Math189R SP19
Homework 6
Monday, Apr 1, 2019

Feel free to work with other students, but make sure you write up the homework and code on your own (no copying homework *or* code; no pair programming). Feel free to ask students or instructors for help debugging code or whatever else, though.

The starter files for problem 2 can be found under the Resource tab on course website. Please print out all the graphs generated by your own code and submit them together with the written part, and make sure you upload the code to your Github repository.

1 (Murphy 11.2 - EM for Mixtures of Gaussians) Show that the M step for ML estimation of a mixture of Gaussians is given by

$$\boldsymbol{\mu}_k = \frac{\sum_i r_{ik}\mathbf{x}_i}{r_k}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{r_k}\sum_i r_{ik}(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^\top = \frac{1}{r_k}\sum_i r_{ik}\mathbf{x}_i\mathbf{x}_i^\top - r_k\boldsymbol{\mu}_k\boldsymbol{\mu}_k^\top.$$

We know the negative log likelihood for a mixture of Gaussians is given by:

$$l(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = -\frac{1}{2}\sum_i r_{ik}(\log|\boldsymbol{\Sigma}_k^{|} + (\mathbf{x}_i - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k))$$

We will take the derivative of the NLL with respect to each parametre and set it to 0 for the M step. For $\boldsymbol{\mu}_k$, this yields:

$$\frac{\partial l(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\partial \boldsymbol{\mu}_k} = \frac{1}{2}\sum_i r_{ik}(\boldsymbol{\Sigma}_k^{-1} + \boldsymbol{\Sigma}_k^{-T})(\mathbf{x}_i - \boldsymbol{\mu}_k)$$

Since $\boldsymbol{\Sigma}$ is symmetric, the above becomes:

$$= \sum_i r_{ik}\boldsymbol{\Sigma}_k(\mathbf{x}_i - \boldsymbol{\mu}_k) = 0$$

Therefore,

$$\boldsymbol{\mu}_k = \frac{\sum_i r_{ik}\mathbf{x}_i}{\sum_i r_{ik}} = \frac{\sum_i r_{ik}\mathbf{x}_i}{r_k}$$

We can write $l(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = -\frac{1}{2}\sum_i r_{ik}(\log|\boldsymbol{\Sigma}_k^{|} + \mathrm{tr}[(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}])$. Therefore,

$$\frac{\partial l(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\partial \boldsymbol{\Sigma}_k} = -\frac{1}{2}\sum_i r_{ik}(\boldsymbol{\Sigma}_k^{-T} - ((\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T)^T) = 0$$

Since $\mathbf{\Sigma}_k$ is symmetric,

$$r_k \mathbf{\Sigma}_k = \sum_i r_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T$$

Which yields:

$$\mathbf{\Sigma}_k = \frac{1}{r_k} \sum_i r_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T$$

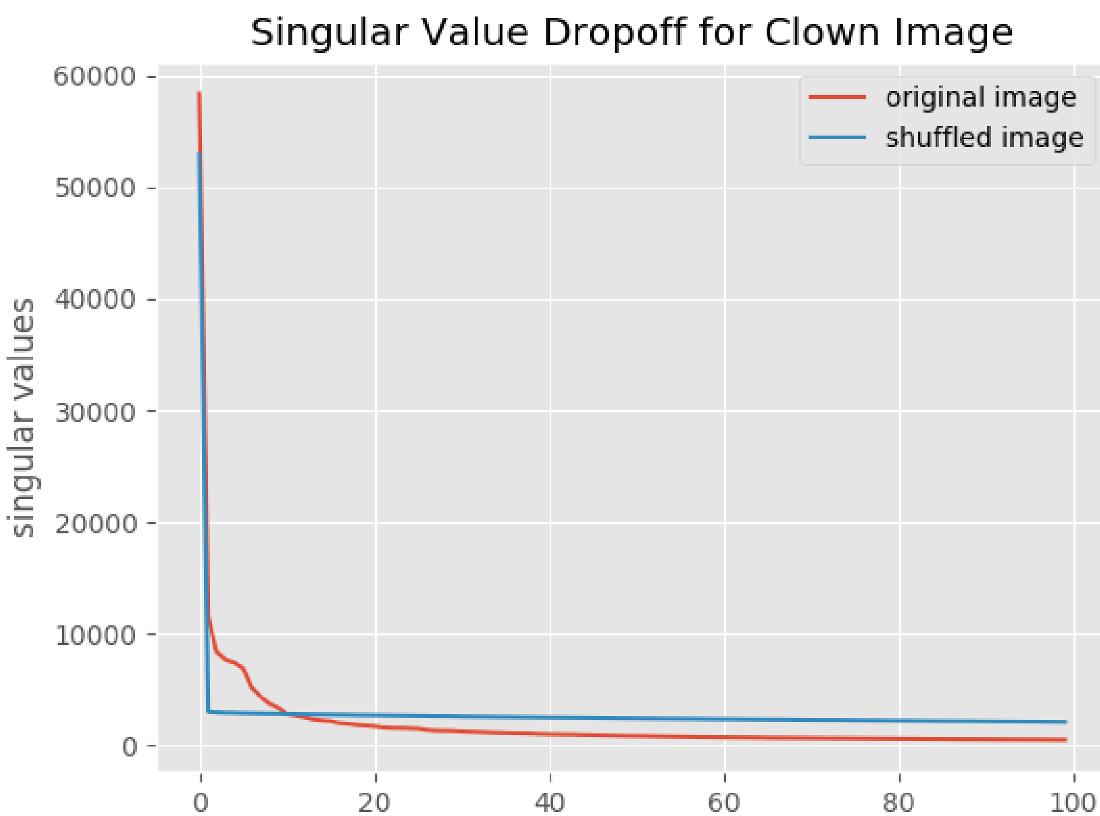
Which is the desired result.

