



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی برق

مقطع کارشناسی

پروژه ی جبر خطی
عنوان پروژه:
شناسایی چهره

نگارش
زهرا عربی
9523083

استاد
دکتر عطریانفر

تیر ماه 1397

در این پروژه می‌خواهیم شناسایی چهره را به کمک PCA در متلب پیاده سازی کنیم.

دلیل استفاده از PCA کاهش دادن ابعاد داده های ورودی است. میدانیم که سائز هر عکس ممکن است عدی بزرگ باشد و هنگامی که مجموعه ای از عکس ها داریم نیاز به ماتریسی داریم که هر سطر آن نمایانگر یک عکس باشد و این در مقیاس نه چندان بزرگ عدی بزرگ می شود و با مشکل memory leak مواجه خواهیم بود.

برای روشن شدن این مطلب در نظر بگیرید که در فایل آموزش ما 60 عکس وجود دارد که ابعاد هر عکس را مطابق زیر به دست می آوریم.

```
1 - clear all
2 - close all
3 - clc
4
5 - originalImage = imread('Train\1.jpg');
6 - [rows, columns] = size(originalImage);
7 - rows
8 - columns
9
```

Command Window

rows =

128

columns =

128

هر عکس سائزی معادل 128*128 دارد. برای نشان دادن هر عکس در برنامه باید هر عکس را معادل یک ماتریس 1*MN در بیاوریم یعنی هر عکس به صورت یک ماتریس یک سطری نشان داده می شود. حال برای نمایش همه ی داده های ورودی که به صورت تصویر هستند باید ماتریسی به تعداد سطر های داده های آموزش در نظر بگیریم و تعداد ستون سائز تصویر ها می باشد. یعنی نیاز به یک ماتریس n*MN داریم.

در این پروژه M=N=128 می باشد و برای کل داده ها حجم ماتریس کل برابر است با
 $983040 = 128 * 128 * 60$

این عدد به دست آمده بسیار بزرگ است و نیاز داریم که سائز مناسبی برای عکس ها در نظر بگیریم به گونه ای که حاصل چندان بزرگ نباشد و دچار overflow نشویم.

برای این منظور سائز مدنظر را 100 در نظر میگیریم.

درون حلقه ای همه ی عکس ها را با دستور `imread` می خوانیم و سپس سایز هر عکس را به 100×100 کاهش می دهیم. حال ماتریس حاصل را به صورت 1 بعدی نوشته و به عنوان یک سطر از ماتریس کلی X در نظر می گیریم.

بعد از یافتن و شکل دادن داده ها باید بردار مناسب جهت کاهش ابعاد داده ها بیابیم به منظور رسیدن به این مهم ابتدا باید میانگین داده های آموزش را پیدا کنیم. میتوان تصویری که به عنوان میانگین نمایش داده می شود را با `imshow` نشان دهیم.

```
clear all
close all
clc

%Number of Train pictures
n = 60;
X_train = TrainMatrix(60);

m=mean(X_train);
MI=reshape(m,[100,100]);
imshow(uint8(imresize(MI,[128,128])),[]);
```



پس از یافتن میانگین تصاویر آن را به عنوان مبدا قرار می دهیم و مختصات بقیه نقاط را نسبت به آن به دست می آوریم که این عمل با کم کردن اعداد هر تصویر از میانگین حاصل می شود. سپس با استفاده از فرمول زیر ماتریس کوواریانس را محاسبه می کنیم.

