Feature Selection

Anna Yeaton

Fall 2018

Feature Selection

Lab Question 1. Can you tell the importance of a predictor from a decision tree? How?

Lab Question 2. Can you tell the importance of a predictor from linear regression? What about from a regularized linear regression?

Regression

Linear Regression

1. Split data into training and test set

```
train_size <- floor(0.75 * nrow(airquality))
set.seed(543)
train_pos <- sample(seq_len(nrow(airquality)), size = train_size)
train_regression <- airquality[train_pos,-c(1,2)]
test_regression <- airquality[-train_pos,-c(1,2)]

dim(train_regression)
```

```
## [1] 114   4
```

```
dim(test_regression)
```

```
## [1] 39  4
```

```
#help(train)
linear_regression <- train(Temp ~ Wind + Month + Day, data=train_regression, method = "lm")
```

```
summary(linear_regression)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.8187  -5.6255  -0.1811   5.4080  18.6658
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 73.91130    4.77303  15.485  < 2e-16 ***
## Wind        -1.03869    0.21113  -4.920 3.06e-06 ***
## Month        2.49179    0.53210   4.683 8.13e-06 ***
## Day         -0.19415    0.08497  -2.285   0.0242 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 7.864 on 110 degrees of freedom
## Multiple R-squared:  0.3736, Adjusted R-squared:  0.3565
## F-statistic: 21.87 on 3 and 110 DF,  p-value: 3.508e-11
```

We can use the Coefficients to compare feature importance if they are standarized so that variance = 1. This is done by subtracting the mean of the feature from each value, and then dividing by the standard deviation.

```
#help(train)
linear_regression <- train(scale(Temp) ~ scale(Wind) + scale(Month) + scale(Day), data=train_regression
summary(linear_regression)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.61367 -0.57386 -0.01847  0.55167  1.90411
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -3.052e-16  7.513e-02   0.000   1.0000
## `scale(Wind)`  -3.803e-01  7.731e-02  -4.920 3.06e-06 ***
## `scale(Month)`  3.607e-01  7.702e-02   4.683 8.13e-06 ***
## `scale(Day)`   -1.731e-01  7.576e-02  -2.285   0.0242 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8022 on 110 degrees of freedom
## Multiple R-squared:  0.3736, Adjusted R-squared:  0.3565
## F-statistic: 21.87 on 3 and 110 DF,  p-value: 3.508e-11
```

or use a package

```
library("QuantPsyc")
```

```
## Loading required package: boot
```

```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:lattice':
##
##     melanoma
```

```
##
## Attaching package: 'QuantPsyc'
```

```
## The following object is masked from 'package:Matrix':
##
##     norm
```

```
## The following object is masked from 'package:base':
##
##     norm
```

```
lm_fit <- lm(Temp ~ Wind + Month + Day, data=train_regression)
scaled_coef <- lm.beta(lm_fit)
scaled_coef
```

```
##       Wind      Month        Day
## -0.3803172  0.3606822 -0.1731034
```

Decision Trees and the importance() function

```
library(mlbench)
data(BreastCancer)

train_size <- floor(0.75 * nrow(BreastCancer))
set.seed(543)
train_pos <- sample(seq_len(nrow(BreastCancer)), size = train_size)

BreastCancer1 <- transform(BreastCancer, Id = as.numeric(Id), Cl.thickness = as.numeric(Cl.thickness),
                           Cell.size = as.numeric(Cell.size),
                           Cell.shape = as.numeric(Cell.shape), Marg.adhesion = as.numeric(Marg.adhesio
                           Epith.c.size = as.numeric(Epith.c.size),
                           Bare.nuclei = as.numeric(Bare.nuclei), Bl.cromatin = as.numeric(Bl.cromatin)
                           Normal.nucleoli = as.numeric(Normal.nucleoli),
                           Mitoses = as.numeric(Mitoses))

train_classification <- BreastCancer1[train_pos, ]
test_classification <- BreastCancer1[-train_pos, ]

dim(train_classification)
```

```
## [1] 524  11
```

```
dim(test_classification)
```

```
## [1] 175  11
```

```
set.seed(30495)
#do not specify mtry. The default for classification is sqrt(p) where p is the number of variables
RF_classification <- randomForest(Class ~ Normal.nucleoli + Epith.c.size + Cell.size , data=train_classi
importance(RF_classification)
```

```
##                   benign malignant MeanDecreaseAccuracy MeanDecreaseGini
## Normal.nucleoli 24.49552 13.354093             27.02063         62.29779
## Epith.c.size    15.55308  8.478846             17.64613         56.64914
## Cell.size       39.07064 35.548773             48.57648         83.66063
```

The first measure is computed from permuting OOB data: For each tree, the prediction error on the out-of-bag portion of the data is recorded (error rate for classification, MSE for regression). Then the same is done after permuting each predictor variable. The difference between the two are then averaged over all trees, and normalized by the standard deviation of the differences.

The second measure is the total decrease in node impurities from splitting on the variable, averaged over all trees. For classification, the node impurity is measured by the Gini index. For regression, it is measured by residual sum of squares.

Wrapper Methods vs Filter methods:

Wrapper methods: Evaulate model after adding or removing features to optimize the model performance using RMSE or AUC as metrics.

