Praktikum 6
Principal Component Analysis

A. Principal Component Analysis

Principal Component Analysis adalah teknik untuk ekstraksi fitur - jadi ini menggabungkan variabel masukan kita dengan cara tertentu, lalu kita dapat menghilangkan variabel "paling tidak penting" sambil tetap mempertahankan bagian paling berharga dari semua variabel! Sebagai manfaat tambahan, masing-masing variabel "baru" setelah PCA semuanya tidak bergantung satu sama lain. Ini menguntungkan karena asumsi model linier mengharuskan variabel independen kami independen satu sama lain. Jika kita memutuskan untuk menyesuaikan model regresi linier dengan variabel "baru" ini (lihat "regresi komponen utama" di bawah), asumsi ini akan terpenuhi. PCA dapat digunakan saat:

1. Ketika Anda ingin mengurangi jumlah variabel, tetapi tidak dapat mengidentifikasi variabel untuk sepenuhnya dihapus,
2. Ketika Anda ingin memastikan variabel Anda tidak bergantung satu sama lain
3. Ketika variabel independen Anda kurang dapat ditafsirkan

B. Cara Kerja PCA

1. Hitung matriks yang merangkum bagaimana semua variabel kita berhubungan satu sama lain.
2. Pecah matriks ini menjadi dua komponen terpisah: arah dan besaran.
3. Ubah data asli agar selaras dengan petunjuk ini

C. Algoritma Metode Principal Component Analysis

Sebelum memulai, Anda harus mengatur data tabel dengan n baris dan kemungkinan p+1 kolom, di mana satu kolom sesuai dengan variabel dependen Anda (biasanya dilambangkan Y) dan kolom p di mana masing-masing sesuai dengan variabel independen (matriks yang biasanya dilambangkan X).

1. Jika variabel Y ada dan merupakan bagian dari data Anda, maka pisahkan data Anda menjadi Y dan X, seperti yang didefinisikan di atas. (Catatan: jika tidak ada kolom untuk Y, tidak apa-apa - lompat ke poin selanjutnya!)

2. Ambil matriks variabel bebas X dan, untuk setiap kolom, kurangi rata-rata kolom tersebut dari setiap entri. (Ini memastikan bahwa setiap kolom memiliki rata-rata nol.)
3. Putuskan apakah akan melakukan standarisasi atau tidak. Panggil matriks Z yang berpusat (dan mungkin terstandarisasi).
4. Ambil matriks Z, ubah urutannya, dan kalikan matriks yang ditransposekan dengan Z. (secara matematis, kita akan menuliskannya sebagai Z^TZ.) Matriks yang dihasilkan adalah matriks kovariansi Z, hingga sebuah konstanta.
5. Hitung vektor eigen dan nilai eigen yang sesuai dari Z^TZ.
6. Ambil nilai eigen λ_1, λ_2,..., λp dan urutkan dari yang terbesar ke terkecil. Dengan demikian, urutkan vektor eigen dalam P yang sesuai. (Misalnya, jika λ_2 adalah nilai eigen terbesar, ambil kolom kedua P dan letakkan di posisi kolom pertama.) Panggil matriks yang diurutkan dari vektor eigen P *. (Kolom P * harus sama dengan kolom P, tetapi mungkin dalam urutan yang berbeda.) Perhatikan bahwa vektor eigen ini tidak bergantung satu sama lain.
7. Hitung Z * = ZP *. Matriks baru ini, Z *, adalah versi X yang dipusatkan / terstandarisasi tetapi sekarang setiap pengamatan adalah kombinasi dari variabel asli, di mana bobot ditentukan oleh vektor eigen. Karena vektor eigen kita di P * tidak bergantung satu sama lain, setiap kolom Z * juga tidak bergantung satu sama lain!
8. Terakhir, kita perlu menentukan berapa banyak fitur yang harus dipertahankan dengan berapa banyak yang akan dihapus. Ada tiga metode umum untuk menentukan hal ini, yang dibahas di bawah ini dan diikuti dengan contoh eksplisit:
    a. Metode 1:

       Secara acak memilih berapa banyak dimensi yang ingin dipertahankan. Mungkin kita ingin merepresentasikan hal-hal secara visual dalam dua dimensi, jadi saya hanya dapat menyimpan dua fitur. Ini bergantung pada kasus penggunaan dan tidak ada aturan yang pasti tentang berapa banyak fitur yang harus saya pilih.

b. Metode 2:

   Hitung proporsi varian yang dijelaskan untuk setiap fitur, pilih ambang batas, dan tambahkan fitur hingga Anda mencapai ambang tersebut.

c. Metode 3:

   Ini terkait erat dengan Metode 2. Hitung proporsi varian yang dijelaskan untuk setiap fitur, urutkan fitur berdasarkan proporsi varian yang dijelaskan, dan plot proporsi kumulatif varian yang dijelaskan saat Anda menyimpan lebih banyak fitur.

Meskipun PCA adalah metode yang sangat teknis yang mengandalkan algoritme aljabar linier yang mendalam, PCA adalah metode yang relatif intuitif saat Anda memikirkannya.

