Práctica 4 - Geometría Computacional

Marcos Herrero Agustín

1. Introducción

En esta práctica se toma un par de sistemas meteorológicos con 365 elementos (días) y 357408 variables de estado y se analizan desde dos puntos de vista distintos. Por un lado, se utiliza la técnica de Análisis de las Componentes Principales (PCA) para reducir el número de elementos del sistema de 2021 capturando la mayor variación posible de los datos. Por otro lado, se usa la analogía para predecir la temperatura de un día de 2022 a partir de 4 días similares en 2021.

2. Datos y condiciones iniciales

Para la realización de la práctica se han utilizado los siguientes datos y condiciones iniciales:

- Los archivos air.2021.nc, hgt.2021.nc, air.2022.nc y hgt.2022.nc que contienen la información de los sistemas a analizar. En concreto, contienen la temperatura T y altura geopotencial Z para cada día, longitud (x), latitud (y) y nivel de presión (p) de 2021 y de 2022.
- Para el apartado i):
  - Las restricciones a aplicar sobre el sistema de 2021 ($p = 500$ hPa).
  - El número de componentes principales a extraer.
- Para el apartado ii):
  - Las restricciones a aplicar sobre el sistema de 2021 ($-20 < x < 20, 30 < y < 50$)
  - El día $a_0 := 11/01/2022$ del que predecir la altura geopotencial.
  - La distancia (euclídea) y los pesos a utilizar para calcular los días análogos.
  - El número de días análogos (4) a utilizar para la predicción.

3. Metodología

Antes de realizar lo que nos pide el enunciado, modificamos el rango de longitudes de $[0°, 360°]$ a $[-180°, 180°]$ para evitar que la península ibérica quede partida entre ambos extremos del mapa. Esto hará más fácil restringir el sistema a las coordenadas pedidas en el apartado ii). Para modificar el rango simplemente hemos usado la función roll de numpy en todos los elementos que lo requieren.

En el apartado i), hemos seguido los sigientes pasos:

1. Restringimos el sistema a $p = 500$ hPa y nos quedamos solo con la altura geopotencial Z.
2. Transformamos el array multidimensional con la información del sistema restringido en una matriz de elementos $\times$ variables de estado.
3. Utilizamos la utilidad PCA de sklearn para obtener el sistema generado con las 4 componentes principales del anterior. La aplicamos sobre la matriz traspuesta del sistema para reducir el número de elementos en lugar del de variables de estado. La misma PCA nos da el porcentaje de varianza explicada por cada componente principal.

En el apartado ii), hemos seguido los siguientes pasos:

1. Obtenemos el estado de las variables del día a_0 del sistema de 2022. Restringimos el sistema (y el día a_0) a $-20 < x < 20$ y $30 < y < 50$.
3. Utilizamos la utilidad NearestNeighbors de sklearn para calcular los 4 días más análogos a a_0 de 2021 basándonos en Z.
4. Para obtener la predicción buscada, calculamos la media de los 4 días más análogos.
5. Calculamos el error absoluto medio de la predicción realizada.

4. Resultados y discusión

4.1. Apartado i)

La primera componente principal explica, por sí sola, el 88.77 % de la varianza. En cambio, la segunda, tercera y cuarta explican únicamente el 5.18 %, el 0.54 % y el 0.36 %, respectivamente.

La figura 1 muestra el sistema resultante de quedarnos con las 4 primeras componentes principales.Se observa que en la primera componente principal la altura geopotencial Z apenas varía con la longitud, sino que está casi perfectamente

