

Lab 1: Least squares, SVD and PCA

Authors Ferran Dalmau Codina: u150401 206744 and Jalal Douiri: U150388 206338¹

^{*} Universitat Pompeu Fabra

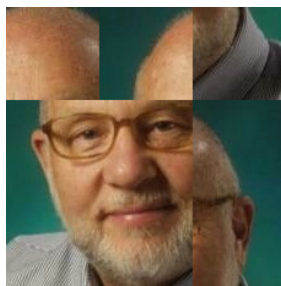
Compiled April 25, 2019

1. PRELIMINARS

A. Exercise 0

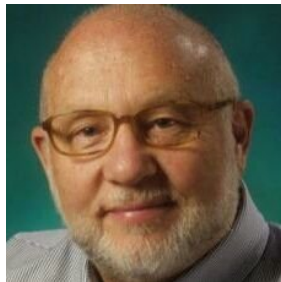


(a) original image



(b) processed image

Fig. 1. RandomPuzzle function input / output.



```
randomPuzzle('cientifico.jpeg')
7     2     4
5     9     8
6     3     1
```

Fig. 2. Matrix output representing the transformation realized.

En la función PuzzleSolver utilizamos esta matriz para reconstruir la imagen original, hacemos uso de la función `find(matriz == valor)` para encontrar en qué posición se encuentra el valor (valores de 1 al 9 indicando qué parte del puzzle es).

2. NORMAL EQUATIONS AS A WAY TO COMPUTE THE MINIMUM OF A LINEAR FUNCTION

A. Exercise 1

B. Exercise 2

Ho satisfà ja que el producte escalar entre r i ImPhi , $\langle r, \text{ImPhi} \rangle$ és un valor que s'aproxima a 0 en els dos casos, ja que el producte escalar de dos vectors és el producte dels mòduls per el coseno de l'angle que formen els dos, quan l'angle es 90 graus o s'aproxima, el coseno es o tendeix a 0, conseqüentment el producte escalar es 0.

3. SINGLE VALUE DECOMPOSITION AND PSEUDO-INVERSE

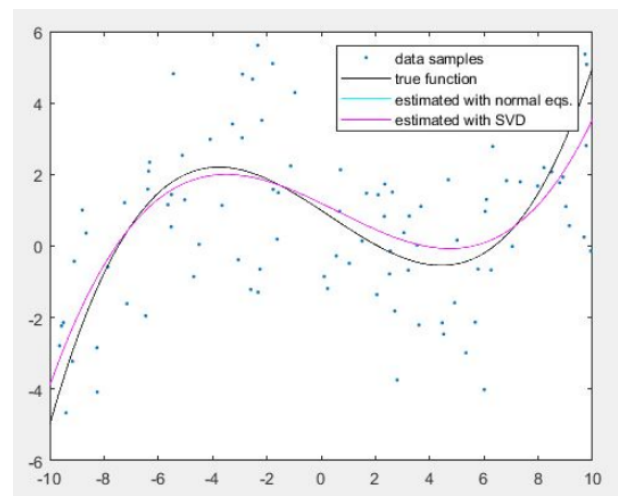
A. Exercise 4

Compute the pseudo-inverse of S , S^\dagger . Use it to compute the pseudo-inverse of A , A^\dagger . Verify that $A^\dagger A$ is the identity matrix of size 3. What happens with AA^\dagger ? Al calcular en matlab AA^\dagger ens dona una matriu 5×5 , degut a que A es 5×3 i que A^\dagger es una matriu 3×5 .

4. SOLVING LEAST SQUARES WITH THE PSEUDO-INVERSE

A. Exercise 5

Complete the MatLab function `polyfit_svd_normal_eq.m` for computing w . Follow the comments provided in the code.



B. Exercise 6

Compare with the solution obtained by inverting the normal equations. Compare also with the solution provided by the MatLab command `pinv`.

El `pinv` retorna la pseudoinversa de Moore-Penrose, que es una matriu que actua com a inversa quan la matriu inversa no existeix

5. PCA

A. Exercise 7

main code of `pca_prin_dir.m`

```
for c=1:m
    mu(1,:) = mu(1,:) + x(c,:);
end
mu = mu/m;
H = H-mu;
% empirical covariance matrix
cov = (transpose(H)*H)/(m-1);
% compute principal directions
[V,L] = eig(cov);
L = abs(sqrt(L));
% This lines are to reverse the order
% of the eigenvectors and eigenvalues
VL = flipud(sortrows([V' diag(L)],n+1));
V = VL(1:n,1:n)';
L = diag(VL(1:n,n+1));
% keep only the p largest eigenvalues
% and eigenvectors
L = L(1:p,1:p);
V = V(:,1:p);
```

B. Exercise 8

main code of `pca_prin_comp.m`

```
for t=1:size(V,2)
    for c=1:m
        out(c,t) = dot(H(c,:), transpose(V(:,t)));
    end
end
```

En el código anterior, se computa el producto escalar entre dos vectores, un eigenvector, y un punto de la muestra centrado en el origen. El resultado es un escalar que representa la magnitud del eigenvector que representa al punto (en vez de representar el punto en bases canónicas se representan en base a los eigenvectors). Este proceso se realiza para cada punto y para cada punto se repite para los diferentes eigenvectors que existan. Los valores escalares que representan los puntos en base a los eigenvectors se almacenan en la variable `out`.

