EX2 - Advanced Practical Course In Machine Learning
Unsupervised Image Denoising

Daniel Afrimi
203865837

November 25, 2020

I was approved for an extension to the exercise by Omri

1 Theoretical Questions

1.1 MLE in the EM algorithm

The expected log-likelihood for Gaussian mixture model is:

$$\mathbb{E}[l(S,\theta)] = \sum_{i=1}^{N}\sum_{y=1}^{k} c_{i,y} \cdot log\left(\pi_y \mathcal{N}\left(x_i;\mu,\sum\nolimits_y\right)\right)$$

We will prove that the mixture weights is:

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

In order to find the value that maximizes the pi_y, We will derive the function according to this variable and find the maximum with the constraint $g(x) = 0$ on given π_y.

$f\left(S,\theta\right) = \mathbb{E}[l(S,\theta)] = \sum_{i=1}^{N}\sum_{y=1}^{k} c_{i,y} \cdot log\left(\pi_y \mathcal{N}\left(x_i;\mu,\sum_y\right)\right)$

$g(x) = \sum_{y=1}^{k}\pi_y - 1$ (Reminder that π_y sums to 1 - distribution vector)

Lagrangian is $L(S,\theta,\lambda) = f\left(S,\theta\right) - \lambda g(S,\theta)$ Therefore we will get:

$$\frac{\partial L(S,\theta,\lambda)}{\partial \pi_j} = \sum_{i=1}^{N}\sum_{y=1}^{k} c_{i,y} \cdot log(\pi_y) + log\left(\mathcal{N}\left(x_i;\mu,\sum\nolimits_y\right)\right) - \lambda\sum_{y=1}^{k}\pi_y - 1$$

$$\Updownarrow$$

$$\sum_{i=1}^{N}\sum_{y=1}^{k} c_{i,y} \cdot log(\pi_y) - \lambda\sum_{y=1}^{k}\pi_y - 1 \text{ (In the first expression the } \pi_y \text{ doesn't appear).}$$

Denote that the derivative according to π_y of this expression $\lambda\sum_{y=1}^{k}\pi_y - 1$ is equal to λ and $\sum_{i=1}^{N}\sum_{y=1}^{k} c_{i,y} \cdot log(\pi_y) = \sum_{i=1}^{N}\frac{c_{i,y}}{\pi_y}$ (log derivative, π_y will appear only on the y Gaussian)

$$\Updownarrow$$

$$\left(\sum_{i=1}^{N}\frac{c_{i,y}}{\pi_y}\right) - \lambda = 0 \Leftrightarrow \left(\sum_{i=1}^{N}\frac{c_{i,y}}{\lambda}\right) = \pi_y \Leftrightarrow \sum_{y=1}^{k}\pi_y\lambda = \sum_{i=1}^{N}\sum_{y=1}^{k} c_{i,y} \Leftrightarrow \lambda = N$$

Therefore we will get:

$$\left(\sum_{i=1}^{N}\frac{c_{i,y}}{\pi_y}\right) - \lambda = 0 =\Leftrightarrow \left(\sum_{i=1}^{N}\frac{c_{i,y}}{\pi_y}\right) - N = 0 \Leftrightarrow \left(\sum_{i=1}^{N}\frac{c_{i,y}}{\mathbb{N}}\right) = \pi_y$$

where λ is a lagrange multiplier ensuring that the mixing sum to 1.

