clustering_sat_image

July 24, 2021

1 Clustering Satellite Images

Our objective is to cluster pixels of a satellite image in order to find some insights.

2 Loading Images

We have 3 differente types of data. The original satellite image with many channel, and also the decomposed to only 2 PCA

```
        b1       b2       b3      b4       b5       b6       b7      b9  \
0  11797.0  10866.0   9907.0  9229.0  13254.0  11461.0   8732.0  5031.0
1  11810.0  10898.0   9933.0  9404.0  13029.0  12277.0   9517.0  5042.0
2  11858.0  10977.0  10068.0  9704.0  13292.0  13226.0  10404.0  5059.0
3  11842.0  10957.0  10053.0  9670.0  12832.0  12994.0  10243.0  5057.0
4  11845.0  10959.0  10050.0  9671.0  13192.0  12669.0  10048.0  5044.0

       b10      b11
0  28591.0  25865.0
1  28564.0  25830.0
2  28580.0  25857.0
3  28643.0  25934.0
4  28729.0  26017.0
```

```
        PC1       PC2       PC3       PC4       PC5       PC6       PC7  \
0 -1.562415  0.609884 -1.899809 -0.078258  0.122494  0.413571 -0.028087
1 -1.208980  0.524218 -0.785919  0.026129 -0.267289  0.513689  0.005252
2 -0.411637  0.522821  0.911681  0.410430 -0.469799  0.513241  0.017641
3 -0.570336  0.384522  0.748034  0.064857 -0.497465  0.493759 -0.013019
4 -0.552427  0.491838 -0.590167 -0.033458 -0.210152  0.407038 -0.035174

        PC8       PC9      PC10
0  0.086845 -0.009167  0.033750
1  0.103105 -0.015590  0.011003
2  0.132162 -0.009331 -0.002152
3  0.127144  0.029468  0.026811
4  0.174953 -0.023369 -0.000859
```

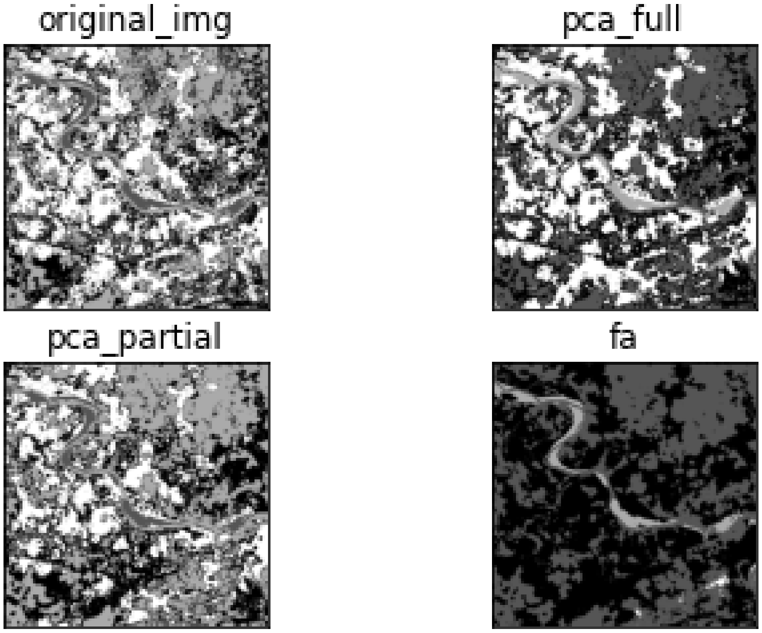
```
        PC1       PC2
```

```
0 -1.562415  0.609884
1 -1.208980  0.524218
2 -0.411637  0.522821
3 -0.570336  0.384522
4 -0.552427  0.491838

         F1        F2
0 -0.423327 -0.574774
1 -0.245363 -0.526909
2  0.122958 -0.489821
3  0.090803 -0.431901
4 -0.116046 -0.345796
```

