LinearModelSelectionRegularization

Ryan Finegan

11/1/2021

```r
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```r
library(MASS)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-1
```

```r
library(class)
set.seed(5)
setwd("/Users/ryanfinegan/Documents")                # my working directory
df<-read.csv("10yearforecasting.csv")                # file for 10 year prediction
dates <- as.POSIXct(df$Dates, format = "%m/%d/%Y")   # converting to get just the year
df$Dates<-format(dates, format="%Y")                 # getting the year in dates
df<-df[-1,]                                          # getting rid of the first row because zeros
df = subset(df, select = -c(thirtyderivative,thirty,movederivative,
                            move,dxyderivative,dxy) )
train <- df[df$Dates < 2015, ]             # splitting at the 2015 mark
test <- df[df$Dates > 2015, ]              # splitting at 2015
train = subset(train, select = -c(Dates) ) # getting rid of Dates
test = subset(test, select = -c(Dates) )   # getting rid of Dates
# tenderivativelag1 + thirtylag5 + dxylag1 + dxylag2 + movelag2 + movelag1 + tenlag2
df<-subset(df, select = -c(Dates) )        # getting rid of Dates
ten.year.lm <- lm(ten ~ ., data = train)   # mlr model on training data
summary(ten.year.lm)                       # regression statistics
```

```
##
## Call:
## lm(formula = ten ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56025 -0.08878 -0.00526  0.07813  0.56549
##
## Coefficients: (16 not defined because of singularities)
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           3.691e-03  4.018e-02   0.092   0.9268
## tenlag1               9.159e-01  7.889e-02  11.609   <2e-16 ***
## tenlag2               1.390e-01  1.052e-01   1.322   0.1864
## tenlag3              -6.029e-02  1.057e-01  -0.571   0.5684
## tenlag4              -1.211e-01  1.058e-01  -1.146   0.2522
## tenlag5               1.181e-01  7.927e-02   1.490   0.1365
## movelag1              7.747e-04  4.424e-04   1.751   0.0802 .
## movelag2             -1.235e-03  5.801e-04  -2.128   0.0335 *
## movelag3              3.912e-04  5.839e-04   0.670   0.5030
## movelag4             -1.197e-04  5.839e-04  -0.205   0.8376
## movelag5              8.511e-05  4.456e-04   0.191   0.8486
## thirtylag1            5.457e-03  8.708e-02   0.063   0.9500
## thirtylag2            1.932e-02  1.166e-01   0.166   0.8684
## thirtylag3            2.982e-02  1.171e-01   0.255   0.7990
## thirtylag4            1.588e-01  1.171e-01   1.356   0.1752
## thirtylag5           -2.072e-01  8.750e-02  -2.368   0.0180 *
## dxylag1               6.342e-03  3.596e-03   1.764   0.0780 .
## dxylag2              -8.511e-03  5.048e-03  -1.686   0.0920 .
## dxylag3               9.712e-04  5.049e-03   0.192   0.8475
## dxylag4               3.933e-03  5.038e-03   0.781   0.4351
## dxylag5              -2.627e-03  3.592e-03  -0.731   0.4646
## tenderivativelag1            NA         NA      NA       NA
## tenderivativelag2            NA         NA      NA       NA
## tenderivativelag3            NA         NA      NA       NA
## tenderivativelag4            NA         NA      NA       NA
## tenderivativelag5    -1.125e-01  7.888e-02  -1.427   0.1539
## movederivativelag1           NA         NA      NA       NA
## movederivativelag2           NA         NA      NA       NA
## movederivativelag3           NA         NA      NA       NA
## movederivativelag4           NA         NA      NA       NA
## movederivativelag5   -3.497e-04  4.364e-04  -0.801   0.4231
## thirtyderivativelag1         NA         NA      NA       NA
## thirtyderivativelag2         NA         NA      NA       NA
## thirtyderivativelag3         NA         NA      NA       NA
## thirtyderivativelag4         NA         NA      NA       NA
## thirtyderivativelag5  5.925e-02  8.725e-02   0.679   0.4972
## dxyderivativelag1            NA         NA      NA       NA
## dxyderivativelag2            NA         NA      NA       NA
## dxyderivativelag3            NA         NA      NA       NA
## dxyderivativelag4            NA         NA      NA       NA
## dxyderivativelag5     7.223e-03  3.599e-03   2.007   0.0450 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.128 on 1273 degrees of freedom
## Multiple R-squared:  0.9949, Adjusted R-squared:  0.9949
## F-statistic: 1.045e+04 on 24 and 1273 DF,  p-value: < 2.2e-16
```

```r
ols.prediction <- predict(ten.year.lm, test)      # getting the MSE for the OLS method
```

```
## Warning in predict.lm(ten.year.lm, test): prediction from a rank-deficient fit
## may be misleading
```

```r
(ols.mse <- mean((ols.prediction - test$ten)^2))  # the predictions on the test data split vs actual
```

```
## [1] 0.008799786
```

```r
library(dplyr)
```

