Practical Machine Learning in R
Regression

Lars Kotthoff
larsko@uwyo.edu

1with slides from Bernd Bischl and Michel Lang
2slides available at http://www.cs.uwyo.edu/~larsko/ml-fac
Goal: Predict a continuous quantity
Linear Models

- assumes linear relationship between features and variable of interest
- prediction is linear combination of feature values with coefficients (similar to logistic regression)
- determine coefficients by minimizing loss (e.g. sum of squared error) wrt training data
Linear Models
Regression Splines

- Multivariate Adaptive Regression Splines (MARS)
- non-parametric technique – no assumptions about the underlying relationship
- prediction is weighted sum of basis functions
- construction similar to trees – repeatedly add basis functions to improve performance (recursive partitioning), then remove some to improve generality (pruning)
Regression Splines
"boost" a weak learner by training set of models that fix each other’s errors (ensemble)

after adding each model, reweigh training data such that examples with higher error get more weight

aggregate by combining weighted predictions from each model

idea similar to random forests

technique not specific to regression
Boosting

blackboost: mstop=1
Train: mse=73.9; CV: mse.test.mean=74.7
blackboost: mstop=2
Train: mse=65.3; CV: mse.test.mean=66.6
blackboost: mstop=3
Train: mse=58.2; CV: mse.test.mean=59.7
blackboost: mstop=5
Train: mse=47.5; CV: mse.test.mean=49.7
blackboost: mstop=10
Train: mse=33.5; CV: mse.test.mean=36.5
blackboost: mstop=100
Train: mse=22.4; CV: mse.test.mean=29.3
blackboost: mstop=1e+03
Train: mse=14.8; CV: mse.test.mean=34.7
Support Vector Machines and Random Forests

▷ regression versions exist
▷ SVMs: minimize error of support vectors
▷ Random Forests: predict constant quantity (or simple linear model) at leaves
Exercises

http://www.cs.uwyo.edu/~larsko/ml-fac/02-regression-exercises.Rmd