1.2 MLE in the GSM Model

We will find the scalars - r_y (for each Gaussian), that maximize the function above:

$$f\left(S,\theta\right)=\mathbb{E}[l(S,\theta)]=\sum_{i=1}^{N}\sum_{y=1}^{k}c_{i,y}\cdot log\left(\pi_y\mathcal{N}\left(x_i;0,r_y^2\sum_y\right)\right)$$

First, we will simplify this expression:

$$\begin{aligned}
\sum_{i=1}^{N}\sum_{y=1}^{k}c_{i,y}\log\left(\pi_y\mathcal{N}\left(x_i;0,r_y^2\Sigma\right)\right)&=\sum_{i=1}^{N}\sum_{y=1}^{k}c_{i,y}\log\left(\pi_y\cdot\frac{1}{\sqrt{\left|2\pi r_y^2\Sigma\right|}}\exp\left(-\frac{1}{2}x_i^Tr_y^2\Sigma^{-1}x_i\right)\right)\\
&=\sum_{i=1}^{N}\sum_{y=1}^{k}c_{i,y}\left(\log\left(\pi_y\right)+\log\left(\frac{1}{\sqrt{\left|2\pi r_y^2\Sigma\right|}}\exp\left(-\frac{1}{2}x_i^Tr_y^2\Sigma^{-1}x_i\right)\right)\right)\\
&=\sum_{i=1}^{N}\sum_{y=1}^{k}c_{i,y}\left(\log\left(\pi_y\right)+\log\left(\frac{1}{\sqrt{\left|2\pi r_y^2\Sigma\right|}}\right)+\log\left(\exp\left(-\frac{1}{2}x_i^T\Sigma^{-1}x_i\right)\right)\right)\\
&=\sum_{i=1}^{N}\sum_{y=1}^{k}c_{i,y}\left(\log\left(\pi_y\right)+\log\left(\frac{1}{\sqrt{\left|2\pi r_y^2\Sigma\right|}}\right)-\frac{1}{2}x_i^T(r_y^2\Sigma)^{-1}x_i\right)\\
&=\sum_{i=1}^{N}\sum_{y=1}^{k}c_{i,y}\left(\log\left(\pi_y\right)+\log\left(\frac{1}{\sqrt{(2\pi r_y^2)\cdot\det(\Sigma)}^d}\right)-\frac{1}{2}x_i^T\frac{\Sigma^{-1}}{r_y^2}x_i\right)\\
&=\sum_{i=1}^{N}\sum_{y=1}^{k}c_{i,y}\left(\log\left(\pi_y\right)+\log\left(\frac{1}{\sqrt{(2\pi r_y^2)\cdot\det(\Sigma)}^d}\right)-\frac{1}{2r_y^2}\left(x_i^T\Sigma^{-1}x_i\right)\right)\\
&=\sum_{i=1}^{N}\sum_{y=1}^{k}c_{i,y}\left(\log\left(\pi_y\right)-\log\left(\sqrt{(2\pi r_y^2)^d\cdot\det(\Sigma)}\right)-\frac{1}{2r_y^2}\left(x_i^T\Sigma^{-1}x_i\right)\right)\\
&=\sum_{i=1}^{N}\sum_{y=1}^{k}c_{i,y}\left(\log\left(\pi_y\right)-\frac{d}{2}\log\left(2\pi r_y^2\right)-log\left(\sqrt{\det(\Sigma)}\right)-\frac{1}{2r_y^2}\left(x_i^T\Sigma^{-1}x_i\right)\right)
\end{aligned}$$

Now, we will take the derivative according to those scalars. and we will get:

$$\sum_{i=1}^{N}c_{i,y}\cdot\left(-\frac{d}{2}\cdot\frac{1}{2\pi r_y^2}\cdot2\pi+\frac{1}{r_y^4}\cdot\frac{\left(x_i^T\Sigma^{-1}x_i\right)}{2}\right)=\sum_{i=1}^{N}c_{i,y}\cdot\left(-\frac{d}{2r_y^2}\cdot2\pi+\frac{1}{r_y^4}\cdot\left(x_i^T\Sigma^{-1}x_i\right)\right)$$

equating to zero:

