Training Options Demo - SPAM Example

Jiachang (Ernest) Xu

6/22/2017

```
## require caret package for machine learning algorithms
require(caret)
```

```
## Loading required package: caret
```

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

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

```
## require kernlab for spam data
require(kernlab)
```

```
## Loading required package: kernlab
```

```
##
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
## data loading
data(spam)
```

```
inTrain <- createDataPartition(y = spam$type, p = 0.75, list = FALSE)
training <- spam[inTrain, ]
testing <- spam[-inTrain, ]
```

```
## generalized linear model
suppressWarnings(model.fit <- train(type ~ ., data = training, method = "glm"))
```

Training Options

trainControl

```
args(trainControl)
```

```
## function (method = "boot", number = ifelse(grepl("cv", method),
##     10, 25), repeats = ifelse(grepl("cv", method), 1, number),
##     p = 0.75, search = "grid", initialWindow = NULL, horizon = 1,
##     fixedWindow = TRUE, skip = 0, verboseIter = FALSE, returnData = TRUE,
##     returnResamp = "final", savePredictions = FALSE, classProbs = FALSE,
##     summaryFunction = defaultSummary, selectionFunction = "best",
##     preProcOptions = list(thresh = 0.95, ICAcomp = 3, k = 5,
##         freqCut = 95/5, uniqueCut = 10, cutoff = 0.9), sampling = NULL,
##     index = NULL, indexOut = NULL, indexFinal = NULL, timingSamps = 0,
##     predictionBounds = rep(FALSE, 2), seeds = NA, adaptive = list(min = 5,
##         alpha = 0.05, method = "gls", complete = TRUE), trim = FALSE,
##     allowParallel = TRUE)
## NULL
```

trainControl resampling

- method
  - "boot" = bootstrapping
  - "boot632" = bootstrapping with adjustment
  - "cv" = cross validation
  - "repeatedcv" = repeated cross validation
  - "LOOCV" = leave one out cross validation
- number
  - for bootstrapping or cross validation
  - number of subsamples to take
- repeats
  - number of times to repeat subsampling
  - if big this can slow things down

Setting the Seed

- It is often useful to set an overall seed.
- You can also set a seed for each resample.
- Seeding each sample is useful for parallel fits.

set.seed() example

```
## set.seed() example
set.seed(1235)
suppressWarnings(model.fit2 <- train(type ~ ., data = training, method = "glm"))
model.fit2
```

```
## Generalized Linear Model
##
## 3451 samples
##   57 predictor
##    2 classes: 'nonspam', 'spam'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 3451, 3451, 3451, 3451, 3451, 3451, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9138136  0.8180775
```

```
## set.seed() example
set.seed(1235)
suppressWarnings(model.fit3 <- train(type ~ ., data = training, method = "glm"))
model.fit3
```

```
## Generalized Linear Model
##
## 3451 samples
##   57 predictor
##    2 classes: 'nonspam', 'spam'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 3451, 3451, 3451, 3451, 3451, 3451, ...
```

```
## Resampling results:
##
##   Accuracy   Kappa
##   0.9138136  0.8180775
```