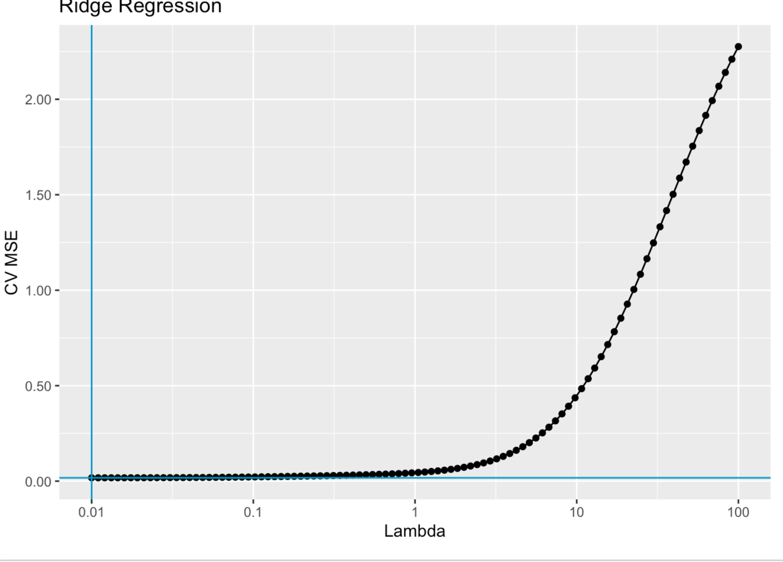
```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# tenderivativelag1 + thirtylag5 + dxylag1 + dxylag2 + movelag2 + movelag1 + tenlag2
train.matrix <- dummyVars(ten ~ ., data = train, fullRank = F) %>%
  predict(newdata = train) %>%
  as.matrix()
# tenderivativelag1 + thirtylag5 + dxylag1 + dxylag2 + movelag2 + movelag1 + tenlag2
test.matrix <- dummyVars(ten ~ ., data = test, fullRank = F) %>%
  predict(newdata = test) %>%
  as.matrix()
# ridge has an alpha of zero and the coefficients never shrink to zero
mod.ridge <- cv.glmnet(y = train$ten,
                        x = train.matrix,
                        alpha = 0,
                        lambda = 10^seq(2,-2, length = 100),
                        standardize = TRUE,
                        nfolds = 5)

data.frame(lambda = mod.ridge$lambda,
           cv_mse = mod.ridge$cvm) %>%
  ggplot(aes(x = lambda, y = cv_mse)) +
  geom_point() +
  geom_line() +
  geom_vline(xintercept = mod.ridge$lambda.min, col = "deepskyblue3") +
  geom_hline(yintercept = min(mod.ridge$cvm), col = "deepskyblue3") +
  scale_x_continuous(trans = 'log10', breaks = c(0.01, 0.1, 1, 10, 100), labels = c(0.01, 0.1, 1, 10, 100)) +
  scale_y_continuous(labels = scales::comma_format()) +
  theme(legend.position = "bottom") +
  labs(x = "Lambda",
       y = "CV MSE",
       col = "Coefficients:",           # Coefficients - Can't be zero because Ridge Regression
       title = "Ridge Regression")      # Lambda Selection with 5 CV
```



