Machine Learning 2019: Feature Selection

Sonali Narang

October 24, 2019

Feature Selection

In machine learning, feature selection is the process of choosing variables that are useful in predicting the response variable. Selecting the right features in your data can mean the difference between mediocre performance with long training times and great performance with short training times that are less computationally intensive.

Often, data can contain attributes that are highly correlated with each other or not useful in helping predict our response variable. Many methods perform better if such variables are removed. Feature selection is usually imporant to implement during the data pre-processing steps of machine learning.

The Breast Cancer Dataset

699 Observations, 11 variables Predictor Variable: Class- benign or malignant

```
data(BreastCancer)
head(BreastCancer)
```

```
##        Id Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size
## 1 1000025            5         1          1             1            2
## 2 1002945            5         4          4             5            7
## 3 1015425            3         1          1             1            2
## 4 1016277            6         8          8             1            3
## 5 1017023            4         1          1             3            2
## 6 1017122            8        10         10             8            7
##   Bare.nuclei Bl.cromatin Normal.nucleoli Mitoses     Class
## 1           1           3               1       1    benign
## 2          10           3               2       1    benign
## 3           2           3               1       1    benign
## 4           4           3               7       1    benign
## 5           1           3               1       1    benign
## 6          10           9               7       1 malignant
```

```
dim(BreastCancer)
```

```
## [1] 699  11
```

```
summary(BreastCancer$Class)
```

```
##    benign malignant
##       458       241
```

