

Project 1, due Oct the 7th, 11:59pm

Rules

- 1) submit your .ipynb file by email, titled "CMP SC ECE: 4001-7001: Project 1: XXX", where XXX is your name.
- 2) your file should be named Project1_XXX.ipynb, where XXX is your name.
- 3) clear all cells before submitting.

You can use

- 1) our class notes (PPTs)
- 2) your book
- 3) any Jupyter pages so far
- 4) webpages on the internet (e.g., Wikipedia)

You CANNOT

- 1) work with anyone else (e.g., someone in the class, on the internet, robot, etc.)

NO late submissions will be accepted.

Do the following tasks with respect to the Indian Pines data set (and its ground truth).

OK, so lets load the Indian Pines

In [1]:

```
import scipy.io
from PIL import Image
import numpy as np
```

```
mat = scipy.io.loadmat('/dsa/git/LecWeek3/Indian_pines.mat')
data = mat['indian_pines']
Xdim, Ydim, Bands = data.shape
print( data.shape )
```

```
(145, 145, 220)
```

Next, load the Indian Pines ground truth

In [2]:

```
mat2 = scipy.io.loadmat('/dsa/git/LecWeek3/Indian_pines_gt.mat')
data2 = mat2['indian_pines_gt']
```

```
NoLabels = np.amax(data2)
print( NoLabels )
```

```
16
```

Here is some simply Python code to help you find which pixels are a certain class

In [3]:

```
# labels
classlabels = [ "No class", "Alfalfa", "Corn-notill", "Corn-mintill", "Corn", "Grass-
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```

pasture", "Grass-trees", "Grass-pasture-mowed", "Hay-windrowed", "Oats", "Soybean-
notill", "Soybean-mintill", "Soybean-clean", "Wheat", "Woods", "Buildings-grass-
trees-dives", "Stone-steel-towers" ]

# which class?
which_class = 12

# what locations?
#
# as an image
WhatLocsImg = np.where( data2 == which_class , 1, 0 ) # so, 1 is our class, 0 is not
(or white and black)
# what specific indices
WhatLocs = np.where( data2 == which_class )
# get in 2D index form
i1 = WhatLocs[0]
i2 = WhatLocs[1]
print( "An example index is: " + str(i1[0]) + " " + str(i2[0]) )
print( "There are " + str(i1.size) + " pixels in this class" )

# show the image
print( classlabels[which_class] )
result = np.uint8(WhatLocsImg*255)
display(Image.fromarray(result).resize((250,250),Image.NEAREST))

An example index is: 2 28
There are 593 pixels in this class
Soybean-clean

```



Few last comments.

The indian pines data set has 220 bands. (what I gave you!)

Wavelengths span 0.4–2.5 μm .

Each pixel is approx 20m (spatial)

Water absorption bands are 104–108, 150–163, and 220

Now, on to the stuff that you need to do!!!

Task 1 (everyone) ¶

- 1) Prompt the user for a class (one of 16) and compute (in Python) its mean and covariance matrix [make this its own cell]
- 2) Plot (via `plt.plot(...)`) the mean signature. Label the x-axis w.r.t. waveform. [make this its own cell]
- 3) Plot (as an image) the covariance matrix of the user selected class. [make this its own cell]

note: I recommend you make cells, but make 1), 2), and 3) functions.

In []:

```
### Prompt the user for which class to work with
which_class = input("What class? {1, 2, ..., 16}: ")
print("Class selected: " + str(which_class) )

# now, on to your code...
```

Task 2 (everyone) ¶

Run ACE, where

$$\text{ACE}(x_{\{(i,j)\}}; s, \mu, \Sigma) = \frac{\left[(s - \mu)^T \Sigma^{-1} (x_{\{(i,j)\}} - \mu) \right] \left[(s - \mu)^T \Sigma^{-1} (x_{\{(i,j)\}} - \mu) \right]}{\left[(s - \mu)^T \Sigma^{-1} (s - \mu) \right] \left[(x_{\{(i,j)\}} - \mu)^T \Sigma^{-1} (x_{\{(i,j)\}} - \mu) \right]}$$

where

- 1) $x_{\{(i,j)\}}$ is the signal at the pixel under investigation
- 2) s is the signal we are trying to match
- 3) (μ, Σ) are statistics we compute "some how" (see below)

In Task 2, I want you to:

- 1) make s the mean of the class we selected in Task 1
- 2) make (μ, Σ) the global (the entire image) mean and covariance

note: show it "raw" and scaled to [0, largest ACE value observed in the image], prompt the user for which to do

Task 3 (everyone) ¶

Run ACE, where

$$\text{ACE}(x_{\{(i,j)\}}; s, \mu, \Sigma) = \frac{\left[(s - \mu)^T \Sigma^{-1} (x_{\{(i,j)\}} - \mu) \right] \left[(s - \mu)^T \Sigma^{-1} (x_{\{(i,j)\}} - \mu) \right]}{\left[(s - \mu)^T \Sigma^{-1} (s - \mu) \right] \left[(x_{\{(i,j)\}} - \mu)^T \Sigma^{-1} (x_{\{(i,j)\}} - \mu) \right]}$$

where

- 1) $x_{\{(i,j)\}}$ is the signal at the pixel under investigation
- 2) s is the signal we are trying to match
- 3) (μ, Σ) are statistics we compute "some how" (see below)

In Task 3, I want you to:

- 1) make s the mean of the class we selected in Task 1
- 2) make (μ, Σ) the local (with respect to a user defined local window (prompt the user for the size of the local window)) mean and covariance

Task 4 (everyone) ¶

Run local RX, i.e., $\text{RX}(x_{\{(i,j)\}}; \mu_l, \Sigma_l) = (x_{\{(i,j)\}} - \mu_l)^t \Sigma_l^{-1} (x_{\{(i,j)\}} - \mu_l)$, where $\$l\$$ denotes local. Prompt the user for the size of the local window.

Task 5 (graduate students) ¶

Go to

http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes

or

<http://lesun.weebly.com/hyperspectral-data-set.html>

Prompt the user for which data set to load and process (Tasks 1-4), i.e., Indian Pines or {Salinas, Pavia, Washington DC MALL, ...}

Task 6 (graduate students) ¶

W.r.t. Tasks 1-4, ask the user if they want to remove the water absorption bands (and do it if so!!!)

Task 7 (graduate students) ¶

W.r.t. Tasks 3 and 4, prompt the user for the size of a buffer window. For example, say we picked a 5 by 5 size local window with a buffer of size 1. The locations we would sample from would look like:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 2 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

where 2) is our pixel location, 0) are buffer pixels we ignore, and the mean and covariance are computed with respect to 1) locations.

Task 8 (extra credit) ¶

Produce (and display) an "RGB" image for the Indian Pines data set

Do it w.r.t.

- 1) the mean spectral band for red, green, and blue
- 2) the full red, green, and blue distributions

Task 9 (extra credit) ¶

Find "a few good bands" (e.g., 2 or 5 or 10 vs. 220) to detect "stone-steel-tower" in Indian Pines, then apply ACE