

Assignment 2

IFT 6085, Winter 2013

Date posted: February 19, 2013

Due: February 26, 2013, at the beginning of class

1. (3/20) Show that applying an orthogonal transformation to data that is white leaves the data white.

2. (5/20) Under the ICA generative model, an image patch $I(x, y)$ is generated as the superposition of basis images $A_i(x, y)$ with statistically independent coefficients s_i as

$$I(x, y) = \sum_{i=1}^m A_i(x, y) s_i$$

Show that under this model, the covariance matrix $C((x, y), (x', y'))$ over images $I(x, y)$ takes the form

$$C((x, y), (x', y')) = \sum_i A_i(x, y) A_i(x', y')$$

3. (12/20) In this question, you will implement the online K -means algorithm we discussed in class to learn features from the CIFAR dataset.

Get the file

http://www.iro.umontreal.ca/~memisevr/teaching/mlvis2013/cifar_mini_images_train.txt

and save it locally. The file contains 2000 rows, each of which contains 3072 integers separated by white space. The 3072 integers in each row, when viewed as array of shape $32 \times 32 \times 3$, represent a 32×32 RGB color image from the CIFAR-10 object recognition dataset. (It is a subset of the CIFAR training set).

Crop 10 or more patches of size 6×6 pixels from each of the 2000 images at random locations. Perform canonical preprocessing (mean-centering and contrast normalization) on the patches, followed by PCA whitening *retaining 90% of the variance*.

Run your K -means implementation on the resulting whitened data using $K = 100$.

Make an image showing the 100 learned filters. Since the filters are defined in the low-dimensional whitened space, you will have to multiply by the inverse whitening matrix to display them.

Hints:

- For preprocessing and whitening, simply treat each $6 \times 6 \times 3$ image patch as a 108-dimensional vector.
- As usual, make sure to represent images as float arrays not integer arrays before doing any further processing on them.
- The K -means clustering solution is sensitive to initialization. Usually, a good way to initialize the filters is by using random training examples.
- In Python, you can read txt-files into numpy arrays using “loadtxt(FILENAME)”
- In Python, you may want to reshape the 2000×3072 array before cropping patches or displaying the images, by using:

```
images = images.reshape(2000, 3, 32, 32).transpose(0,2,3,1)
```

- The Pylab function “imshow” will interpret an array of shape $X \times Y \times 3$ as an RGB color image.
- To display multiple color images that are given by a 4D-“tensor” like above, you may use the function defined in

http://www.iro.umontreal.ca/~memisevr/teaching/mlvis2013/dispims_color.py

To display some of the images above, for example, you can use

```
dispims_color.dispims_color(images[:100])
```

- If you prefer (and have a fast enough machine available), you may use the full CIFAR-10 training set at

<http://www.cs.toronto.edu/~kriz/cifar.html>

for this assignment instead of the mini CIFAR subset from the file above. All other parameters (preprocessing, retaining 90% variance, etc.) must be exactly the same.

What to hand in:

- A color image showing the 100 learned K -means filters.
- A color image showing the PCA whitening filters *that you kept* to retain 90% of the variance.
- Answers to the following questions:
 - Which learning rate did you use to train your K -means model?
 - How many iterations did you train it for?

Do *not* hand in any program code.