STA 2260 Project Guidelines

Kevin Bailey

Due 12/11/2020

General Comments

- Counts as an **extra** 5% towards your overall grade.
- You will hand in your code and data set, along with a ∼3-5 page paper (including **reasonably sized** tables/figures).
    - Compress all of these to a `.zip` file and submit to blackboard. If the file is too large, DM it to me on Discord *and* e-mail it to me with "[**STA 2260 Project**]" in the title.
- There will be a video corresponding to these guidelines to show you how to import data in `R` among other things. I will also provide an example of a project here.
- You do not need to spend a ton of time on this, but good analysis and model creation can look good for your portfolio!

Data

There are a few places you can get your data from. I'll provide two sources, but if you have your own data set you would like to work with and are interested in, feel free to message me and ask about it. **You MUST confirm your data set with me before you actually start working on this.**

- **UCI Machine Learning Repository**: http://archive.ics.uci.edu/ml/datasets.php
    - On the left side under *Default Task*, click **Regression**.
    - Pick a data set that is **NOT** a time series. Check for this under the column *Data Types*, if it mentions time-series then do not use it.
    - Many of the data sets here have descriptions in the data set itself that you can read about. This description will commonly mention the response variable and describe the predictors, but not always.
        - The **response** is what you plan to predict. The **predictors** are what you plan to *use* to predict the response.
    - I will try to scoop up some data sets from here and post them in their own channel, potentially making your choice of data easier.
- **Kaggle**: https://www.kaggle.com/datasets
    - Basically, just find a data set that seems interesting to you here.
    - You can apply filters for data here and look at tags you may be curious about, I also suggest looking only for `.csv` (comma-separated-values) files and applying that filter as well.

Methods

Your analysis should perform the following:

- Identify both your **response** as well as what variables could theoretically be **predictors**.

  - The predictors do not *have* to make sense, but they should be included in the analysis if it is possible from a data standpoint to use them in a future analysis.
    - ∗ Sometimes, you may not think certain variables will be useful as predictors, but you may be surprised when actually analyzing them and making models.
  - You may want to pick a data set with a small number of attributes (predictors/columns) for this reason.

- For all possible predictors, produce scatter plots to see its relationship with the response.

Model 1

- Fit a *one-predictor* model. See the guide for help on this.
  - Consider a few possible models here. Which predictors seem they would do the best to predict the response if *only* that predictor would be used?
- Call this model `(m1)`.
  - Compute both the R^2 and AIC values.
  - Show a scatterplot and produce the fitted line.

Model 2

- Compare *at least two new models* that include **multiple** predictors.
  - You can include quadratic terms, cubic terms, interaction terms, or similar if you think it will help.
  - Compute the R^2 and AIC for *both* of these models.
- Out of the two models considered in this part, call the model you choose `(m2)`.
  - Usually, the model with the lower AIC is the "better" model, but this is not a binary decision. If the AIC are similar for both models, but one has a much better R^2, you may wish to choose the one with the better R^2 even if its AIC is slightly worse.
- Compare `(m1)` and `(m2)` on the basis of both R^2 and AIC. You should clearly state which model explains more variability in the data, and which model should do a better job at predicting future data.

Grade Breakdown

- (20%) Correct identification of a response and viable predictors
- (20%) Determining reasonable set of predictors (scatter plot analysis, discussion)
- (20%) Analysis of Model 1.
- (20%) Analysis of Model 2.
- (20%) Overall clarity and required information in the paper.
  - Paper should include the final fitted plot for Model 1, along with R^2 and AIC values for both Model 1 and 2.

  - Include the final formula for both Model 1 and 2.
  - The final formula for Model 2 has potential to be quite ugly if you have a lot of predictors. It is acceptable to me to just paste the model summary from `R` instead of writing out the whole model.

Helpful Tips

Linear Regression

Linear Regression hypothesizes a **linear relationship**, i.e. one of the form $mx+b$. Specifically, the assumption is stated

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i,$$

where ϵ_i is random noise (uncontrollable).