Recursive Feature Selection: Create model, measure variable importance, remove features of low importa

```
set.seed(134)
BreastCancer1[is.na(BreastCancer1)] <- 0
#help(rfe)
svmProfile <- rfe(BreastCancer1[,3:ncol(BreastCancer1)-1], BreastCancer1[,ncol(BreastCancer1)],
                  sizes = c(2, 5, 9),
                  rfeControl = rfeControl(functions = caretFuncs,
                                          number = 2),
                  ## pass options to train()
                  method = "svmRadial")
svmProfile
```

```
##
## Recursive feature selection
##
## Outer resampling method: Bootstrapped (2 reps)
##
## Resampling performance over subset size:
##
##  Variables Accuracy  Kappa AccuracySD  KappaSD Selected
##          2   0.9426 0.8710   0.001086 0.005411
##          5   0.9542 0.8985   0.009949 0.020269        *
##          9   0.9485 0.8873   0.012545 0.025816
##
## The top 5 variables (out of 5):
##    Cell.size, Cell.shape, Bl.cromatin, Bare.nuclei, Cl.thickness
```

```
svmProfile$variables
```

```
##       benign malignant   Overall             var Variables  Resample
## 1  0.9760908 0.9760908 0.9760908      Cell.shape         9 Resample1
## 2  0.9750839 0.9750839 0.9750839       Cell.size         9 Resample1
## 3  0.9496997 0.9496997 0.9496997     Bl.cromatin         9 Resample1
## 4  0.9362083 0.9362083 0.9362083     Bare.nuclei         9 Resample1
## 5  0.9236751 0.9236751 0.9236751 Normal.nucleoli         9 Resample1
## 6  0.9231938 0.9231938 0.9231938    Epith.c.size         9 Resample1
## 7  0.9179694 0.9179694 0.9179694    Cl.thickness         9 Resample1
## 8  0.8954602 0.8954602 0.8954602   Marg.adhesion         9 Resample1
## 9  0.7372946 0.7372946 0.7372946         Mitoses         9 Resample1
## 10 0.9760908 0.9760908 0.9760908      Cell.shape         5 Resample1
## 11 0.9750839 0.9750839 0.9750839       Cell.size         5 Resample1
## 12 0.9496997 0.9496997 0.9496997     Bl.cromatin         5 Resample1
## 13 0.9362083 0.9362083 0.9362083     Bare.nuclei         5 Resample1
## 14 0.9236751 0.9236751 0.9236751 Normal.nucleoli         5 Resample1
## 15 0.9760908 0.9760908 0.9760908      Cell.shape         2 Resample1
## 16 0.9750839 0.9750839 0.9750839       Cell.size         2 Resample1
## 17 0.9761703 0.9761703 0.9761703       Cell.size         9 Resample2
## 18 0.9743950 0.9743950 0.9743950      Cell.shape         9 Resample2
## 19 0.9434155 0.9434155 0.9434155     Bare.nuclei         9 Resample2
## 20 0.9348878 0.9348878 0.9348878     Bl.cromatin         9 Resample2
## 21 0.9198993 0.9198993 0.9198993    Cl.thickness         9 Resample2
## 22 0.9103913 0.9103913 0.9103913   Marg.adhesion         9 Resample2
## 23 0.9100026 0.9100026 0.9100026    Epith.c.size         9 Resample2
## 24 0.8766340 0.8766340 0.8766340 Normal.nucleoli         9 Resample2
## 25 0.7069290 0.7069290 0.7069290         Mitoses         9 Resample2
```

```
## 26 0.9761703 0.9761703 0.9761703       Cell.size         5 Resample2
## 27 0.9743950 0.9743950 0.9743950      Cell.shape         5 Resample2
## 28 0.9434155 0.9434155 0.9434155     Bare.nuclei         5 Resample2
## 29 0.9348878 0.9348878 0.9348878     Bl.cromatin         5 Resample2
## 30 0.9198993 0.9198993 0.9198993    Cl.thickness         5 Resample2
## 31 0.9761703 0.9761703 0.9761703       Cell.size         2 Resample2
## 32 0.9743950 0.9743950 0.9743950      Cell.shape         2 Resample2
```

Filter Methods: Remove features based on metrics such as variance, and correlation with outcome

From Max Khun: The caret function sbf (for selection by filter) can be used to cross-validate such feature selection schemes. Univariate examples are give by anovaScores for classification and gamScores for regression. anovaScores treats the outcome as the independent variable and the predictor as the outcome. In this way, the null hypothesis is that the mean predictor values are equal across the different classes. For regression, gamScores fits a smoothing spline in the predictor to the outcome using a generalized additive model and tests to see if there is any functional relationship between the two. In each function the p-value is used as the score.

```
set.seed(154)
help(sbf)
filteredNB <- sbf(BreastCancer1[,3:ncol(BreastCancer1)-1], BreastCancer1[,ncol(BreastCancer1)],
                  sbfControl = sbfControl(functions = caretSBF,
                                          verbose = FALSE,
                                          method = "repeatedcv",
                                          repeats = 1))

filteredNB
```

```
##
## Selection By Filter
##
## Outer resampling method: Cross-Validated (10 fold, repeated 1 times)
##
## Resampling performance:
##
##  Accuracy  Kappa AccuracySD KappaSD
##    0.9729 0.9407    0.02553  0.0557
##
## Using the training set, 9 variables were selected:
##    Cl.thickness, Cell.size, Cell.shape, Marg.adhesion, Epith.c.size...
##
## During resampling, the top 5 selected variables (out of a possible 9):
##    Bare.nuclei (100%), Bl.cromatin (100%), Cell.shape (100%), Cell.size (100%), Cl.thickness (100%)
##
## On average, 9 variables were selected (min = 9, max = 9)
```

Homework Feature Selection:

Choose a dataset, and run a general linear model, a tree based method, and a neural network, on the unfiltered feature set, the feature set after recursive feature elimination, and the feature set after filtering based on the characteristics of the data. Which models do you think would be most impacted by the feature selection? Did your models do better after feature selection?