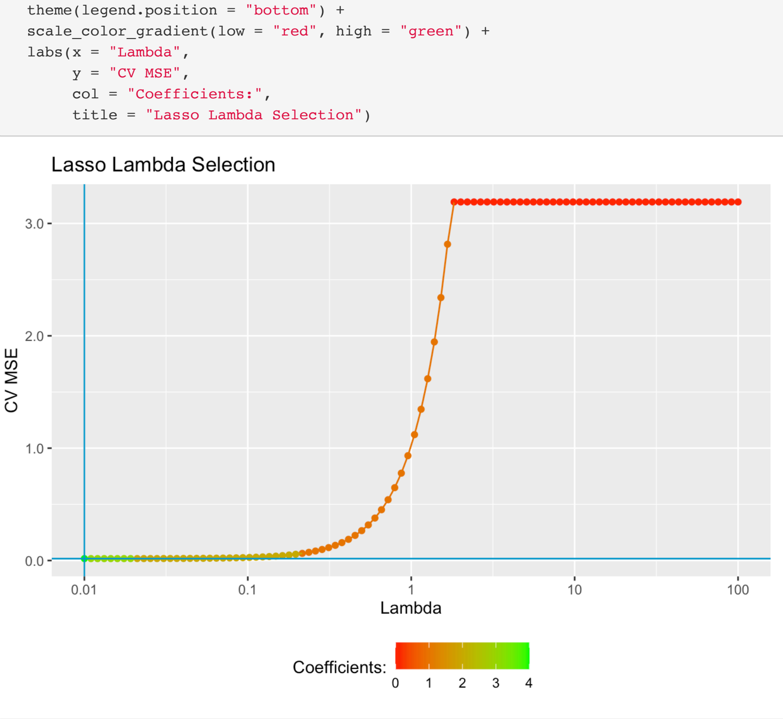

```r
### Using lambda selection model above
mod.ridge.best <- glmnet(y = train$ten,
                          x = train.matrix,
                          alpha = 0,
                          lambda = 10^seq(2,-2, length = 100))

ridge.prediction <- predict(mod.ridge.best, s = mod.ridge$lambda.min, newx = test.matrix)
(ridge.coef <- predict(mod.ridge.best, type = "coefficients", s = mod.ridge$lambda.min))
```

```
## 41 x 1 sparse Matrix of class "dgCMatrix"
##                                  1
## (Intercept)          -9.644693e-02
## tenlag1               2.305448e-01
## tenlag2               1.998801e-01
## tenlag3               1.807223e-01
## tenlag4               1.678480e-01
## tenlag5               1.592517e-01
## movelag1              4.055365e-05
## movelag2             -6.999793e-05
## movelag3             -4.785335e-05
## movelag4             -6.147263e-05
## movelag5             -5.550702e-05
## thirtylag1           -2.713635e-02
## thirtylag2            2.561110e-03
## thirtylag3            1.980840e-02
## thirtylag4            3.155623e-02
## thirtylag5            3.974564e-02
## dxylag1              -1.627403e-04
## dxylag2               1.251479e-05
## dxylag3               1.625469e-04
## dxylag4               2.601627e-04
## dxylag5               3.001955e-04
## tenderivativelag1     6.287387e-01
## tenderivativelag2     5.667343e-01
## tenderivativelag3     3.408576e-01
## tenderivativelag4     7.322600e-02
## tenderivativelag5    -8.111355e-02
## movederivativelag1    7.338204e-04
## movederivativelag2   -4.280930e-04
## movederivativelag3    1.105378e-06
## movederivativelag4   -7.012943e-05
## movederivativelag5   -2.986203e-04
## thirtyderivativelag1  9.171953e-02
## thirtyderivativelag2  1.083925e-01
## thirtyderivativelag3  1.032968e-01
## thirtyderivativelag4  2.092852e-01
## thirtyderivativelag5  2.543348e-02
## dxyderivativelag1     7.489751e-03
## dxyderivativelag2    -9.587432e-04
## dxyderivativelag3    -3.404088e-04
## dxyderivativelag4     3.187494e-03
## dxyderivativelag5     7.406089e-03
```

```r
(ridge.mse <- mean((ridge.prediction - test$ten)^2))
```

```
## [1] 0.008894181
```

```r
model.lasso <- cv.glmnet(y = train$ten, x = train.matrix, alpha = 1, lambda = 10^seq(2, -2,length = 100), standar
dize = TRUE, nfolds = 5, thresh = 1e-12)
data.frame(lambda = model.lasso$lambda,
           cv_mse = model.lasso$cvm,
           nonzero_coeff = model.lasso$nzero) %>%
  ggplot(aes(x = lambda, y = cv_mse, col = nonzero_coeff)) +
  geom_point() +
  geom_line() +
  geom_vline(xintercept = model.lasso$lambda.min, col = "deepskyblue3") +
  geom_hline(yintercept = min(model.lasso$cvm), col = "deepskyblue3") +
  scale_x_continuous(trans = 'log10', breaks = c(0.01, 0.1, 1, 10, 100), labels = c(0.01, 0.1, 1, 10, 100)) +
  scale_y_continuous(labels = scales::comma_format()) +
  theme(legend.position = "bottom") +
  scale_color_gradient(low = "red", high = "green") +
  labs(x = "Lambda",
       y = "CV MSE",
       col = "Coefficients:",
       title = "Lasso Lambda Selection")
```




```r
### Lambda Selection for Lasso Regression
```

```r
model.lasso.best <- glmnet(y = train$ten,x = train.matrix,alpha = 1, lambda = 10^seq(2,-5, length = 100))    # al
pha at one is a lasso regression
lasso.prediction <- predict(model.lasso.best, s = model.lasso$lambda.min, newx = test.matrix)
(lasso.mse <- mean((lasso.prediction - test$ten)^2))   # MSE for Lasso Regression
```

```
## [1] 0.008903584
```

```r
# getting the coefficients below
lasso.weights <- predict(model.lasso.best, type = "coefficients", s = model.lasso$lambda.min)
lasso.weights
```

```
## 41 x 1 sparse Matrix of class "dgCMatrix"
##                                 1
## (Intercept)          0.0357731056
## tenlag1              0.9800292598
## tenlag2              0.0110089311
## tenlag3              .
## tenlag4              .
## tenlag5              .
## movelag1             .
## movelag2             .
## movelag3             .
## movelag4             .
## movelag5             .
## thirtylag1           .
## thirtylag2           0.0006813501
## thirtylag3           .
## thirtylag4           .
## thirtylag5           .
## dxylag1              .
## dxylag2              .
## dxylag3              .
## dxylag4              .
## dxylag5              .
## tenderivativelag1    .
## tenderivativelag2    .
## tenderivativelag3    .
## tenderivativelag4    .
## tenderivativelag5    .
## movederivativelag1   .
## movederivativelag2   .
## movederivativelag3   .
## movederivativelag4   .
## movederivativelag5   .
## thirtyderivativelag1 .
## thirtyderivativelag2 .
## thirtyderivativelag3 .
## thirtyderivativelag4 0.0027976196
## thirtyderivativelag5 .
## dxyderivativelag1    .
## dxyderivativelag2    .
## dxyderivativelag3    .
## dxyderivativelag4    .
## dxyderivativelag5    .
```

```r
# principal components regression
library(pls)
```

```
##
## Attaching package: 'pls'
```

