

BLM6105-2 Uzaktan Algılama Dersi Ödev 7

24501100 - Aleyna Nil Uzunoğlu

Giriş

Bu çalışmada, uzaktan algılama verilerinin analizinde temel bir adım olan veri matrisleri üzerinde istatistiksel analizler ve benzerlik/uzaklık ölçümleri incelenmiştir.

Python programlama dili kullanılarak, 10 örnek (sample) ve 5 özellikten (attribute) oluşan bir veri matrisi rastgele değerlerle oluşturulmuştur. (Bu matris konsolda görülebilir) Her bir özelliğin farklı sayısal aralıklarda değer alması sağlanmıştır. Özelliklerin sayısal aralıkları kod üzerinde görülebilir.

Ödev kapsamında aşağıdaki adımlar gerçekleştirilmiştir:

1. Her bir öznenin (özellik) ortalama, standart sapma ve varyans değerleri kütüphane kullanmadan ve NumPy kütüphanesi fonksiyonları kullanılarak hesaplanmış ve doğrulanmıştır.
2. Veri matrisinin kovaryans matrisi hesaplanmıştır.
3. Belirlenen üç farklı örnek çifti için Euclidean, Cosine, Manhattan ve Mahalanobis uzaklıkları hem formüllerle hem de ilgili kütüphane fonksiyonları (NumPy, SciPy) yardımıyla hesaplanarak karşılaştırılmıştır.

Adımlar

Bu ödev aşağıdaki ana adımları içermektedir:

(0) Veri Matrisinin Oluşturulması

- 10 örnek ve 5 özellikten oluşan bir NumPy dizisi oluşturulmuştur. Her bir özellik (sütun) için aşağıdaki gibi farklı sayı aralıkları belirlenmiş ve bu aralıklarda rastgele sayılar üretilmiştir:
- Özellik 1: 0-1 aralığı, Özellik 2: 1-10 aralığı, Özellik 3: 50-150 aralığı, Özellik 4: -5 ile 5 aralığı, Özellik 5: 1000-2000 aralığı

(1) Öznitelik İstatistiklerinin Hesaplanması

- Veri matrisindeki her bir özellik (sütun) için ayrı ayrı:
 - Ortalama değer, standart sapma değeri, varyans değeri hesaplanmıştır.
- Bu hesaplamalar hem temel matematiksel formüllerin NumPy operasyonları ile uygulanmasıyla hem de NumPy'nin mean(), std(), var() fonksiyonları kullanılarak doğrulanmıştır.

(2) Kovaryans Matrisinin Hesaplanması

- Oluşturulan veri matrisinin özellikler arasındaki ilişkiyi gösteren kovaryans matrisi, NumPy'nin cov() fonksiyonu kullanılarak hesaplanmıştır.

(3) Uzaklık Metriklerinin Hesaplanması

- Veri matrisinden rastgele seçilen (veya sabit olarak belirlenen, örneğin: (0,1), (3,7), (5,9) numaralı örnekler) üç örnek çifti için aşağıdaki uzaklık metrikleri hesaplanmıştır:

- **Euclidean Uzaklığı:** İki vektör arasındaki geometrik düz çizgi mesafesi.
- **Cosine Uzaklığı:** İki vektör arasındaki açının kosinüsünden türetilen, yön benzerliğini ölçen bir metrik (1 - Cosine Benzerliği).
- **Manhattan (City Block) Uzaklığı:** İki vektör arasındaki farkların mutlak değerlerinin toplamı.
- **Mahalanobis Uzaklığı:** Özellikler arasındaki korelasyonu ve farklı ölçekleri dikkate alan, kovaryans matrisinin tersini kullanan bir uzaklık ölçüsü.
- Tüm uzaklık hesaplamaları, hem temel formüllerin NumPy operasyonları ile uygulanmasıyla "elle" hem de scipy.spatial.distance modülündeki (veya np.linalg.norm gibi) ilgili fonksiyonlar kullanılarak doğrulanmıştır.