$$Q = \frac{X^T \cdot X}{n - 1}$$

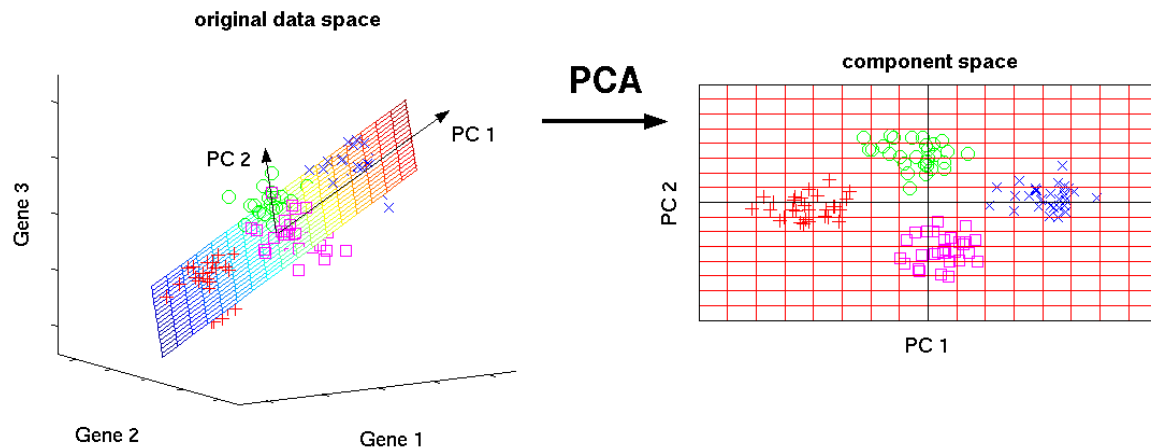
سپس باید مقادیر ویژه و بردار ویژه Q را محاسبه کنیم که این امر با استفاده از دستور `eig` در متلب امکان پذیر است. به وسیله ی این دستور ماتریس مربعی قطری حاوی مقادیر ویژه به دست می آید.

```
for i=1:n
    X_train(i,:)= X_train(i,:)-m;
end
%Covariance matrix
Q=(X_train'*X_train)/(n-1);
[eVecMatrix,eValMatrix]=eig(Q);
%Sort all eigen values
eVal=diag(eValMatrix);
[eValSorted,ind]=sort(eVal,'descend');
eVecSorted=Evec(:,ind);
```

تعریف PCA:

یکی از کاربردهای اصلی PCA در عملیات کاهش ویژگی (Dimensionality Reduction) است. PCA همان طور که از نامش پیداست می تواند مولفه های اصلی را شناسایی کند و به ما کمک می کند تا به

جای اینکه تمامی ویژگی‌ها را مورد بررسی قرار دهیم، یک سری ویژگی‌هایی را ارزش بیشتری دارند، تحلیل کنیم. در واقع PCA آن ویژگی‌هایی را که ارزش بیشتری فراهم می‌کنند برای ما استخراج می‌کند. برای مثال به جای جایگزین کردن هر عکس با بردار یک بعدی $1 \times MN$ (که مشاهده شد که حتی با کاهش سایز هر عکس نیز عددی بزرگ است) فقط ویژگی‌های معروف را در هر عکس نگه داریم و تعداد این ویژگی‌ها L در نظر می‌گیریم و در این پروژه با آزمون و خطا مقدارش را محاسبه می‌کنیم.



مقادیر ویژه را بر اساس مقدار آن‌ها به صورت نزولی مرتب کرده و L مقدار بیشتر آن را به عنوان ماتریس pca در نظر می‌گیریم.

در مرحله ی بعد باید ماتریس انتقال را محاسبه کنیم که از طریق رابطه ی زیر به دست می آید:

$$T_{n \times L} = [X - m]_{n \times MN} \cdot [P_{PCA}]_{MN \times L}$$

کد قسمت آموزش در این مرحله به اتمام میرسد و باید قسمت تست نوشته شود.

تست پروژه:

در قسمت تست ابتدا باید فایل تست انتخاب شود سپس ابعاد آن به ابعاد مورد نظر کاهش یابد. ماتریس تصویر مربوطه باید به فضای PCA تبدیل شود و برای هر تصویر جمع قدر مطلق هر دو نقطه محاسبه میشود و تصویری که دارای کمترین اختلاف است یعنی نزدیک ترین تصویر می باشد و به عنوان جواب بازگردانده می شود.

```

function Test(M,m,n,Pca,T)

[fileN,pathN]=uigetfile('*.','Select the File to be Tested');
filewithpath=strcat(pathN,fileN);
img=imread(filewithpath);
img=imresize(img,[M,M]);
img=double(reshape(img,[1,M*M]));
imgpca=(img-m)*Pca;
dist=zeros(n,1);
for i=1:n
    dist(i)=sum(abs(T(i,:)-imgpca));
end
[result,ind]=min(dist);
resultimg=imread(sprintf('Train\\%d.jpg',ind));
subplot(121)
imshow(imgo);
title('test file');
subplot(122)
imshow(resultimg);
title('found file');

```