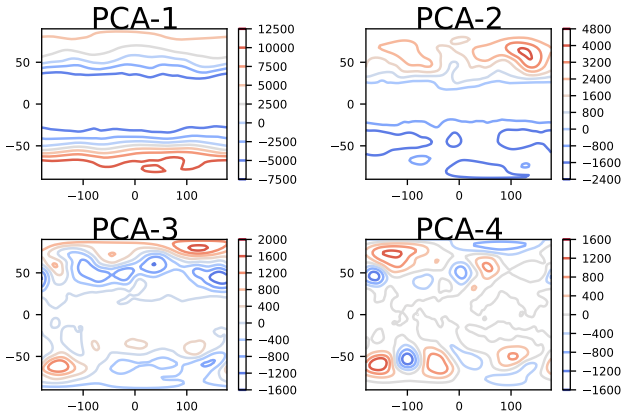

Figura 1

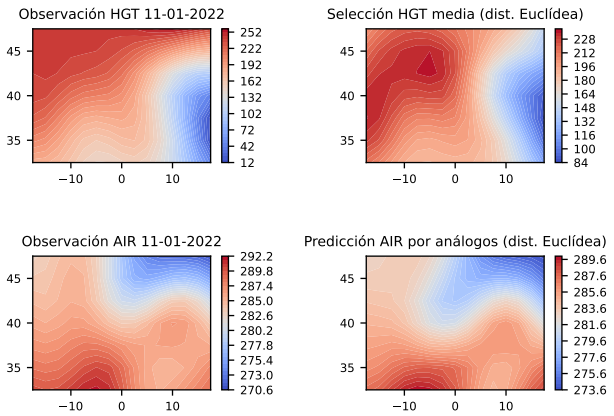

Figura 2

determinada por la latitud. Además, se observa un área amplia en torno al ecuador en la que Z es constante. En la segunda componente principal, se sigue observando un área en torno al ecuador con Z constante, aunque más reducida que en la anterior. El resto del diagrama es mucho más caótico. En las restantes dos componentes apenas se observa regularidad.

Como la varianza explicada de la primera componente principal es del 88.77 %, la altura geopotencial del sistema presenta un comportamiento mayoritariamente regular (el capturado por esta componente) y determinado con bastante precisión a partir de la latitud.

4.2. Apartado *ii*)

Los 4 días de 2021 más similares a $a_0 := 11/01/2022$, basándonos en la altura geopotencial, son:

- 23/03, que está a distancia 396.42
- 16/01, que está a distancia 467.18
- 12/01, que está a distancia 524.74
- 16/03, que está a distancia 525.95

A partir de estos días, se predice la temperatura del día a_0.

La figura 2 resume toda la información de la predicción. En la subgráfica superior izquierda se muestra el valor de Z del día a_0. En la superior derecha observamos la media de los valores de Z de los 4 días más similares a a_0. Como era de esperar, ambas subgráficas son similares. En la subgráfica inferior izquierda se muestra la verdadera temperatura de a_0, y en la inferior derecha la predicción que hemos realizado a partir de los análogos. Se observa que, de nuevo, ambas gráficas son muy similares, así que la predicción ha sido correcta.

El error absoluto medio en la predicción de la temperatura con $p = 1000$ es de 1.42 K. Este es un error bastante reducido, lo cual constituye otro indicador de que la predicción ha funcionado bien.

5. Conclusiones

En el apartado *i*), hemos visto como las primeras componentes principales presentan una regularidad mucho mayor que las últimas. Parece ser que estas capturan la parte más regular del sistema meteorológico. Un 88.77 % de varianza explicada es una cantidad elevada, de donde se deduce que la altura geopotencial es una variable que habitualmente se puede predecir con bastante precisión a partir de las variables de estado x, y, p. En particular, para la presión $p = 500$ hPa fijada, la mayoría de los datos tratados presentan altura geopotencial que se estima bien a partir de su latitud, como se observa en la gráfica PCA-1 de la figura 1.

En el apartado *ii*), hemos obtenido una predicción de la temperatura en a_0 muy cercana al dato real y un error absoluto medio reducido. Esto quiere decir que pequeñas variaciones en la altura geopotencial no conllevan grandes variaciones en la temperatura . Por tanto, el método de predicción utilizado, la predicción por análogos, es fiable para determinar la temperatura a partir de la altura geopotencial.

