

POLITECHNIKA WROCŁAWSKA

INTELIGENCJA OBLICZENIOWA I JEJ ZASTOSOWANIA

Ćwiczenie 1
Metody redukcji wymiarowości
Analiza składowych głównych

Autorzy:

Paweł ANDZIUL 200648

Robert CHOJNACKI 200685

Marcin SŁOWIŃSKI 200638

Prowadzący:

dr hab. inż. Rafał ZDUNEK

7 czerwca 2017

Spis treści

1	Zadanie 1	2
1.1	Opis metody	2
1.2	Algorytm	2
1.3	Realizacja	2
1.4	Wyniki	3
2	Zadanie 2	3
2.1	Wczytanie obrazów twarzy	4
2.2	Wyznaczenie cech holistycznych (twarzy własnych)	5
2.3	Grupowanie metodą k-średnich	5
2.4	Klasyfikacja za pomocą klasyfikatora k-NN	5
3	Zadanie 3	8
3.1	Algorytm Powera	8
3.2	Algorytm Lanczosa	8
3.3	Wyniki	8
4	Podsumowanie	8

1 Zadanie 1

Dla danych:

$$X = [2.50.52.21.93.12.3211.51.1; 2.40.72.92.232.71.61.11.60.9];$$

- Zaimplementować metodę PCA w Matlabie. Do wyznaczenia par własnych macierzy kowariancji można zastosować wbudowaną funkcję `eig(.)` lub `eigs(.)`.
- Wyznaczyć składowe główne i wektory cech.
- Pokazać na rysunku punkty obserwacji oraz wyznaczone wielkości.

1.1 Opis metody

Metoda PCA (ang. *Principal Component Analysis*) jest jedną ze statystycznych metod analizy czynnikowej, która pozwala na odnajdywanie pewnych struktur w zbiorze zmiennych losowych. Wielowymiarowe dane z reguły nie są równomiernie rozrzucone wzdłuż wszystkich kierunków układu współrzędnych, ale koncentrują się w pewnych podprzestrzeniach oryginalnej przestrzeni. Analiza składowych głównych oparta jest na wykorzystaniu pojęć ze statystyki, jakimi są m.in. korelacja i wariancja. Celem PCA jest znalezienie tych podprzestrzeni w postaci składowych głównych, tzn. kierunków przy których wartość wariancji (lub korelacji) jest zmaksymalizowana. Analiza może być oparta albo na macierzy korelacji, albo macierzy kowariancji utworzonej ze zbioru wejściowego.

Metoda PCA jest zazwyczaj używana do redukcji rozmiaru danych statystycznych, poprzez odrzucenie ostatnich czynników, tzn. takich, których wariancja jest najniższa. Oznacza to, że n -wymiarowy zbiór danych możemy ograniczyć wyznaczając n -wektorów własnych, z których wybieramy tylko k -wektorów, tak aby kn . Zrzutowanie danych na przestrzeń rozpiętą przez k -wybranych wektorów pozwala zredukować wymiarowość danych.

1.2 Algorytm

1.3 Realizacja

Rozwiązanie zadania polega kolejno na...