1) Pertama, matriks kovariansi Z^TZ adalah matriks yang berisi perkiraan tentang bagaimana setiap variabel di Z berhubungan dengan setiap variabel lain di Z. Memahami bagaimana satu variabel dikaitkan dengan variabel lain cukup ampuh.
2) Kedua, nilai eigen dan vektor eigen penting. Vektor eigen mewakili arah. Pikirkan untuk memplot data Anda pada sebar multidimensi. Kemudian orang dapat menganggap vektor eigen individu sebagai "arah" tertentu dalam diagram sebar data Anda. Nilai eigen mewakili besaran, atau kepentingan. Nilai eigen yang lebih besar berkorelasi dengan arah yang lebih penting.
3) Ketiga, membuat asumsi bahwa lebih banyak variabilitas dalam arah tertentu berkorelasi dengan menjelaskan perilaku variabel dependen. Banyak variabilitas biasanya menunjukkan sinyal, sedangkan variabilitas kecil biasanya menunjukkan noise. Jadi, semakin banyak variabilitas yang ada dalam arah tertentu, secara teoritis, menunjukkan sesuatu yang penting yang ingin kita deteksi.

Jadi, PCA adalah metode yang menyatukan:

a) Ukuran bagaimana setiap variabel dikaitkan satu sama lain. (Matriks kovarian.)
b) Arah penyebaran data. (Vektor eigen.)
c) Kepentingan relatif dari arah yang berbeda ini. (Nilai Eigen.)

Contoh:

PCA untuk Visualisasi Data

Untuk banyak aplikasi pembelajaran mesin, ada baiknya jika Anda dapat memvisualisasikan data Anda. Memvisualisasikan data 2 atau 3 dimensi tidaklah terlalu menantang. Namun, bahkan dataset Iris yang digunakan di bagian tutorial ini adalah 4 dimensi. PCA dapat Anda gunakan untuk mereduksi data 4 dimensi tersebut menjadi 2 atau 3 dimensi sehingga Anda dapat memplot dan semoga dapat memahami data tersebut dengan lebih baik.

Load Dataset Iris

Dataset Iris adalah salah satu set data yang dilengkapi scikit-learn yang tidak memerlukan pengunduhan file apa pun dari beberapa situs web eksternal. Kode di bawah ini akan memuat dataset iris.

```
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
# Load dataset into Pandas DataFrame
df = pd.read_csv(url, names=['sepal length','sepal width','petal length','petal width','target'])
```

Standarisasi Data

PCA dipengaruhi oleh skala sehingga Anda perlu menskalakan fitur dalam data Anda sebelum menerapkan PCA. Gunakan StandardScaler untuk membantu Anda menstandarisasi fitur set data ke dalam skala unit (mean = 0 dan variance = 1) yang merupakan persyaratan untuk performa optimal dari banyak algoritma pembelajaran mesin.

```
features = ['sepal length', 'sepal width', 'petal length', 'petal width']
# Separating out the features
x = df.loc[:, features].values
# Separating out the target
y = df.loc[:,['target']].values
# Standardizing the features
x = StandardScaler().fit_transform(x)
```

Proyeksi PCA ke 2D

Data asli memiliki 4 kolom (sepal length, sepal width, petal length, and petal width). Pada bagian ini kode memproyeksikan data asli yang 4 dimensi menjadi 2 dimensi. Perlu diingat bahwa setelah reduksi dimensi, biasanya tidak ada makna tertentu yang ditetapkan untuk setiap komponen utama. Komponen baru hanyalah dua dimensi utama variasi.

```
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(x)
principalDf = pd.DataFrame(data = principalComponents
             , columns = ['principal component 1', 'principal component 2'])
```

```
finalDf = pd.concat([principalDf, df[['target']]], axis = 1)
```

Menggabungkan DataFrame di sepanjang sumbu = 1. finalDf adalah DataFrame terakhir sebelum memplot data.

Visualisasikan Proyeksi 2D

Bagian ini hanya memplot data 2 dimensi.

```
fig = plt.figure(figsize = (8,8))
ax = fig.add_subplot(1,1,1)
ax.set_xlabel('Principal Component 1', fontsize = 15)
ax.set_ylabel('Principal Component 2', fontsize = 15)
ax.set_title('2 component PCA', fontsize = 20)
targets = ['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']
colors = ['r', 'g', 'b']
for target, color in zip(targets,colors):
    indicesToKeep = finalDf['target'] == target
    ax.scatter(finalDf.loc[indicesToKeep, 'principal component 1']
               , finalDf.loc[indicesToKeep, 'principal component 2']
               , c = color
               , s = 50)
ax.legend(targets)
ax.grid()
```

Varians yang Dijelaskan

Varians yang dijelaskan memberi tahu Anda berapa banyak informasi (varian) yang dapat dikaitkan dengan masing-masing komponen utama. Hal ini penting karena meskipun Anda dapat mengubah ruang 4 dimensi menjadi ruang 2 dimensi, Anda akan kehilangan sebagian varians (informasi) saat melakukannya. Dengan menggunakan atribut menjelaskan_varians_rasio_, Anda dapat melihat bahwa komponen utama pertama berisi 72,77% varian dan komponen utama kedua berisi 23,03% varian. Bersama-sama, kedua komponen tersebut mengandung 95,80% informasi.

```
pca.explained_variance_ratio_
```

```
array([0.72770452, 0.23030523])
```

TUGAS:

1. Lakukan Principal Component Analysis untuk dataset yang anda pilih di tugas 1!