

3. Сучасні телекомунікації: Мережі, технології, безпека, економіка, регулювання. – Видання друге (доповнене). – /За загальною ред. Довгого С.О. – К.: «Азимут – Україна». – 2013. – 608 с.

Осипова О. І.

к.е.н.

*ДВНЗ «Київський національний економічний університет
імені Вадима Гетьмана», м. Київ*

Ткачук Д. М.

Міжнародна платіжна платформа Fondy.eu, м. Київ

ФОРМУВАННЯ МАСИВУ ВХІДНОЇ ІНФОРМАЦІЇ ДЛЯ ПОБУДОВИ ПРОСТОРОВОЇ МАТРИЦІ ВАГ ЗА ДОПОМОГОЮ ПРОГРАМНОГО КОДУ НА МОВІ PYTHON

Побудова просторової матриці ваг є обов'язковим етапом попереднього аналізу даних та економетричного моделювання в просторовій економетриці. Дана матриця призначена для врахування взаємного просторового впливу досліджуваних об'єктів та являє собою квадратну симетричну матрицю розміру $N \times N$, кожний елемент якої (w_{ij}) характеризує міру близькості об'єктів i та j у просторі ($i = \overline{1, N}$; $j = \overline{1, N}$). Для виключення впливу об'єкту самого на себе діагональні елементи матриці дорівнюють нулю [1].

На сьогодні розроблено достатньо програмних модулів для роботи з просторовими економетричними моделями, у тому числі в яких реалізовано низку підходів до побудови просторової матриці ваг (набір інструментів для просторового економетричного аналізу в Stata, Matlab, R, бібліотека PySal в Python). Водночас дослідники часто стикаються з проблемою збору вхідної інформації для побудови такої матриці, адже цей процес може бути довгим та трудомістким: необхідно сформувати матрицю, в якій кожний елемент (за виключенням діагональних елементів) дорівнює відстані між i -тим та j -тим об'єктом. Тому автоматизація процесу збору вхідної інформації для побудови просторової матриці ваг дозволить прискорити та дещо спростити проведення просторового економетричного аналізу даних.

Розглянемо приклад формування масиву вхідної інформації для побудови одного із найпоширеніших видів просторових вагових матриць – матриці зворотних відстаней, в якій недиагональні елементи розраховуються за таким правилом: $w_{ij} = 1 / d_{ij}$, де d_{ij} – відстань між об'єктами i та j у просторі. В нашому прикладі у якості вхідної інформації для побудови просторової матриці ваг ми обрали відстані між 28 столицями країн-учасниць ЄС. Початковою інформацією для розрахунку відстаней між обраними місцями є географічні координати цих міст (географічна широта та довгота). Інформація про географічні координати більшості населених пунктів світу знаходиться у відкритому доступі в мережі Internet. Наприклад, ми завантажили файл формату json, в якому містяться назви та географічні координати всіх столиць з бази даних <http://techslides.com> (файл завантажено та збережено під назвою country-capitals.json).

Програмний код, написаний на мові Python 3.6, має такий вигляд:

1. завантажуюємо потрібні для написання програмного коду бібліотеки:

```
import json
import csv
import math
```

2. функція `def read_data ()` зчитує дані із завантаженого файлу `country-capitals.json` та повертає словник з назвами столиць та їх координатами:

```
def read_data(countries):
    data = json.load(open('country-capitals.json'))
    target_dict = {}
    for i in data:
        if i['ContinentName'] in countries:
            target_dict[i['CapitalName']] = [float(i['CapitalLatitude']),
            float(i['CapitalLongitude'])]
    return target_dict
```

3. функція `def get_distance()` приймає 2 кортежі з координатами різних міст та за формулою гаверсинусів визначає відстань між ними [2]:

```
def get_distance(city1, city2):
    x1, y1 = map(math.radians, city1)
```

```

x2, y2 = map(math.radians, city2)
d_x = x2 - x1
d_y = y2 - y1
d_sigma = 2 * math.asin(math.sqrt(math.cos(x1) * math.cos(x2) *
math.sin(d_y / 2) ** 2 + math.sin(d_x / 2) ** 2))
distance = 63718 * d_sigma
return distance

```

4. обробляємо дані, робимо розрахунки та зберігаємо до csv-файлу:

```

def save_to_file(countries):
    capitals = read_data(countries)
    with open('distances.csv', 'w', encoding='utf-8', newline='') as
csvfile:
        file_writer = csv.writer(csvfile)
        file_writer.writerow([''] + sorted(list(capitals)))
        for i in sorted(capitals):
            file_writer.writerow([i] + [round(get_distance(capitals[i],
capitals[j]), 1) for j in sorted(capitals)])
    if __name__ == '__main__':
        save_to_file(['Europe'])

```

Отриманий в результаті реалізації програмного коду csv-файл містить інформацію у вигляді матриці розміру 28×28 про відстані в км між столицями країн-учасниць ЄС та в подальшому може використовуватись для побудови просторової матриці ваг. Незначне коригування програмного коду дозволяє змінювати перелік населених пунктів, для яких необхідно побудувати просторову матрицю ваг.

Список використаних джерел

1. Introduction to spatial regression in Python [Електронний ресурс]. – Режим доступу: <https://www.earthdatascience.org/tutorials/intro-to-spatial-regression/> (дата звернення 07.09.2018).
2. Вычисление расстояния и начального азимута между двумя точками на сфере [Електронний ресурс]. – Режим доступу: <http://gis-lab.info/qa/great-circles.html> (дата звернення 07.09.2018).

⁸ Радіус Землі – 6371 км.