Apéndice A Código utilizado

```
"""
Práctica 4 de Geometría Computacional
Autor: Marcos Herrero
"""

import datetime as dt  # Python standard library datetime  module
import numpy as np
import matplotlib.pyplot as plt
from netCDF4 import Dataset
from sklearn.decomposition import PCA
from sklearn.neighbors import NearestNeighbors

"""
Cargamos T y Z de 2021 y 2022
"""

#Fichero de T (temperatura) en 2021
f = Dataset("air.2021.nc", "r", format="NETCDF4")

#Rangos :
# time mapea cada día del año 2021 a horas desde el 01/01/1800
# level mapea cada nivel de presión considerado al número de hPa que representa
# lats mapea cada nivel de latitud al número de grados respecto al ecuador (entre 90 y -9
# lons mapea cada nivel longitud(meridiano) al número de grados respecto al
# meridiano de Greenwich (entre 0 y 360)
time21 = f.variables['time'][:].copy()
level = f.variables['level'][:].copy()
lats = f.variables['lat'][:].copy()
lons = f.variables['lon'][:].copy()

#Temperatura en cada posición en Kelvin
air21 = f.variables['air'][:].copy()
f.close()

#Fichero de Z (altura) en 2021
f = Dataset("hgt.2021.nc", "r", format="NETCDF4")
hgt21 = f.variables['hgt'][:].copy()
f.close()

#Fichero de T en  2022 (level,lats y lons son los mismos que en 2021)
f = Dataset("air.2022.nc", "r", format="NETCDF4")
time22 = f.variables['time'][:].copy()

#Temperatura en cada posición en Kelvin
air22 = f.variables['air'][:].copy()
f.close()

#Fichero de Z (altura) en 2022
f = Dataset("hgt.2022.nc", "r", format="NETCDF4")
hgt22 = f.variables['hgt'][:].copy()
f.close()

'''
Vamos a modificar el rango de lons para que España no quede partida.
Necesitamos que lons vaya de -180 a 180, y para ello hemos de reorganizar los
datos de air21 y hgt21
'''

pos180 = np.argmax(lons >= 180)
```

```
air21 = np.roll(air21,axis = 3, shift = -pos180)
hgt21 = np.roll(hgt21,axis = 3, shift = -pos180)
air22 = np.roll(air22,axis = 3, shift = -pos180)
hgt22 = np.roll(hgt22,axis = 3, shift = -pos180)
lons = np.roll(lons, shift= -pos180)
lons[lons >= 180] -= 360

'''
Apartado i): Estimar las 4 componentes principales del sistema fijando
p = 500 hPa
'''

print('Apartado i):')
print()

n_components = 4

#Nos quedamos con los datos con p = 500 hPa
hgt21b = hgt21[:,level==500.,:,:].reshape(len(time21),len(lats)*len(lons))
air21b = air21[:,level==500.,:,:].reshape(len(time21),len(lats)*len(lons))

#Aplicaremos pca sobre la matriz traspuesta para reducir el número de elementos
# (días) y no el de variables de estado
X = hgt21b.transpose()
pca = PCA(n_components= n_components)


Element_pca = pca.fit_transform(X)
Element_pca = Element_pca.transpose(1,0).reshape(n_components,len(lats),len(lons))

print("Fracción de varianza explicada: {}".format(pca.explained_variance_ratio_))

fig = plt.figure()
fig.subplots_adjust(hspace=0.4, wspace=0.4)

for i in range(1, 5):
    ax = fig.add_subplot(2, 2, i)
    ax.text(0.5, 90, 'PCA-'+str(i),fontsize=18, ha='center')
    plt.contour(lons, lats, Element_pca[i-1,:,:],cmap = 'coolwarm')
    plt.colorbar()

fig = plt.gcf()
fig.savefig("contornosi.pdf",format='pdf')
plt.show()

print()
print()

'''
Apartado ii): consideramos el subsistema con x en (-20,20) e y en (30,50).
Buscar los 4 días de 2021 más análogos de a0 = 2022/01/11 considerando solo Z.
Calcular el error absoluto medio de la  temperatura (con p = 1000 hPa) prevista
para el elemento a0 segun la media de dichos análogos. Para la analogía,
consideramos la distancia euclídea con pesos 1 para x,y , 0,5 para las presiones
de 500 y 1000 hPa y 0 para el resto.
'''

print('Apartado ii)')
print()

#Buscamos el dia a0
```

```