Ek-1: Program Çıktısı

OLUŞTURULAN VERİ MATRİSİ

```
[[ 0.36  5.048 98.963  4.435 1868.535], [ 0.863  8.864 135.258  4.374 1912.608], [ 0.898  8.372 126.186  4.598 1578.899]
 [ 0.98  3.653 66.224  1.241 1974.042], [ 0.965  1.248 62.458  2.804 1073.235], [ 0.612  8.012  85.166  0.973 1790.892]
 [ 0.92  6.929 107.688 -3.103 1760.434], [ 0.981  1.527 134.716  1.139 1836.679], [ 0.585  7.228 131.064 -3.456 1057.262]
 [ 0.011  2.8  53.321  1.428 1854.88 ]]
```

1) ÖZNİTELİK İSTATİSTİKLERİ HESAPLANIYOR

Özellik 1 (Aralık: 0-1):

Hesaplanan Ortalama: 0.718 (NumPy: 0.718), Hesaplanan Std. Sapma: 0.308 (NumPy: 0.308), Hesaplanan Varyans: 0.095 (NumPy: 0.095)

Özellik 2 (Aralık: 1-10):

Hesaplanan Ortalama: 5.368 (NumPy: 5.368), Hesaplanan Std. Sapma: 2.748 (NumPy: 2.748), Hesaplanan Varyans: 7.549 (NumPy: 7.549)

Özellik 3 (Aralık: 50-150):

Hesaplanan Ortalama: 100.104 (NumPy: 100.104), Hesaplanan Std. Sapma: 30.180 (NumPy: 30.180), Hesaplanan Varyans: 910.839 (NumPy: 910.839)

Özellik 4 (Aralık: -5-5):

Hesaplanan Ortalama: 1.443 (NumPy: 1.443), Hesaplanan Std. Sapma: 2.727 (NumPy: 2.727), Hesaplanan Varyans: 7.437 (NumPy: 7.437)

Özellik 5 (Aralık: 1000-2000):

Hesaplanan Ortalama: 1670.747 (NumPy: 1670.747), Hesaplanan Std. Sapma: 318.644 (NumPy: 318.644), Hesaplanan Varyans: 101534.122 (NumPy: 101534.122)

--- 2) KOVARYANS MATRİSİ ---

```
[[ 0.105  0.032  3.69 -0.009 -13.381]
 [ 0.032  8.388 47.991 -0.304  72.661]
 [ 3.69  47.991 1012.044 -5.712 -542.202]
 [-0.009 -0.304 -5.712  8.264  291.662]
 [-13.381 72.661 -542.202 291.662 112815.691]]
```

