

GeoPandas (geoprocessing): Overlays

Dr. Cornelis Stal

April 28, 2022

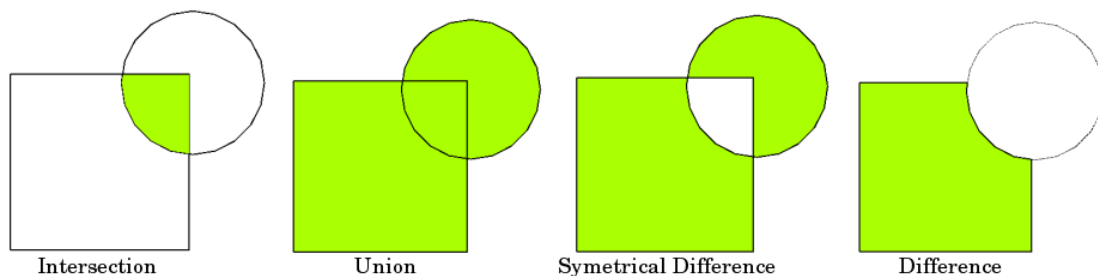
1 Overlays

Ruimtelijke overlappingsen (of in het Engels vertaald naar *Spatial overlays*) laten toe om twee `GeoDataFrames` met geometriën van het type polygoon of multi-polygoon met elkaar te vergelijken. De verschillende methodes die bij een overlay gebruikt kunnen worden, resulteren ieder in een nieuwe `GeoDataFrame` met nieuwe geometriën die overeenkomen met een bepaalde ruimtelijke combinatie van de oorspronkelijke twee inputdatasets, aangevuld met de oorspronkelijke maar samengevoegde attributen. Hiermee verschilt deze techniek van een selectie (waarbij een subset van ruimtelijke objecten wordt aangemaakt) en van een clip (waarbij enkel de attributen van de bronlaag worden meegenomen: zie verder).

Het berekenen van ruimtelijke overlappingsen laat toe om vragen te stellen zoals:

Wat zijn de karakteristieken van alle zones met biologische waardering binnen een bepaalde gemeente? Wat is per gemeente de oppervlakte van verschillende type bestemmingen in het gewestplan?

Het idee achter het berekenen van ruimtelijke overlappen zal hieronder geïllustreerd worden aan de hand van het kanaal Gent-Terneuzen en de gemeentegrens van de stad Gent.



1.1 Data inladen en voorbereiden

We starten met het importeren van de vereiste bibliotheken en het inladen van de data. Uit de dataset met Vlaamse gemeentegrenzen (offline beschikbare ESRI Shapefile) wordt enkel gewerkt met de polygoon corresponderend met de stad Gent. Voor de dataset met wateroppervlakten wordt gewerkt met de [CORINE dataset](#), waarvoor een beperkt aantal objecten aan de hand van de ID worden gedownload door middel van een ESRI REST service.

We bekijken bij het importeren van de CORINE-dataset zeker ook eens de verschillende velden in de `params`-dictionary, en hoe het resulterende object wordt omgezet naar een `GeoDataFrame`-object:

```
[ ]: # Bibliotheken importeren
from shapely.geometry import Point
import geopandas as gpd
import matplotlib.pyplot as plt
import requests

# Instelling om figuren wat groter af te beelden
plt.rcParams['figure.figsize'] = [7, 10]

# Gent uit de dataset met gemeentegrenzen
refgem = gpd.read_file("data/refgem_2018.shp")
gent = refgem[refgem["NAAM"] == "Gent"]

# Grote wateroppervlakken (CORINE 2018)
base = "https://image.discomap.eea.europa.eu/arcgis/rest/services/Corine/
↳CLC2018_WM/MapServer/0/query"
params = {"objectIds": "595867,596425,596417,596444,596422",
          "geometryType": "esriGeometryPolygon",
          "spatialRel": "esriSpatialRelIntersects",
          "units": "esriSRUnit_Meters",
          "returnGeometry": "true",
          "featureEncoding": "esriDefault",
          "outSR": "31370",
          "f": "geojson"}
r = requests.get(base, params=params)
water = gpd.GeoDataFrame.from_features(r.json()['features'])
```