a0 = dt.date(2022, 1, 11)
timea0 = (a0 - dt.date(1800,1,1)).total_seconds()/3600
aira0 = air22[time22 == timea0,:,:,:][0]
hgta0 = hgt22[time22 == timea0,:,:,:][0]

#Restringimos el sistema y el día a los rangos pedidos
condlats = np.logical_and(lats > 30,lats < 50)
condlons = np.logical_and(lons > -20, lons < 20)

airsigma = air21[:,:,condlats,:]
airsigma = airsigma[:,:,:,condlons]
hgtsigma = hgt21[:,:,condlats,:]
hgtsigma = hgtsigma[:,:,:,condlons]
aira0rest = aira0[:,condlats,:]
aira0rest = aira0rest[:,:,condlons]
hgta0rest = hgta0[:,condlats,:]
hgta0rest = hgta0rest[:,:,condlons]

latssigma = lats[condlats]
lonssigma = lons[condlons]

#Calculamos los 4 elementos más análogos a a0 en hgt2021
n_neighbours = 4

weights = np.zeros((len(level),len(latssigma),len(lonssigma)))
weights[level == 500,:,:] = 0.5
weights[level == 1000,:,:] = 0.5

neigh = NearestNeighbors(n_neighbors= n_neighbours, metric_params = {'w':weights.flatten(
neigh.fit(hgtsigma.reshape(len(time21),len(level)*len(latssigma)*len(lonssigma)))
distsa0, neighboursa0 = neigh.kneighbors([hgta0rest.flatten()])
distsa0 = distsa0[0]
neighboursa0 = neighboursa0[0]

print("Días más próximos a a0 := {}  :".format(a0))
print()
for i in range(n_neighbours):
    fecha = dt.date(1800, 1, 1) + dt.timedelta(hours= time21[neighboursa0[i]])
    print("Dia {}: {}; Distancia : {}".format(i,fecha,distsa0[i]))

print()
#Calculamos la media de los días más análogos

hgtmedio = np.zeros((len(level),len(latssigma),len(lonssigma)))
airmedio = np.zeros((len(level),len(latssigma),len(lonssigma)))
for i in range(n_neighbours):
    hgtmedio += hgtsigma[neighboursa0[i]]
    airmedio += airsigma[neighboursa0[i]]

hgtmedio /= n_neighbours
airmedio /= n_neighbours

#Calculamos el error absoluto medio de la temperatura

MAE = np.mean(abs(airmedio[level==1000][0] - aira0rest[level==1000][0]))
print("Error absoluto medio de T: {}".format(MAE))

#Dibujamos las gráficas
plt.rcParams.update({'font.size': 7})

fig = plt.figure()
fig.subplots_adjust(hspace=0.7, wspace=0.5)
```

```
ax = fig.add_subplot(2, 2, 1)
ax.title.set_text('Observación HGT {}-{}-{}'.format(str(a0.day).zfill(2),
                                                    str(a0.month).zfill(2),a0.year))

plt.contourf(lonssigma, latssigma, hgta0rest[level == 1000,:,:][0],
             levels = 40, cmap = 'coolwarm')
plt.colorbar()

ax = fig.add_subplot(2, 2, 2)
ax.title.set_text('Selección HGT media (dist. Euclídea)')
plt.contourf(lonssigma, latssigma,hgtmedio[level == 1000,:,:][0],
             levels = 40,cmap = 'coolwarm')
plt.colorbar()

ax = fig.add_subplot(2, 2, 3)
ax.title.set_text('Observación AIR {}-{}-{}'.format(str(a0.day).zfill(2),
                                                      str(a0.month).zfill(2),a0.year))
plt.contourf(lonssigma, latssigma,aira0rest[level == 1000,:,:][0],
             levels = 40, cmap = 'coolwarm')
plt.colorbar()

ax = fig.add_subplot(2, 2, 4)
ax.title.set_text('Predicción AIR por análogos (dist. Euclídea)')
plt.contourf(lonssigma, latssigma,airmedio[level == 1000,:,:][0],
             levels = 40, cmap = 'coolwarm')
plt.colorbar()

fig = plt.gcf()
fig.savefig("obsypredii.pdf",format='pdf')
plt.show()
```

Apéndice B Resultado de la ejecución

```
Apartado i):

Fracción de varianza explicada: [0.8877314  0.05177601 0.00543984 0.00357637]


Apartado ii)

Días más próximos a a0 := 2022-01-11  :

Dia 0: 2021-03-23; Distancia : 396.4154622993407
Dia 1: 2021-01-16; Distancia : 467.1849874514377
Dia 2: 2021-01-12; Distancia : 524.7436874322549
Dia 3: 2021-03-16; Distancia : 525.9499738568298

Error absoluto medio de T: 1.419346182686942
```