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

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

We have the complete data log likelihood $l(\boldsymbol{\mu_k}, \boldsymbol{\Sigma_k}) = \boldsymbol{\Sigma}_k \boldsymbol{\Sigma}_i r_{ik} log \mathbb{P}(\mathbf{x}_i | \boldsymbol{\theta_k})$
$= \frac{-1}{2} \boldsymbol{\Sigma}_i r_{ik} (log|\boldsymbol{\Sigma}_k| + (\mathbf{x}_i - \boldsymbol{\mu_k}^T) \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu_k}))$ to a constant.

Differentiate with respect to $\boldsymbol{\mu_k}$. The $log|_k|$ term becomes a constant that gets eliminated. The $\boldsymbol{\Sigma}_k^{-1}$ term becomes a constant that remains due to similar operations as the fact that for example, $\frac{\partial}{\partial x} x^T A x = 2Ax$.

$\frac{\partial l}{\partial \boldsymbol{\mu_k}} = \boldsymbol{\Sigma}_i r_{ik} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu_k}) = \boldsymbol{\Sigma}_k^{-1} \boldsymbol{\Sigma}_i r_{ik} (\mathbf{x}_i - \boldsymbol{\mu_k}) = 0$ since $\boldsymbol{\Sigma}_k^{-1}$ is linear.

Simplify in order to achieve optimization.
First, $\boldsymbol{\Sigma}_k^{-1}$ can be eliminated as a constant.
Then, $\boldsymbol{\Sigma}_i r_{ik} \mathbf{x}_i = \sum_i r_{ik} \boldsymbol{\mu_k}$.

At optimality, this becomes $\boldsymbol{\Sigma}_i r_{ik} \mathbf{x}_i = \boldsymbol{\mu_k} \boldsymbol{\Sigma}_i r_{ik}$.

Isolate for $\boldsymbol{\mu_k}$, so $\boldsymbol{\mu_k} = \frac{\boldsymbol{\Sigma}_i r_{ik} \mathbf{x}_i}{r_k}$ since $\boldsymbol{\Sigma}_i r_{ik} = r_k$, as desired.

Next, differentiate with respect to $\boldsymbol{\Sigma_k}$, so

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

We did this by using the fact that $\frac{\partial^{-1}}{\partial x} = -\boldsymbol{Y}^{-1} \frac{\partial Y}{\partial x} \boldsymbol{Y}^{-1}$. In this case, Y^{-1} is $\boldsymbol{\Sigma}_k^{-1}$, and x is $\boldsymbol{\Sigma}_k$.

This gives the optimality condition that $\boldsymbol{\Sigma}_i r_{ik} I = (\boldsymbol{\Sigma}_i r_{ik} (\mathbf{x}_i - \boldsymbol{\mu_k})(\mathbf{x}_i - \boldsymbol{\mu_k})^T) \boldsymbol{\Sigma}_k^{-1}$.

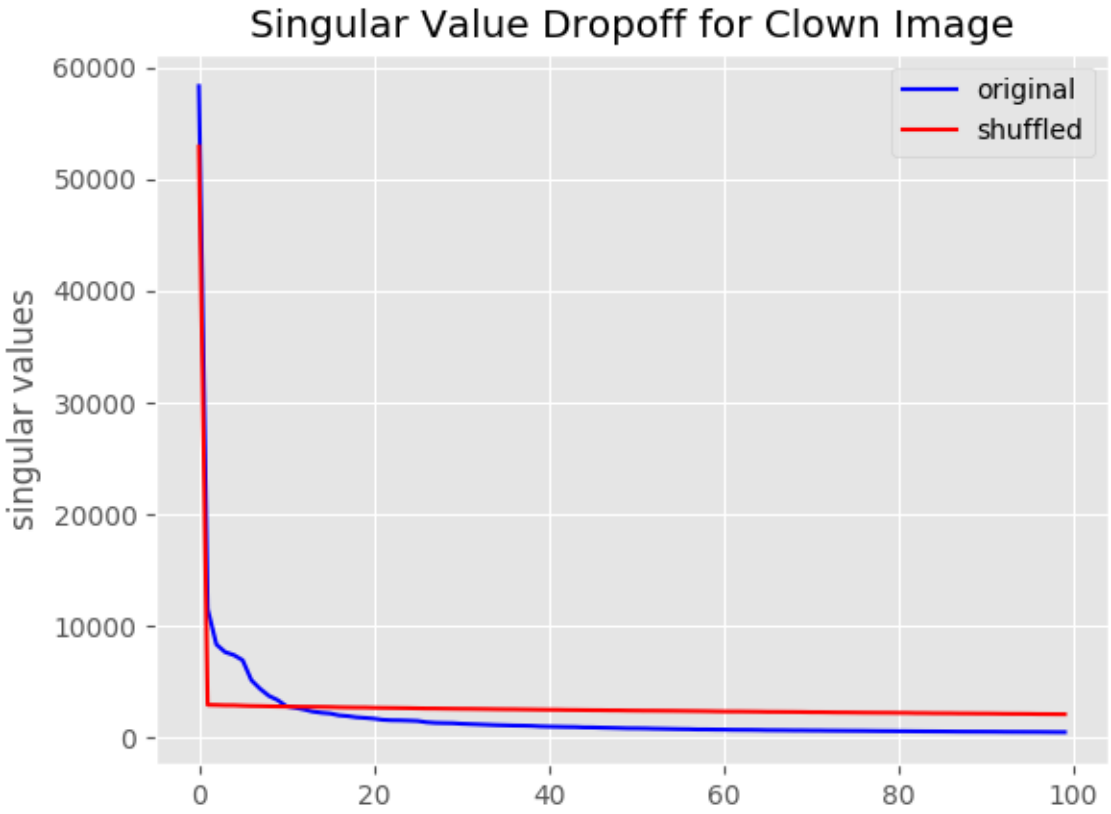
Multiply by $\boldsymbol{\Sigma}_k$ on the right and divide by $r_k = \boldsymbol{\Sigma}_i r_{ik}$ to get

$\frac{\boldsymbol{\Sigma}_i r_{ik} I}{\boldsymbol{\Sigma}_i r_{ik}} = (\boldsymbol{\Sigma}_i r_{ik}(\mathbf{x}_i - \boldsymbol{\mu_k})(\mathbf{x}_i - \boldsymbol{\mu_k})^T)\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\Sigma}_k$, which becomes
$\boldsymbol{\Sigma}_k = (\boldsymbol{\Sigma}_i r_{ik}(\mathbf{x}_i - \boldsymbol{\mu_k})(\mathbf{x}_i - \boldsymbol{\mu_k})^T)$, as desired.

■

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.

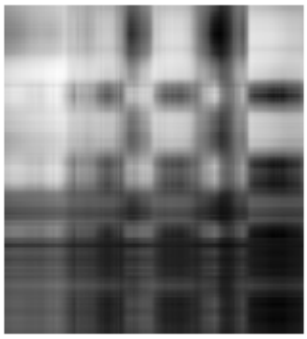
Plot the progression of the 100 largest singular values for the original image and a randomly shuffled version of the same image (all on the same plot). In a single figure plot a grid of four images: the original image, and a rank k truncated SVD approximation of the original image for $k \in \{2, 10,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



The reconstruction plot of the original image and rank 2, 10, and 20 truncated stochastic gradient descent approximations of the original image are below:

Original Image

Rank 2 Approximation



Rank 10 Approximation

Rank 20 Approximation

■