Training options

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

SPAM Example

```
library(caret); library(kernlab); data(spam)
inTrain <- createDataPartition(y=spam$type,
                               p=0.75, list=FALSE)
training <- spam[inTrain,]
testing <- spam[-inTrain,]
modelFit <- train(type ~.,data=training, method="glm")
```

Genau gleich wie in den vorherigen zwei Lectures.

Train options

```
args(train.default)
```

um gewissen Datensaetze zu gewichten

```
function (x, y, method = "rf", preProcess = NULL, ..., weights = NULL,
    metric = ifelse(is.factor(y), "Accuracy", "RMSE"), maximize = ifelse(metric ==
        "RMSE", FALSE, TRUE), trControl = trainControl(), tuneGrid = NULL,
    tuneLength = 3)
NULL
```

…siehe spaeter…

Metric options

Continous outcomes:

- *RMSE* = Root mean squared error
- *RSquared* = R^2 from regression models

Categorical outcomes:

- *Accuracy* = Fraction correct `default for categorial variables`
- *Kappa* = A measure of concordance

trainControl

```
args(trainControl)
```

```
                                  number of times to repeat the bootstrap
function (method = "boot", number = ifelse(method %in% c("cv",
    "repeatedcv"), 10, 25), repeats = ifelse(method %in% c("cv",
    "repeatedcv"), 1, number), p = 0.75, initialWindow = NULL,
    horizon = 1, fixedWindow = TRUE, verboseIter = FALSE, returnData = TRUE,
    returnResamp = "final", savePredictions = FALSE, classProbs = FALSE,
    summaryFunction = defaultSummary, selectionFunction = "best",
    custom = NULL, preProcOptions = list(thresh = 0.95, ICAcomp = 3,
        k = 5), index = NULL, indexOut = NULL, timingSamps = 0,
    predictionBounds = rep(FALSE, 2), seeds = NA, allowParallel = TRUE)
NULL
```

p: size of training set

…see next slide for some explanations

trainControl resampling

- *method*
  - *boot* = bootstrapping
  - *boot632* = bootstrapping with adjustment `for the fact that multiple samples are repeatedly resampled.`
  - *cv* = cross validation
  - *repeatedcv* = repeated cross validation
  - *LOOCV* = leave one out cross validation
- *number*
  - For boot/cross validation
  - Number of subsamples to take
- *repeats*
  - Number of times to repeate subsampling
  - If big this can *slow things down*

Setting the seed

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

seed example

```
set.seed(1235)
modelFit2 <- train(type ~.,data=training, method="glm")
modelFit2
```

```
Generalized Linear Model

3451 samples
  57 predictors
   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  Accuracy SD  Kappa SD
  0.9       0.8    0.007        0.01
```

seed example

modelFit3 has exactly the same results because we used the same seed!

```
set.seed(1235)
modelFit3 <- train(type ~.,data=training, method="glm")
modelFit3
```

```
Generalized Linear Model

3451 samples
  57 predictors
   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  Accuracy SD  Kappa SD
  0.9       0.8    0.007        0.01
```

Further resources

- Caret tutorial
- Model training and tuning