Listing 1: Skrypt w Matlabie

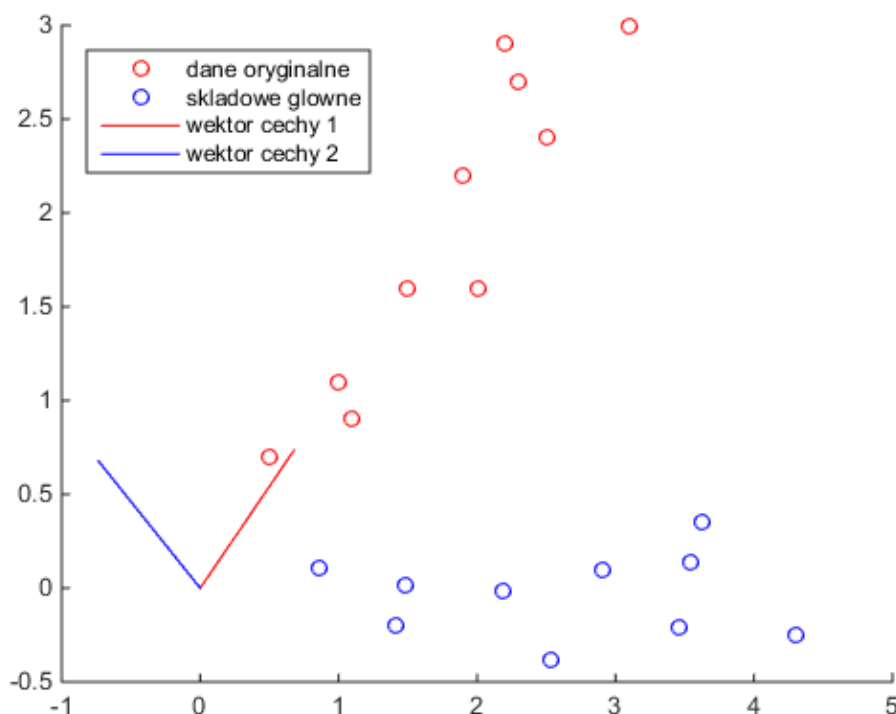
```
1 x = [2.5 0.5 2.2 1.9 3.1 2.3 2 1 1.5 1.1; 2.4 0.7 2.9 2.2 3 2.7 1.6 1.1 1.6
2     0.9];
3 s = cov(x');
4 [eigenvectors, eigenvalues] = eigs(s, 2);
5 pcs = (x') * eigenvectors;
6
7 figure;
8 hold on
9 plot(x(1:1,:), x(2:2,:), 'or', pcs(:,1:1), pcs(:,2:2), 'ob')
10 plotv(eigenvectors(:,1), '-r');
11 plotv(eigenvectors(:,2), '-b');
12 hold off
```

```
13 legend('dane oryginalne','składowe glowne','wektor cechy 1','wektor cechy
    2','Location','northwest','Orientation','vertical')
```

Powyższy kod przedstawia zaimplementowaną metodę PCA dla zadanego zbioru danych.

W celu obliczenia par własnych macierzy kowariancji zastosowana została dwuargumentowa metoda `eigs`, w której pierwszym argumentem jest macierz danych, drugim ilość wektorów (wierszy macierzy).

1.4 Wyniki



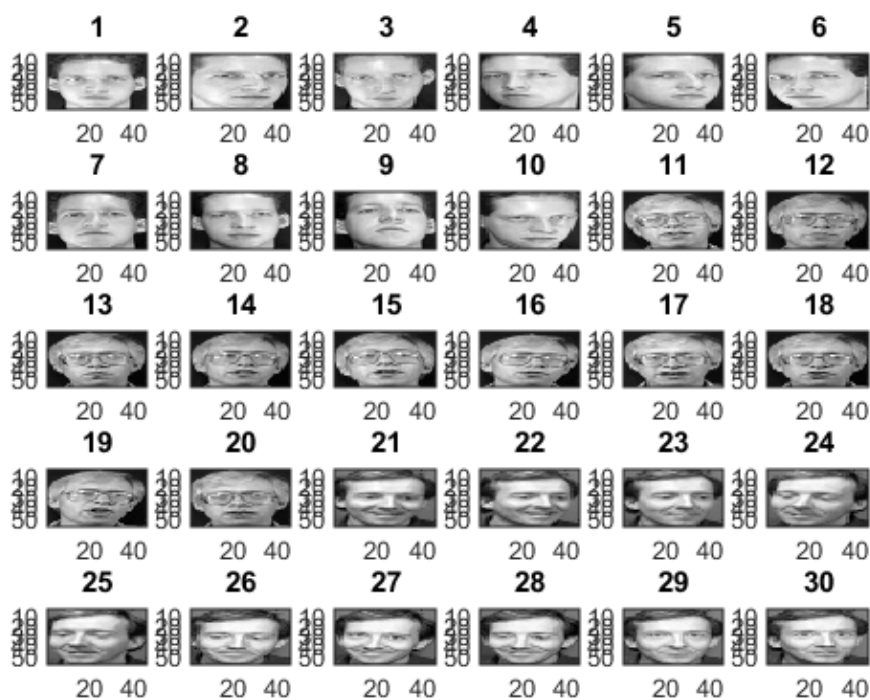
Rysunek 1: Ilustracja punktów obserwacji i składowych głównych

Na wykresie można zauważyć poprawne działanie funkcji PCA. Dane rozkładają się wzdłuż oznaczonych wektorów.

