Practical Machine Learning in R
Classification

Lars Kotthoff\textsuperscript{1,2}
laruko@uwyo.edu

\textsuperscript{1}with slides from Bernd Bischl and Michel Lang
\textsuperscript{2}slides available at http://www.cs.uwyo.edu/~laruko/ml-fac
Classification

Goal: Predict a class (discrete quantity), or membership probabilities
Logistic Regression

- estimates probability of binary response
- i.e. predict whether example belongs to one class or another
- logistic function has output between 0 and 1, can be interpreted as probability
- essentially determines coefficients (importance) of each feature
Logistic Regression

logreg: model=FALSE
Train: acc=0.94; CV: acc.test.mean=0.94

\[ f(x) = -40.2447695 - 1.7247776x_1 - 5.0182373x_2 + 8.0163583x_3 + 15.500357x_4 \]
Binary vs. Multi-class

▷ some learners can handle only two classes (e.g. logistic regression)
▷ can distinguish between more classes with more models
▷ e.g. one-vs-all approach:
  ▷ for each class, learn to predict score of how likely data point is in class
  ▷ aggregate scores over all models (classes)
Linear Discriminant Analysis

▷ finds linear combination of features that separate classes
▷ maps feature space into lower-dimensional space (dimensions are linear combinations of features)
▷ determines centroid for each class in mapped space
▷ classifies by assigning data point to centroid
Linear Discriminant Analysis

3-class feature data

worst 1D subspace

best 1D subspace

Support Vector Machines
Support Vector Machines

- data points with minimal margin are the **support vectors (SV)**
- finding a hyperplane to maximize the margin is a straightforward optimization problem
Support Vector Machines

Non-separable data

maximal margin

minimal margin violations
Kernels allow to extend SVMs to non-linear separation and non-vectorial data
- maps the original feature space into higher-dimensional space
- classes become linearly separable in this higher-dimensional space
- input and output spaces can be infinite-dimensional
Support Vector Machines – Kernel Trick

The diagram shows the original data points in the $x$-$y$ plane and the transformed data points in the $x^2$-$y^2$-$xy$ space after applying the kernel trick. The transformation is represented by the function $\phi(x, y)$, which maps the original points into a higher-dimensional space where they can be more easily separated by a linear classifier.
Support Vector Machines – Kernel Trick

Examples:
▷ linear kernel
▷ polynomial kernel
▷ Gaussian kernel
Classification Trees

▷ divide the feature space into rectangles and fit simple models (i.e. constant) in each
▷ prediction is class distribution / most frequent label in subspace
▷ rectangles can be further subdivided
Tree Induction Algorithms

▷ Greedy: Pick the best feature and its best split point in each iteration
▷ Binary splits vs. multi-way splits
▷ Criteria for the selection of a variable and its split point(s) (e.g. entropy)
▷ Stopping criteria (e.g. minimum number of data points)
▷ Handling missing values
▷ Pruning
Tree Building Example
Tree Building Example
Tree Building Example
Tree Building Example
Classification Forests

Random Forests:
▷ Example of an ensemble method: instead of a single model, use several and combine the results
▷ train trees on different subsamples (with replacement) of the data/features
▷ aggregate predictions across trees by counting “votes” for each class
▷ general method for improving unstable learners
▷ usually done without pruning to increase variance
Classification Forests
Exercises

http://www.cs.uwyo.edu/~larsko/ml-fac/01-classification-exercises.Rmd