6 Partial Least Squares Regression

Aim

Modeling and mapping continuous biophysical variables based on spectral data

Data

This training is based on simulated EnMAP data (which were frequently used before actual EnMAP data became available). These data cover the area west of Lake Ammer, Bavaria. The simulation was accomplished by Vista Geowissenschaftliche Fernerkundung GmbH using a Landsat 5 TM image acquired on July 22nd, 2006. To increase the spectral information content of the Landsat image from the original six spectral bands to the expected content of 244 EnMAP bands, an elaborate procedure including the combined use of a radiative transfer model and a land cover classification was applied. The radiative transfer model (see next training session) is able to physically model the scattering, transmission, and absorption of light within a vegetation stand if some basic parameters are known. Here, these parameters are estimated based from the results of the land cover classification. It is important to note that such simulations never capture the full complexity of an actual image; for development and testing purposes they are still suitable. The image and the accompanying information we are using in this session could be downloaded from the EnMAP web page (http://www.enmap.org/?q=node/21, data currently not available anymore). The data provide a spatial resolution of 30 m x 30 m pixel size on the ground. The pixel values are reflectance [% * 100]. We will only use a subset of the image for our training. Further, a pseudo-ground truth data set with values of the leaf area index (LAI, one sided leaf area [m^2] / area on the ground [m^2]) generated from the LAI layer provided alongside the image is used. LAI is an important biophysical variable frequently used, for example, in global climate models. Time series of remotely sensed data are used to derive spatio-temporally continuous information on LAI to cover the variability caused by the phenological development of vegetation.
We use a Partial Least Squares regression (PLSR) model to regress the continuous LAI information collected in some training sites against the spectral information of the corresponding image pixels. The regression model is subsequently applied onto the image data to generate a spatial prediction of the LAI value per pixel. The script code for this training is available in the file 'PLSR.r.'

Loading and preparing the data

```
library (terra)
```

Amongst other R packages that include functions for PLS regression, the pls-package offers a robust and powerful implementation of the PLSR algorithm:

```
install.packages ("pls")
library (pls)
```

The image data are provided as GeoTiff 'enmap_ammersee.tif', the calibration and validation points come as *.txt files. These files contain a matrix with three columns which represent the LAI values, X-, and Y-coordinates of the points.

```
enmap <- rast ("enmap_ammersee.tif")
calpoints <- read.table ("calpoints.txt", header=T, dec=".")
valpoints <- read.table ("valpoints.txt", header=T, dec=".")
LAI <- calpoints[,1]
```

As always, we first take a look at the image and the calibration points. We use the 'cex' argument of the points() function to illustrate the distribution of the LAI values (the larger the point the higher the LAI).

```
plotRGB (enmap, 60, 45, 30, stretch="lin")
points (calpoints[,2:3], cex=LAI / 6 + 0.3, lwd=2)
```

Q6.1 (1 pt): Use your expert knowledge on visual image interpretation. Which vegetation type (deciduous forest, needle-leaf forest, grassland) features the highest LAI values?

Q6.2 (3 pts): PLSR model optimization is known to be affected by a skewed distribution of the response variable. How are the LAI values statistically distributed? Perform a test, visualization or suitable quantification to assess whether you are going to face problems (or have to think about some kind of transformation).

We further need the wavelength information for each band.

```
wl <- read.table ("enmap_wl.txt")[,1]
```

Q6.3 (1 pt): Take a look at the wavelength information. Is there anything strange about it (Of course, there is. But what?)

Extracting the calibration spectra

The reflectance spectra of the calibration points are then extracted from the image using the coordinates of the points in columns 2 and 3. Before we do that, we assign the wavelength information to the bands of the image.

```
names (enmap) <- paste0 ("nm", wl)
calref <- extract (enmap, calpoints[, 2:3], ID=F)
```

Not all spectra regions are equally sensitive to the LAI. Some show a good correlation with LAI, others not. We plot the spectra and use a color coding scheme to illustrate the distribution of the LAI for a better understanding.

```
dev.off()
gc <- colorRampPalette (c ("#784421", "#00FF00", "#008000"))
plot (wl[-c (90:99)], calref[1, -c (90:99)] / 100, type="l", ylim=c (0, 50),
      col=gc(10)[round (LAI[1]) + 1], xlab ="Wavelength / nm",
      ylab="Reflectance / %")
for (i in 2:65)
  lines (wl[-c (90:99)], calref[i, -c (90:99)] / 100,
         col=gc (10)[round (LAI[i]) + 1])
legend ("topleft", legend=0:9, col=gc(10), lty=1, lwd=2)
```

Q6.4 (1 pt): According to your visual examination - in which spectral regions is the LAI value related to the reflectance?

