3 Random Forest Classification

Aim

- Detailed land cover classification based on AVIRIS data using the Random Forest algorithm.
- Assessment of the importance of different spectral bands for the classification.

Data

The data set for this training is an AVIRIS image covering the area of the Kennedy Space Center (KSC, the NASA rocket launch site) near Titusville, FL, USA. The data were acquired on March 23rd, 1996. All water absorption and otherwise very noisy bands were removed from the original 224 bands, leaving 176 bands for our analyses. The airborne sensor was operated at an elevation of approximately 20 km above ground, resulting in a spatial resolution of 18 m on the ground. The data and all information are provided trough the Center for Space Research at the University of Texas in Austin, TX, USA (http://www.csr.utexas.edu/hyperspectral/data/KSC/) and the Universidad del Pais Vasco, Spain (http://www.ehu.es/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes). The data set is not georeferenced. The pixel values represent reflectance values in %*100.

In addition, ground truth data were sampled for the study area. These data contain 13 land use and land cover classes (see table 3.1). The delineation of these classes is considered a difficult task because they show some overlap in the spectral feature space. Further information on the data is provided in Ham et al. (2005). Please pay attention: The groundtruth data for calibration (i.e. training) and validation (i.e. testing) were sampled by simply delineating polygons representing the 13 classes, which results in a clustered distribution of the corresponding points. This is bad practice because the sample is not statistically independent. A random or stratified random sample that covers multiple occurrences of each class would have been a better approach from my point of view. Still, this is what we have.

Q3.1 (2 pts) Consider the possible effects of the clustered sampling design on model calibration and validation. What are possible effects on the model transferability and general applicability, if the training data are sampled in a clustered distribution? Will the validation results be rather over- or underoptimistic when the validation points are sampled from the same polygon? Provide a brief assessment in 2-3 sentences for each question.

Table 3.1: The 13 land cover classes taken from Ham et al. (2005).

Class ID	Name	Description
1	Scrub	Shrub dominated vegetation
2	Willow swamp	Swamp dominated by willow shrubs
3	Cabbage palm hammock	Dry and elevated areas with palm trees
4	Cabbage palm / oak hammock	Dry and elevated areas with palms and oaks
5	Slash pine	Stands of slash pine trees
6	Oak / broadleaf hammock	Dry and elevated areas with oak and other broadleaf trees
7	Hardwood swamp	Swamp with hardwood broadleaf trees
8	Graminoid marsh	Grass dominated marsh
9	Spartina marsh	Marsh dominated by cordgrass
10	Cattail marsh	Marshland dominated by Typha
11	Salt marsh	Coastal salt marsh
12	Mudflats	Tidal mud flats
13	Water	Open water

Loading the data

Copy the data from moodle to your personal directory and load them in your workspace. The data are provided in the ENVI-format, consisting of the image file itself and a *.hdr-file containing the meta information.

```
setwd ("PATH_TO_YOUR_FOLDER") ## set the path to your folder
library (terra)
image <- rast ("aviris_ksc")
```

Take a look a the image as R/G/B and NIR/R/G composite to get an idea of its content.

```
plotRGB (image, 25, 15, 5, stretch="lin") ## R/G/B
plotRGB (image, 40, 25, 15, stretch="lin") ## NIR/R/G
```

The Western parts of the image cover the city of Titusville, the middle displays the Indian River which is here at least 1 km wide, and the Eastern parts cover the KSC marshlands. These marshlands are divided by two major highways.

In addition to the image, we need the wavelength information of the spectral bands and the ground truth points, which is provided in second column of the file 'aviris_ksc_bands.txt' and was taken from the *.hdr file that came with the image dataset.

```
wl <- read.table ("aviris_ksc_bands.txt", header=T, sep=" ", dec=".")[,2]
names (image) <- paste0 ("nm", round (wl, 0)) ## label bands with wavelengths
```

The ground truth data come as point shapefile. The sampling points were divided in two equal subsets (50 points per class), a calibration data set (cal) for training the Random Forest algorithm and a validation data set (val) for accuracy assessment.

```
cal.gt <- vect ("aviris_ksc_gt_calpoints.shp")
val.gt <- vect ("aviris_ksc_gt_valpoints.shp")
```