- Each ϵ_i is independent of the others.
- $\mathbb{E}[\epsilon_i] = 0$, $\text{Var}[\epsilon_i] = \sigma^2$

The goal is to find the line of best fit based on the observed data.

- Choose β_0 and β_1 optimally, to “best fit” the data.
- If the resulting “best fit” estimates are $\hat{\beta_0}$ and $\hat{\beta_1}$, then predictions are done using the fitted line:

  $$\hat{y} = \hat{\beta_0} + \hat{\beta_1} x.$$

  Note that there is no random error here (ϵ_i) because this is a plug-in function that gives an estimate for a given x.

Residual Sum of Squares

Usually the “best” line is chosen to minimize the **residual sum of squares**(RSS), sometimes also called *squared error* or L_2 loss.

- y_i = observed value, $\hat{y_i}$ = predicted value (value on the line).

  $$\implies RSS = \sum_{i=1}^{n} (y_i - \hat{y_i}^2)$$

  is the **sum of squared distances from the observations to the line.**
- I won’t go too far into details, but basically you re-write $\hat{y_i}$ as $\hat{\beta_0} + \hat{\beta_1} x_i$, and then take the partial derivatives of the RSS with respect to $\hat{\beta_0}, \hat{\beta_1}$, set them equal to 0 and solve to get the optimal estimates.

Going Beyond Linearity

The above is using one predictor, but we can simply add in multiple predictors like so:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \epsilon.$$

where p is your number of predictors; throw some hats in there and we have a model. We need to be careful with this though, because some possible issues are **overfitting** (adding more and more predictors) and collinearity (confounding variables, where two or more variables are correlated and used in tandem for prediction which can lead to issues.

Some issues that can arise from collinearity are:

- Model interpretability is hurt significantly when *both* variables are included.
- The confounding variables have nearly no effect on model performance.
- When the confounding variables are highly correlated, computers may run into numerical issues when solving for estimates.

Model Performance

Linear models fit on the basis of *squared error*,

- Choose the β to minimize the distance from the predicted values to the observed values.
- Adding more terms will **always** produce a better model (in-sample) on this basis.

So if we were to fit

$$\hat{y} = \hat{\beta_0} + \hat{\beta_1} X_1,$$

and then also fit

$$\hat{y} = \hat{\beta_0} + \hat{\beta_1} X_1 + \hat{\beta_2} X_2^2,$$

the second model will **always** do better on the basis of reducing in-sample RSS. This is because we always have the option of simply reusing the $\hat{\beta_0}$ and $\hat{\beta_1}$ estimates from the first model. While tweaking the parameters, we can just set $\hat{\beta_2} = 0$ and obtain the first model. So we will always get *at least* the same squared error as the first model, with a potentially better one from the second model. Overfitting might occur when doing this, meaning you are predicting *very* well in-sample as you add more and more estimates, but will predict *horribly* out of sample.

Penalize Overfitting!

The RSS above doesn't penalize overfitting, it actually encourages it. Because of this, you will usually want to compare models from the basis of their AIC (Akike Information Criterion).

- In the case of linear regression with normally distributed errors, the AIC is

  $$\text{AIC} = \frac{1}{n\hat{\sigma}^2}(RSS + 2p\hat{\sigma}^2),$$

  where

  - p is the number of parameters included in the model
  - RSS is the sum of squared errors from before

  - $\hat{\sigma}^2$ is an estimate of the noise variance

- The lower AIC is better when comparing two models.
  - Worst fit to the data $\implies$ RSS increases $\implies$ AIC increases
  - More terms added to the model (possibly overfitting) $\implies$ p increases $\implies$ AIC increases

There are many other tools to assess model performance (cross-validation is the more universal and popular one, but I will not cover it as it is more complicated, though very important in Machine Learning). The R^2 is another metric, but is more about the proportion of variance that can be explained by our model (the close to 1 it is, the better the model explains the variance). R^2 does not penalize overfitting, so **Adjusted** R^2 is more commonly reported (seen in my example and in the summary output of an `lm(...)` object in `R`.