2 Zadanie 2

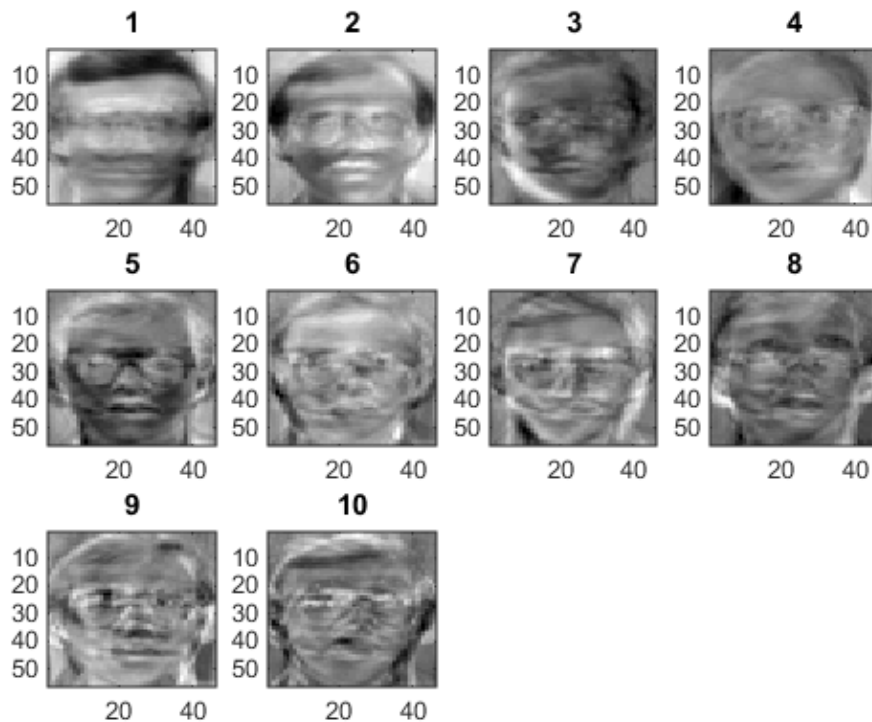
Dla obrazów twarzy z bazy ORL (lub podobnej) wyznaczyć cechy holistyczne (twarze własne) dla różnej liczby estymowanych komponentów głównych ($J = 4, 10, 20, 30$). Pogrupować obrazy stosując metodę k -średnich, do obrazów oryginalnych oraz zredukowanych. Badania przeprowadzić dla różnej liczby grup. Porównać dokładność i czas grupowania. Następnie dokonać klasyfikacji obrazów w obu przestrzeniach (oryginalnej i zredukowanej) za pomocą klasyfikatora k -NN. Porównać efekty klasyfikacji z efektami grupowania.

2.1 Wczytanie obrazów twarzy



Rysunek 2: Ilustracja twarzy wejściowych...

2.2 Wyznaczenie cech holistycznych (twarzy własnych)



Rysunek 3: Ilustracja ejgenfejsów...

2.3 Grupowanie metodą k-średnich

2.4 Klasyfikacja za pomocą klasyfikatora k-NN

Zgodnie z dokumentacją metody `knnclassify` [3] umożliwia ona klasyfikację pewnej próbki danych na podstawie danych dla których klasyfikacja jest znana.

$$Class = knnclassify(Sample, Training, Group) \quad (1)$$

, parametry *Training* oraz *Group* zawierają informację o podziale danych na grupy natomiast *Sample* jest zbiorem którego klasyfikację pragniemy przeprowadzić. Wynikiem jest *Class* będący tego samego typu co *Group* tzn. dziedzina klasyfikacji jest taka sama.

Poniżej zamieszczono..