C. Exercise 9

main code of `pca_prin_comp.m`

```
for c=1:m
    for t=1:size(V,2)
        out(c,:) = out(c,:) + transpose(V(:,t))*z(c);
    end
end
```

(la variable `z` contiene la posición de los puntos en base eigenvectors, y `V` contiene los eigenvectors)

En la reconstrucción se aplica el proceso inverso al **Ejercicio 8**, primero se suman todos los eigenvectors que componen la base multiplicados por la magnitud correspondiente en cada punto para obtener los puntos en base canónica, y finalmente se añade la media que fue sustraída para centrar los puntos.

D. Exercise 10

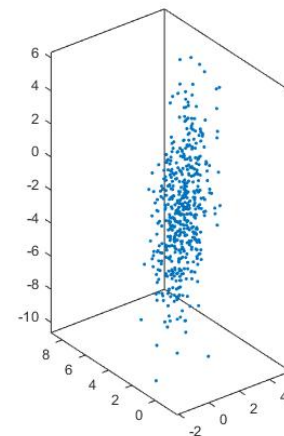


Fig. 3. Scatter plot of points.

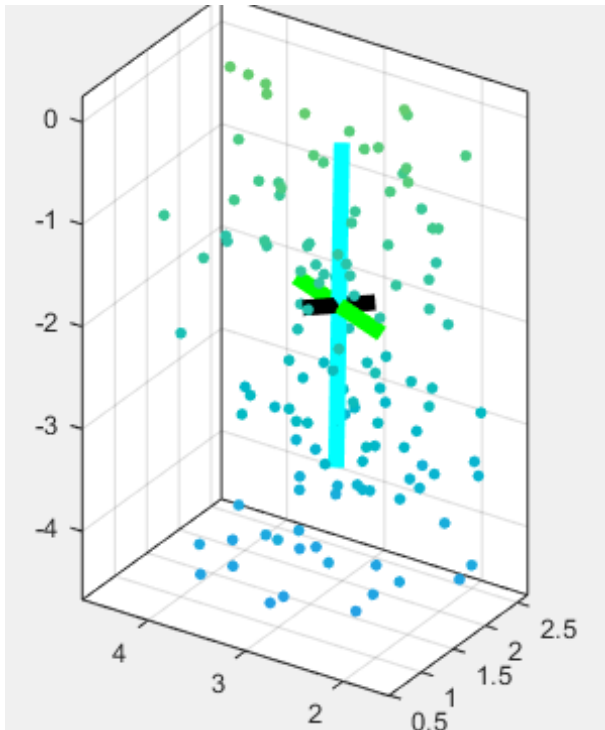


Fig. 4. Axis representing eigenvectors of the sample

la ultima imagen el color indica el componente eigenvector perpendicular a los otros.

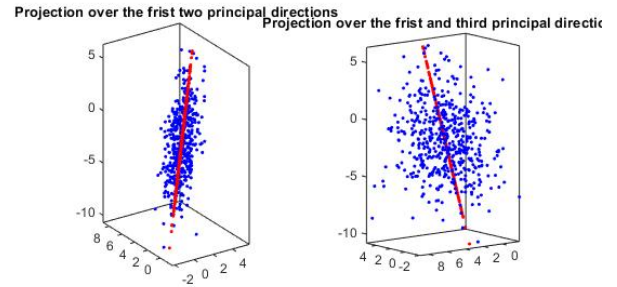


Fig. 6. Projection over axis

En este plot podemos se observa en color rojo los puntos que estan cerca de los dos ejes representados(con valores mas bajos (tendiendo a zero) en el eje no representado), el eje principal y el secundario en el primer plot, y el eje principal con el tercer eje en el segundo plot.