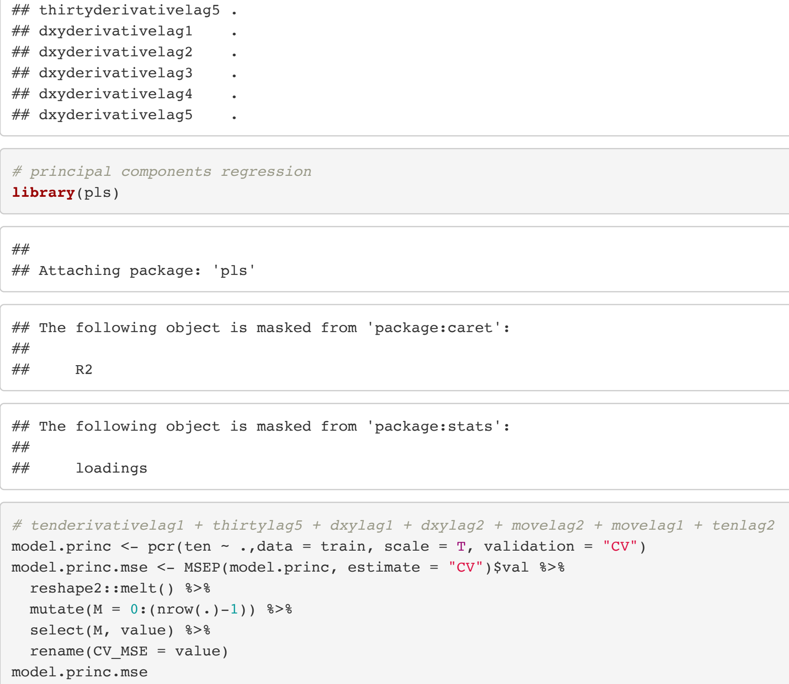
```
## The following object is masked from 'package:caret':
##
##     R2
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```