Listing 2: Fragment skryptu w Matlabie obejmujący klasyfikację k-NN

```
1 clc;  
2 clear;  
3 close all;  
4
```

```

5 load('yalefaces.mat');
6 load('FaceData_56_46.mat');
7
8 %imagesc(yalefaces(:,:,1))
9
10 % -- FRAGMENT Z WIKIPEDII --
11
12 % J = 30;
13 % [h,w,n] = size(yalefaces);
14 % d = h*w;
15 % % vectorize images
16 % x = reshape(yalefaces,[d n]);
17 % x = double(x);
18 % %subtract mean
19 % x=bsxfun(@minus, x, mean(x,2));
20 % % calculate covariance
21 % s = cov(x');
22 % % obtain eigenvalue & eigenvector
23 % [V,D] = eigs(s,J);
24 % eigval = diag(D);
25 % % sort eigenvalues in descending order
26 % eigval = eigval(end:-1:1);
27 % V = fliplr(V);
28 % % show 0th through 15th principal eigenvectors
29 % eig0 = reshape(mean(x,2), [h,w]);
30 % figure,subplot(6,6,1)
31 % imagesc(eig0)
32 % colormap gray
33 % for i = 1:J
34 %     subplot(6,6,i+1)
35 %     imagesc(reshape(V(:,i),h,w))
36 % end
37
38
39 % -- NASZ KOD --
40
41 Persons = 3;
42 ImagesPerPerson = 10;
43 J = 10;
44
45 Group = [];
46 M = [];
47 for p=(1:Persons)
48     for i=(1:ImagesPerPerson)
49         x = FaceData(p, i).Image;
50         [irow, icol] = size(x);
51         x = double(x);
52         temp = reshape(x', irow * icol, 1);
53         M = [M temp];
54         Group = [Group p];
55     end
56 end
57
58
59 %[eigenvectors, eigenvalues] = eigs(M*(M'), J);
60 %V = eigenvectors;

```

```

61
62
63 %subtract mean
64 x=bsxfun(@minus, M, mean(M,2));
65 % calculate covariance
66 s = cov(x');
67 % obtain eigenvalue & eigenvector
68 [V,D] = eigs(s,J);
69 Z = (x') * V;
70
71
72 figure; %originalne
73 nOfImages = Persons*ImagesPerPerson;
74 for i=(1:nOfImages)
75     C = M(:,i);
76     CC = reshape(C, [46, 56]);
77     subplot(round(sqrt(nOfImages)), round(sqrt(nOfImages)) + 1, i);
78     imagesc(CC');
79     title(i)
80     colormap gray;
81 end
82
83 figure; %eigenfaces
84 for i=(1:J)
85     C = V(:,i);
86     CC = reshape(C, [46, 56]);
87     subplot(round(sqrt(J)), round(sqrt(J)) + 1, i);
88     imagesc(CC');
89     title(i)
90     colormap gray;
91 end
92
93 id = (1:J);
94
95 kmeans_result = kmeans(Z', Persons);
96 groups = sortrows([id', kmeans_result], 2);
97
98 known_groups = [1 2 3 2 1 2 3 2 3 1];
99
100 confusionmat_result = confusionmat(known_groups, kmeans_result);
101
102 [acc_eigen, rand_index_eigen, match_eigen] = AccMeasure(known_groups,
    kmeans_result);
103
104 kmeans_result_original = kmeans(M', Persons);
105 [acc_orig, rand_index_orig, match_orig] = AccMeasure(Group,
    kmeans_result_original);
106
107 Class = knnclassify(V', M', Group);

```

Poniżej confusion matrix:

2	0	1
3	1	0
3	0	0

3 Zadanie 3

Wyznaczyć pary własne macierzy kowariancji za pomocą algorytmów: Powera oraz Lanczosa. Zaimplementować algorytmy i zastosować je do rozwiązania powyższych zadań. Porównać wyniki.

3.1 Algorytm Powera

Listing 3: Fragment skryptu w Matlabie

```
1
2 A=[1 1 -2;-1 2 1; 0 1 -1]
3 x=[1 1 1];
4 x=x';
5
6 [m,y_final]=power_method(A,x)
```

3.2 Algorytm Lanczosa

3.3 Wyniki

4 Podsumowanie

Podczas zajęć laboratoryjnych...

Z przeprowadzonych badań wynika...

Literatura

- [1] <https://www.mathworks.com/help/nnet/ref/plotconfusion.html>
- [2] <https://www.mathworks.com/help/stats/confusionmat.html>
- [3] <https://www.mathworks.com/help/bioinfo/ref/knnclassify.html>
- [4] <http://www.kmg.zut.edu.pl/opt/wyklad/bezgrad/powell.html>
- [5] https://en.wikipedia.org/wiki/Lanczos_algorithm