$$\begin{gathered}
\sum_{i=1}^{N}c_{i,y}\cdot\left(-\frac{d}{2r_y^2}\right)=\sum_{i=1}^{N}\frac{1}{2r_y^4}\cdot\left(x_i^T\Sigma^{-1}x_i\right)\cdot c_{i,y}\\
\Updownarrow\\
\sum_{i=1}^{N}c_{i,y}\cdot d\cdot r_y^2=\sum_{i=1}^{N}c_{i,y}\cdot\left(x_i^T\Sigma^{-1}x_i\right)\\
\Updownarrow\\
r_y^2=\frac{\sum_{i=1}^{N}c_{i,y}\cdot\left(x_i^T\Sigma^{-1}x_i\right)}{d\cdot\sum_{i=1}^{N}c_{i,y}}
\end{gathered}$$

1.3 EM Initialization

for all y $\pi_y=\frac{1}{k},\mu_y=\mu,\sum_y=\sum$
Which means that for any sample, the Gaussian's would give the same output.

After the first iteration, all the Gaussian's will share the same updated parameters.
Formally $\forall \mathrm{y}_1, y_2 \in [k]$:

$$\begin{aligned} c_{i,y_1} = c_{i,y_2} &= \tfrac{1}{k} \\ \pi_{y_1} = \pi_{y_2} &= \tfrac{1}{k} \\ \mu_{y_1} = \mu_{y_2} &= \frac{\sum_{i=1}^{N} x_i}{N} \\ \sum_{y_1} = \sum_{y_2} &= \frac{\sum_{i=1}^{N} \left(x_i - \frac{\sum_{i=1}^{N} x_i}{N} \right) \cdot \left(x_i - \frac{\sum_{i=1}^{N} x_i}{N} \right)^T}{N} \end{aligned}$$

Because all the Gaussian's share the same parameters, in the seconds iterations, the updated parameters will be exactly the same as above.
The EM Algorithm will converge after two iterations (because we didn't change any value of the updated parameters), though the result are meaningless and we didn't learn nothing, if we will initialize the Gaussian's with the same parameters.

2 Models Comparison

2.1 Learning Run Time

I sampled 10,000 patches, and each model learned on those patches.

MVN Runtime is: 0.0061528682708740234 seconds

GSM (with K=6 gaussains) Runtime is: 2.687565803527832 seconds
Testing runtime was for MVN 1.258 seconds. and for GSM 9.430 seconds

2.2 MSE of the reconstruction - Test Images

When testing the models on the test set, we have got that the image denoising by both of the models decrease the MSE. Those are the results of each model with different noises.

MVN Model:

noisy MSE for noise = 0.01: 9.940963489542536e-05
denoised MSE for noise = 0.01: 9.81697335008661e-05

noisy MSE for noise = 0.05: 0.0024959006261331923
denoised MSE for noise = 0.05: 0.0009458099252639308

noisy MSE for noise = 0.1: 0.009998222421297208
denoised MSE for noise = 0.1: 0.001731135570114278

noisy MSE for noise = 0.2: 0.04002684645373435
denoised MSE for noise = 0.2: 0.00283920270909617

Denoise the patches with MVN took in avg 0.4320838451385498 seconds.

GSM Model:

noisy MSE for noise = 0.01: 9.989594872223444e-05
denoised MSE for noise = 0.01: 6.279260482011952e-05

noisy MSE for noise = 0.05: 0.0024996985429824163
denoised MSE for noise = 0.05: 0.0005002815424208118

noisy MSE for noise = 0.1: 0.009984753226575539
denoised MSE for noise = 0.1: 0.0009941064627784903

noisy MSE for noise = 0.2: 0.040110396801731
denoised MSE for noise = 0.2: 0.001983371562890489