3) UZAKLIK HESAPLAMALARI

--- Örnek Çifti 1 (Örnek 1 ve Örnek 2) ---

Euclidean Uzaklığı (Hesaplanan): 57.223

Euclidean Uzaklığı (NumPy linalg.norm): 57.223

Euclidean Uzaklığı (SciPy): 57.223

Cosine Uzaklığı (1 - Benzerlik_Hesaplanan): 0.000

Cosine Uzaklığı (SciPy): 0.000

Manhattan Uzaklığı (Hesaplanan): 84.747

Manhattan Uzaklığı (NumPy sum(abs)): 84.747

Manhattan Uzaklığı (SciPy cityblock): 84.747

Mahalanobis Uzaklığı (Hesaplanan): 2.013

Mahalanobis Uzaklığı (SciPy): 2.013

--- Örnek Çifti 2 (Örnek 4 ve Örnek 8) ---

Euclidean Uzaklığı (Hesaplanan): 153.507

Euclidean Uzaklığı (NumPy linalg.norm): 153.507

Euclidean Uzaklığı (SciPy): 153.507

Cosine Uzaklığı (1 - Benzerlik_Hesaplanan): 0.001

Cosine Uzaklığı (SciPy): 0.001

Manhattan Uzaklığı (Hesaplanan): 208.084

Manhattan Uzaklığı (NumPy sum(abs)): 208.084

Manhattan Uzaklığı (SciPy cityblock): 208.084

Mahalanobis Uzaklığı (Hesaplanan): 3.315

Mahalanobis Uzaklığı (SciPy): 3.315

--- Örnek Çifti 3 (Örnek 6 ve Örnek 10) ---

Euclidean Uzaklığı (Hesaplanan): 71.667

Euclidean Uzaklığı (NumPy linalg.norm): 71.667

Euclidean Uzaklığı (SciPy): 71.667

Cosine Uzaklığı (1 - Benzerlik_Hesaplanan): 0.000

Cosine Uzaklığı (SciPy): 0.000

Manhattan Uzaklığı (Hesaplanan): 102.100

Manhattan Uzaklığı (NumPy sum(abs)): 102.100

Manhattan Uzaklığı (SciPy cityblock): 102.100

Mahalanobis Uzaklığı (Hesaplanan): 2.635

Mahalanobis Uzaklığı (SciPy): 2.635

Ek-2 : main.py

```
import numpy as np
import random
import logging
from scipy.spatial import distance # ÖZELLİK HESAPLARI doğrulaması için
logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s - %(message)s')
logging.info("7.Odev - Veri Matrisinin Oluşturulması ve Uzaklık Hesaplamaları")

ornek_sayisi = 10
ozellik_sayisi = 5
ozellik_aralıkları = [ (0, 1), (1, 10), (50, 150), (-5, 5), (1000, 2000) ]
veri_matrisi = np.zeros((ornek_sayisi, ozellik_sayisi))
for j in range(ozellik_sayisi):
    min_val, max_val = ozellik_aralıkları[j]
    for i in range(ornek_sayisi):
        veri_matrisi[i, j] = random.uniform(min_val, max_val)
print("OLUŞTURULAN VERİ MATRİSİ")
np.set_printoptions(precision=3, suppress=True)
print(veri_matrisi)
print("-" * 40)

# 1) Her bir özneliğin ortalama, standart sapma ve varyans değerleri hesaplanması
print("\n 1) ÖZNİTELİK İSTATİSTİKLERİ HESAPLANIYOR")
for j in range(ozellik_sayisi):
    ozellik_verisi = veri_matrisi[:, j]
    ortalama_hesaplanan = np.sum(ozellik_verisi) / ornek_sayisi
    ortalama_numpy = np.mean(ozellik_verisi)
    varyans_hesaplanan = np.sum((ozellik_verisi - ortalama_hesaplanan)**2) / ornek_sayisi
    std_sapma_hesaplanan = np.sqrt(varyans_hesaplanan)
    std_sapma_numpy = np.std(ozellik_verisi)
```

```
varyans_numpy = np.var(ozellik_verisi)
```

```
print(f"\nÖzellik {j+1} (Aralık: {ozellik_araliklari[j][0]}-{ozellik_araliklari[j][1]}):")
```

```
print(f" Hesaplanan Ortalama: {ortalama_hesaplanan:.3f} (NumPy: {ortalama_numpy:.3f})")
```

```
print(f" Hesaplanan Std. Sapma: {std_sapma_hesaplanan:.3f} (NumPy: {std_sapma_numpy:.3f})")
```

```
print(f" Hesaplanan Varyans: {varyans_hesaplanan:.3f} (NumPy: {varyans_numpy:.3f})")
```

```
print("-" * 40)
```

```
print("\n--- 2) KOVARYANS MATRİSİ ---")
```

```
kovaryans_matrisi = np.cov(veri_matrisi, rowvar=False)
```

```
print(kovaryans_matrisi)
```

```
print("-" * 40)
```

```
print("\n 3) UZAKLIK HESAPLAMALARI")
```

```
ornek_indeksleri = list(range(ornek_sayisi))
```

```
random.shuffle(ornek_indeksleri)
```

```
if ornek_sayisi >= 6:
```

```
    secilen_ciftler_indeks = [  
        (ornek_indeksleri[0], ornek_indeksleri[1]),  
        (ornek_indeksleri[2], ornek_indeksleri[3]),  
        (ornek_indeksleri[4], ornek_indeksleri[5])  
    ]
```

```
else:
```

```
    secilen_ciftler_indeks = [(0,1), (1,2), (0,2)]
```

```
    if ornek_sayisi < 3:
```

```
        secilen_ciftler_indeks = [(0,1)]
```

```
secilen_ciftler_indeks = [ (0, 1),(3, 7),(5, 9)]
```

```
for i, cift in enumerate(secilen_ciftler_indeks):
```

```
    ornek1_idx, ornek2_idx = cift
```

```
    vektor1 = veri_matrisi[ornek1_idx, :]
```

```
    vektor2 = veri_matrisi[ornek2_idx, :]
```

```

print(f"\n--- Örnek Çifti {i+1} (Örnek {ornek1_idx+1} ve Örnek {ornek2_idx+1}) ---")

# a) Euclidean Uzaklığı

fark_kare_toplam = 0

for k in range(ozellik_sayisi):

    fark_kare_toplam += (vektor1[k] - vektor2[k])**2

euclidean_hesaplanan = np.sqrt(fark_kare_toplam)

euclidean_numpy = np.linalg.norm(vektor1 - vektor2)

euclidean_scipy = distance.euclidean(vektor1, vektor2)

print(f" Euclidean Uzaklığı (Hesaplanan): {euclidean_hesaplanan:.3f}")

print(f" Euclidean Uzaklığı (NumPy linalg.norm): {euclidean_numpy:.3f}")

print(f" Euclidean Uzaklığı (SciPy): {euclidean_scipy:.3f}")

# b) Cosine Uzaklığı (1 - Cosine Benzerliği)

dot_product = np.dot(vektor1, vektor2)

norm_vektor1 = np.linalg.norm(vektor1)

norm_vektor2 = np.linalg.norm(vektor2)

if norm_vektor1 == 0 or norm_vektor2 == 0:

    cosine_benzerligi_hesaplanan = 0 if dot_product == 0 else 1

else:

    cosine_benzerligi_hesaplanan = dot_product / (norm_vektor1 * norm_vektor2)

cosine_uzakligi_hesaplanan = 1 - cosine_benzerligi_hesaplanan

# SciPy ile (doğrulama):

cosine_scipy = distance.cosine(vektor1, vektor2) # Bu direkt uzaklığı verir

print(f" Cosine Uzaklığı (1 - Benzerlik_Hesaplanan): {cosine_uzakligi_hesaplanan:.3f}")

print(f" Cosine Uzaklığı (SciPy): {cosine_scipy:.3f}")

# c) Manhattan (City Block) Uzaklığı

mutlak_fark_toplam = 0

for k in range(ozellik_sayisi):

    mutlak_fark_toplam += abs(vektor1[k] - vektor2[k])

manhattan_hesaplanan = mutlak_fark_toplam

manhattan_numpy = np.sum(np.abs(vektor1 - vektor2))