```r
# tenderivativelag1 + thirtylag5 + dxylag1 + dxylag2 + movelag2 + movelag1 + tenlag2
model.princ <- pcr(ten ~ .,data = train, scale = T, validation = "CV")
model.princ.mse <- MSEP(model.princ, estimate = "CV")$val %>%
  reshape2::melt() %>%
  mutate(M = 0:(nrow(.)-1)) %>%
  select(M, value) %>%
  rename(CV_MSE = value)
model.princ.mse
```

```
##     M     CV_MSE
## 1   0 3.18845019
## 2   1 0.25573558
## 3   2 0.16155876
## 4   3 0.05979789
## 5   4 0.05974783
## 6   5 0.05880521
## 7   6 0.05111413
## 8   7 0.04416015
## 9   8 0.03836433
## 10  9 0.03827377
## 11 10 0.03819785
## 12 11 0.03830370
## 13 12 0.03836243
## 14 13 0.03834401
## 15 14 0.03807190
## 16 15 0.03782049
## 17 16 0.03773281
## 18 17 0.03762043
## 19 18 0.03759524
## 20 19 0.03805554
## 21 20 0.03674128
## 22 21 0.02453695
## 23 22 0.02330149
## 24 23 0.02020085
## 25 24 0.01712792
## 26 25 0.01714076
## 27 26 0.01714102
## 28 27 0.01707050
## 29 28 0.01712820
## 30 29 0.01707121
## 31 30 0.01704061
## 32 31 0.01709226
## 33 32 0.01714851
## 34 33 0.01719928
## 35 34 0.01739254
## 36 35 0.01778277
## 37 36 0.01798999
## 38 37 0.01811921
## 39 38 0.01830499
## 40 39 0.01855466
## 41 40 0.01849195
```

```r
model.princ.mse %>%
  mutate(min_CV_MSE = as.numeric(min(CV_MSE) == CV_MSE)) %>%
  ggplot(aes(x = M, y = CV_MSE)) +
  geom_line(col = "grey55") +
  geom_point(size = 2, aes(col = factor(min_CV_MSE))) +
  scale_y_continuous(labels = scales::comma_format()) +
  scale_color_manual(values = c("deepskyblue3", "green")) +
  theme(legend.position = "none") +
  labs(x = "M",
       y = "Cross-Validation MSE",
       col = "Non-Zero Coefficients:",
       title = "PCR - M Selection (Using 10-Fold Cross-Validation)")
```




