Chapter 7 + 8 - Cluster analysis + taxonomy

November 10, 2020

[ ]:

```
import geopandas as gpd
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import preprocessing
import numpy as np
from sklearn.mixture import GaussianMixture
```

[ ]:

```
path = 'files/contextual.parquet'
```

[ ]:

```
data = pd.read_parquet(path)
```

[ ]:

```
# normalise data

x = data.values
scaler = preprocessing.StandardScaler()
cols = list(data.columns)
data[cols] = scaler.fit_transform(data[cols])
```

We have now normalised data, let's save them.

[ ]:

```
data.to_parquet('files/contex_data_norm.parquet')
```

[ ]:

```
bic = pd.DataFrame(columns=['n', 'bic', 'run'])
ix = 0

n_components_range = range(2, 40)
gmmruns = 3
```

Measure BIC to estimate optimal number of clusters.

[ ]:

```
sample = data
for n_components in n_components_range:
    for i in range(gmmruns):
        gmm = GaussianMixture(n_components=n_components,␣
 ↪covariance_type="full", max_iter=200, n_init=1, verbose=1)
        fitted = gmm.fit(sample)
        bicnum = gmm.bic(data)
        bic.loc[ix] = [n_components, bicnum, i]
        ix += 1

        print(n_components, i, "BIC:", bicnum)
```

[ ]:

```
bic.to_csv('files/complete_BIC.csv')
```

Plot BIC values

[ ]:

```
import seaborn as sns
import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(16, 16))
sns.lineplot(ax=ax, x='n', y='bic', data=bic)
plt.savefig('files/complete_BIC.pdf')
```

0.1 Clustering

[ ]:

```
n = 30

gmm = GaussianMixture(n_components=n, covariance_type="full", max_iter=200,␣
 ↪n_init=5, verbose=1)
fitted = gmm.fit(data)
```

[ ]:

```
data['cluster'] = gmm.predict(data)
```

[ ]:

```
data.reset_index()[['cluster', 'uID']].to_csv('files/
 ↪200309_clusters_complete_n30.csv')
```

0.2 Dendrogram

[ ]:

```
from scipy.cluster import hierarchy
import matplotlib.pyplot as plt
```

[ ]:

```
clusters = data.reset_index()[['cluster', 'uID']]
```

Save to pdf.

[ ]:

```
group = data.groupby('cluster').mean()
Z = hierarchy.linkage(group, 'ward')
plt.figure(figsize=(25, 10))
dn = hierarchy.dendrogram(Z, color_threshold=30, labels=group.index)

plt.savefig('tree.pdf')
```