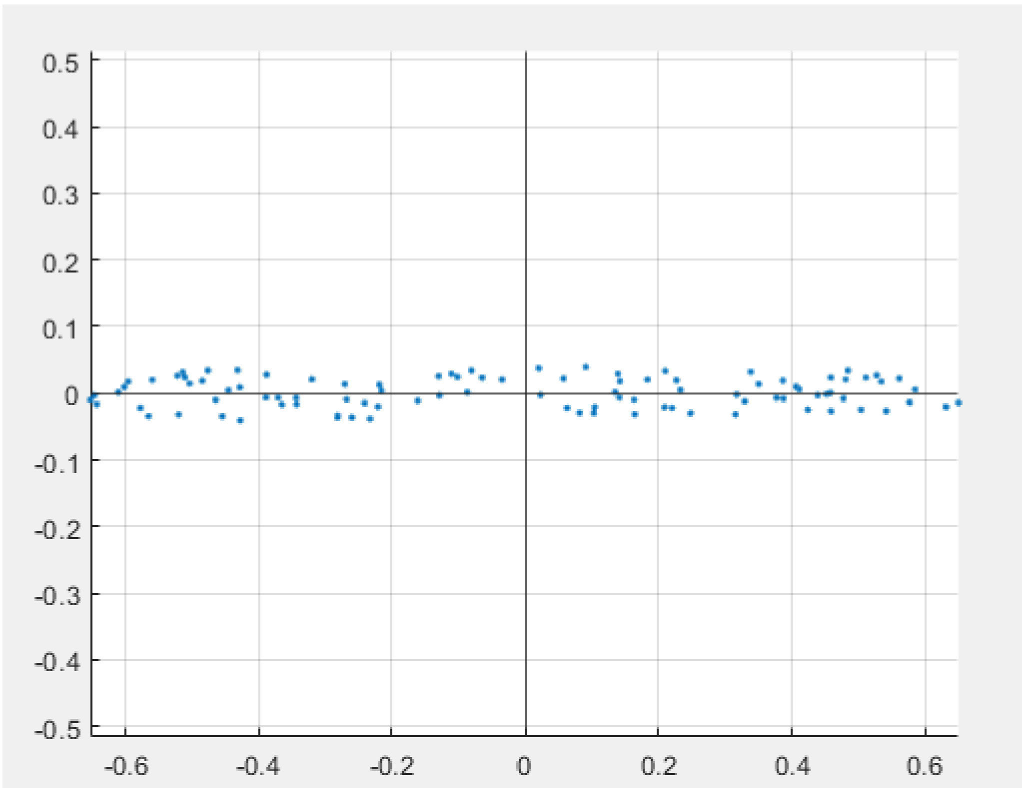
VC: Informe de Laboratori 1

Transformació d'un núvol de punts



Pere Ginebra Solanellas

21/2/2021 – Q2 Curs 2020-21

Visió per Computador, FIB UPC

1. Introducció

En aquest laboratori pretenem començar a familiaritzar-nos amb l'entorn de Matlab així com estudiar com manipular o transformar certs conjunts de dades, en concret un núvol de punts representats en un gràfic 2D, el qual rotarem per deixar-lo centrat a l'eix horitzontal.

2. Explicació del procés:

Seguint els passos descrits per l'enunciat, comencem generant un núvol aleatori a partir de 100 valors aleatoris de x i 100 de y en funció dels de x (i un pendent i offset aleatori). Amb això obtenim un núvol de 100 punts distribuïts al voltant d'una recta amb el pendent aleatori obtingut:

```
% Point cloud creation
x = rand(1,100) + rand();    % 100 punts aleatoris amb un offset aleatori
y = rand().*x  + rand(1,100)/10; % pendent i offset aleatoris

scatter(x,y,'.');
grid on;
axis('equal');
```

Figura 2.1. Generació d'un núvol de punts aleatoris distribuïts al voltant d'un pendent aleatori

A continuació centrem el núvol de punts a l'origen, restant-li a les coordenades de cada punt el valor de la mitjana del conjunt:

```
% Cloud point centering
xp = x - mean(x);
yp = y - mean(y);
```