The class membership of the points can be extracted from the attribute table of the shapefile.

```
cal.class <- as.factor (cal.gt$DN)
val.class <- as.factor (val.gt$DN)
```

We add the points to the image for a visual examination. For a more intuitive understanding, a distinct color value is assigned to each class.

```
plotRGB (image, 25, 15, 5, stretch="lin")
cl <- c ('#AA0000', '#00FF00', '#FFa500', '#FFD700', '#000000', '#FFFF00',
         '#006400', '#6B8E23', '#D2B48C', '#8FBC8F', '#BEBEBE', '#8B5A2B',
         '#0000FF')
points (cal.gt, pch=19, cex=0.2, col=rep (cl, each=50))
```

For the analyses and in the attribute table, all classes are assigned to numerical codes (1-13, see table 2.1). However, we will still need the class names and therefore create a vector containing their description.

```
class.names <- c ("Scrub (1)", "Willow swamp (2)", "Cabbage palm hammock (3)",
                  "Cabbage palm/oak hammock (4)", "Slash pine (5)",
                  "Oak/broadleaf hammock (6)", "Hardwood swamp (7)",
                  "Graminoid marsh (8)", "Spartina marsh (9)",
                  "Cattail marsh (10)", "Salt marsh (11)", "Mudflats (12)",
                  "Water (13)")
add_legend ("topleft", legend=class.names, pch=19, cex=0.7, col=unique (cl),
          ncol=3)
```

Q3.2 (1 pt) Which areas of the image are covered by the ground truth and which are under-represented or not covered at all?

The coordinates of the cal-points are used to extract the corresponding reflectance values from the imagery. This reflectance sample is subsequently used to train the Random Forest classifier. The extraction process may take some seconds.

```
cal.ref <- extract (image, cal.gt, ID=FALSE)
```

Some bands still contain weird values (> 65530) that interfere with the analyses. We remove these bands from the data, leaving a total of 170 bands.

```
cal.ref[cal.ref>65530] <- NA
missing <- is.na (apply (cal.ref, 2, sum))==F
cal.ref <- cal.ref[,missing]
wl <- wl[missing] ## and likewise from the wavelength vector
```

Spectral library

Even though Random Forest is not dealing with spectral libraries, combining the data to a spectral library is useful to gain further insights. We build a library and take a brief look at the spectra of each class.

```
speclib <- list (cal.ref, wl)
```

To plot the mean and standard deviation (i.e., the heterogeneity) of all spectra belonging to the same class, we write a tiny function plotspecs():

```
plotspecs <- function (spectra, wl, class.name) {
 m <- apply (spectra, 2, mean)
 s <- apply (spectra, 2, sd)
 plot (wl, m, main=class.name, ylim=c (0, 450), ylab="Reflectance (%*100)",
       xlab="Wavelength / nm", lwd=2, type="o", pch=19, cex=0.5)
 polygon (c (wl, wl[length (wl):1]), c (m+s, (m-s)[length (wl):1]), density=30,
       angle=45)}

dev.off() ## Close graphic device to reset paremeters (bug in plotRGB)
```

This function is then applied to each class, while Sys.sleep() lets the loop take a 3 sec break after each plot:

```
for (i in 1:13){
 plotspecs (speclib[[1]][(i*50-49):(i*50),], speclib[[2]], class.names[i])
 Sys.sleep(3)}
```

Q3.3 (2 pts) Try to remember what you have learned about spectra in the last sessions. Do the spectra of the classes meet your expectation (e.g., do the spectra of a class representing a vegetation type look like a vegetation spectrum)? Which classes are spectrally heterogeneous (i.e., the spectra vary in shape and magnitude) and which are homogeneous? With a little experience, this information enables a rough estimation which classes can be mapped accurately and which not.

Q3.4 (2 pts) The inital removal of bands did not eliminate all noise in the spectra. Apply the Savitzky-Golay filter on the spectra (without using derivatives, you decide which polynom and window size is suitable) and store the result as object 'cal.ref.smooth' for further analysis. Please provide the code that you have used for this operation.

