WAVeTrends_classify

August 16, 2019

1 West Africa Vegetation Trends (WAVeTrends) Data Companion Script

Python code to be used in obtaining per-pixel categorical vegetation change, at any user-defined level of statistical significance (p-value), from supplementary data contained in the WAVeTrends.tif source file. These data are a combination of continuous variables separately representing long-term woody and herbaceous tendencies in West African Savannas, their respective p-values, and a quality flag indicating the validity of the underlying concept used to assess trends. See User Guide documentation as well as Anchang et al.(https://doi.org/10.3390/rs11050576) for more details. Prerequisites: Python 3.x (Anaconda distributon recommended), Rasterio.

1.1 Import modules

[1]:
```
import os, rasterio
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Patch
from matplotlib.colors import ListedColormap
```

1.2 Set working directory (folder containing the downloaded "WAVeTrends.tif" data file)

[2]:
```
os.chdir("C:/trends_data")
```

1.3 Open the raster data file and check metadata

[3]:
```
src_file = rasterio.open('WAVeTrends.tif')
print (src_file.meta)
```

```
{'driver': 'GTiff', 'dtype': 'float32', 'nodata': -3.4028234663852886e+38,
'width': 649, 'height': 149, 'count': 6, 'crs': CRS.from_epsg(4326),
'transform': Affine(0.049999999999999996, 0.0, -17.143186187593336,
       0.0, -0.049999999999999996, 17.414762969350303)}
```

1.4 Read required bands as numpy arrays (Bands 2 - 6)

b2: difference in mean Rain Use Efficiency between first and last decade (+ve/-ve values = woody increase/decrease); b3: difference in slope of NPP-rainfall model between first and last decade (+ve/-ve values = herb increase/decrease); b4: p-values associated with b2 (significance of woody trends derived from independent t-test); b5: p-values associated with b3 (significance of herb trends derived from ANCOVA); b6: Quality flag based on type of NPP-Rainfall relationship (1 = non valid, 2 & 3 = valid).

[4]:
```
b2 = src_file.read(2, masked=True)
b3 = src_file.read(3, masked=True)
b4 = src_file.read(4, masked=True)
b5 = src_file.read(5, masked=True)
b6 = src_file.read(6, masked=True)
```

1.5 Define desired signficance level (p-value) to assess vegetation change

This will be used to separate change in each vegetation type (whether +ve or -ve) from no change. Valid range of 0 - 1.

[5]:
```
pv_woody = 0.05
pv_herb = 0.05
```

1.6 Classify pixels into vegetation change categories using consecutive nested conditional statements.

1 = No Change; 2 = Vegetation Loss; 3 = Woody Loss Only; 4 = Herbaceous loss only; 5 = Herbaceous Gain/Woody Loss; 6 = Woody Gain/Herbaceous Loss; 7 = Herbaceous Gain Only; 8 = Woody Gain Only; 9 = Vegetation Gain. Note that in the original WAVeTrends.tif file, we have provided already classified trends (Band_1 or b1) using a 0.05 significance threshold.

[6]:
```
class_1 = np.where((b4>pv_woody)&(b5>pv_herb),1,0)
class_2 = np.where((b2<0)&(b3<0)&(b4<pv_woody)&(b5<pv_herb),2,class_1)
class_3 = np.where((b2<0)&(b4<pv_woody)&(b5>pv_herb),3,class_2)
class_4 = np.where((b3<0)&(b4>pv_woody)&(b5<pv_herb),4,class_3)
class_5 = np.where((b2<0)&(b3>0)&(b4<pv_woody)&(b5<pv_herb),5,class_4)
class_6 = np.where((b2>0)&(b3<0)&(b4<pv_woody)&(b5<pv_herb),6,class_5)
class_7 = np.where((b3>0)&(b4>pv_woody)&(b5<pv_herb),7,class_6)
class_8 = np.where((b2>0)&(b4<pv_woody)&(b5>pv_herb),8,class_7)
class_9 = np.where((b2>0)&(b3>0)&(b4<pv_woody)&(b5<pv_herb),9,class_8)
veg_cat = np.where(b6>1,class_9, 0)#apply validity mask
```

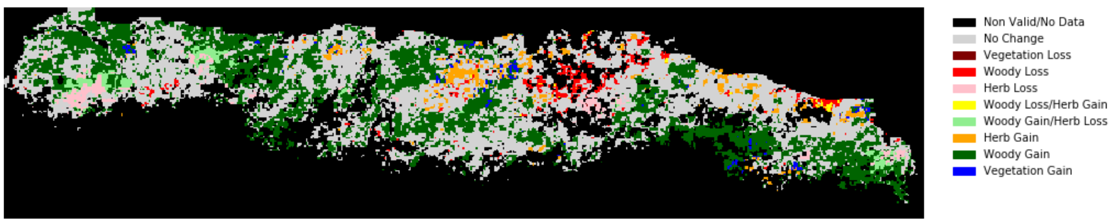
1.7 Display ouput

[7]:
```
colors = ['black','lightgray',␣
↪'maroon','red','pink','yellow','lightgreen','orange','darkgreen','blue']
labels = ['Non Valid/No Data','No Change', 'Vegetation Loss', 'Woody␣
↪Loss','Herb Loss','Woody Loss/Herb Gain',
```

```
        'Woody Gain/Herb Loss', 'Herb Gain', 'Woody Gain', 'Vegetation Gain']
patches = [Patch(color=c, label=l)
                  for c, l in zip(colors, labels)]

cmap = ListedColormap(colors)

fig, ax = plt.subplots(figsize=(15, 15))
ax.imshow(veg_cat,
          cmap=cmap)
ax.legend(handles=patches,
         facecolor ="white",
         edgecolor = "white",
         bbox_to_anchor = (1.21,1))
ax.set_axis_off()
```



1.8 Save raster ouput to TIF

[8]:
```
with rasterio.Env():
    # retrieve and update metadata from original source file
    profile = src_file.profile
    profile.update(
        dtype=rasterio.uint8,#now set to 8-bit unsigned integer
        count=1,#new number of bands
        nodata=0,#new nodata value
        compress='lzw')
    with rasterio.open('classified_trends.tif', 'w', **profile) as dest_file:
        dest_file.write(veg_cat.astype(rasterio.uint8), 1)
```