APML Image Denosing Report

General notes: I think that the exercise was very interesting. Thank you for that. I think that there are parts that were very difficult to understand because of some mixed up things in lecture and recitation. I spent lots of time to try to understand what is the first model that we are implementing and its likelihood functions. It was not clear that MVN model is simple and no EM is needed.

I am sorry that my report is poor, I had many bugs in the code and I spent too much time on it. Thanks, Sean.

Theoretical part in hand writing in the end of the file – sorry for that.

Practical Exercise:

1. MVN Model:
   a. Log likelihood:
      MVN PDF function is:

$$f(x) = \frac{1}{\sqrt{(2\pi)^d \det \Sigma}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

      We are calculating the Likelihood of *theta = (μ, sigma)* by multiplying the PDF with each *x in X*:

$$\mathcal{L}(\mu, \Sigma) = (2\pi)^{-\frac{np}{2}} \prod_{i=1}^{n} \det(\Sigma)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_i - \mu)^{\mathrm{T}} \Sigma^{-1}(x_i - \mu)\right)$$

      We taking the log of this equation and receiving the log likelihood for single sample x in X:

$$\ln L = -\frac{1}{2}\left[\ln(|\mathbf{\Sigma}|) + (\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}} \mathbf{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) + k \ln(2\pi)\right],$$

      But we want the log likelihood of the whole dataset
   b. Learning:
      In the implemented MVN, as discussed in the Piazza forum, we got a single Gaussian. This means that we can differentiate the log likelihood function and find the MLE of *theta* without using EM. Given X, the training dataset I learnt the following:

$$\hat{\mu}_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i = \bar{x}$$

$$\hat{\Sigma}_{MLE} = \frac{1}{n}\sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T$$

c. Denoising:

For denoising, I simply used Weiner filter. The fact that we got no mixture of Gaussians make it easier and we do not need to calculate the c_i posterior probabilities. So given Y – denoised image, we already learnt *theta = (μ, sigma)* and now we can calculate:

$$Weiner(y) = \left(\Sigma^{-1} + \frac{1}{\sigma^2} I\right)^{-1} \left(\Sigma^{-1}\mu + \frac{1}{\sigma^2} y\right)$$

2. GSM

a. Log likelihood of GMM:

$$l(S, \theta) = \sum_{i=1}^{n} \log\left(\sum_{y=1}^{k} \pi_y \, N(x; \mu_y, \Sigma_y)\right)$$

b. Learning:

I implemented the EM algorithm while estimating r_y, pi_y and $c_{i,y}$ while the covariance matrix is the sample covariance matrix and mean is zero.

c. Denoising:

I implemented Weighted Weiner filter as explained in the exercise pdf.

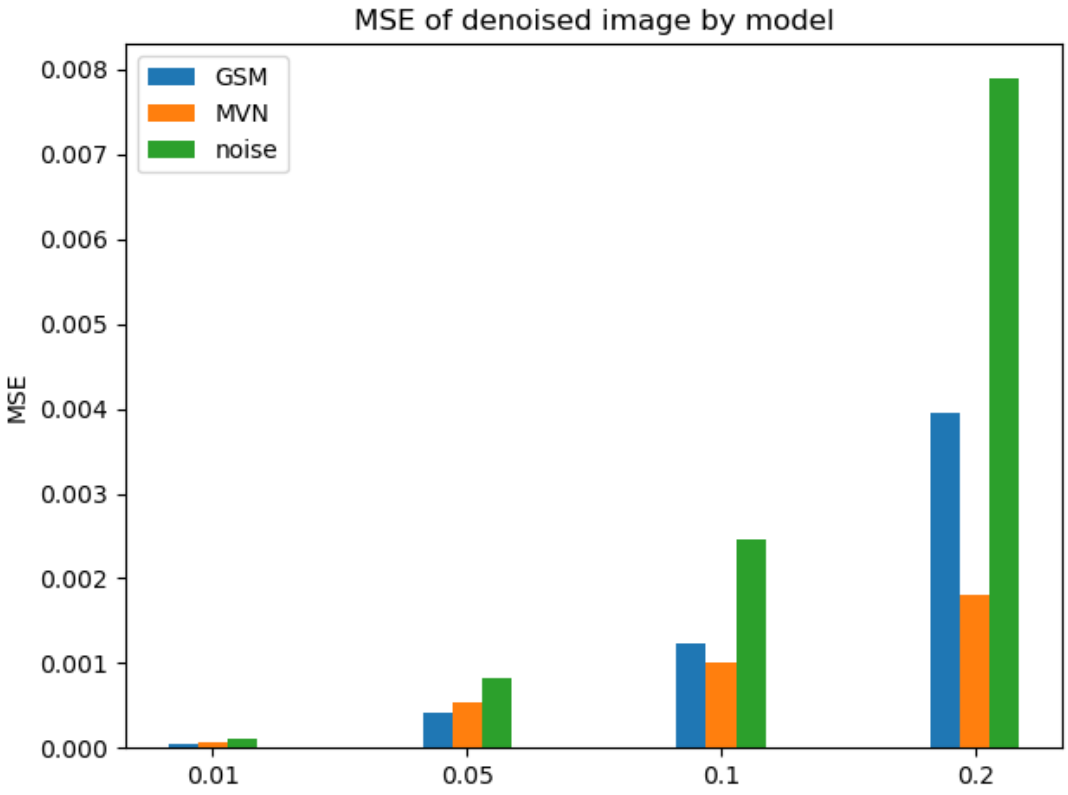
d. Implementation notes:

- In order to implement GSM, I had to implement almost the whole process in log-space to avoid overflow.
- Vectorized notation significantly improves the running time. Unfortunately, I did not had a time to convert the whole code to vectorized notation and I assume it would improve the GSM running time.
- In some points I had to deal with determinant that equals zero (I do not know whether it is a bug or not – I assume it is but do not have time). The zero determinant is damage the PDF calculation (division by zero or taking log of zero) and I had to find a solution. When I tackled this

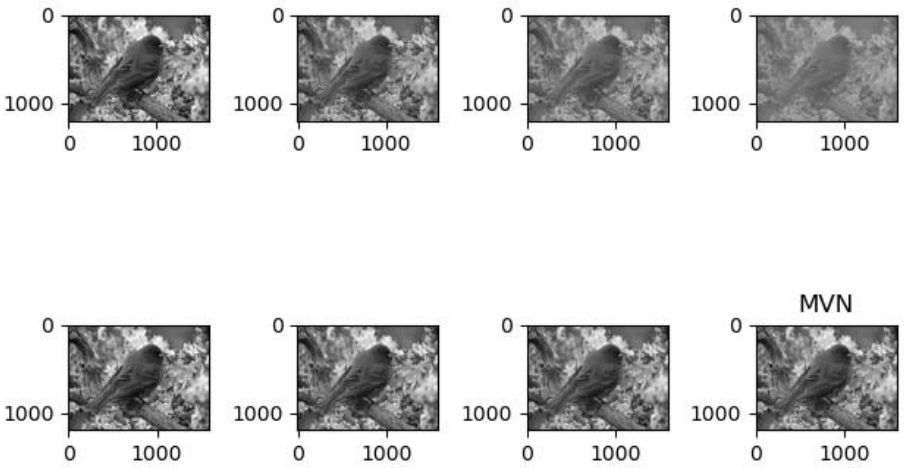
problem during the learning process of the model, I inserted random values instead of damaged PDF values because I wanted to avoid convergence in such state. During the denoising process, I just gave it a very small value.

- I noticed I had a bug with the gsm r scales that I saved it as log and I did not convert it when finished training. When I fixed this bug, the zero determinant issue became very rare and the model performance became much better (reasonable because I multiplied the covariance matrix by the log values of the scales).