Random Forest Classification

The Random Forest algorithm is implemented in R in the randomForest package. This package must also be installed and loaded to make the algorithm available.

```
install.packages ("randomForest") ## Install - only before first use!
library (randomForest) ## Load the package
```

The application is pretty simple. A single line of code is needed to build and train the Random Forest model. Random Forest is the most user friendly approach that we will deal with in this class. Just make sure that (for classification) y is provided as 'factor'.

```
rfmod <- randomForest (cal.ref, cal.class, importance=T)
```

The resulting model object rfmod provides the out-of-bag error assessment and a OOB-confusion matrix (based on the calibration data only) with the class error.

```
rfmod
```

Per default option, randomForest() builds a model based on 500 trees. Plotting the model reveals whether this number results in stable overall and class-specific errors. Here, 500 trees are sufficient.

```
plot (rfmod)
```

Q3.5 (2 pts) Build a second Random Forest model (rfmod2) using the smoothed reflectance data. Compare the results in terms of overall and class specific errors as well as in the stability of the errors to the first model. Does the smoothing improve the model performance?

The model can be easily applied onto the image to get a spatial prediction assigning each pixel to a class. This may again take some seconds. If rfmod2 resulted in a much better performance than rfmod, you can likewise apply rfmod2 to the image. Applying rfmod2 requires, however, that you apply the smoothing to all image pixels, which may take some computation time and additional scripting (you decide).

```
map <- predict (image, rfmod)
```

Before we take a look at the classification result, the 13 colors used above to represent the classes are converted into a color gradient.

```
cl2 <- colorRampPalette (cl)
plot (map, col=cl2 (13), legend=F)
add_legend ("topleft", legend=class.names, pch=19, cex=0.7, col=unique (cl),
            ncol=3)
```

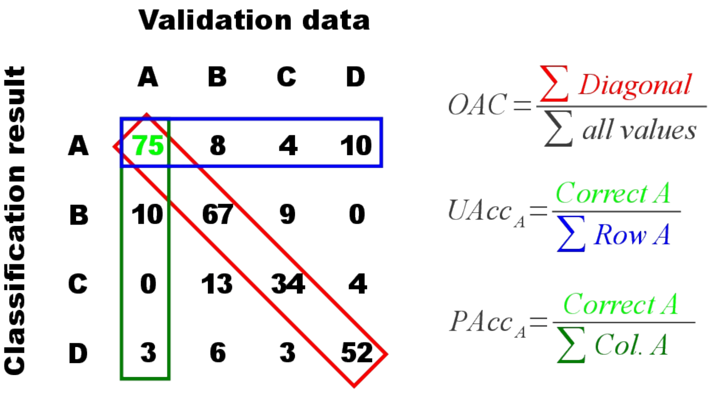
Q3.6 (2 pts) To which class has the built-up area in Titusville been assigned? Can you imagine why the built-up area has been assigned to this particular class?

The resulting map can be written as a GeoTiff to your hard drive. The layer can then be used in a GIS for further analyses. Because the data are not georeferenced, our possibilities on this end are limited.

```
writeRaster (map, "rfmap.tif", format="GTiff")
```

Accuracy Assessment

A classification is only meaningful in combination with an accuracy assessment to evaluate whether the predicted class memberships per pixel are correct and accurate. This assessment is based on a comparison of the predicted and actual (i.e., observed) class memberships of the validation points. The comparison is summarized to a confusion matrix, from which several accuracy measures can be calculated (see fig. 3.1). The predicted class



	A	B	C	D
A	75	8	4	10
B	10	67	9	0
C	0	13	34	4
D	3	6	3	52

$$OAC = \frac{\sum Diagonal}{\sum all\ values}$$

$$UAcc_A = \frac{Correct\ A}{\sum Row\ A}$$

$$PAcc_A = \frac{Correct\ A}{\sum Col.\ A}$$

Fig. 3.1: A confusion matrix and the derived measures of accuracy.

memberships are extracted from the map using the location of the validation points; the actual memberships are listed in the attribute table. The confusion matrix summarizes the frequency of all possible combinations of 'actual vs. predicted.' The rows display the classification result (predicted), the columns list the actual class membership as determined during the sampling of the validation points. The diagonal of the square confusion matrix contains all points that were correctly classified (i.e., the actual membership meets the predicted one). All values outside the diagonal represent misclassified points or pixels.

```
cfm <- table (t (extract (map, val.gt, ID=FALSE)), val.class) ## Confusion-
                                                              ## matrix
  cfm <- cfm[match (colnames (cfm), rownames (cfm)),]
          ## reorder columns to numerical order
cfm
```

