
	Politechnika Bydgoska im. J.J. Śniadeckich Wydział Telekomunikacji, Informatyki i Elektrotechniki		
Przedmiot	Algorytmy i eksploracja danych		
Prowadzący	dr inż. Michał Kruczkowski		
Temat	Redukcja wymiarowości		
Studenci	Adam Szreiber, Cezary Naskręt		
Nr ćw.	2	Data wykonania	15.10.2023

1. Cel ćwiczenia

Celem dzisiejszego ćwiczenia było zapoznanie się z różnymi aspektami redukcji wymiarowości danych i wykonywanie praktycznych kroków związanych z tym procesem, takich jak obliczanie wymiarów wczytanych danych, zapisywanie przetworzonych danych do pliku csv, implementacja algorytmów redukcji wymiarowości takich jak PCA, obliczanie wariancji zbioru i wartości wariancji wyjaśnionej, obliczanie sumy składowej i poznanie innych sposobów redukcji wymiarowości.

2. Przebieg laboratorium

2.1. Zadanie 1

Wczytaj zbiór danych pochodzący z biblioteki sklearn za pomocą komendy:

```
from sklearn.datasets import load_breast_cancer
print(load_breast_cancer().data)
print(load_breast_cancer().target)
```

Zapoznaj się z informacjami dotyczącymi wczytanych danych znajdującymi się w dokumentacji sklearn.datasets.

Jako wynik działania programu wypisz następujące informacje:

1. Wymiar wczytanych danych
2. Ilość wartości unikatowych w wektorze target
3. Zaproponuj usunięcie tych kolumn ze zbioru danych, które Twoim zdaniem dostarczają najmniej informacji.
4. Zapisz w pliku dataset_cut.csv wypadkowy zbiór danych jak poniżej:
ColName1; ColName2; ...; TARGET

Do poznania wymiarów można wykorzystać atrybut `shape`. Następnie aby zliczyć liczbę wymiarów należy użyć funkcji `len()` mierzącej długość tablicy `shape`.

Aby zliczyć liczbę wartości unikatowych można użyć metody `np.unique()`.

```
from sklearn.datasets import load_breast_cancer
import numpy as np

data = load_breast_cancer().data
target = load_breast_cancer().target

print("Wymiar wczytanych danych: " + str(len(data.shape)))

print("Ilość wartości unikalnych w wektorze target: " +
      str(len(np.unique(target))))
```

Kolejnym zadaniem było zdecydowanie które naszym zdaniem kolumny można usunąć, aby zredukować rozmiar danych. Ja zdecydowałem usunąć dwie kolumny, które mają najmniejszą wariancję. W tym celu napisałem funkcję `reduceData()`, która liczy wariancję za pomocą metody `np.var()`, a następnie usuwa kolumny o najmniejszej wariancji.

Na końcu programu zredukowane dane są zapisywane do pliku zgodnie z formatem podanym w poleceniu.

```
def reduceData(data, columns):
    def getIndexOfMin(arr):
        m = min(arr)
        for i in range(len(arr)):
            if arr[i] == m:
                return i

    for i in range(columns):
        variance = np.var(data, 0)
        index = getIndexOfMin(variance)
        data = np.delete(data, index, axis=1)

    return data

reducedData = reduceData(data, 2)
outputData = np.hstack((reducedData, np.array([target]).T))
np.savetxt('dataset_cut.csv', outputData, delimiter=';')
```

2.2. Zadanie 2

Wykonaj redukcję wymiarowości za pomocą algorytmu PCA – analiza głównych składowych.

1. Wprowadź zbiór danych do algorytmu PCA
2. Wykonaj redukcję wymiarowości do 5 wymiarów wypadkowego zbioru danych
3. Zapisz w pliku dataset_pca_5.csv wypadkowy zbiór danych jak poniżej:
COMP1; COMP2; COMP3; COMP4; COMP5; TARGET
4. Oblicz wariancję zbioru danych po redukcji wymiarowości
5. Oblicz wartość wariancji wyjaśnionej dla wygenerowanych składowych (ang. explained variance ratio)

Pierwszym krokiem, jaki wykonuje jest import wszystkich bibliotek jakie będę używał do realizacji zadania drugiego. Następnie pobieram dane i odpowiednio je skaluje. Redukuje wymiar danych do 5.

```
# Import necessary libraries
from matplotlib import pyplot as plt
import pandas as pd # to load the dataframe
import numpy as np # to save in csv
from sklearn.preprocessing import StandardScaler # to standardize data
from sklearn.decomposition import PCA # to apply PCA
from sklearn.datasets import load_breast_cancer # data to work

# Load the Dataset
cancer = load_breast_cancer()
# convert the dataset into a pandas data frame
df = pd.DataFrame(cancer['data'], columns=cancer['feature_names'])

scalar = StandardScaler()
scaled_data = pd.DataFrame(scalar.fit_transform(df)) # scaling the data

pca = PCA(n_components=5)
data_pca = pca.fit_transform(scaled_data)
```

Otrzymany wynik zapisuję do pliku CSV zgodnie z formatem podanym w poleceniu. Tak jak w pierwszym zadaniu na końcu dopisuję kolumnę z wartościami target.

