# Mini-project 1: Independent Component Analysis

Hao REN, Han JU

10/26/2012

|  |  |  |
| --- | --- | --- |
| **originDataPoints.png**  Figure 1 : original data points | **mixedDataPoints.png**  Figure 2 : mixed data points | **whitenedDataPoints.png**  Figure 3 : whitened data points |

## Exercise 1: Find a single IC

1. Convergence measurement:

*norm(abs(wNew) - abs(w)) / norm(w) < tol,*where *tol* represent the error tolerance which is very small, say, 1.0e-6

Initially, we use the convergence measurement below:

*norm(wNew - w) / norm(w) > tol,*

However, the problem is that, in some cases, the w vector will converge randomly to two directions which are mutually opposite. They are both the acceptable direction of w vector representing the natural axe, but this convergence measurement will never be satisfied. That's why we use the version with absolute value to fix this problem. As a result, the problem mentioned before will be eliminated.

To simplify the measurement, thanks to the normalization of the old w vector, we could write:

*norm(abs(wNew) - abs(w)) < tol*

1. The function quality measurement

For this part, we use the algorithm with the 3 different non-linear functions to process the same data. We come up with these 2 measurements of quality for comparing them:

1. The average correlation coefficient between the reconstructed image and the original one
2. Number of step of iteration

The first criterion measures the quality of the reconstructed image data with respect to the original one, so the correlation between the two data vectors does perfectly this job.

The second criteria aims at the speed of convergence, the ideal non-linear function should converge fast with the second order Newton method.

A result is showed below:

--> g1:

counter = 6

quality = 0.997689371618089

--> g2:

counter = 6

quality = 0.997633716285635

--> g3:

counter = 7

quality = 0.998010445592914

In most cases, the iteration counters seem to be almost the same (between 5 and 7 iterations for the 3 functions). Sometimes, the counter of g3 is bigger than the one of g2, sometimes not. But the average correlation coefficient of g3 is always the biggest one. The difference between g1 and g2 is negligible. In conclusion, g3 has the best quality of reconstruction among these three functions.

3)

|  |  |
| --- | --- |
| C:\Users\JH\Documents\My Dropbox\EPFL\Unsupervised Learning and Reinforce Learning in Neuron Networks\project\ICA\reconstructedImage1.png | C:\Users\JH\Documents\My Dropbox\EPFL\Unsupervised Learning and Reinforce Learning in Neuron Networks\project\ICA\reconstructedImage2.png |

For reconstructing images using ICA, firstly we should figure out what represent the two natural axes. In fact, the same as the blind signal separation problem mentioned in the lecture, the natural axe for this problem is the axe on which the projection of the pre-processed (centered and whitened) mixture data reveals the original value of one element of the mixture. Thus, we can just project the mixture onto the w vector to get the corresponding original picture.

In this project we find out that the first image is sometimes color-reversed. When it's the case, we will use the reversed w vector to reconstruct the image. That's the appropriate weights. In order to recover the second signal, we just use a vector which is orthogonal to the right w vector. Here, we have also two choices. In the same way, we decide the right orthogonal vector to recover the first signal.

We should point out that the converged w vector might not exactly recover the original signal, since we cannot decide the final w vector is the one which point to the positive (negation) direction of first (second) signal. But we can still recover the information in the original signals by trying the different sign of the w vector and its orthogonal vector

## Ex2: Find all ICs

1) The role of symmetric orthogonalization is to make sure that the w vectors are orthogonal to each other. In fact, after every step of the factICA, the w vectors will lose their orthogonality, which leads to the fact that there is several w vectors point to the same direction. As a result, we will get several identical images in the recovered images set.

1. A set of reconstructed images.

|  |  |  |  |
| --- | --- | --- | --- |
| recvImg1.png  Recovered Image 1 | recvImg2.png  Recovered Image 2 | recvImg3.png  Recovered Image 3 | recvImg4.png  Recovered Image 4 |
| recvImg5.png  Recovered Image 5 | recvImg6.png  Recovered Image 6 | recvImg7.png  Recovered Image 7 | recvImg8.png  Recovered Image 8 |

3) Assign the restored signals to the corresponding sources

Description:

The idea is that by calculating the correlation coefficient between one restored image and the 8 original images, the original picture corresponding to the maximum of the 8 coefficient is the one should be assigned to the restored image.

We iterate the procedure for 8 times, thus, we finally get the assignment of all the reconstructed pictures.

The result according to the restored image sets in 2)

1 3 5 2 7 4 6 8

% File: assign.m %

**function** res **=** assign**(**imgsRecons**,** imgsOrigin**)**

**[**N**,~]** **=** size**(**imgsRecons**);**

max **=** 0**;**

res **=** ones**(**N**,**1**);**

% find the most correlated image to each w vector %

**for** i **=** 1**:**N

**for** j **=** 1**:** N

temp **=** corrcoef**(**imgsRecons**(**i**,:),** imgsOrigin**(**j**,:));**

**if(** abs**(**temp**(**1**,**2**))** **>** max**)**

max **=** abs**(**temp**(**1**,**2**));**

index **=** j**;**

**end**

**end**

res**(**i**)** **=** index**;**

max **=** 0**;**

**end**

**end**

## Conclusion

This mini-project helped us to fully understand the principal ideas of ICA as well as the fastICA algorithm within the context of a concrete application. We are also exposed to some implementation technics of the MatLab platform.