The first accuracy measure that can be derived from this matrix is the overall accuracy (OAC). This measure is simply the percentage of points or pixels in the validation data set that were classified correctly. It is calculated by dividing the number of points in the diagonal by the total number of points in the confusion matrix.

```
oac <- round (sum (diag (cfm)) / sum (cfm), 3) ## Overall Accuracy
oac
```

Q3.7 (1 pt) How large is thus the overall error of the classification? Is this error in line with the OOB-error that is given when you call rfmod (or rfmod2)?

The error of a classification is seldom evenly distributed across all classes. In general, some classes show a larger error than others. The OAC gives no information on this distribution across the classes. It is desirable to know which classes have been mapped accurately and which not. We thus need some additional measures that quantify the error per class. These measures are the user's and producer's accuracy that offer information on the accuracy per class from two different points of view. The user's accuracy describes the percentage with which a pixel classified as, for example, 'water' is actually water. The producer's accuracy changes the perspective and provides the information on the percentage of actual water bodies that were correctly mapped as 'water.' Both measures are calculated by dividing the number of correctly classified points per class by the sum of the respective row or column.

```
users.acc <- round (diag (cfm) / apply (cfm, 1, sum), 2) ## User's Accuracy
 names (users.acc) <- class.names
users.acc
producers.acc <- round (diag (cfm) / apply (cfm, 2, sum), 2) ## Producer'S Acc.
  names (producers.acc) <- class.names
producers.acc
```

Q3.8 (1 pt) Which classes show the lowest user's and producer's accuracy and feature thus the largest error?

Importance of spectral bands for the classification

As outlined in the introduction to Random Forest, this technique provides insights into the importance of distinct spectral bands for the classification. For other approaches, this information can only be achieved from separate spectral separability analyses. Not all spectral bands are evenly suitable to delineate the classes in the spectral feature space. The 'variable importance measure' of Random Forest allows for an assessment which bands are most suitable. This assessment offers two advantages:

1. Unimportant bands can be eliminated in a so-called backward selection. The resulting subset of important bands can be used to build a new, refined Random Forest model. Such an opimized model has often an even higher classification accuracy.
2. An identification of important spectral bands eases the interpretation of the modeled relationships. Because wavelength regions are linked to distinct properties of the Earth's surface, the importance information enables to draw conclusions on the relevant properties that make objects spectrally distinct. For example, a high importance of spectral bands in the red edge region and near the chlorophyll absorption features point towards distinct differences in pigmentation, vegetation density, and biomass of the (vegetation) classes to be delineated. The bandwise importances can either be extracted for all bands or hierarchically ordered for the most important bands.

```
imp <- as.data.frame (rfmod$importance) ## or rfmod2
plot (wl, imp$MeanDecreaseAccuracy, type="h", xlab="Wavelength / nm",
      ylab="Mean Decrease in Accuracy") ## Band importance for classification
plot (wl, imp$MeanDecreaseGini, type="h", xlab="Wavelength / nm",
      ylab="Mean Decrease Gini") ## Band importance for classification
varImpPlot (rfmod) ## or rfmod2, only the most important bands
```

Q3.9 (3 pts) Which spectral bands/wavelengths contribute most to the classification? Where do the two metrics agree in their importance assessment, where do they differ? Can you give an explanation for their importance?

Literature and data source:

Ham J, Chen Y, Crawford MM, Ghosh J (2005). Investigation of the Random Forest framework for the classification of hyperspectral data. IEEE Transactions on Geoscience and Remote Sensing 43, 492-501.

Q3.10 (4 pts) Please read the paper on the Random Forest application case study provided in moodle and post in the moodle forum one (or more) question(s) that address(es) the paper and its methodology for our joint discussion. Deadline for posting the question is on the Tuesday before the discussion, 23h59.

Compile the answers to all ten problems including your own question(s). You can again get a maximum of 20 points for all answers being correct plus two extra points for providing an individual solution. Grading will follow the scheme provided in training #2. Have fun!