Wypisuję wariancję zbioru i wariancję wyjaśnioną. Poniżej pokazuje również sposób obliczania wariancji metodą `np.var` z biblioteki `numpy`. Jest to alternatywa dla odczytywania wariancji z obiektu PCA z biblioteki `sklearn`.

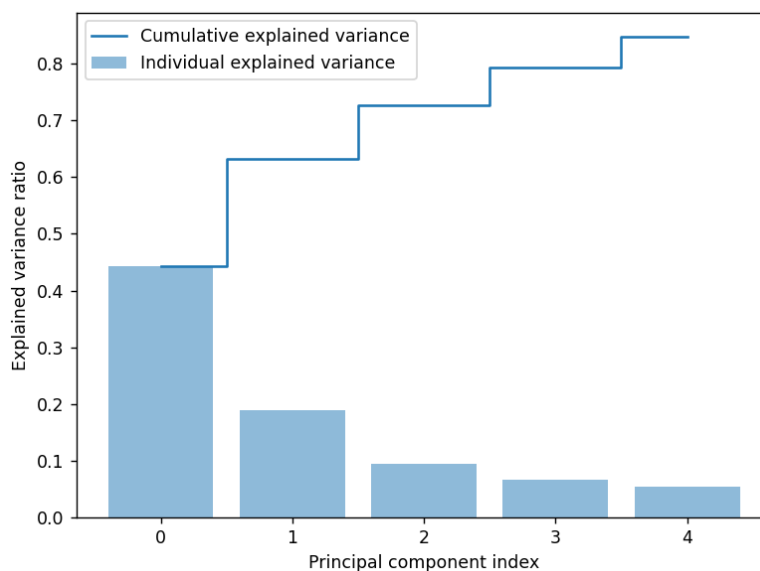
```
outputData = np.hstack((data_pca, np.array([cancer.target]).T))
np.savetxt('dataset_pca_5.csv', outputData, delimiter=';')

print(pca.explained_variance_) # variance
print(pca.explained_variance_ratio_) # variance ratio
variance = np.var(data_pca, 0)
print(variance) # variance
```

Dla lepszego zobrazowania danych wykorzystuję wizualizację graficzną z biblioteki `matplotlib`.

```
exp_var_pca = pca.explained_variance_ratio_
cum_sum_eigenvalues = np.cumsum(exp_var_pca)

# Create the visualization plot
plt.bar(range(0, len(exp_var_pca)), exp_var_pca, alpha=0.5,
        align='center', label='Individual explained variance')
plt.step(range(0, len(cum_sum_eigenvalues)), cum_sum_eigenvalues,
        where='mid', label='Cumulative explained variance')
plt.ylabel('Explained variance ratio')
plt.xlabel('Principal component index')
plt.legend(loc='best')
plt.tight_layout()
plt.show()
```



2.3. Zadanie 3

Wykonaj redukcję wymiarowości za pomocą algorytmu PCA – analiza głównych składowych.

1. Uzasadnij optymalny wymiar zbioru danych zakładając, że w zbiorze wypadkowym suma składowych będzie stanowiła minimum 90% wariancji.
2. Wykonaj redukcję wymiarowości za pomocą algorytmu PCA.
3. Zapisz w pliku dataset_pca_n.csv wypadkowy zbiór danych jak poniżej:
COMP1; COMP2; COMP3; COMP4; ...; COMPN; TARGET
4. Oblicz wartość wariancji wyjaśnionej dla wygenerowanych składowych (ang. explained variance ratio)
5. Skomentuj otrzymane rezultaty.

