Clasification with Cross Validation - use of CARET

We show the use of 5 resampling methods on the `iris` data, with the most simple classifier model, *naive Bayes*. The methods used are:

1. Train-test split.
2. Bootstrap.
3. k-fold CV.
4. k-fold CV with repeats.
5. LOOCV.

Data

Load the data and split into train-test with a ratio 80-20.

```
# load the libraries
library(caret)
```

```
## Loading required package: ggplot2
```

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

```
library(klaR) # required for 'naive Bqyes'
```

```
## Loading required package: MASS
```

```
# load  `iris` data
data(iris)
# define the train/test split as 80%/20%
split=0.80
trainIndex <- createDataPartition(iris$Species, p=split, list=FALSE)
data_train <- iris[ trainIndex,]
data_test  <- iris[-trainIndex,]
# fit a "naive bayes" model
model <- NaiveBayes(Species~., data=data_train)
# predictions on the test set
x_test <- data_test[,1:4]
y_test <- data_test[,5]
predictions <- predict(model, x_test)
# print the results
confusionMatrix(predictions$class, y_test)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         10          0         0
##   versicolor      0          9         1
##   virginica       0          1         9
##
## Overall Statistics
##
```

```
##                Accuracy : 0.9333
##                  95% CI : (0.7793, 0.9918)
##     No Information Rate : 0.3333
##     P-Value [Acc > NIR] : 8.747e-12
##
##                   Kappa : 0.9
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                 1.0000            0.9000           0.9000
## Specificity                 1.0000            0.9500           0.9500
## Pos Pred Value              1.0000            0.9000           0.9000
## Neg Pred Value              1.0000            0.9500           0.9500
## Prevalence                  0.3333            0.3333           0.3333
## Detection Rate              0.3333            0.3000           0.3000
## Detection Prevalence        0.3333            0.3333           0.3333
## Balanced Accuracy           1.0000            0.9250           0.9250
```

Resampling by Bootstrap

Sampling with replacement.

```
# define control parameters for the training
train_control <- trainControl(method="boot", number=100)
# fit the model
model <- train(Species~.,
               data=iris,
               trControl=train_control,
               method="nb")
# print the results
print(model)
```

```
## Naive Bayes
##
## 150 samples
##   4 predictor
##   3 classes: 'setosa', 'versicolor', 'virginica'
##
## No pre-processing
## Resampling: Bootstrapped (100 reps)
## Summary of sample sizes: 150, 150, 150, 150, 150, 150, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.9500463  0.9244324
##    TRUE      0.9513457  0.9263860
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = TRUE and adjust
```

```
##  = 1.
```

k-fold Cross-Validation

We use the deafult, 10-fold CV.

```
# define control parameters for the training
train_control <- trainControl(method="cv", number=10)
# fix tuning parameters of the algorithm
grid <- expand.grid(.fL=c(0), .usekernel=c(FALSE),.adjust=FALSE)
# fit the model
model <- train(Species~.,
               data=iris,
               trControl=train_control,
               method="nb",
               tuneGrid=grid)
# print the results
print(model)
```

```
## Naive Bayes
##
## 150 samples
##   4 predictor
##   3 classes: 'setosa', 'versicolor', 'virginica'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 135, 135, 135, 135, 135, 135, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9533333  0.93
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'usekernel' was held constant at a value of FALSE
## Tuning
##  parameter 'adjust' was held constant at a value of FALSE
```

Repeated k-fold Cross-Validation

10-fold with 3 repeats.

```
# define control parameters for the training
train_control <- trainControl(method="repeatedcv", number=10, repeats=3)
# fit the model
model <- train(Species~.,
               data=iris,
               trControl=train_control,
               method="nb")
# print the results
print(model)
```

```
## Naive Bayes
##
## 150 samples
```

```
##   4 predictor
##   3 classes: 'setosa', 'versicolor', 'virginica'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 135, 135, 135, 135, 135, 135, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.9533333  0.9300000
##    TRUE      0.9577778  0.9366667
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = TRUE and adjust
##  = 1.
```

LOOCV

```
# define control parameters for the training
train_control <- trainControl(method="LOOCV")
# fit the model
model <- train(Species~.,
               data=iris,
               trControl=train_control,
               method="nb")
# print the results
print(model)
```

```
## Naive Bayes
##
## 150 samples
##   4 predictor
##   3 classes: 'setosa', 'versicolor', 'virginica'
##
## No pre-processing
## Resampling: Leave-One-Out Cross-Validation
## Summary of sample sizes: 149, 149, 149, 149, 149, 149, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.9533333  0.93
##    TRUE      0.9600000  0.94
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = TRUE and adjust
##  = 1.
```