Building a first PLSR model

Building the PLSR model basically requires only a single line of code. In this model, the LAI values are regressed against the corresponding reflectance spectra. For the sake of convenience, the LAI values and the reflectance data are merged to one data frame (caldat), which is called in the model function. The plsr() function builds models for all possible numbers of latent vectors (LVs). It is our turn to identify the number of LVs that results in the smallest RMSE in cross-validation. The function offers two cross-validation options:

Q6.5 (1 pt): Which cross-validation methods are available for plsr() or the plsr implementation you are using?

```
caldat <- data.frame (LAI, calref)
plsmod <- plsr (LAI~., data=caldat, val="CV", jackknife=T, scale=T)
```

With ‘jackknife=T’ we perform an assessment of the stability of the regression coefficients across the cross-validation

steps. With 'scale=T' the reflectance values are standardized to get an equal weight in the model. This enables an additional interpretation opportunity of their importance in the model.

Q6.6 (1 pt): What happens if we refrain from standardizing the bands? Which bands would automatically get a higher importance in the model?

To identify the number of LVs that lead to the best fitting model, we need to consult the RMSE values in cross-validation. The RMSE values for all candidate models can be extracted from the plsmod.object for calibartion and cross-validation.

```
rmse <- RMSEP (plsmod, "all")
rms <- rmse$val[1:2,,-1]
colnames (rms) <- c (1:ncol (rms))
rownames (rms) <- c ("RMSEcal", "RMSEval")
```

We plot the RMSE values as barplots and take a closer look at their trends across an increasing number of LVs.

```
barplot(rms[2,1:20], space=0, col=4, ylim=c (0, max (rms[2,])), las=2,
        cex.names=0.6, xlab="# latent vectors", ylab="RMSE")
barplot(rms[1,1:20], space=0, col=2, add=T, axes=F, axisnames=F)
legend ("topright", legend=c ("cal", "val"), pch=15, col=c ('#FF0000',
       '#0000FF'), bty="n", horiz=T)
```

Q6.7 (2 pts): Why is the RMSE in calibration (displayed in red) decreasing monotonically while the RMSE in validation (blue) shows a more complex behavior? Which number of LVs leads to the smallest RMSE in cross-validation?

We assign the number of LVs leading to the smallest RMSE to the object 'nlv'.

```
nlv <-9 ## Change the number if not appropriate
```

How well does this model fit? A good measure for that performance is the combination of R^2 (the closer to 1 the better) and RMSE (the lower the better) in cross-validation. In addition the difference between these performance indicators and the corresponding metrics for the calibration data should not be too large. All metrics need to be extracted from the plsmod-object and are subsequently visualized for inspection.

```
fit.val <- plsmod$validation$pred[,,nlv]
fit.cal <- plsmod$fitted.values[,,nlv]
plot (LAI, fit.val, xlab="observed", ylab="predicted", pch=19,
      col='#0000FF', main="predicted vs. observed")
pred.obs.val <- lm (fit.val~LAI)
abline (pred.obs.val, col='#0000FF')
points (LAI, fit.cal, pch=19, col='#FF0000')
pred.obs.cal <- lm (fit.cal~LAI)
abline (pred.obs.cal, col='#FF0000')
leg <- paste (c ("RMSEcal", "RMSEval", "R2val", "R2cal"), '=',
               c (round(rms[,nlv],3), round (summary (pred.obs.val)$r.squared,
                  2), round (summary (pred.obs.cal)$r.squared, 2)))
legend ("bottomright", legend=leg, text.col=c ('#0000FF', '#FF0000', '#0000FF',
       '#FF0000'), bty="n")
```

Take a note of the R^2- and RMSE-values.