3 K-Means

```
0
1
2
3
```



4 PAM (Partition Around Medoids)

```
---------------------------------------------------------------------------
MemoryError                               Traceback (most recent call last)
/tmp/ipykernel_195685/2717583691.py in <module>
```

```
      2     data = data.values
      3 
----> 4     plot_clusters(get_model_results_kmedoids(data)['cluster'])
      5     plt.tick_params(left = False, right = False , labelleft = False ,
      6                     labelbottom = False, bottom = False)

/tmp/ipykernel_195685/1728678069.py in get_model_results_kmedoids(data, n_clusters)
      6                         init='k-medoids++')
      7     model_results = data
----> 8     model_results['cluster'] = kmedoids.fit_predict(data)
      9 
     10     return kmedoids.fit_predict(data)

~/Documents/projects/stats-img-processing/venv/lib/python3.8/site-packages/sklearn/
 ↪base.py in fit_predict(self, X, y)
    581         # non-optimized default implementation; override when a better
    582         # method is possible for a given clustering algorithm
--> 583         self.fit(X)
    584         return self.labels_
    585 

~/Documents/projects/stats-img-processing/venv/lib/python3.8/site-packages/
 ↪sklearn_extra/cluster/_k_medoids.py in fit(self, X, y)
    194             )
    195 
--> 196         D = pairwise_distances(X, metric=self.metric)
    197         medoid_idxs = self._initialize_medoids(
    198             D, self.n_clusters, random_state_

~/Documents/projects/stats-img-processing/venv/lib/python3.8/site-packages/sklearn/
 ↪utils/validation.py in inner_f(*args, **kwargs)
     61             extra_args = len(args) - len(all_args)
     62             if extra_args <= 0:
---> 63                 return f(*args, **kwargs)
     64 
     65             # extra_args > 0

~/Documents/projects/stats-img-processing/venv/lib/python3.8/site-packages/sklearn/
 ↪metrics/pairwise.py in pairwise_distances(X, Y, metric, n_jobs,␣
 ↪force_all_finite, **kwds)
   1788         func = partial(distance.cdist, metric=metric, **kwds)
   1789 
-> 1790     return _parallel_pairwise(X, Y, func, n_jobs, **kwds)
   1791 
   1792 

~/Documents/projects/stats-img-processing/venv/lib/python3.8/site-packages/sklearn/
 ↪metrics/pairwise.py in _parallel_pairwise(X, Y, func, n_jobs, **kwds)
```

```
   1357
   1358     if effective_n_jobs(n_jobs) == 1:
-> 1359         return func(X, Y, **kwds)
   1360
   1361     # enforce a threading backend to prevent data communication overhead

~/Documents/projects/stats-img-processing/venv/lib/python3.8/site-packages/sklearn/
 ↪utils/validation.py in inner_f(*args, **kwargs)
     61             extra_args = len(args) - len(all_args)
     62             if extra_args <= 0:
---> 63                 return f(*args, **kwargs)
     64
     65             # extra_args > 0

~/Documents/projects/stats-img-processing/venv/lib/python3.8/site-packages/sklearn/
 ↪metrics/pairwise.py in euclidean_distances(X, Y, Y_norm_squared, squared,␣
 ↪X_norm_squared)
    311     else:
    312         # if dtype is already float64, no need to chunk and upcast
--> 313         distances = - 2 * safe_sparse_dot(X, Y.T, dense_output=True)
    314         distances += XX
    315         distances += YY

~/Documents/projects/stats-img-processing/venv/lib/python3.8/site-packages/sklearn/
 ↪utils/validation.py in inner_f(*args, **kwargs)
     61             extra_args = len(args) - len(all_args)
     62             if extra_args <= 0:
---> 63                 return f(*args, **kwargs)
     64
     65             # extra_args > 0

~/Documents/projects/stats-img-processing/venv/lib/python3.8/site-packages/sklearn/
 ↪utils/extmath.py in safe_sparse_dot(a, b, dense_output)
    150             ret = np.dot(a, b)
    151     else:
--> 152         ret = a @ b
    153
    154     if (sparse.issparse(a) and sparse.issparse(b)

MemoryError: Unable to allocate 466. GiB for an array with shape (250000, 250000)␣
 ↪and data type float64
```

5 Agglomerative Hierarchical

```
  File "/tmp/ipykernel_195685/2844372854.py", line 13
    plt.savefig('data/output/img/PAM_cluster_img.png')
    ^
IndentationError: expected an indented block
```