Het eerste object `gent` bevat zoals verwacht slechts één object, namelijk de polygoon en attributen voor de stad Gent.

```
[ ]: gent.plot()
```

De tweede `GeoDataFrame` is een verzameling polygoonen die rechtstreeks zijn ingelezen vanuit de online databron. De individuele polygoonen krijgen een aparte kleur in functie van het veld `Code_18` en aan de hand van een welbepaald [kleurenpalet](#).

```
[ ]: water.plot(column='Code_18', cmap='tab20b')
```

1.2 De operator

De `geopandas.overlay`-functie vereist drie argumenten:

- `gdf1`: eerste geodataframe
- `gdf2`: tweede geodataframe
- `how`: methode voor de uitvoering van de overlay

Het argument `how` is hierbij een van de volgende waarden:

```
['intersection',
```

```
'union',  
'identity',  
'symmetric_difference',  
'difference']
```

Het resultaat van de verschillende waarden wordt hieronder geïllustreerd.

```
[ ]: df_intersection = gpd.overlay(gent, water, how="intersection")  
df_intersection.plot(column='Code_18', cmap='tab20b')
```

Merk op dat in het resultaat van de vorige cel een waarschuwing verscheen over een verschil tussen de verschillende coördinaatreferentiesystemen van de beide datasets (CRS mismatch between the CRS of left geometries and the CRS of right geometries). Dit kan als volgt opgelost worden.

```
[ ]: # Optie 1: CRS toekennen bij aanmaken  
water = gpd.GeoDataFrame.from_features(r.json()['features'], crs="epsg:31370")  
  
# Optie 2: EPSG:31370 expliciet toekennen  
gent = gent.set_crs(epsg=31370, allow_override=True)  
water = water.set_crs(epsg=31370, allow_override=True)  
  
# Optie 3: CRS van Gent overdragen op water  
water = water.set_crs(gent.crs, allow_override=True)  
  
kanaalzone = gpd.overlay(gent, water, how="intersection")  
kanaalzone.plot(column='Code_18', cmap='tab20b')
```

Als we nu kijken naar de attributen van de oorspronkelijke datasets en van de attributen van de nieuwe GeoDataFrame, stellen we vast dat de oorspronkelijke attributen behouden zijn in deze nieuwe dataset.

```
[ ]: gent.head()
```

```
[ ]: water.head()
```

```
[ ]: kanaalzone.head()
```

1.3 De how-operator

Het aanpassen van de how-operator geeft uiteraard verschillende resultaten:

Union:

```
[ ]: unie = gpd.overlay(gent, water, how="union")  
unie.plot(column='Code_18', cmap='tab20b')
```

Identity:

```
[ ]: gelijk = gpd.overlay(gent, water, how="identity")
gelijk.plot(cmap='tab20b')
```

Symmetric_difference:

```
[ ]: symetrischVerschil = gpd.overlay(gent, water, how="symmetric_difference")
symetrischVerschil.plot(cmap='tab20b')
```

Difference:

```
[ ]: verschil = gpd.overlay(gent, water, how="difference")
verschil.plot(cmap='tab20b')
```

Opdracht: OpenStreetMap (OSM) biedt een zeer uitgebreide dataset met ruimtelijke objecten van overal op aarde. De dataset wordt continu aangevuld door een zeer actieve community en is geheel open-source. Verschillende desktop GIS-programma's, zoals [QGIS](#) laten het downloaden van OSM-data toe. Via [GeoFabrik](#) is het ook mogelijk om specifieke continenten, landen of regio's te downloaden als ESRI Shapefile. De complete dataset van België heeft tijdens het voorbereiden van deze notebook een grootte van 840 Mb en bevat een groot aantal lagen en objecten.

Vermits de gehele dataset van België niet altijd erg performant is om te gebruiken bij studies met betrekking tot kleinere studiegebieden, valt het aan te bevelen om de dataset uit te knippen aan de hand van een ruimtelijke begrenzing. De opdracht is daarom om de dataset van België te downloaden als ESRI Shapefile, en vervolgens alle data uit te knippen voor de stad Gent.

```
[ ]: ## UW CODE HIER ##
```