Aby wiedzieć do ilu wymiarów mogę zmniejszyć danę muszę policzyć skumulowaną wariancję, czyli kumulującą się sumę wariancji dla kolejnych wymiarów. Następnie sprawdzam dla ilu wymiarów suma jest mniejsza niż 90%. W kolejnym kroku redukuję dane do takiej liczby wymiarów, aby ich skumulowana wariancja była większa niż 90%. Następnie zapisuje dane do pliku w odpowiednim formacie.

```
# Import necessary libraries
from matplotlib import pyplot as plt
import pandas as pd # to load the dataframe
import numpy as np # to save in csv
from sklearn.preprocessing import StandardScaler # to standardize data
from sklearn.decomposition import PCA # to apply PCA
from sklearn.datasets import load_breast_cancer # data to work

# Load the Dataset
cancer = load_breast_cancer()
# convert the dataset into a pandas data frame
df = pd.DataFrame(cancer['data'], columns=cancer['feature_names'])

scalar = StandardScaler()
scaled_data = pd.DataFrame(scalar.fit_transform(df)) # scaling the data

pca = PCA()
pca.fit_transform(scaled_data)

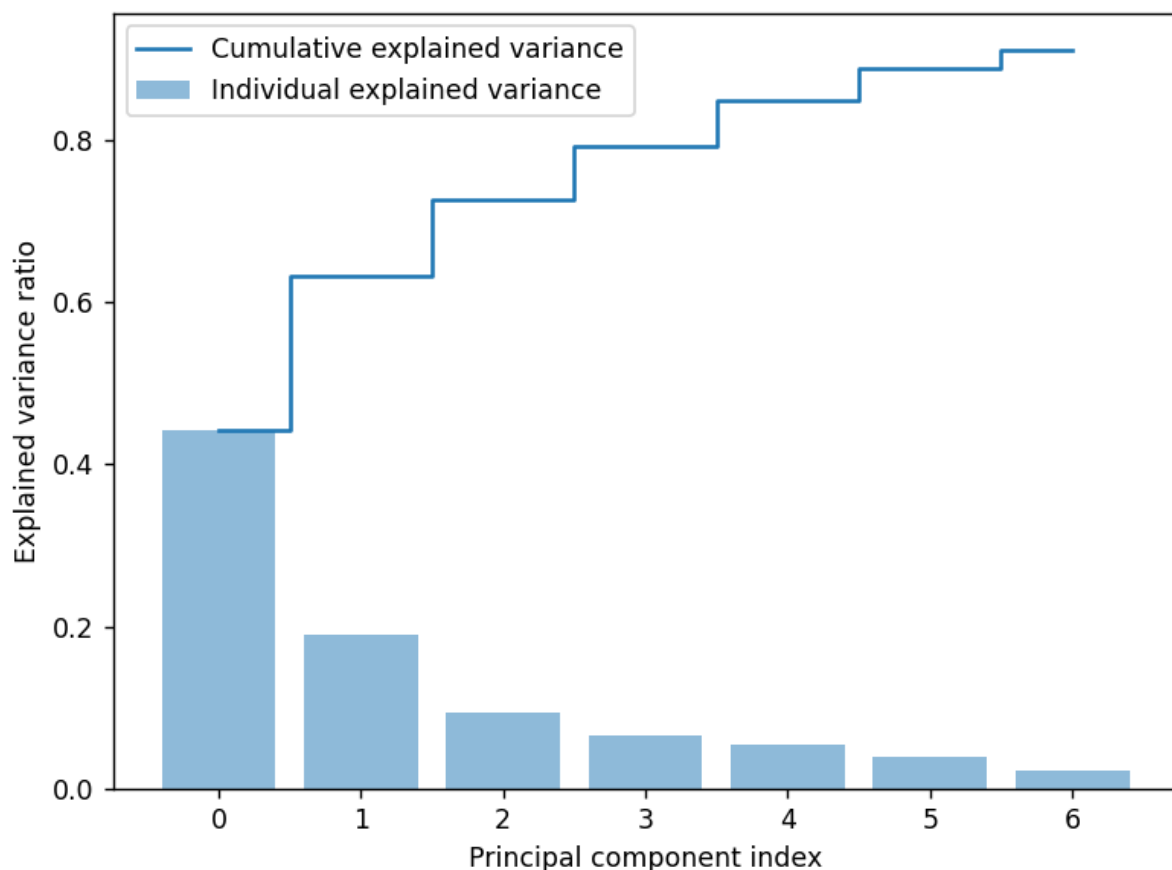
# calculate cumulate sum of variance
cum_sum_eigenvalues = np.cumsum(pca.explained_variance_ratio_)

components = len(list(filter(lambda v: v < 0.9, cum_sum_eigenvalues))) + 1
pca2 = PCA(n_components=components)
data_pca = pca2.fit_transform(scaled_data)
outputData = np.hstack((data_pca, np.array([cancer.target]).T))
np.savetxt('dataset_pca_' + str(components) + '.csv', outputData,
delimiter=';')
```