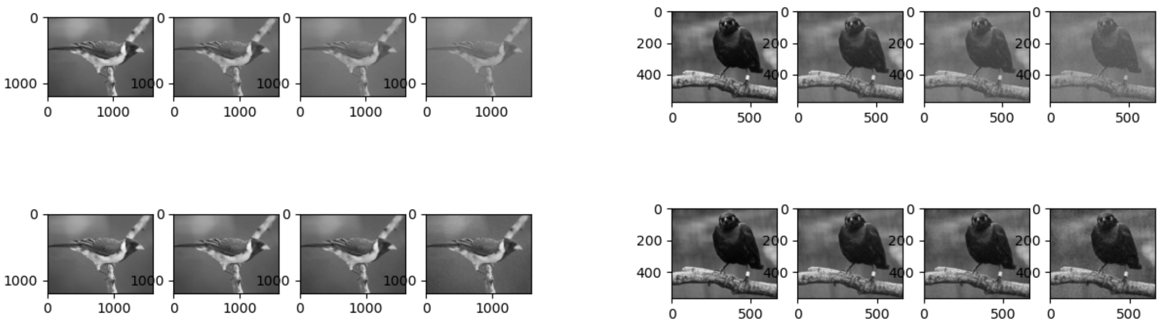
Denoise the patches with GSM took in avg 4.391088962554932 seconds.

2.3 Log-Likelihood

In Figures 2 we can see the Likelihood changes during the train of the GSM Model (using EM algorithm). In the EM algorith is used epsilon = 0.001 (for convergence).

2.4 Image Denoising

In Figures 1 we can see how the models denoised the noisy image (with different noises).



(a) MVN Denoising

(b) GSM Denoising

Figure 1: Denoised image by different models

2.5 Best Model

There is no unequivocal answer. The training is faster for MVN, so if we training the models on a large amount of patches, this is important. In terms of performance The MSE (the error) is smaller for MVN, but in terms of likelihood GSM is better. In Addition, visually the image Denoising looks similar for both models, perhas MVN still quite noisy, therefore GSM model is better than the MVN.

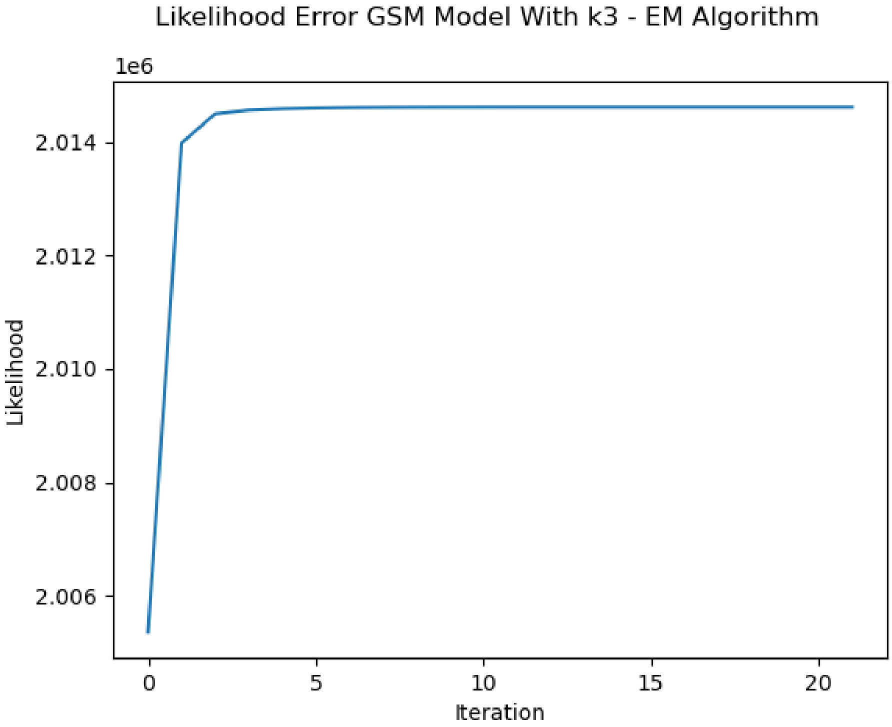


Figure 2: Likelihood of GSM during training