Regression for Spatial Data

Introduction

In this exercise we will be estimating the chlorophyll content from a hyperspectral image. More specifically, you will use data points measured on the ground by a Chlorophyll content meter instrument (Minolta CCM-200) and the corresponding hyperspectral values as a dataset to train two different models.
You will then use the trained models on the hyperspectral image.

The provided hyperspectral image has been acquired by the CHRIS/PROBA spaceborne sensor and contains 62 bands in the visible and near-infrared (NIR) region (400–1000 nm) at a spatial resolution of 34m.

Background

In this exercise, we will learn how to use two different models to perform regression:

- Linear model:
  The goal is to find the coefficients β_i of the equation:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_N X_N \tag{1}$$

  where Y is the variable to predict (the chlorophyll content in our case) and X_i are the predictor variables (in our case, the hyperspectral values). So in the exercise, we have $N = 62$ (as we have 62 bands).
  In the book (James, G., Witten, D., Hastie, T., Tibshirani, R., 2017: "An Introduction to Statistical Learning, with Applications in R." Vol. 112, Springer, New York), the main chapter of interest is chapter 3, and more specifically chapter 3.2.
- KNN:
  This is a simple non-parametric approach to regression. Here, Y is predicted by averaging the values of the K nearest neighbors to $\mathbf{X}$.
  In the book, this approach is described in chapter 3.5.

To evaluate and compare our models, we will use several metrics: the Root Mean Square Error (RMSE, chapter 2.2.1), Mean Absolute Error (MAE) and the coefficient of determination (R^2, chapter 3.1.3).

Instructions

As stated above, our goal is to estimate the chlorophyll content from an hyperspectral image. As a ground truth, we will use 135 data points (with hyperspectral values and chlorophyll content) which have been acquired *in situ*. In this exercise, we will only provide you the image and the data points.
A quick explanation about the structure of the pdf:

- Whenever there is a grey box, you have to provide something.
- An item in the box starting with a bold **R** requires you to fill in some code in the provided .R file.
- The letter **Q** marks a question that you have to answer in your report.

Tasks

1 Setup

1.1 Launch your R software environment and copy the data provided to a folder on your local disk. Open the provided .R file.

1.2 For this exercise, you will need to load the following libraries: FNN (for the KNN regression) and raster. Optionnally, you might want to load ggplot2 (to make nice figures).

2 Data Preparation

2.1 In the provided material, you have a file named *J_SPARC_one_day.csv*. Open it in Excel.
This file is organized as follow:

- each column (after the first one) correspond to a sample
- the first 3 rows indicates the chlorophyll, leaf area index and the Fraction of Vegetation Cover (FCover) for each sample
- the following rows indicate the hyperspectral responses of each sample.

R Open the csv in R and format it as a data frame with meaningful columns and rows names.

Q Plot the histogram of the chlorophyll content of the sample points and comment it.

Q Plot the hyperspectral values of samples 11 and 100. Look at their respective Chlorophyll content and comment it.

2.2 As we are dealing with supervised algorithms, we need to split our dataset in training and validation set.

R Split the dataset in a training and a validation step.
Q Which methodology did you use, and why?

3 Linear models, part 1

Start by training a linear model on your data (corresponding function *lm()*).

R Train and evaluate (using R^2, MAE and RMSE) the linear regression model.
Q Report these values and comment them.

4 Data normalization

In the previous part, we took the raw data. However, you can see from the last question of 2.1 that the raw hyperspectral value might not be a good indicator of the chlorophyll content. What people generally do with this data, is to normalize the values of each spectrum separately with respect to the value found in the 54^{th} band.

R Perform the normalization.
Q Once again, plot the normalized hyperspectral values of samples 11 and 100. What can you say with respect to the plot that you made in 2.1?

5 Linear models, part 2

Re-train a linear model on the normalized data.

R Train and evaluate (using R^2, MAE and RMSE) the linear regression model on the normalized data.
Q Report these values and comment them.
Q Make a scatter plot of the predictions (both with the normalized and un-normalized data) against the ground truth.

6 K-Nearest Neighbors

6.1 In this task, we will train a KNN model for regression with a fixed value for K.

Q What value for K would you guess being reasonable?
R Run the KNN (function *knn.reg()*)
Q Comment the scores of the KNN and compare to the ones previously obtained by the linear models.

6.2 We will now tune the hyper-parameter of the model (K). This means you will loop (use *for()*) the KNN with different values of K and retain the one that gives the best performances (cross-validation, chapter 5.1). We will use the 'validation set approach' (chapter 5.1.1, take a moment to read it), i.e. we will sacrify part of our training samples to find K.):

1. Run the KNN by using as training samples the first 70 samples of the training set and as test samples the remaining ones of the training set. **You won't use the test set here, otherwise you will bias your final results!**
2. Compare the predictions with the true Chl values of those remaining samples. The one with the lowest RMSE is the one you will use for the final evaluation.

Q What range of values would be suitable for K in this problem?
R Try all the values in the range you chose and choose the one giving the best RMSE.
Q Make a scatter plot of the predictions of the KNN and linear model against the ground truth. What are the advantages and the shortcomings of such omethod?

7 Qualitative evaluation on hyperspectral image

We will now evaluate our two models on the provided image named *CHRIS.tif*. We start by opening and displaying the image.

R load the image as a raster. Use function *stack()*.
Q What is the size of the image you loaded?
R using the function *plotRGB()*, plot the image in RGB (bands corresponding to 450, 550 and 650 nanometers correspond to the three channels) and describe it.

7.1 Linear regression

First we predict and produce the map for the linear regression.

R Convert the raster to a data frame (using function *as.data.frame()*). This converts the original image in a rows-by-columns $\times$ bands matrix.
R Don't forget to normalize the spectra and to rename the columns.
R Apply the linear model to the image.
Q Display the image (you will have to create a raster with the function *raster()*) and comment the results

7.2 KNN

R Convert the raster into a matrix using function *raster::as.matrix()*. This converts the original image in a rows-by-columns $\times$ bands matrix.
R Don't forget to normalize the spectra
R Apply the KNN to the image.
Q Display the image (you will have to create a raster with the function *raster()*) and comment the results with respect to the linear regression one.