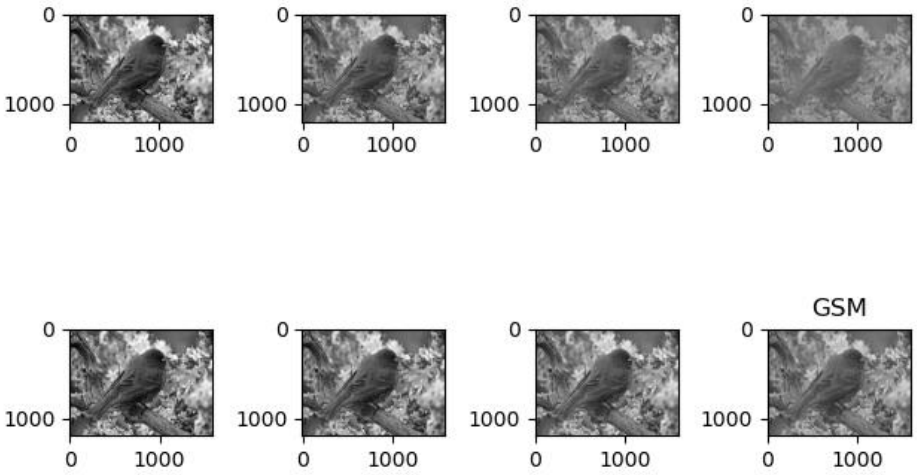
3. Model Comparison
   a. In aspects of denoising performance, there is a close battle between the two models. I think GSM works better (I ran it with five different scales) but MVN sometimes works better than GSM on certain inputs – especially on the high noise inputs.
      - The log likelihood of trained models of the test set is pretty similar, but still it seems to be an advantage to GSM:
        - GSM LL - 1443987.097527962
        - MVN LL - 1428835.197546069
      - The MSE:



- Running Time – It is easy to see and feel that GSM takes lots of time while MVN is very fast. Unfortunately, I do not have time to show it visually.
- Example:
  MVN:



GSM:



שין כהן
203466425

APML תרגיל 2

1) בכיתה הראנו כי

$$E(\ell(S,\theta)) = \sum_{i=1}^{N}\sum_{y=1}^{k} c_{i,y}\log\left(\pi_y N(x_i,\mu_y,\Sigma_y)\right)$$

נזכור כי יש אילוץ על π כך ש $\sum_{y=1}^{k}\pi_y = 1$.

נגדיר פונ' f, g כך:

$$f(\pi) = E(\ell(S,\theta))$$

$$g(\pi) = \sum_{y=1}^{k}\pi_y - 1$$

כעת נבנה לגרנז'יאן:

$$\ell(\pi,\lambda) = E(\ell(S,\theta)) - \lambda\sum_{y=1}^{k}\pi_y + \lambda$$

נגזור ונשווה ל-0 לפי π ואת לפי λ:

$$\frac{\partial \ell}{\partial \pi_y} = \left(\frac{c_{1,y}}{\pi_y} + \frac{c_{2,y}}{\pi_y} + \dots + \frac{c_{n,y}}{\pi_y}\right) - \lambda = \sum_{i=1}^{n}\frac{c_{i,y}}{\pi_y} - \lambda = 0$$

$$\Rightarrow \quad \pi_y = \frac{\sum_{i=1}^{n} c_{i,y}}{\lambda} \qquad \text{נסמן } (*)$$

$$\frac{\partial \ell}{\partial \lambda} = \sum_{y=1}^{k}\pi_y - 1 = 0$$

נשלב את שתי המשוואות:

$$\sum_{y=1}^{k}\pi_y = \frac{\sum_{y=1}^{k}\sum_{i=1}^{n} c_{i,y}}{\lambda} = 1$$

לפי הגדרת $c_{i,y}$ נקבל כי:

$$\sum_{y=1}^{k}\sum_{i=1}^{n} c_{i,y} = \sum_{y=1}^{k}\sum_{i=1}^{n}\frac{\pi_y N(x_i,\mu_y,\Sigma_y)}{\sum_{j=1}^{k}\pi_j N(x_i,\mu_j,\Sigma_j)} = \sum_{i=1}^{n}\frac{\sum_{y=1}^{k}\pi_y N(\dots)}{\sum_{j=1}^{k}\pi_j N(\dots)} = \sum_{i=1}^{n} 1 = n$$

אזי $\lambda = n$ וקיבלנו כי

$$MLE_{\pi_y} = \frac{1}{n}\sum_{i=1}^{n} c_{i,y}$$

לפי $(*)$. ∎

שוב, נגזור $E(\ell(S,\theta))$ לפי r_y עבור y כלשהו.

