

# Text as Data: Homework 4

Jeff Ziegler

August 22, 2017

In this problem set we're going to make a slight detour from text as data to predict student's drinking habits. While this might seem like a big departure, we're going to learn more about some methods that are fundamental to classification problems. Further, we're going to see that the methods for learning for text as data problems are more generally useful. Trust me, this is going to help you analyze text better (but always keep in mind what would be similar...) The data come from a public health study of Portuguese students. You can read more about the variables (and their interpretation) [here](#). We're going to model the sum of weekday and weekend drinking activity. The data are stored in `StudentDrinking.RData` on canvas. The dependent variable is, `alcohol`, which is a measure of alcohol consumption. The bigger `alcohol` is, the more students drink. The covariates are stored in `X`.

## 1 Comparing Coefficients from OLS, LASSO, Ridge, and Elastic Net

We first want to explore the behavior of OLS, LASSO, and Ridge applied to the data.

- i) Fit a linear regression of alcohol on the covariates in the included data
- ii) Using `cv.glmnet` fit a LASSO regression of alcohol on the covariates
- iii) Using `cv.glmnet` fit a Ridge regression of alcohol on the covariates
- iv) Using `cv.glmnet` fit an elastic-net regression of alcohol on the covariates, with  $\alpha = 0.5$ . Explain what  $\alpha = 0.5$  implies about the model you're fitting.

```
1 # Problem 1
2
3 # load libraries and .csv files
4 library(glmnet); library(data.table)
5 load("~/Documents/Git/WUSIL_textAnalysis/StudentDrinking.RData")
6
7 # i) fit a linear regression of alcohol on the covariates in the included data
8 olsModel <- lm(alcohol~X)
9 # ii) fit a lasso regression of alcohol on the covariates
10 # alpha = 1 for lasso , alpha=0 for ridge
```

```

11 lassoModel <- cv.glmnet(x = X, y = alcohol, alpha = 1)
12 # iii fit a ridge regression of alcohol on the covariates
13 ridgeModel <- cv.glmnet(x = X, y = alcohol, alpha = 0)
14 # iv) fit an elastic-net regression of alcohol on the covariates (alpha = 0.5)
15 elasticModel <- cv.glmnet(x = X, y = alcohol, alpha = 0.5)
16 # alpha = 0.5 implies a "balance" between the penalty occurred
17 # under ridge (lambda*sum(beta_j)) and lasso (lambda*sum(abs(beta_j)))

```

v) Using your models from (i-iv) let's examine the behavior of the coefficient on **male** as  $\lambda$  increases

