    0.03086   2.780  0.00554
## X11         -0.06139    0.03149  -1.950  0.05149
## X16         -0.04919    0.03057  -1.609  0.10795
## X19          0.06933    0.03091   2.243  0.02510
##
## Residual standard error: 0.9843 on 995 degrees of freedom
## Multiple R-squared:  0.01881,    Adjusted R-squared:  0.01486
## F-statistic: 4.769 on 4 and 995 DF,  p-value: 0.0008177
```

Back to our activity...

```
set.seed(123); n <- 1000; p <- 20; y <- rnorm(n, 0, 1)
x <- matrix(rnorm(n * p), nrow = n); dat <- data.frame(y, x)
m4 <- lm(y ~ X1 + X11 + X16 + X19, data = dat); summary(m4)
```

```
##
## Call:
## lm(formula = y ~ X1 + X11 + X16 + X19, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.83423 -0.65687 -0.00603  0.64656  3.06008
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01436    0.03123   0.460  0.64570
## X1           0.08579    0.03086   2.780  0.00554
## X11         -0.06139    0.03149  -1.950  0.05149
## X16         -0.04919    0.03057  -1.609  0.10795
## X19          0.06933    0.03091   2.243  0.02510
##
## Residual standard error: 0.9843 on 995 degrees of freedom
## Multiple R-squared:  0.01881,    Adjusted R-squared:  0.01486
## F-statistic: 4.769 on 4 and 995 DF,  p-value: 0.0008177
```

Back to our activity...

Remember, if the null hypothesis is true, then p-values should have a uniform distribution on (0, 1).

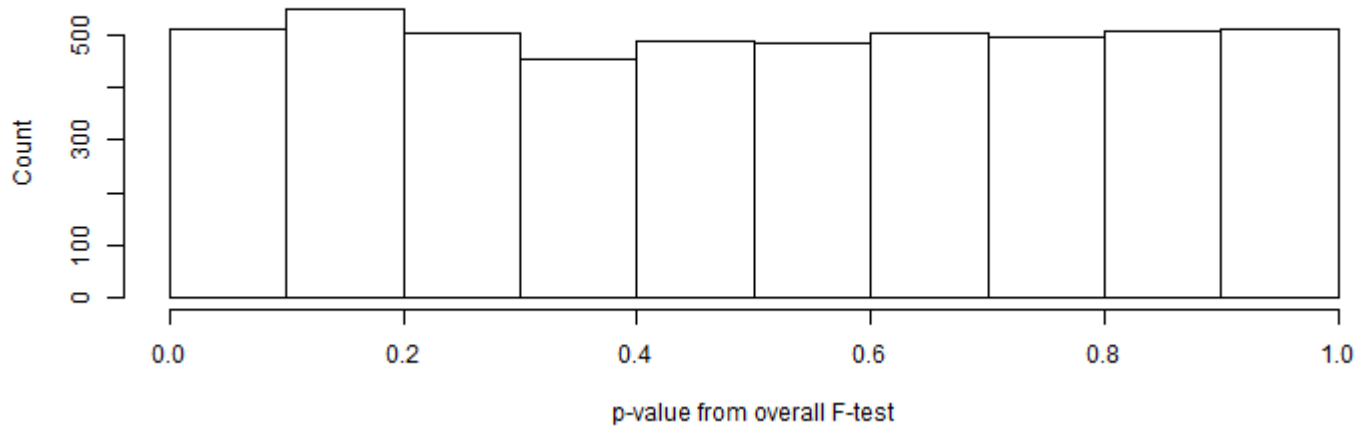
```
temp <- numeric(5000)
for(i in 1:5000){
  set.seed(i)
  y <- rnorm(n, 0, 1); x <- matrix(rnorm(n * p), nrow = n);
  dat <- data.frame(y, x)
  mod <- summary(lm(y ~ ., data = dat))
  temp[i] <- pf(mod$fstatistic[1], mod$fstatistic[2], mod$fstatist
}
```

Back to our activity...

```
round(temp[1:25], 3)
```

```
##  [1] 0.827 0.093 0.366 0.062 0.894 0.330 0.152 0.589 0.348 0.600 0.459
## [13] 0.840 0.388 0.052 0.724 0.899 0.291 0.197 0.126 0.302 0.437 0.035
## [25] 0.405
```

```
hist(temp, xlab = "p-value from overall F-test",
     ylab = "Count", main = "")
```



Back to our activity...

