

# Métodos PCA e ICA para separación de imágenes

Cedric Prieels

17 de enero 2017

## Resumen

Tenemos a nuestra disposición tres imágenes que resultan de la mezcla de diferentes maneras de tres imágenes de base. El objetivo de este ejercicio consiste en encontrar o escribir un programa implementando los métodos de PCA y ICA vistos en clase para poder separar estas imágenes mezcladas y volver a encontrar las tres imágenes de base.

## Resolución del ejercicio

Encontré en internet (<https://github.com/xindongzhang/FastICA-for-Image-Separation/tree/master/src>) un programa de Matlab que tiene ya implementado el método fastICA y que permite separar directamente las tres imágenes que tenemos. Solo hay que hacer algunas modificaciones a la macro encontrada, porque de momento, hay que darle tres imágenes y el programa mezcla estas imágenes, antes de volver a separarlas. Por lo tanto, hay que quitar toda la primera parte de este programa. El programa final que nos hace todas las operaciones matemáticas necesarias es el siguiente.

```
%Programa Matlab que permite separar tres imágenes mezcladas, usando un  
%método de ICA.
```

```
clc;  
clear all;  
close all;
```

```
%Cargamos las imágenes proporcionadas
```

```
I1=load('imagenmezclada_uno.dat');  
I2=load('imagenmezclada_dos.dat');  
I3=load('imagenmezclada_tres.dat');
```

```
%Pintamos las imágenes iniciales mezcladas que tenemos
```

```
figure()  
subplot(2,3,1),imshow(I1),title('Imagen inicial 1'),  
subplot(2,3,2),imshow(I2),title('Imagen inicial 2'),  
subplot(2,3,3),imshow(I3),title('Imagen inicial 3'),
```

```
%Operaciones matemáticas para lograr la separación
```

```
[x1,y1,z1]=size(I1);  
[x2,y2,z2]=size(I2);  
[x3,y3,z3]=size(I3);
```

```
s1=[reshape(I1,[1,x1*y1*z1])];  
s2=[reshape(I2,[1,x2*y2*z2])];
```

```

s3=[reshape(I3,[1,x3*y3*z3])];

S_all=[s1;s2;s3];
S=double(S_all);
Sweight=rand(size(S_all,1));
MixedS=Sweight*S

ms1=reshape(MixedS(1,:),[x1,y1,z1]);
ms2=reshape(MixedS(2,:),[x2,y2,z2]);
ms3=reshape(MixedS(3,:),[x3,y3,z3]);

MixedS_bak=MixedS;
MixedS_mean=zeros(3,1);

for i=1:3
    MixedS_mean(i)=mean(MixedS(i,:));
end

for i=1:3
    for j=1:size(MixedS,2)
        MixedS(i,j)=MixedS(i,j)-MixedS_mean(i);
    end
end

MixedS_cov=cov(MixedS');
[E,D]=eig(MixedS_cov);
Q=inv(sqrt(D))*(E)';
MixedS_white=Q*MixedS;
IsI=cov(MixedS_white');
X=MixedS_white;
[VariableNum,SampleNum]=size(X);
numofIC=VariableNum;
B=zeros(numofIC,VariableNum);

for r=1:numofIC
    i=1;maxIterationsNum=150;
    b=2*(rand(numofIC,1)-.5);
    b=b/norm(b);
    while i<=maxIterationsNum+1
        if i == maxIterationsNum
            fprintf('No convergence!\n', r,maxIterationsNum);
            break;
        end
        bOld=b;
        u=1;
        t=X'*b;
        g=t.^3;
        dg=3*t.^2;
        b=((1-u)*t'*g*b+u*X*g)/SampleNum-mean(dg)*b;

        b=b-B*B'*b;
        b=b/norm(b);
        if abs(abs(b'*bOld)-1)<1e-10
            B(:,r)=b;
            break;
        end
        i=i+1;
    end
end

ICAedS=B'*Q*MixedS_bak;
ICAedS_bak=ICAedS;
ICAedS=abs(55*ICAedS);

is1=reshape(ICAedS(1,:),[x1,y1,z1]);
is2=reshape(ICAedS(2,:),[x2,y2,z2]);
is3=reshape(ICAedS(3,:),[x3,y3,z3]);

```

```

I1_icaed =uint8 (round(is1));
I2_icaed =uint8 (round(is2));
I3_icaed =uint8 (round(is3));

%Pintamos las imágenes obtenidas después de todo el proceso

subplot(2,3,4),imshow(I1_icaed),title('Imagen restaurada 1');
subplot(2,3,5),imshow(I2_icaed),title('Imagen restaurada 2');
subplot(2,3,6),imshow(I3_icaed),title('Imagen restaurada 3');

```

El output del programa es la figura siguiente.

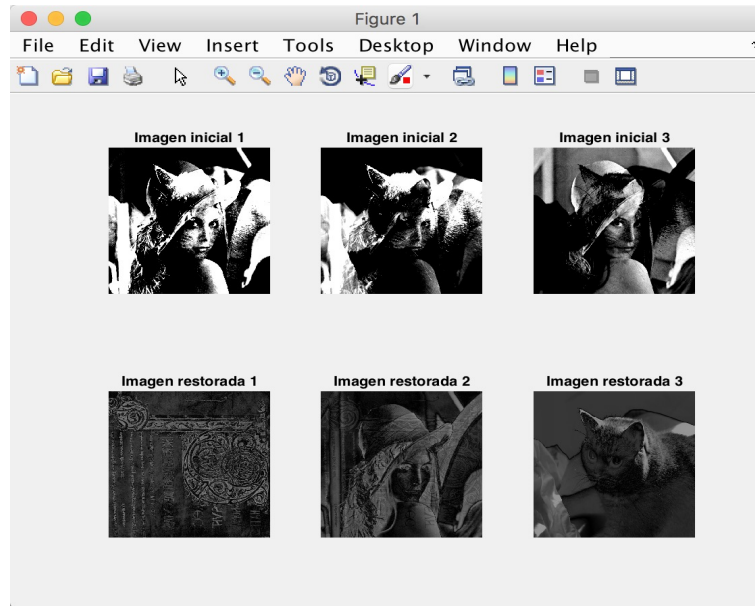


Figura 1: Output del programa Matlab implementado el método fastICA

Vemos claramente que las tres imágenes mezcladas del principio han sido separadas, y que podemos con este método volver a obtener las tres imágenes usadas al principio.