$\blacksquare$

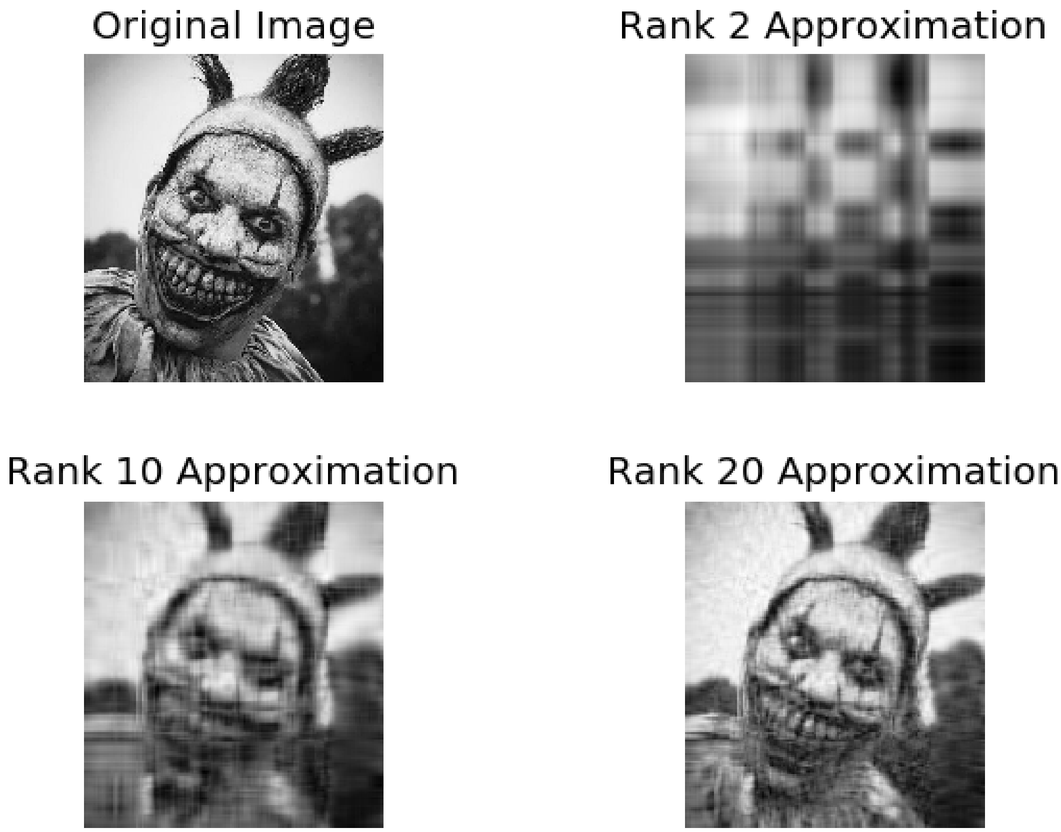
2 (SVD Image Compression) In this problem, we will use the image of a scary clown online to perform image compression. In the starter code, we have already load the image into a matrix/array for you. However, you might need internet connection to access the image and therefore successfully run the starter code. The code requires Python library Pillow in order to run.

Plot the progression of the 100 largest singular values for the original image and a randomly shuffled version of the same image (all on the same plot). In a single figure plot a grid of four images: the original image, and a rank k truncated SVD approximation of the original image for $k \in \{2, 10, 20\}$.

Given below are the plots for the dropoff of singular values for the original and shuffled images:



Below this is the grid of reconstructed images for $k \in \{2, 10, 20\}$


Original Image
Rank 2 Approximation
Rank 10 Approximation
Rank 20 Approximation

∎