

1. Jelaskan cara kerja dari algoritma tersebut!

Jawab:

Principal Component Analysis (PCA) adalah teknik reduksi dimensi yang digunakan untuk mengubah dataset berdimensi tinggi menjadi dimensi yang lebih rendah, sambil mempertahankan informasi yang penting dalam data. PCA bekerja dengan menemukan principal components, yang merupakan vektor dengan variansi maksimum di data. Langkah-langkahnya adalah sebagai berikut:

- Standarisasi dataset sehingga setiap fitur memiliki mean 0 dan variansi 1 karena PCA sensitif terhadap skala fitur.
- Hitung matriks kovarian dari data. Matriks ini menunjukkan bagaimana fitur saling berkorelasi.
- Cari eigenvalues dan eigenvectors dari matriks kovarian. Eigenvector menunjukkan arah dari principal components, sementara eigenvalue menunjukkan variansi yang dijelaskan oleh setiap komponen.
- Urutkan eigenvectors berdasarkan eigenvalue-nya. Semakin besar eigenvalue, semakin banyak variansi yang dijelaskan oleh principal component yang terkait.
- Transformasi data asli ke ruang dimensi yang baru menggunakan principal components. Jumlah principal components yang dipilih menentukan dimensi hasil akhir dari data yang direduksi.
- PCA menghitung explained variance dari setiap komponen, yaitu persentase variansi oleh setiap principal component.

4. Bandingkan hasil evaluasi model pada nomor 2 dan 3, bagaimana hasil perbandingannya? Jika ada perbedaan, jelaskan alasannya!

Jawab:

Menggunakan Implementasi PCA dari Scratch

```
import pca
importlib.reload(pca)

from pca import PCAFromScratch

pca_scratch = PCAFromScratch(n_components=2)
pca_scratch.fit(X)

# Data hasil transformasi
X_transformed = pca_scratch.transform(X)
print("Data hasil PCA (dari scratch):\n", X_transformed)

# Explained variance
print("\nExplained Variance (dari scratch):\n", pca_scratch.explained_variance_)
✓ 0.0s

Data hasil PCA (dari scratch):
[[ 3.18125490e+02 -2.44852708e+00]
 [ 3.03140430e+02 -6.99307461e+00]
 [ 4.38107229e+02  9.77670011e-01]
 [ 7.33118887e+02  9.55551741e-01]
 [-1.19006212e+01  1.17915145e+00]
 [ 7.03123140e+02 -9.42105762e-01]
 [ 5.43125043e+02 -2.56831955e+00]
 [ 5.48109437e+02  2.74031947e-01]
 [ 2.98132324e+02 -4.10094147e+00]
 [ 2.98126877e+02 -1.72386738e+00]
 [ 7.63106494e+02  1.66229724e+00]
 [ 5.33113087e+02 -5.92914547e-01]
 [ 5.73117396e+02 -1.01540910e+00]
 [ 4.03143810e+02 -6.09156160e+00]
 [ 8.00139240e+02 -3.39471318e+00]
 [ 5.63115754e+02  5.88425711e-01]
 [ 5.33101692e+02  2.83898792e+00]
 [ 3.83104435e+02  2.32055391e+00]
 [ 9.33117960e+02  1.72182824e+00]
 [ 9.81268924e+01 -3.62188460e+00]
 [ 3.31279522e+01 -3.18736565e+00]
 [ 2.31092057e+01 -8.04570847e-01]
 [ 2.88115564e+02 -2.13393113e+00]
 [ 2.68108834e+02 -1.04233862e+00]
 ...
 [-1.86904420e+02  5.53553647e+00]]

Explained Variance (dari scratch):
[numpy.float64(0.9998327515536855), numpy.float64(9.5990139649115e-05)]
```

Menggunakan Implementasi PCA dari Scikit-Learn

```
pca_sklearn = PCA(n_components=2)
X_transformed_sklearn = pca_sklearn.fit_transform(X)

# Data hasil transformasi
print("Data hasil PCA (Scikit-learn):\n", X_transformed_sklearn)

# Explained variance
print("\nExplained Variance (Scikit-learn):\n", pca_sklearn.explained_variance_ratio_)
✓ 0.0s

Data hasil PCA (Scikit-learn):
[[ 3.18125490e+02 -2.44852708e+00]
 [ 3.03140430e+02 -6.99307461e+00]
 [ 4.38107229e+02  9.77670011e-01]
 [ 7.33118887e+02  9.55551741e-01]
 [-1.19006212e+01  1.17915145e+00]
 [ 7.03123140e+02 -9.42105762e-01]
 [ 5.43125043e+02 -2.56831955e+00]
 [ 5.48109437e+02  2.74031947e-01]
 [ 2.98132324e+02 -4.10094147e+00]
 [ 2.98126877e+02 -1.72386738e+00]
 [ 7.63106494e+02  1.66229724e+00]
 [ 5.33113087e+02 -5.92914547e-01]
 [ 5.73117396e+02 -1.01540910e+00]
 [ 4.03143810e+02 -6.09156160e+00]
 [ 8.00139240e+02 -3.39471318e+00]
 [ 5.63115754e+02  5.88425711e-01]
 [ 5.33101692e+02  2.83898792e+00]
 [ 3.83104435e+02  2.32055391e+00]
 [ 9.33117960e+02  1.72182824e+00]
 [ 9.81268924e+01 -3.62188460e+00]
 [ 3.31279522e+01 -3.18736565e+00]
 [ 2.31092057e+01 -8.04570847e-01]
 [ 2.88115564e+02 -2.13393113e+00]
 [ 2.68108834e+02 -1.04233862e+00]
 ...
 [-1.86904420e+02  5.53553647e+00]]

Explained Variance (Scikit-learn):
[9.99832752e-01 9.59901396e-05]
```

Tidak ada perbedaan dalam hasil evaluasi antara implementasi dari scratch dan dari Scikit-Learn.