```
temp <- numeric(25)
for(i in 1:25){
  set.seed(i)
  y <- rnorm(n, 0, 1); x <- matrix(rnorm(n * p), nrow = n);
  dat <- data.frame(y, x)
  mod <- summary(stepAIC(lm(y ~ ., data = dat), direction = "both"
  temp[i] <- pf(mod$fstatistic[1], mod$fstatistic[2], mod$fstatist
}
round(temp, 3)
```

```
##  [1] 0.119 0.001 0.026 0.001 0.035 0.009 0.001 0.018 0.013 0.032 0.011
## [13] 0.075 0.003 0.001 0.055 0.120 0.002 0.001 0.003 0.006 0.002 0.000
## [25] 0.005
```

What do you notice? What's going on?

The problem with post-selection inference

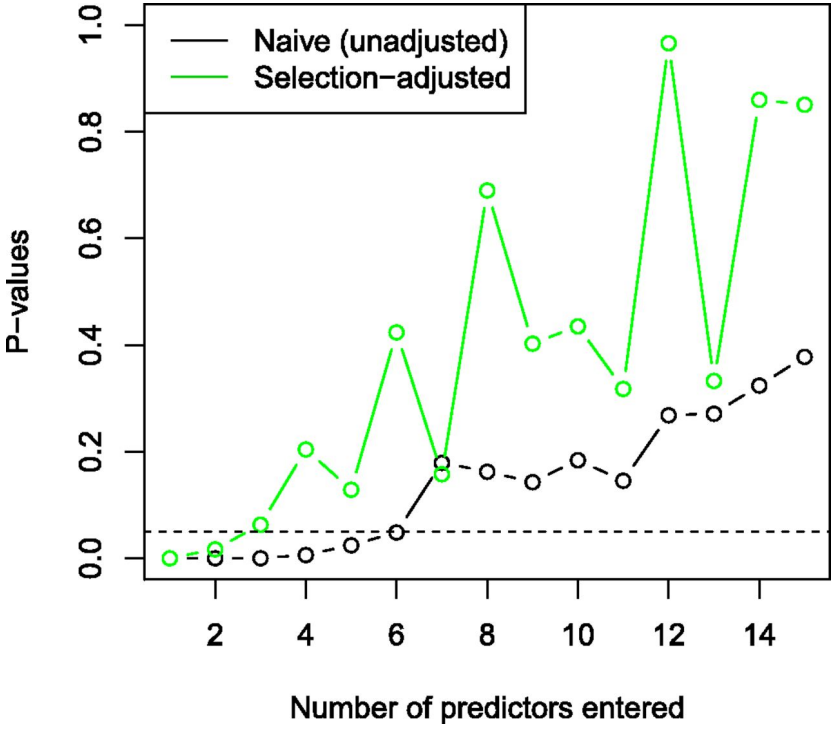
Note that we chose our final model *because it did a good job with these data*.

- In classical statistical theory, data is assumed to be generated from a known model, with inference for parameters performed on this known model.
- In the real world, it's often the case that the generating mechanism is unknown, *with a model being chosen based on the data*.
- The selection process that produces a model is itself random, which affects the sample distriutions of the post-selection parameter estimates. These may change dramatically after conditioning on model selection procedures.

Performing statistical inference on the same data we used to choose our model is not appropriate - *even if* asymptotically your selection procedure correctly chooses variables almost surely!

The problem with post-selection inference

- Rhee et al. (2003) studied HIV drugs and predicted drug resistance based on location of genetic mutations.
- Using forward selection, naive p-values suggest six significant predictors.
- However, selection-adjusted p-values only suggest two significant predictors (figure below from Taylor and Tibshirani, 2015, *PNAS*):



The problem with post-selection inference

So what can we do?

- Split the data - perform model selection using only part of the data, and then use that model on the remaining data.
- Post-selection inferential methods - e.g., relying on polyhedral lemma for LASSO, etc.
- Ignore the problem entirely.

Post-selection inference is an active field of research and many powerful and complex tools are available. In your own analyses, just be cautious of running into this situation.