$$E(\ell(S,\theta)) = \sum_{i=1}^{n}\sum_{y=1}^{k} c_{i,y} \log\left(\mathcal{N}(x, \mu_y, r_y^2\Sigma)\right) =$$

$$= \sum_{i=1}^{n}\sum_{y=1}^{k} c_{i,y} \log\left(\frac{1}{\sqrt{(2\pi r_y^2\Sigma)}} e^{-\frac{1}{2r_y^2} x_i^T\Sigma^{-1}x_i}\right) = \left\{\leftarrow \left((r_y^2\Sigma)^{-1} = (r_y^2)^{-1}\Sigma^{-1} = \frac{\Sigma^{-1}}{r_y^2}\right)\right.$$

$$= \sum_{i=1}^{n}\sum_{y=1}^{k} c_{i,y}\left(\log(1) - \log\left(\sqrt{(2\pi r_y^2\Sigma)}\right) + \left(-\frac{1}{2r_y^2} x_i^T\Sigma^{-1}x_i\right)\right) =$$

$$= \sum_{i=1}^{n}\sum_{y=1}^{k} c_{i,y}\left(-d\log(r_y) - \frac{x_i^T\Sigma^{-1}x_i}{2r_y^2}\right)$$

כדי למצוא נקודות קיצון נגזור ונשווה ל-0 לפי שיטת MLE:

$$\frac{\partial E(\ell(S,\theta))}{\partial r_y} = \sum_{i=1}^{n} c_{i,y}\left(\frac{-d}{r_y} + \frac{x_i^T\Sigma^{-1}x_i}{r_y^3}\right) = 0$$

$$\frac{\sum_{i=1}^{n} c_{i,y}\cdot(-d)}{r_y} + \frac{\sum_{i=1}^{n} c_{i,y}\, x_i^T\Sigma^{-1}x_i}{r_y^3} = \frac{-dr_y^2\sum_{i=1}^{n} c_{i,y}}{r_y^3} + \frac{\sum_{i=1}^{n} c_{i,y}\, x_i^T\Sigma^{-1}x_i}{r_y^3} = 0$$

~~$\sum_{i=1}^{n} c_{i,y}\left(x_i\Sigma^{-1}x_i - dr_y^2\right) = 0 \iff x_i\Sigma^{-1}dr_y^2 = 0$~~

$$dr_y^2\sum_{i=1}^{n} c_{i,y} = \sum_{i=1}^{n} c_{i,y}\, x_i^T\Sigma^{-1}x_i$$

$$r_y^2 = \frac{\sum_{i=1}^{n} c_{i,y}\, x_i^T\Sigma^{-1}x_i}{d\sum_{i=1}^{n} c_{i,y}}$$

1.3) אתחול:

$$\forall y: \quad \pi_y = \frac{1}{k}$$
$$\mu_y = \mu$$
$$\Sigma_y = \Sigma$$

איטרציה 1

$$c_{i,y} = \frac{\frac{1}{k} N(x_i, \mu_y, \Sigma_y)}{\sum_{l=1}^{k} \frac{1}{k} N(x_i, \mu_l, \Sigma_l)}$$

but for all y, l: $N(x_i, \mu_y, \Sigma_y) = N(x, \mu_l, \Sigma_l)$ because of initialization

$$= \frac{\frac{1}{k} X}{\frac{1}{k} \sum_{l=1}^{k} X} = \frac{X}{kX} = \frac{1}{k}$$

$$\pi_y = \frac{1}{N} \sum_{i=1}^{N} c_{i,y} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{k} = \frac{1}{N} \cdot \frac{N}{k} = \frac{1}{k}$$

$$\mu_y = \frac{\sum_{i=1}^{N} \frac{1}{k} x_i}{\sum_{i=1}^{N} \frac{1}{k}} = \frac{\frac{1}{k} \sum_{i=1}^{N} x_i}{\frac{N}{k}} = \frac{k}{Nk} \sum_{i=1}^{N} x_i = \frac{1}{N} \sum_{i=1}^{N} x_i = \mu$$

$$\Sigma_y = \frac{\sum_{i=1}^{N} \frac{1}{k} (x_i - \mu)(x_i - \mu)^T}{\sum_{i=1}^{N} \frac{1}{k}} = \frac{\frac{1}{k} \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T}{\frac{N}{k}} =$$

$$= \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T = \Sigma$$

באיטרציה השניה נשארנו עם משתנים זהים לאתחול,

ולכן עדיין לכל y,l $N(x, \mu_y, \Sigma_y) = N(x, \mu_l, \Sigma_l)$

ולכן שוב נקבל $c_{i,y} = \frac{1}{k}$ ואז נקבל שוב אותם ערכים

עבור π_y, μ_y, Σ_y, ולמעשה אף ערך לא ישתנה.