Excercise_5

October 9, 2019

1 ICAT3190, Module 5, Excercises

1.1 Satellite image classification

Following satellite image is obtained from the Söderfjärden meteorite crater near Vaasa at 10:00 in 28.9.2019. The image is acquired from European Sentinell 2 satellite by means of multispectral imaging device (MSI). The multispectral camera has acquired the image using 13 different wavelength bands instead of three (RGB) in the normal camera. The image was searched and dowloaded using Copernicus Open Access Hub, and preprosessed by using ESA's SNAP tool.

The bands used are

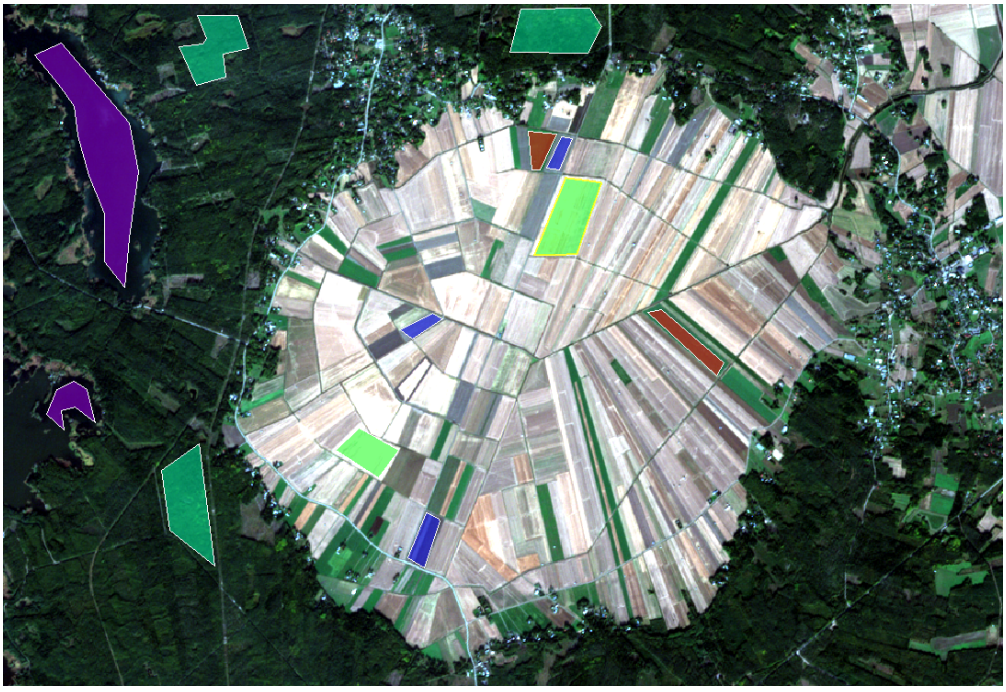
Band number	Band name	Wavelength	Region	Remarks
1	B1	443 nm	Violet	Chlorophyll-A
2	B2	490 nm	Cyan	
3	B3	560 nm	Green	
4	B4	665 nm	Red	Chlorophyll_A
5	B5	705 nm	Red	
6	B6	740 nm	Red	
7	B7	783 nm	Deep red	
8	B8	842 nm	NIR	
9	B8A	865 nm	NIR	
10	B9	945 nm	NIR	
11	B10	1375 nm	NIR	
12	B11	1610 nm	NIR	
13	B12	2190 nm	NIR	

The channels listed above can be used for creating a natural looking RGB-image, as shown below.

Even though, only three channels are used for RGB image, all 13 can be usefull features for land type and crops classification.

1.2 Training data

In the image above, an expert has manually segmented some areas of the image and labelled them according to land use. The labelled areas are:



machine_learning.svg

Segment no.	Segment name	Segment color
1	Water	Violet
2	Forest	Green
3	HayField	Red
4	StrawField	Light Green
5	DarkSoil	Blue

1.3 Task 1

1.3.1 Read the data

Open the training data, which contains the pixel values of the segmented regions shown in the image above. Each pixel contains 13 floating point values between 0 and 1, one for each channel.

The training data is stored in a HDF5 file `Soderfjarden_training_data.h5`, it has following data sets:

- `spectra`: a matrix, which dimensions are (23754, 15). Notice that two first values are the x and y coordinates of the pixel. Do not use them for classification, but use only following 13 channels. Construct a design matrix X (23754,13) by slicing this data.
- `labels`: The expert segmented labels (23754, 1), the true values, the ground truth. This is the y-vector which you try to predict
- `colnames`: The name of the columns (15,1), if you need them. Contains the band names, shown in he table above, plus the name of the x and y coordinate columns.
- `wavelengths`: The wavelengts of each band (13,1), the same shown in the table above.
- The HDF5 includes also a data attribute called `timestamp`, which contains the time when the image was acquired.

If you can't remember how to read HDF5 file, refer to excercise 3.

1.3.2 Make training set and test set

Separate your data X and y to training set (X_train, y_train) which contains 75% of the data and to

```
<ipython-input-1-dae54fb501a2> in <module>()
  1 ## >> Some tests, do not change
----> 2 assert(X.shape==(23754,13))
  3 assert(y.shape==(23754,))


NameError: name 'X' is not defined
```

1.4 Task 2

- Study the data, select a classifier, and train it using the training data.
- Use cross validatin to test the performance of the classifier and tune it's parameters as good as you can
- Finally test the classifier with the test set
- Report the classification accuracy in the training set, cross validation score and accuracy in the test set
- Plot also the confusion matrix when predicting the test set.
- What is your opinion of the performance? Is there signs of overfitting?

```
In [127]: ### >>>>> Write your code here

In [129]: ### >>>> Some testing
          # Your classifier should reach above 90% of accuracy in the test set
```

1.5 Task 3, Select the best channels

Like it often is, some features are more important for classification than the others.

- Study which channels are the most important for the classification. You can use LASSO or Elastic Nets to select variables (SelectFromModel), as shown in the lecture notes of Module 6, or ir you use random forests (Extratrees, or boosted trees), they already keep account on most often used features, see also lecture notes of Module 6.
- Plot of scatter plot of pixels in the training set, using two most important features

```
In [108]: # >> Write your code here

In [121]: ### >>> Some testing
          # It seems that there are only a few bands which are sufficient for classification.
          # Threfore you should get pretty nice separation of clusters in the scatter plot
          # You can also try to plot a scatter plot with PCA
```

1.6 Task 4, Extra task

Now you have trained a classifier to perform well in the trainig data. It is time to put it in actual work. - Read the data of the whole image from another HDF5 file `Soderfjarden_image.h5`. It has the same structure as the training data file, except few differences - There is no labels available, since the whole image is not labeled - There is no pixel coordinates in the begining of spectra

matrix, but only n samples and 13 channels - There are also two additional attributes (width=924, height=630) which shows the image dimensions - Predict all values in the image, and store the results in vector y - Convert y to numerical, so that each class is described by a unique integer - Reshape y to image shape `image=y.reshape(630,924)` - Plot the image `plt.imshow(img)`

In [1]:
```
###>>> Write your code here
```

In [3]:
```
##>>> SOme testing
# You should see that all pixels are now assigned to some classes. You can compare wit
# image and try to find out if the classification looks good.
# If it does not, how it could be improved?
```