```r
### Cross Validation picked M = 7
```

```r
princ.pred <- predict(model.princ, test, ncomp = 24)
(princ.mse <- mean((princ.pred - test$ten)^2))
```

```
## [1] 0.008799786
```

```r
# tenderivativelag1 + thirtylag5 + dxylag1 + dxylag2 + movelag2 + movelag1 + tenlag2
mod.partial <- plsr(ten ~ .,data = train, scale = T, validation = "CV")
mod.partial.mse <- MSEP(mod.partial, estimate = "CV")$val %>%
  reshape2::melt() %>%
  mutate(M = 0:(nrow(.)-1)) %>%
  select(M, value) %>%
  rename(CV_MSE = value)
mod.partial.mse
```

```
##     M     CV_MSE
## 1   0 3.18845019
## 2   1 0.12529373
## 3   2 0.04100189
## 4   3 0.03565166
## 5   4 0.03430089
## 6   5 0.02789918
## 7   6 0.02019443
## 8   7 0.01745352
## 9   8 0.01728339
## 10  9 0.01724992
## 11 10 0.01722295
## 12 11 0.01716162
## 13 12 0.01709459
## 14 13 0.01709570
## 15 14 0.01709386
## 16 15 0.01710393
## 17 16 0.01710446
## 18 17 0.01710418
## 19 18 0.01710347
## 20 19 0.01710338
## 21 20 0.01710341
## 22 21 0.01710345
## 23 22 0.01710344
## 24 23 0.01710344
## 25 24 0.01710344
## 26 25 3.25630363
## 27 26 4.04443141
## 28 27 4.38790331
## 29 28 4.30788888
## 30 29 4.78510642
## 31 30 4.66942535
## 32 31 4.46184921
## 33 32 4.73325347
## 34 33 4.70962676
## 35 34 4.58374683
## 36 35 4.41253727
## 37 36 4.33308771
## 38 37 4.52621131
## 39 38 4.93537046
## 40 39 4.47430789
## 41 40 4.39642085
```

```r
mod.partial.mse %>%
  mutate(min_CV_MSE = as.numeric(min(CV_MSE) == CV_MSE)) %>%
  ggplot(aes(x = M, y = CV_MSE)) +
  geom_line(col = "grey55") +
  geom_point(size = 2, aes(col = factor(min_CV_MSE))) +
  scale_y_continuous(labels = scales::comma_format()) +
  scale_color_manual(values = c("deepskyblue3", "green")) +
  theme(legend.position = "none") +
  labs(x = "M",
       y = "Cross-Validation MSE",
       title = "PLS - M Selection (Using 10-Fold Cross-Validation)")
```


```r
partial.pred <- predict(mod.partial, test, ncomp = 14)
(partial.mse <- mean((partial.pred - test$ten)^2))
```

```
## [1] 0.0087965
```

```r
### Model Comparison
tss <- sum((test$ten - mean(test$ten))^2)    # total sum of squares
### data frame with the five models used before
data.frame(method = c("OLS", "Ridge", "Lasso", "PCR", "PLS"),
           test.mean.squared.errors = c(ols.mse, ridge.mse, lasso.mse, princ.mse, partial.mse),
           test.r2 = c(1 - sum((test$ten - ols.prediction)^2) / tss,
                       1 - sum((test$ten- ridge.prediction)^2) / tss,
                       1 - sum((test$ten - lasso.prediction)^2) / tss,
                       1 - sum((test$ten - princ.pred)^2) / tss,
                       1 - sum((test$ten - partial.pred)^2) / tss)) %>%
  arrange(test.mean.squared.errors)
```

```
##   method test.mean.squared.errors   test.r2
## 1    PLS              0.008796500 0.9823851
## 2    PCR              0.008799786 0.9823785
## 3    OLS              0.008799786 0.9823785
## 4  Ridge              0.008894181 0.9821895
## 5  Lasso              0.008903584 0.9821707
```