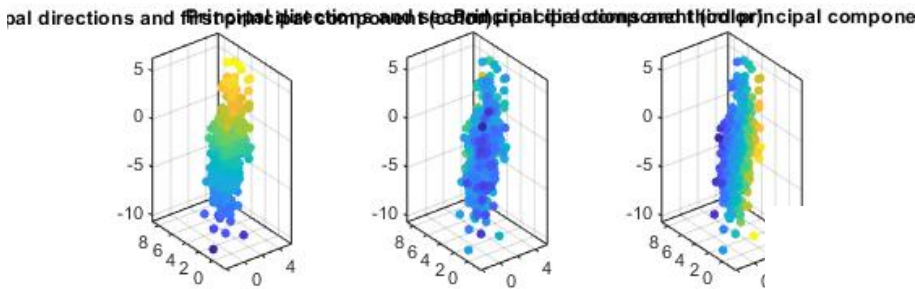


Fig. 5. Sample color representing

En este grafico el color representa la importancia que tiene el eigenvector en representar el punto. Los puntos con pca dejan de estar expresados en base canonica y se expresan en base de eigenvectors, cuando un punto es representado con un mayor valor para un eigenvector el color del punto es mas amarillo. En este caso para un eigenvector vertical en la primera imagen, un eigenvector en profundidad para la imagen intermedia y

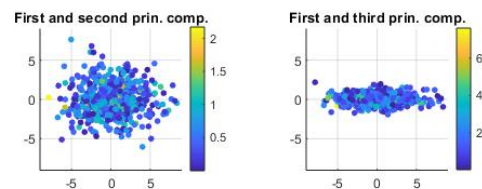


Fig. 7. color graph

La magnitud del eigenvector restante esta representado con el codigo de color, el tercer componente en el primer grafo y el segundo eigenvector en el segundo grafo.

6. HANDWRITTEN DIGIT RECOGNITION

A. Exercise 11

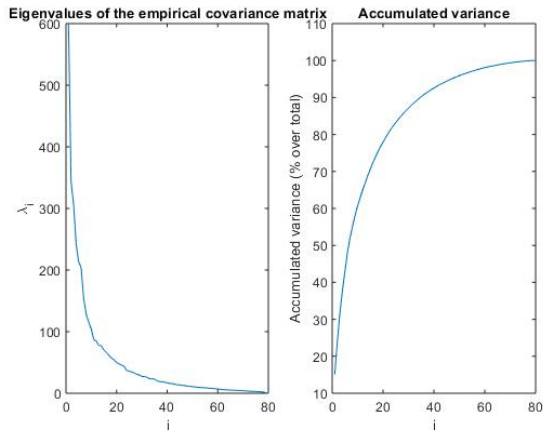


Fig. 8. eigenvalues vs variance

En este grafo se muestra como los eigenvalues y la varianza respecto el eigenvector es inversamente proporcional.

Visualization of principal directions 1 to 6

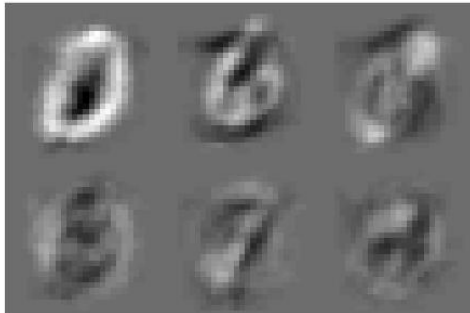
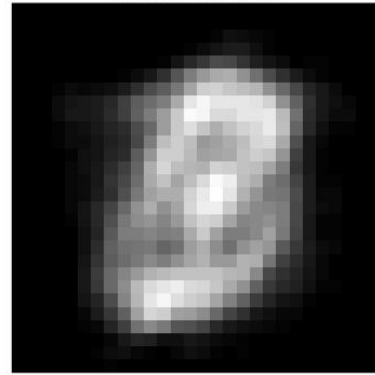


Fig. 9. eigenvectors impact.

Mean handwritten digit



que muestra la media de todas las imagenes.

Fig. 10. mean

Displacement along 3 principal direction

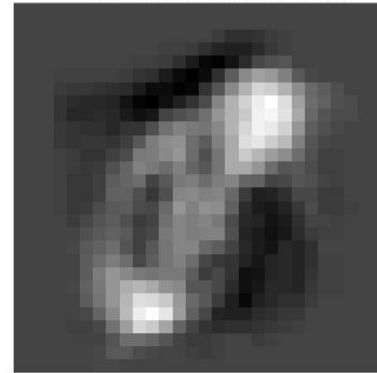


Fig. 11. 3 axis image representation.

Imagen representada solo con 3 eigenvectors.

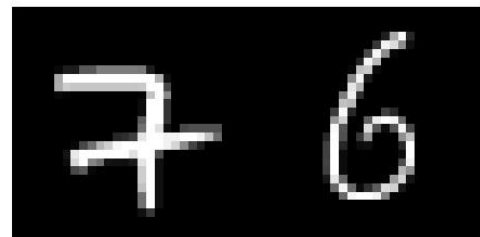


Fig. 12. image recognition

Como se puede observar en este caso el reconocimiento ha fallado.



Fig. 13. image recognition

En este caso el reconocimiento ha acertado el numero.

Aproximadamente 8 de cada 10 imagenes son acertadas, una proporcion que puede ser mejorada si en vez de utilizar los 10 eigenvectors principales (p), se utilizaran mas eigenvectors.