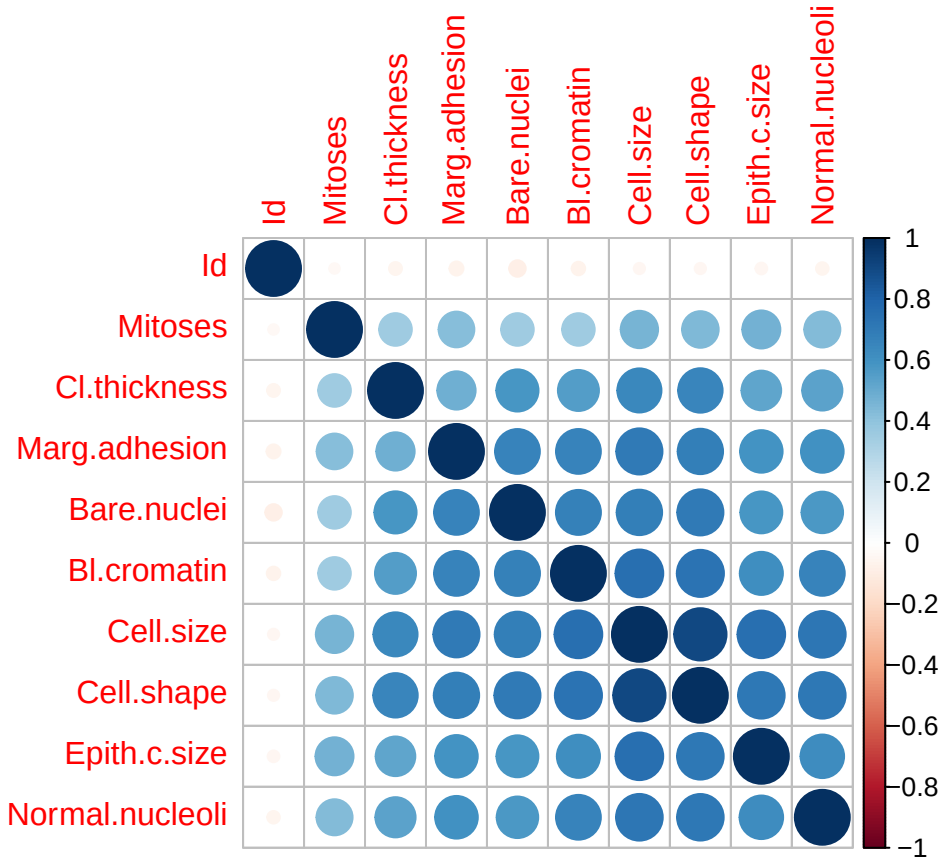
Feature Selection Using Correlation

Remove attributes that are highly correlated.

```
BreastCancer_num = 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.adhesion),
                         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))

BreastCancer_num[is.na(BreastCancer_num)] = 0

#calculate correlation matrix using pearson correlation (others include spearman and kendall)
correlation_matrix = cor(BreastCancer_num[,1:10])

#visualize correlation matrix
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
corrplot(correlation_matrix, order = "hclust")
```



```r
#apply correlation filter of 0.7
highly_correlated <- colnames(BreastCancer[, -1])[findCorrelation(correlation_matrix, cutoff = 0.7, verb
```

```
## Compare row 3  and column  4 with corr  0.907
##   Means:  0.631 vs 0.477 so flagging column 3
## Compare row 4  and column  8 with corr  0.736
##   Means:  0.588 vs 0.447 so flagging column 4
## All correlations <= 0.7
```

```r
#which features are highly correlated and can be removed
highly_correlated
```

```
## [1] "Cell.shape"    "Marg.adhesion"
```

Feature Selection Using Wrapper Methods: Recursive Feature Elimination (RFE)

This function implements backwards selection of predictors based on predictor importance ranking. The predictors are ranked and the less important ones are sequentially eliminated prior to modeling. The goal is to find a subset of predictors that can be used to produce an accurate model.

```r
data(BreastCancer)
BreastCancer_num = 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.adhesion),
                         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))

BreastCancer_num[is.na(BreastCancer_num)] = 0

#define the control
control = rfeControl(functions = caretFuncs, number = 2)

# run the RFE algorithm
results = rfe(BreastCancer_num[,1:10], BreastCancer_num[,11], sizes = c(2,5,9), rfeControl = control, me

results
```

```
##
## Recursive feature selection
##
## Outer resampling method: Bootstrapped (2 reps)
##
## Resampling performance over subset size:
##
##  Variables Accuracy  Kappa AccuracySD KappaSD Selected
```

```
##          2   0.9367 0.8636   0.004005 0.01367
##          5   0.9385 0.8687   0.009952 0.01509
##          9   0.9523 0.8976   0.012403 0.02215        *
##         10   0.9365 0.8644   0.012803 0.02213
##
## The top 5 variables (out of 9):
##    Cell.shape, Cell.size, Bare.nuclei, Bl.cromatin, Epith.c.size
```

```
results$variables
```

```
##       benign malignant   Overall             var Variables  Resample
## 1  0.9764468 0.9764468 0.9764468       Cell.size        10 Resample1
## 2  0.9710240 0.9710240 0.9710240      Cell.shape        10 Resample1
## 3  0.9468238 0.9468238 0.9468238     Bl.cromatin        10 Resample1
## 4  0.9432893 0.9432893 0.9432893     Bare.nuclei        10 Resample1
## 5  0.9207441 0.9207441 0.9207441    Epith.c.size        10 Resample1
## 6  0.9141196 0.9141196 0.9141196    Cl.thickness        10 Resample1
## 7  0.8889730 0.8889730 0.8889730 Normal.nucleoli        10 Resample1
## 8  0.8653318 0.8653318 0.8653318   Marg.adhesion        10 Resample1
## 9  