```

```

manhattan_scipy = distance.cityblock(vektor1, vektor2)

print(f" Manhattan Uzaklığı (Hesaplanan): {manhattan_hesaplanan:.3f}")
print(f" Manhattan Uzaklığı (NumPy sum(abs)): {manhattan_numpy:.3f}")
print(f" Manhattan Uzaklığı (SciPy cityblock): {manhattan_scipy:.3f}")

# d) Mahalanobis Uzaklığı

try:
    ters_kovaryans_matrisi = np.linalg.inv(kovaryans_matrisi)

    fark_vektoru = vektor1 - vektor2

    mahalanobis_hesaplanan = np.sqrt(np.dot(np.dot(fark_vektoru.T, ters_kovaryans_matrisi),
fark_vektoru))

    mahalanobis_scipy = distance.mahalanobis(vektor1, vektor2, ters_kovaryans_matrisi)

    print(f" Mahalanobis Uzaklığı (Hesaplanan): {mahalanobis_hesaplanan:.3f}")

    print(f" Mahalanobis Uzaklığı (SciPy): {mahalanobis_scipy:.3f}")

except np.linalg.LinAlgError:

    print(" Mahalanobis Uzaklığı: Kovaryans matrisi tekil (singular), tersi alınamıyor.")

print("-" * 40)

```