

BIFX 553 - Discussion 6

Randy Johnson

February 23, 2017

Setup

```
library(shrink)
library(tidyverse)
library(broom)
theme_set(theme_classic() +
  theme(axis.line.x = element_line(color = 'black'),
        axis.line.y = element_line(color = 'black'),
        text = element_text(size = 15)))
```

Multivariable Modeling Strategies

Prespecification of predictor complexity

- ▶ Deciding on a model before analysis of the data is the best policy when model building. Unless you have a good reason to remove something from the model after beginning the analysis, you probably shouldn't.
- ▶ If you don't know much about the system you are modelling, decide before hand how complex you want your model to be.

Variable selection

There are automated methods for building your model (e.g. forward or backward selection). These are best avoided because:

- ▶ r^2 values will be biased too high.
- ▶ Test statistics do not have the correct distribution.
- ▶ Standard errors / confidence intervals are biased smaller than they should be.
- ▶ p-values will be too small
- ▶ Collinearity makes variable selection arbitrary
- ▶ It allows you to turn off your brain.

Overfitting

Overfitting is likely to occur if the number of parameters is too high.

At some point, new parameters added to the model will fit sampling noise rather than the information we are trying to make inferences about.

A common rule of thumb for picking the number of variables is $\frac{\text{sample size}}{10}$ or $\frac{\text{sample size}}{20}$ for continuous variables (see pg 61 of Harrell, 2001 for more information).

For genetic markers, this is more like $\frac{MAF}{10}$ or $\frac{MAF}{20}$, where MAF = minor allele frequency.

Regression to the Mean

When using a model to predict outcomes in independent data, predicted values that are far away from the mean will tend to over estimate the deviation from the mean, while predicted values near the mean will tend to be more accurate.

```
set.seed(28347) # this is the first seed I tried
n_pred <- 100
pred <- data_frame(x1 = rnorm(n_pred),
                   x2 = rnorm(n_pred),
                   x3 = rnorm(n_pred),
                   x4 = rnorm(n_pred),
                   x5 = rnorm(n_pred),
                   y = x1 + x2 + x3 + rnorm(n_pred))

model <- lm(y ~ x1 + x2 + x3 + x4 + x5,
           data = pred, y = TRUE, x = TRUE)
```

Regression to the Mean

```
tidy(model)
```

##	term	estimate	std.error	statistic	p.value
## 1	(Intercept)	-0.09277074	0.09253871	-1.0025073	3.186723e-01
## 2	x1	1.06597935	0.09177622	11.6149837	7.225899e-20
## 3	x2	1.05377814	0.08049772	13.0907824	6.649894e-23
## 4	x3	1.07985388	0.08990097	12.0115926	1.082171e-20
## 5	x4	-0.07176524	0.09221576	-0.7782318	4.383869e-01
## 6	x5	0.24741839	0.09523223	2.5980530	1.088397e-02

Shrinkage

The regression coefficient for x_5 is too big, and the coefficients for x_1 , x_2 and x_3 are all slightly too big. Predictions of y using this model will be slightly too big, and this effect will be magnified for points lying further from the mean.

Shrinkage can be used to compensate some for this (models with more parameters will benefit more).

```
shrink(model)
```

```
## Error in eval(expr, envir, enclos): could not find function
```

Collinearity

- ▶ Standard errors will be inflated.
- ▶ Correlated coefficients are difficult to estimate, because data provide limited information when holding another, correlated parameter constant.
- ▶ It is difficult to sort out the important variable(s)

Data reduction

Using what is known about the system you are modelling can be one of the best ways to reduce the number of parameters. Read the literature / consult experts.

Clustering or summarizing (e.g. using principal components) can be another powerful tool to reduce the dimensionality of your data (more on this to come in future discussions).

Overly influential variables

Sources include: - Data entry errors - Measurement error (if a particular variable is too noisy, it may be best to just throw out all observations of that particular predictor) - Data need to be transformed - Not enough data to model a complex system - Model doesn't fit a specific subset of the data

Comparing two models

- ▶ Consider model assumptions / all of the above.
- ▶ A smaller model isn't always preferred if important statistically non-significant variables are omitted.
- ▶ Does it agree with our understanding of the system (e.g. our disease model)?

Summary: Developing predictive models

- ▶ Collect lots of data.
- ▶ Do your homework when developing the disease model/hypothesis.
- ▶ Impute missing data if there isn't much. Will people using your model have trouble collecting all the variables you are using?
- ▶ Reduce the number of predictors if necessary/practical.
- ▶ Check all assumptions.
- ▶ Influential observations are especially bad in predictive models.
- ▶ Validate your final model.
- ▶ Use parameter shrinkage if possibility of over fitting.
- ▶ Develop some simplified alternative models.

Summary: Developing models for effect estimation

- ▶ Parsimony isn't as important in this context.
- ▶ Interactions should still receive careful consideration.
- ▶ Imputation of predictors is less helpful.
- ▶ Check all assumptions.

Summary Developing models for hypothesis testing

- ▶ Check all assumptions.
- ▶ Interactions should still receive careful consideration.
- ▶ Imputation might be helpful, but often results in no net benefit.

Binary Logistic Regression

Model assumptions

- ▶ Assessment of model fit
- ▶ Collinearity
- ▶ Overly influential variables

Odds ratio, Risk ratio and Risk difference

- ▶ Definition
- ▶ Interpretation of model parameters

Estimation, test statistics, residuals

Quantifying predictive ability

Validating the predictive ability

Describing the model

Survival Analysis

Ordinal Logistic Regression

Proportional odds model