0.7145888 0.7145888 0.7145888         Mitoses        10 Resample1
## 10 0.5691938 0.5691938 0.5691938              Id        10 Resample1
## 11 0.9764468 0.9764468 0.9764468       Cell.size         9 Resample1
## 12 0.9710240 0.9710240 0.9710240      Cell.shape         9 Resample1
## 13 0.9468238 0.9468238 0.9468238     Bl.cromatin         9 Resample1
## 14 0.9432893 0.9432893 0.9432893     Bare.nuclei         9 Resample1
## 15 0.9207441 0.9207441 0.9207441    Epith.c.size         9 Resample1
## 16 0.9141196 0.9141196 0.9141196    Cl.thickness         9 Resample1
## 17 0.8889730 0.8889730 0.8889730 Normal.nucleoli         9 Resample1
## 18 0.8653318 0.8653318 0.8653318   Marg.adhesion         9 Resample1
## 19 0.7145888 0.7145888 0.7145888         Mitoses         9 Resample1
## 20 0.9764468 0.9764468 0.9764468       Cell.size         5 Resample1
## 21 0.9710240 0.9710240 0.9710240      Cell.shape         5 Resample1
## 22 0.9468238 0.9468238 0.9468238     Bl.cromatin         5 Resample1
## 23 0.9432893 0.9432893 0.9432893     Bare.nuclei         5 Resample1
## 24 0.9207441 0.9207441 0.9207441    Epith.c.size         5 Resample1
## 25 0.9764468 0.9764468 0.9764468       Cell.size         2 Resample1
## 26 0.9710240 0.9710240 0.9710240      Cell.shape         2 Resample1
## 27 0.9703761 0.9703761 0.9703761      Cell.shape        10 Resample2
## 28 0.9623496 0.9623496 0.9623496       Cell.size        10 Resample2
## 29 0.9576802 0.9576802 0.9576802     Bare.nuclei        10 Resample2
## 30 0.9458362 0.9458362 0.9458362     Bl.cromatin        10 Resample2
## 31 0.9403424 0.9403424 0.9403424    Epith.c.size        10 Resample2
## 32 0.9109810 0.9109810 0.9109810   Marg.adhesion        10 Resample2
## 33 0.9032170 0.9032170 0.9032170    Cl.thickness        10 Resample2
## 34 0.8816381 0.8816381 0.8816381 Normal.nucleoli        10 Resample2
## 35 0.7140397 0.7140397 0.7140397         Mitoses        10 Resample2
## 36 0.5469294 0.5469294 0.5469294              Id        10 Resample2
## 37 0.9703761 0.9703761 0.9703761      Cell.shape         9 Resample2
## 38 0.9623496 0.9623496 0.9623496       Cell.size         9 Resample2
## 39 0.9576802 0.9576802 0.9576802     Bare.nuclei         9 Resample2
## 40 0.9458362 0.9458362 0.9458362     Bl.cromatin         9 Resample2
## 41 0.9403424 0.9403424 0.9403424    Epith.c.size         9 Resample2
## 42 0.9109810 0.9109810 0.9109810   Marg.adhesion         9 Resample2
## 43 0.9032170 0.9032170 0.9032170    Cl.thickness         9 Resample2
```

```
## 44 0.8816381 0.8816381 0.8816381 Normal.nucleoli         9 Resample2
## 45 0.7140397 0.7140397 0.7140397         Mitoses         9 Resample2
## 46 0.9703761 0.9703761 0.9703761      Cell.shape         5 Resample2
## 47 0.9623496 0.9623496 0.9623496       Cell.size         5 Resample2
## 48 0.9576802 0.9576802 0.9576802     Bare.nuclei         5 Resample2
## 49 0.9458362 0.9458362 0.9458362     Bl.cromatin         5 Resample2
## 50 0.9403424 0.9403424 0.9403424    Epith.c.size         5 Resample2
## 51 0.9703761 0.9703761 0.9703761      Cell.shape         2 Resample2
## 52 0.9623496 0.9623496 0.9623496       Cell.size         2 Resample2
```

