

Математички методи за машинско учење 2023

Домаћи задатак број 6

```
import numpy as np
import numpy.random as rndm
import matplotlib as mplb
import matplotlib.pyplot as plt
```

Задатак 1. Одредити спектрални полупречник, Фробенијусову норму, норму трага и L_1, L_2 индуковане матричне норме за матрицу A генерисану доњим кодом. Одредити најмању по модулу сингуларну вредност матрице A . Да ли је матрица A регуларна?

(10 поена)

```
import numpy as np
import numpy.linalg as la
import numpy.random as rndm

n = rndm.randint(231, 345)
A = rndm.rand(n, n)

# Spektralni poluprečnik
eigenvalues = la.eigvals(A)
spectral_radius = np.max(np.abs(eigenvalues))

# Frobeniusova norma
frobenius_norm = la.norm(A, 'fro')

# Norma traga
trace_norm = la.norm(A, 'nuc')

# L1 indukovana matricna norma
l1_norm = la.norm(A, 1)

# L2 indukovana matricna norma
l2_norm = la.norm(A, 2)

# Najmanja po modulu singularna vrednost matrice A
singular_values = la.svd(A, compute_uv=False)
min_singular_value = np.min(singular_values)

# Provera regularnosti matrice A
is_regular = la.matrix_rank(A) == n

print(f"Spectral radius: {spectral_radius}")
print(f"Frobenius norm: {frobenius_norm}")
```

```

print(f"Trace norm: {trace_norm}")
print(f"L1 induced matrix norm: {l1_norm}")
print(f"L2 induced matrix norm: {l2_norm}")
print(f"Smallest singular value: {min_singular_value}")
print(f"Is matrix A regular? {is_regular}")

```

```

Spectral radius: 142.80464387357102
Frobenius norm: 164.77950711780906
Trace norm: 1313.6233405720573
L1 induced matrix norm: 158.13033331368314
L2 induced matrix norm: 142.98246825903917
Smallest singular value: 0.0048722523106274505
Is matrix A regular? True

```

Задатак 2. У колико значајних цифара можемо имати поверења у поступку решавања система линеарних једначина $Ax=b$, за матрицу A генерисану доњим кодом.

(10 поена)

```

import numpy as np
import numpy.random as rndm

n = rndm.randint(17, 35)
A = rndm.rand(n, n)

kondicioni_broj = np.linalg.cond(A)
print("Kondicioni broj matrice A je:", kondicioni_broj)

broj_znacajnih_cifara = abs(-np.log10(kondicioni_broj))
print("Broj značajnih cifara u kojima možemo imati poverenja je:",
broj_znacajnih_cifara)

```

```

Kondicioni broj matrice A je: 1437.7043752292348
Broj značajnih cifara u kojima možemo imati poverenja je:
3.1576695943879467

```

Задатак 3. Самостално истражити и описати улогу SVD у поступку Latent Semantic Analysis у процесирању природних језика. Направити мали пример на коме се то показује.

(30 поена)

Latent Semantic Analysis (LSA) je tehnika za analizu teksta koja se koristi u obradi prirodnih jezika (NLP) kako bi se identifikovale skrivene teme i semantičke veze između dokumenata i reči. Singular Value Decomposition (SVD) je ključni matematički postupak u LSA koji omogućava redukciju dimenzionalnosti i otkrivanje skrivenih semantičkih struktura.

U postupku LSA, koraci su sledeći:

Kreirati matricu termin-dokument. Ova matrica predstavlja broj pojavljivanja reči (termina) u svakom dokumentu. Redovi matrice predstavljaju reči, a kolone predstavljaju dokumente. Element matrice a_{ij} predstavlja broj pojavljivanja reči i u dokumentu j .

Primeniti SVD na matricu termin-dokument. Ako je matrica M , SVD dekompozicija izgleda ovako: $M=U \Sigma V^T$, gde su U i V ortogonalne matrice, a Σ je dijagonalna matrica singularnih vrednosti.

Redukcija dimenzionalnosti: Odabrati k najvećih singularnih vrednosti iz matrice Σ i zadržati samo prvih k kolona matrica U i V . Ovo će rezultirati aproksimacijom matrice M ranga k .

Tumačenje rezultata: Redukovane matrice U i V sadrže informacije o "latentnim" temama koje povezuju reči i dokumente. Možemo izračunati sličnost između reči ili dokumenta koristeći kosinusnu sličnost između redova ili kolona ovih matrica.

```
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import TruncatedSVD

# Primer dokumenata
documents = [
    "Ovo je prvi dokument koji govori o mačkama.",
    "Drugi dokument je o psima i mačkama.",
    "Ovde je treći dokument o pticama.",
    "Četvrti dokument sadrži informacije o psima i pticama."
]

# Kreiranje matrice termin-dokument
vectorizer = CountVectorizer()
td_matrix = vectorizer.fit_transform(documents).toarray()
print("Matrica termin-dokument:\n", td_matrix)

# Primeniti SVD na matricu termin-dokument
k = 2
svd = TruncatedSVD(n_components=k)
svd.fit(td_matrix)

# Redukovane matrice U i V
U = svd.transform(td_matrix)
V = svd.components_

print("Redukovana matrica U:\n", U)
print("Redukovana matrica V:\n", V)

from sklearn.metrics.pairwise import cosine_similarity

# Izračunavanje kosinusne sličnosti između dokumenata
doc_sim = cosine_similarity(U)
```

```
# Prikazivanje sličnosti između dokumenata u procentima
num_docs = len(documents)
for i in range(num_docs):
    for j in range(i + 1, num_docs):
        similarity_percentage = round(doc_sim[i, j] * 100, 2)
        print(f"Sličnost između dokumenta {i+1} i dokumenta {j+1}:
{similarity_percentage}%")
```

Matrica termin-dokument:

```
[[1 0 1 0 1 1 1 0 1 1 0 0 0 0 0]
 [1 1 0 0 1 0 1 0 0 0 1 0 0 0 0]
 [1 0 0 0 1 0 0 1 0 0 0 1 0 1 0]
 [1 0 0 1 0 0 0 0 0 0 1 1 1 0 1]]
```

Redukovana matrica U:

```
[[ 2.03546437 -1.49144948]
 [ 1.77257096 -0.14153452]
 [ 1.54271707  0.45404959]
 [ 1.47999042  1.74744522]]
```

Redukovana matrica V:

```
[[ 0.576168      0.14951502  0.17168988  0.12483608  0.45133193
 0.17168988
 0.32120491  0.13012702  0.17168988  0.17168988  0.2743511
 0.2549631
 0.12483608  0.13012702  0.12483608]
 [ 0.10328712 -0.02571401 -0.27096673  0.31747607 -0.21418895 -
 0.27096673
 -0.29668074  0.08249179 -0.27096673 -0.27096673  0.29176206
 0.39996785
 0.31747607  0.08249179  0.31747607]]
```

Sličnost između dokumenta 1 i dokumenta 2: 85.11%

Sličnost između dokumenta 1 i dokumenta 3: 60.69%

Sličnost između dokumenta 1 i dokumenta 4: 7.03%

Sličnost između dokumenta 2 i dokumenta 3: 93.38%

Sličnost između dokumenta 2 i dokumenta 4: 58.35%

Sličnost između dokumenta 3 i dokumenta 4: 83.55%