Wypisuje na ekranie wariancje i wariancję wyjaśnioną.

```
print(pca2.explained_variance_) # variance
print(pca2.explained_variance_ratio_) # variance ratio

# to visualization
exp_var_pca = pca2.explained_variance_ratio_
cum_sum_eigenvalues = np.cumsum(exp_var_pca)
# Create the visualization plot
plt.bar(range(0, len(exp_var_pca)), exp_var_pca, alpha=0.5,
        align='center', label='Individual explained variance')
plt.step(range(0, len(cum_sum_eigenvalues)), cum_sum_eigenvalues,
        where='mid', label='Cumulative explained variance')
plt.ylabel('Explained variance ratio')
plt.xlabel('Principal component index')
plt.legend(loc='best')
plt.tight_layout()
plt.show()
```



2.4. Zadanie 4

Wykonaj redukcję wymiarowości na podstawie dowolnego innego algorytmu a wynik zapisz w pliku dataset_algorithm.csv. Skomentuj otrzymane rezultaty.

Do redukcji wielowymiarowości postanowiłem wybrać metodę ICA. Na początku importuje wszystkie niezbędne biblioteki. Następnie wczytuje dane. Skaluje odpowiednio dane. W kolejnym kroku dokonuje redukcji wymiarów za pomocą algorytmu ICA. Zapisuje otrzymane dane do pliku CSV.

```
# Import necessary libraries
import pandas as pd # to load the dataframe
import numpy as np # to save in csv
from sklearn.preprocessing import StandardScaler # to standardize data
from sklearn.datasets import load_breast_cancer # data to work
from sklearn.decomposition import FastICA # to apply ICA

# Load the Dataset
cancer = load_breast_cancer()
# convert the dataset into a pandas data frame
df = pd.DataFrame(cancer['data'], columns=cancer['feature_names'])

scalar = StandardScaler()
scaled_data = pd.DataFrame(scalar.fit_transform(df)) # scaling the data

transformer = FastICA(n_components=7)
data_ica = transformer.fit_transform(scaled_data)
print(data_ica.shape)
print(data_ica)

outputData = np.hstack((data_ica, np.array([cancer.target]).T))
np.savetxt('dataset_ica_7.csv', outputData, delimiter=';')
```

1. Wnioski

Posiadanie dużej ilości danych jest korzystne, ale nadmiar kolumn może spowolnić tworzenie modeli. Dlatego czasami trzeba ograniczyć liczbę wymiarów w zbiorze danych. Wybór odpowiedniej metody jest istotny, aby uniknąć utraty ważnych informacji. Istnieją różne sposoby redukcji wymiarowości, zarówno poprzez usuwanie kolumn, jak i przekształcanie istniejących w nowe.

Pierwszym sposobem wyboru, które kolumny usunąć, jest analiza tych zawierających duże ilości brakujących wartości. Jeśli dana kolumna nie jest niezbędna do celów przewidywania, usunięcie kolumn z dużym odsetkiem brakujących wartości spowoduje mniejszą utratę informacji niż usunięcie całych kolumn. W naszym przypadku dane nie posiadały braków.

Jeśli kolumna ma bardzo niską wariancję, to prawdopodobnie nie jest przydatna do prognoz. Niewielkie zmiany w wartości nie powodują znaczących zmian w zmiennej wynikowej. Istnieje duża szansa, że ta cecha nie ma istotnego wpływu na nasz model dlatego możemy ją usunąć. Jest to kolejny sposób na redukcję wymiarowości.

PCA to kolejny algorytm do redukcji wymiarów. Próbuje on przekształcić dwie lub kilka istniejących kolumn w jedną, zachowując wariancję z oryginalnego zestawu danych.

Wariancja to miara rozproszenia danych wokół ich średniej wartości. Oznacza, jak bardzo punkty danych różnią się od średniej. Wyższa wariancja wskazuje na większą różnorodność danych, a niższa na mniejszą zmienność. Wariancja oblicza się jako średnią arytmetyczną kwadratów odchyleń od średniej.

Współczynnik wyjaśnionej wariancji to miara w analizie głównych składowych (PCA) informująca, ile procent całkowitej wariancji danych jest wyjaśnione przez daną składową. Pomaga ocenić, jakie znaczenie ma dana składowa w kontekście redukcji wymiarów. Wartości współczynnika sumują się do 100%, co pozwala określić, ile informacji jest zachowywane lub utracone w procesie redukcji wymiarów. Jest używany do wyboru odpowiedniej liczby składowych do zachowania w celu uproszczenia modelu. Składowe z wyższym współczynnikiem wyjaśnianej wariancji są istotne dla analizy danych.

Ostatnim algorytmem, jaki użyłem do redukcji wymiarów jest ICA (Independent Component Analysis). To technika analizy sygnałów, która oddziela niezależne składowe z mieszanego sygnału. Opiera się na transformacjach, aby wyodrębnić źródła sygnałów.