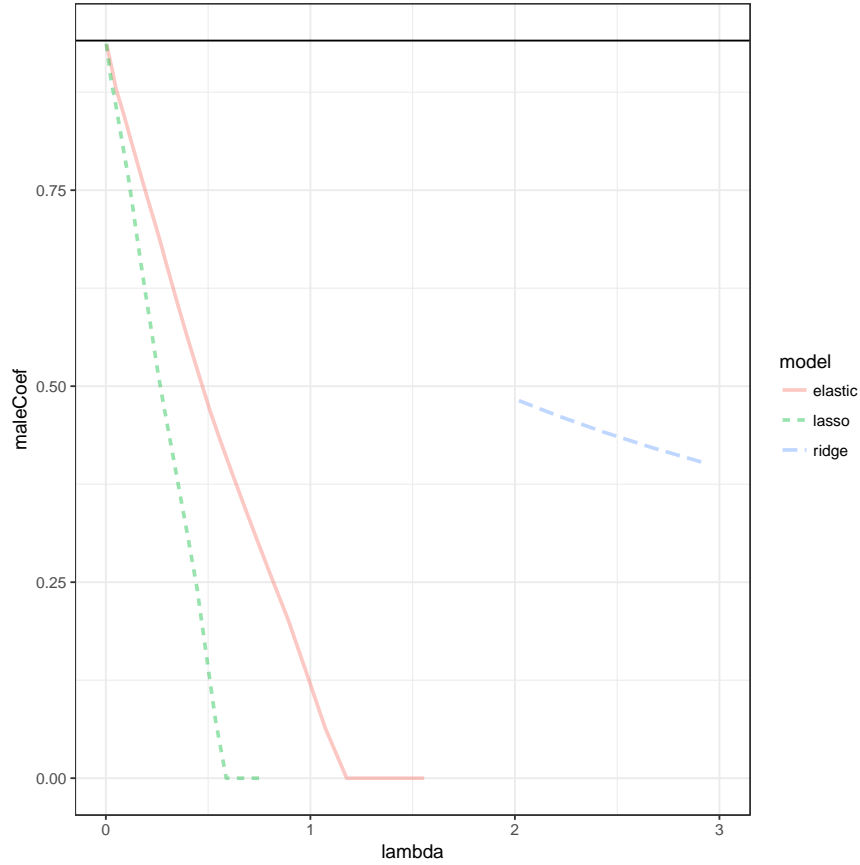
- a) Suppose `glmnet.obj` contains the results from applying `cv.glmnet`. To obtain the coefficient values for the sequence of  $\lambda$  values tested in `cv.glmnet`, we use the coefficient function `coef(glmnet.obj, s = glmnet.obj$lambda)`. Use this function to obtain a matrix of coefficients for the models used in (ii-iv).
- b) Using the matrix for each method, plot the coefficient on **male** against the value of  $\lambda$  from the models in ii-iv. Include the coefficient from OLS as a flat line. What do you notice as  $\lambda$  increases?

```

1 # v) a) task: obtain a matrix of coefficients for the models used in (ii-iv).
2 lassoLambda <- data.table("maleCoef" = t(as.matrix(coef(lassoModel, s =
  lassoModel$lambda)))[,3],
3                           "lambda"=lassoModel$lambda, "model"="lasso")
4 ridgeLambda <- data.table("maleCoef" = t(as.matrix(coef(ridgeModel, s =
  ridgeModel$lambda)))[,3],
5                           "lambda"=ridgeModel$lambda[1:65], "model"="ridge")
6 elasticLambda <- data.table("maleCoef" = t(as.matrix(coef(elasticModel, s =
  elasticModel$lambda)))[,3],
7                            "lambda"=elasticModel$lambda[1:65], "model"="
  elastic")
8
9 # b) plot the coefficient on male against the value of lambda from the models
  in ii-iv
10 # and include the coefficient from OLS as a flat line
11 # merge data
12 plotDataframe <- merge(lassoLambda, ridgeLambda, by=c("maleCoef", "lambda", "
  model"), all=T)
13 plotDataframe <- cbind(olsModel$coefficients[3], merge(elasticLambda,
  plotDataframe, by=c("maleCoef", "lambda", "model"), all=T))
14
15 pdf("~/Documents/Git/WUSTL_textAnalysis/HW4lambda.pdf")
16 # we can see from the plot that as lambda -> 0, we get closer to the ols coef
17 # which is displayed as the solid, horizontal black line at 0.94
18 ggplot(plotDataframe, aes(x = lambda, y = maleCoef, col = model, linetype =
  model)) +
19   geom_line(lwd = 1, alpha=.4) + geom_hline(yintercept=plotDataframe$V1) +
20   theme_bw() + scale_x_continuous(limits = c(0, 3))
21 dev.off()

```

Figure 1: The estimated coefficient of `male` as  $\lambda$  varies.



## 2 Cross-Validation, Super Learning and Ensembles

We're going to assess the performance of five models, an unweighted average, and a super-learning average of the methods.

- i) First, set the first 20 rows to the side for use as the validation set.
- ii) We'll first estimate the (unconstrained) super learner weights.
  - a) On the training data (all but the first 20 rows) perform ten fold cross validation, including (1) linear regression, (2) LASSO, (3) Ridge, (4) Elastic-Net, and (5) Random Forest. Obtain 5 predictions for each observation in the training set, one from each observation
  - b) Regress the dependent variable on the out of sample prediction, (without including an intercept).
  - c) Store those weights as  $\mathbf{w}$
- iii) Now, fit all 5 models from (ii)-(a) to the entire training data set and predict the drinking level from the validation set (the data put off to the side).

- iv) Obtain two ensemble predictions.
  - a) Take the unweighted average of the predictions from the methods
  - b) Take the weighted average, using the weights  $\mathbf{w}$ .
- v) You should have 7 predictions. Store those in a matrix and report the correlation between the predictions
- vi) Using the average absolute difference as a loss function assess the performance of each method. Which method performs best? Which performs the worst?

The average absolute difference for method  $k$  is defined as

$$L(\mathbf{Y}, \hat{\mathbf{Y}}_k) = \sum_{i=1}^{N_{\text{validation}}} \frac{|Y_i - \hat{Y}_{ik}|}{N_{\text{validation}}}$$

where  $N_{\text{validation}}$  refers to the number of observations in the validation set  $\hat{\mathbf{Y}}_k$  refers to the predictions from the  $k^{\text{th}}$  method,