Next we can check the regression coefficients. The absolute value of the coefficients for the standardized spectral bands (i.e. their value regardless the sign) is an indicator of their importance in the model. The jack-knifing offers a p-

value that is a metric for the variability of the coefficients during the cross-validation. A low variability, indicated by a small p-value, is desirable since the respective coefficient is stable across the different subsets of the data and hence rather robust. A user-defined threshold can be used to convert the p-values into a binary significant/not significant classification.

```
reg.coef <- coef (plsmod, ncomp=nlv)
p <- 0.1 ## define the threshold for the level of significance; adjust, if
necessary
jt <- suppressWarnings (jack.test (plsmod, ncomp=nlv))$pvalues
sig <- jt<p
```

We now plot the regression coefficients and highlight the significant bands.

```
plot (wl, reg.coef, type="h",xlab="Wavelength / nm", ylab="Coefficient")
points (wl, reg.coef, pch=c (1, 19)[sig+1])
legend ("topleft", legend=paste0 (c ('p<', 'p>='), p), pch=c (19, 1), bty="n")
```

Q6.8 (2 pts): Which spectral bands feature a high influence (i.e., a high absolute value of the regression coefficient)? Is this selection in line with your visual interpretation of the spectra?

Optimizing the model through backward-selection of spectral bands

Now we can try to refine the model by removing all bands that have not been labeled as significant from the analysis. This – in theory – eliminates unnecessary information and noise from the data and makes the model more parsimonious. For practical reasons it is technically more straightforward to remove the bands from the image data and to extract the spectral calibration data again than to eliminate the bands in all data sets.

```
wl2 <- wl[sig]
enmap2 <- subset (enmap, (1:244)[sig==F])
calref2 <- extract (enmap2, calpoints[,2:3], ID=F)
caldat2 <- data.frame (LAI, calref2)
```

Based on this subset, we build a new model, select the number of LVs to be included and evaluate the model performance.

```
plsmod2 <- plsr (LAI~., data=caldat2, val="CV", jackknife=T, scale=T)

rmse2 <- RMSEP (plsmod2, "all")
rms2 <- rmse2$val[1:2,,-1]
colnames (rms2) <- c (1:ncol (rms2))
rownames (rms2) <- c ("RMSEcal", "RMSEval")
barplot(rms2[2,1:20], space=0, col='#0000FF', ylim=c (0, max (rms2[2,])),
             las=2, cex.names=0.6, xlab="# latent vectors", ylab="RMSE")
barplot(rms2[1,1:20], space=0, col='#FF0000', add=T, axes=F, axisnames=F)
legend ("topright", legend=c ("cal", "val"), pch=15, col=c ('#FF0000',
       '#0000FF'), bty="n", horiz=T)

nlv <- 11 ## Change the number if not appropriate

fit.val2 <- plsmod2$validation$pred[,,nlv]
fit.cal2 <- plsmod2$fitted.values[,,nlv]
plot (LAI, fit.val2, xlab="observed", ylab="predicted", pch=19,
      col='#0000FF', main="predicted vs. observed")
```

```
pred.obs.val <- lm (fit.val2~LAI)
abline (pred.obs.val, col='#0000FF')
points (LAI, fit.cal2, pch=19, col='#FF0000')
pred.obs.cal <- lm (fit.cal2~LAI)
abline (pred.obs.cal, col='#FF0000')
leg <- paste (c ("RMSEcal", "RMSEval", "R2val", "R2cal"), '=',
              c (round (rms2[,nlv], 3), round (summary (pred.obs.val)$r.squared,
                 2), round (summary (pred.obs.cal)$r.squared, 2)))
legend ("bottomright", legend=leg, text.col=c ('#0000FF', '#FF0000', '#0000FF',
       '#FF0000'), bty="n")
```

Q6.9 (1 pt): Compare the model fit to the results of the first model. How did RMSE and R^2 change?