Feature Selection Using Embedded Methods: Lasso

Least Absolute Shrinkage and Selection Operator (LASSO) regression

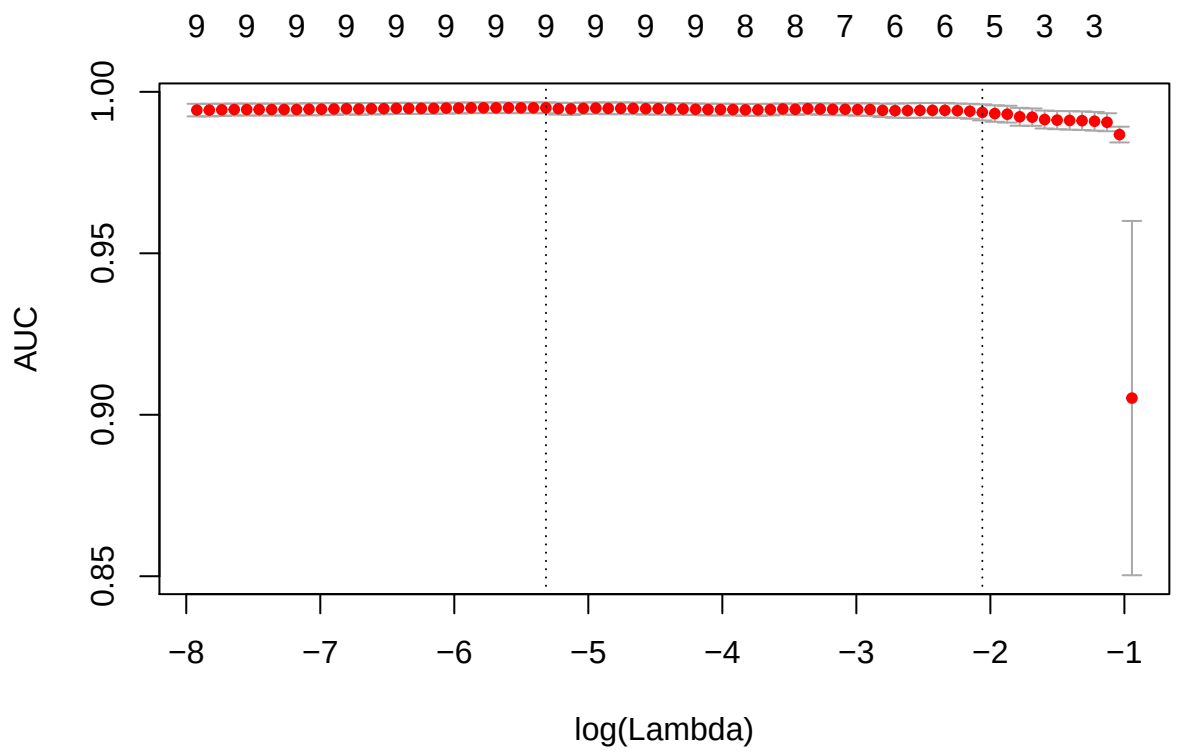
```
library(glmnet)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loading required package: foreach

##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##     accumulate, when

## Loaded glmnet 2.0-18
```

```
set.seed(24)

#convert data
x = x <- as.matrix(BreastCancer_num[,1:10])
y = as.double(as.matrix(ifelse(BreastCancer_num[,11]=='benign', 0, 1)))

#fit Lasso model
cv.lasso <- cv.glmnet(x, y, family='binomial', alpha=1, parallel=TRUE, standardize=TRUE, type.measure='a
```

```
## Warning: executing %dopar% sequentially: no parallel backend registered
```

```
plot(cv.lasso)
```




```
cat('Min Lambda: ', cv.lasso$lambda.min, '\n 1Sd Lambda: ', cv.lasso$lambda.1se)
```

```
## Min Lambda:  0.0049116
##  1Sd Lambda:  0.1274572
```

```
df_coef <- round(as.matrix(coef(cv.lasso, s=cv.lasso$lambda.min)), 2)

# See all contributing variables
df_coef[df_coef[, 1] != 0, ]
```

```
##     (Intercept)    Cl.thickness       Cell.size      Cell.shape
##           -7.97            0.44            0.06            0.28
##   Marg.adhesion    Epith.c.size     Bare.nuclei     Bl.cromatin
##            0.16            0.05            0.37            0.33
## Normal.nucleoli         Mitoses
##            0.14            0.23
```

Feature Selection Using Embedded Methods: RandomForest

Random Forest Importance function and caret package's varImp functions perform similarly.

```
#data
data(BreastCancer)
```

```r
train_size <- floor(0.75 * nrow(BreastCancer))
set.seed(24)
train_pos <- sample(seq_len(nrow(BreastCancer)), size = train_size)

#convert to numeric
BreastCancer_num = 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.adhesion),
                         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))

BreastCancer_num[is.na(BreastCancer_num)] = 0

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

#fit a model
rfmodel = randomForest(Class ~ Id + Cl.thickness + Cell.size + Cell.shape + Marg.adhesion + Epith.c.siz

#rank features based on importance
importance(rfmodel)
```

```
##                    benign malignant MeanDecreaseAccuracy MeanDecreaseGini
## Id              -0.6895607  6.306423             5.356937         4.753537
## Cl.thickness    20.1614358 21.864276            24.672793        15.817233
## Cell.size       13.2922349 16.005349            20.854493        51.850109
## Cell.shape       9.9205845 15.663444            18.547503        52.135873
## Marg.adhesion    6.9732478  8.757817            11.298839         7.495039
## Epith.c.size     8.2669770  3.558679             8.988390        14.948179
## Bare.nuclei     18.7963652 27.340734            28.123604        42.322414
## Bl.cromatin      8.5784618 14.394368            16.261622        28.772624
## Normal.nucleoli 11.9915409  9.888276            14.484327        19.497268
## Mitoses          6.3442746  2.271364             6.768548         1.615123
```

Homework

1. Compare the most important features from at least 2 different classes of feature selection methods covered in this tutorial with any reasonable machine learning dataset from mlbench. Do these feature selection methods provide similar results?

2. Attempt a feature selection method not covered in this tutorial (backward elimination, forward propogation, etc.)