Thitaree Tanprasert (Mint)
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

$$\mu_k = \frac{\sum_i r_{ik} \mathbf{x}_i}{r_k}$$

$$\Sigma_k = \frac{1}{r_k} \sum_i r_{ik} (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^\top = \frac{1}{r_k} \sum_i r_{ik} \mathbf{x}_i \mathbf{x}_i^\top - r_k \mu_k \mu_k^\top.$$

According the textbook section 11.4.2.3, we know that

$$\ell(\mu_k, \Sigma_k) = -\frac{1}{2} \sum_i r_{ik} [\log |\Sigma_k| + (\mathbf{x}_i - \mu_k)^\top \Sigma_k^{-1} (\mathbf{x}_i - \mu_k)]$$

Firstly, we will find optimal parameter μ_k by setting the derivative of the log-likelihood to be 0:

$$0 = \frac{\partial \ell}{\partial \mu_k} = -\frac{1}{2} \sum_i r_{ik} \Sigma_k^{-1} (\mathbf{x}_i - \mu_k)$$

Since $-\frac{1}{2}$ and Σ_k are constants with respect to i, we can discard them and rearranging the equation as follow:

$$\sum_i r_{ik} \mathbf{x}_i = \sum_i r_{ik} \mu_k$$

Finally, we know that $\sum_i r_{ik} = r_k$ by definition, so we have:

$$\mu_k = \frac{\sum_i r_{ik} \mathbf{x}_i}{r_k}$$

as desired.

Secondly, to derive the optimal value of Σ_k, we can take the derivative of the log-likelihood equation with respect to Σ_k instead.

$$0 = \frac{\partial \ell}{\partial \Sigma_k} = -\frac{1}{2} \sum_i r_{ik} (\Sigma_k^{-1} - \Sigma_k^{-1} (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^\top \Sigma_k^{-1})$$

1

Again, we can discard $\frac{1}{2}$ and rearrange the equation to get:

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

Then, we can multiply both sides by $\mathbf{\Sigma}_k$ from the left and by $\mathbf{\Sigma}_k$ from the right to get:

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

Equivalently, we can rearrange the equation and expand the terms to get the desired formula:

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

■

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 $k \in \{2, 10, 20\}$.
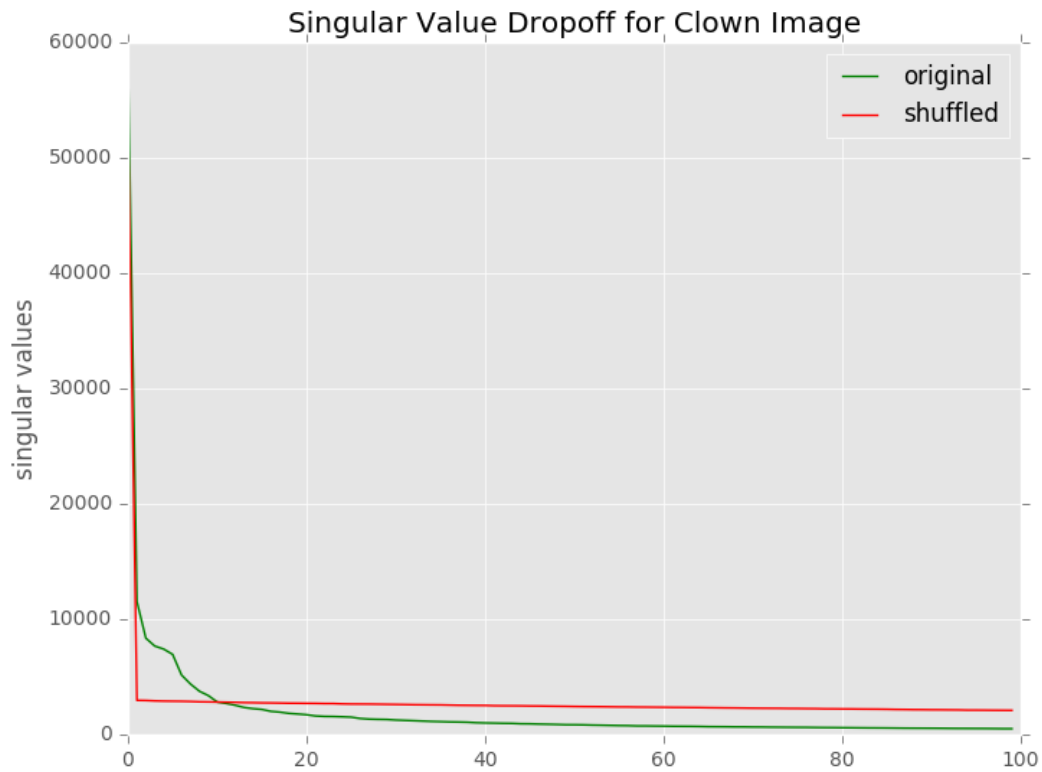
Original Image

Rank 2 Approximation

Rank 10 Approximation

Rank 20 Approximation