Mapping the LAI distribution

The model is then applied on the image data for a pixel-wise prediction of LAI.

```
pixval <- values (enmap)
mapvalues <- predict (plsmod, pixval,ncomp=nlv)
map <- subset (enmap, 1)
values (map) <- mapvalues
```

For an intuitive reading of the map, we define a color gradient.

```
cl <- colorRampPalette (c ("#F4EED7", "#00FF00", "#008000", "#000000"))
plot (map, col=cl (255), axes=F)
```

The application of the regression equation results in a LAI prediction for each pixel. This prediction includes pixels that are free from vegetation, such as in the Lake Ammer. The model was, however, only calibrated for pixels covered by vegetation. We thus need to mask all pixels that are not represented by the calibration data, i.e., all non-vegetation pixels. A mask separating vegetation from non-vegetation can be easily derived from the Normalized Difference Vegetation Index (NDVI). The NDVI describes the difference between the red and NIR spectral region in the spectrum. Vegetation pixels show a steep increase from the red to the NIR, non-vegetation spectra do not feature such pronounced differences or even feature a decrease in reflectance from the red to the NIR. To mitigate effects of the overall brightness of the spectrum, the difference is normalized, resulting in index values [-1, 1]. Vegetated pixels show a positive NDVI value (the denser the vegetation, the closer to 1). Non-vegetated pixels have index values close to 0 or even negative values. If we calculate the NDVI for each pixel and apply a threshold, the result can be used as binary mask for the LAI layer. For EnMAP data, the NDVI is calculated following equation 6.1 using bands 65 (Ref800) and 47 (Ref670).

$$NDVI = \frac{Ref_{800} - Ref_{670}}{Ref_{800} + Ref_{670}} \qquad \text{eq. 6.1}$$

```
ndvi <- (enmap[[65]] - enmap[[47]]) / (enmap[[65]] + enmap[[47]])
plot (ndvi, axes=F)
```

Thresholding (NDVI > 0) converts the NDVI layer into a binary mask (NA / 1) that is subsequently applied multiplicatively to the LAI layer.

```
ndvi.mask <- classify (ndvi, matrix (c (-Inf, 0, NA, 0, Inf, 1), 2, 3,
                          byrow=T))
map2 <- map * ndvi.mask
```

```
plot (map2, col=cl (255), axes=F)
```

After masking of non-vegetated pixels some pixels not represented in the calibration data set will still remain. For some of these pixels, negative values are predicted (add some leaf area to get a leaf area of zero...) or the predicted leaf area exceeds the calibration range. These values should be removed from the prediction, too.

range (calpoints[,1])

```
map2 <- classify (map2, matrix (c (-Inf, 0, NA, 9, Inf, NA), 2, 3, byrow=T))
plot (map2, col=cl (255), axes=F)
```

Voilà, the final LAI map.

Accuracy assessment

While the model has already been validated through cross-validation, it lacks an independent validation of the prediction. This independent validation allows for a more accurate assessment whether the model is indeed transferable. We use the validation data and unmasked map for this independent accuracy assessment. For regression models, the predicted values are plotted against and correlated with the actual observations. The correlation R^2 describes the model fit in validation.

```
predval <- extract (map, valpoints[,2:3])
par (pin=c (3, 3))
plot (valpoints[,1], predval, xlim=c (0, 9), ylim=c (0, 9), xlab="Observed LAI",
      ylab="Predicted LAI")
abline (0, 1)
vm <- lm (predval~valpoints[,1])
abline (vm, lty=2)
text (0, 8, paste0 ("R2val=", round (summary (vm)$r.squared, 2)), pos=4)
rmse <- sqrt(mean((predval-valpoints[,1])^2))
text (0, 7, paste0 ("RMSE=", round (rmse, 2)), pos=4)
```

Q6.10 (1 pts): Compare the R^2 and RMSE of the independent validation to the reults of the cross-validation. Do they differ? If yes, can you guess why?

Q6.11 (3 pts): Search the web for the 'ratio of performance to deviation (RPD)', a measure to interpret the prediction ability of a model. Calculate the RPD for our second model. How do you judge the prediction ability of your LAI model?

Read the paper by Gomez et al. (2008) and post a related question in the discussion forum. Put the answers to all eleven problems and the question you have posted in the forum (3 pts) into a single document and upload it to Moodle. You can get a maximum of 20 points for all answers being correct and for asking a relevant question plus two bonus points for deviating from the solution/code provided in the working assignment.

References

Gomez C, Viscarra Rossel RA, McBratney AB (2008). Soil organic carbon prediction by hyperspectral remote sensing and field visNIR spectroscopy: An Australian case study. Geoderma 146, 403411.