Figura 2.2. Centrar el núvol de punts a l'origen

Ara toca girar el núvol per distribuir els punts al voltant de l'eix horitzontal de l'origen, per això ens cal determinar l'angle que forma la pendent del núvol amb l'eix horitzontal per realitzar una rotació adient. Aquesta pendent l'obtindrem a partir dels eigenvectors de la matriu de covariància del conjunt/núvol, en concret, l'eigenvector amb l'eigenvalue més gran és el vector que representa el pendent del nostre núvol (en el nostre cas, Matlab ens dona el vector unitari en concret):

```
% Covariance and eigen values
c = cov(xp, yp);
[evectors, evalues] = eig(c);

% Determine which dimension has the major variance
[val,ind] = max(diag(evalues));
```

Figura 2.3. Càlcul dels eigenvectors de la matriu de covariància del núvol

Obtingut l'eigenvector adequat, només ens queda calcular l'angle de rotació adient i la seva matriu corresponent (en aquest cas una rotació en el sentit de les agulles del rellotge) i aplicar-la al conjunt de punts:

```
% Extract the angle of the 'major axis'
% obtenim l'angle a partir de l'arctangent dels components de
% l'eigenvector adequat
theta = -pi/2-atan2(evectors(ind,1),evectors(ind,2));

% Create clockwise rotation matrix
R = [cos(theta) sin(theta); -sin(theta) cos(theta)];

% Rotate the points
rp = R * [xp;yp];
```

Figura 2.4. Càlcul de l'angle de rotació i la seva matriu

Finalment, representem el nou núvol transformat en una nova figura, en la qual afegirem els eixos de l'origen per una millor visualització de la modificació:

```
% Draw the points
figure
scatter(rp(1,:),rp(2,:),'.');
axis('equal');
xline(0);
yline(0);
grid on
```

Figura 2.5. Representació del núvol transformat en una nova figura

3. Resultats

Alguns dels resultats obtinguts al executar l'script:

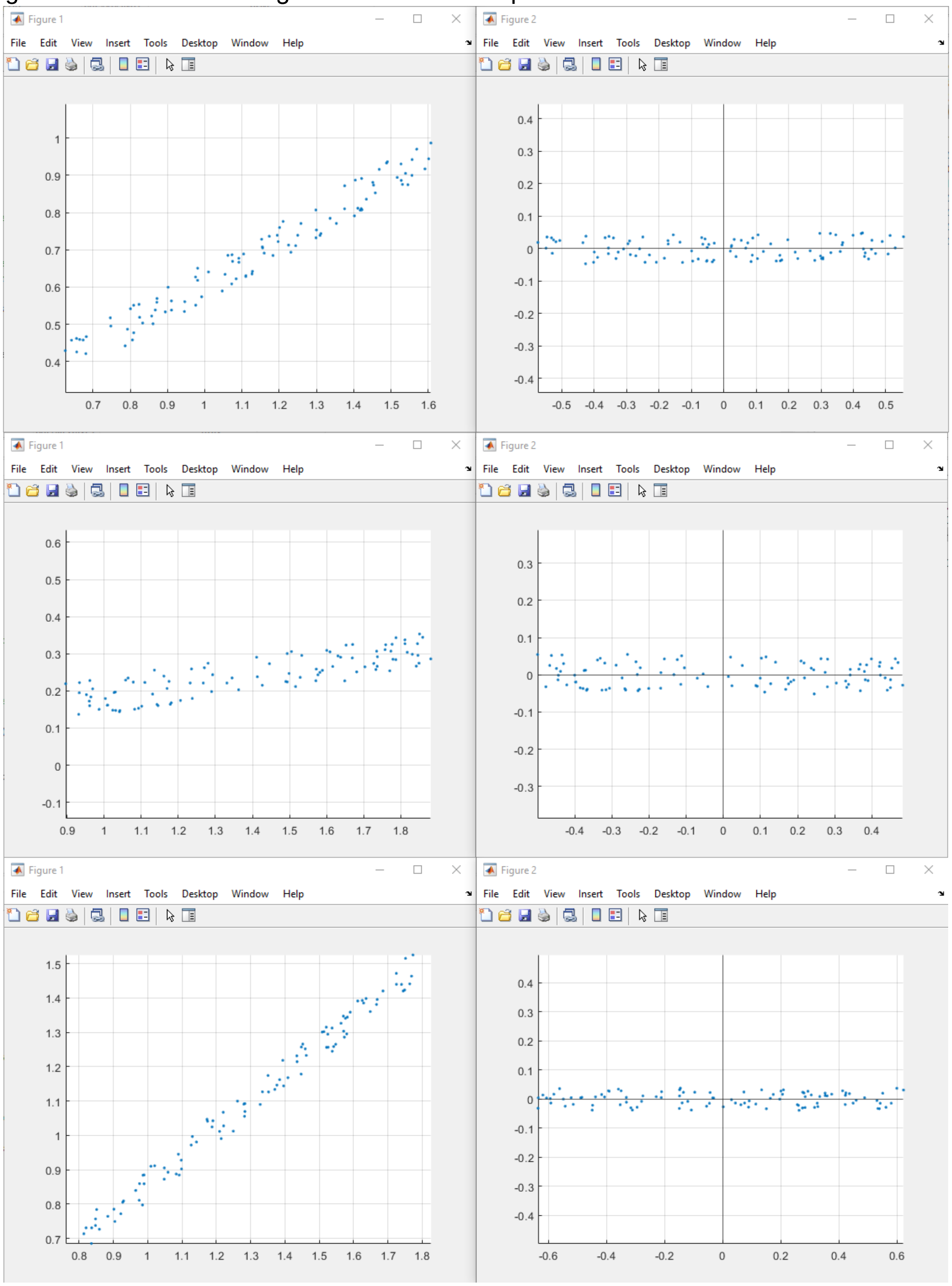


Figura 3.1. Resultats obtinguts en diferents execucions del script
(a l'esquerra els núvols originals i a la dreta els transformats)

Com podem veure la transformació efectivament ens transporta el núvol de punts a l'eix horitzontal de l'origen. Aquesta nova distribució ens permetrà, per exemple, estudiar el grau de similitud entre imatges diferents.

Aquest procés també es podria aplicar sobre conjunts de dades en 3 dimensions adaptant les matrius utilitzades (covariança i de rotació) i les seves operacions (com l'obtenció dels eigenvalues i els angles respecte els eixos) per a 3 conjunts de dades x, y i z.

4. Annexos (Script)

```
clear
close all

% Point cloud creation
x = rand(1,100) + rand();    % 100 punts aleatoris amb un offset aleatori
y = rand().*x  + rand(1,100)/10; % pendent i offset aleatoris

scatter(x,y,'.');
grid on;
axis('equal');

% Cloud point centering
xp = x - mean(x);
yp = y - mean(y);

% Covariance and eigen values
c = cov(xp, yp);
[evectors, evalues] = eig(c);

% Determine which dimension has the major variance
[val,ind] = max(diag(evalues));

% Extract the angle of the 'major axis'
% obtenim l'angle a partir de l'arctangent dels components de
% l'eigenvector adequat
theta = -pi/2-atan2(evectors(ind,1),evectors(ind,2));

% Create clockwise rotation matrix
R = [cos(theta) sin(theta); -sin(theta) cos(theta)];

% Rotate the points
rp = R * [xp;yp];

% Draw the points
figure
scatter(rp(1,:),rp(2,:),'.');
axis('equal');
xline(0);
yline(0);
grid on
```

Figura 4.1. Script utilitzat per la sessió

5. Bibliografia / documentació

Documentació sobre el procés de transformació d'un núvol de punts:

- https://alyssaq.github.io/2015/computing-the-axes-or-orientation-of-a-blob/
- http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

Documentació per altres conceptes utilitzats en l'exercici:

- https://atenea.upc.edu/pluginfile.php/3776482/mod_resource/content/1/review.PDF
- https://www.youtube.com/watch?v=PFDu9oVAE-g&ab_channel=3Blue1Brown
- https://en.wikipedia.org/wiki/Covariance