In the Name of God

Introduction to Machine Learning (25737-2)

Project Phase 1

Spring Semester 1401-02

Department of Electrical Engineering

Sharif University of Technology

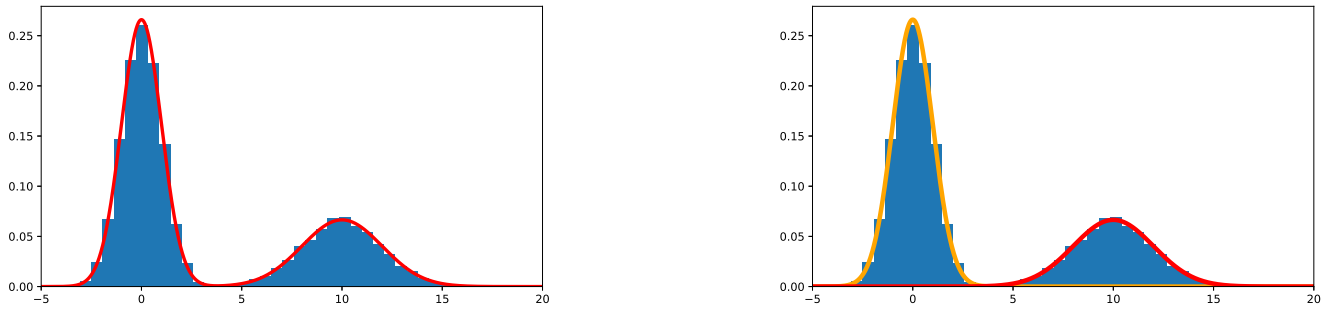
Instructor: Dr. S. Amini

Due on Farvardin 25, 1402 at 23:55

(∗) Should you have any questions concerning the project, please feel free to ask via Telegram.

1 Introduction to Mixture Models

In the lecture notes (and probably in other courses!), you have learned about different probability distributions. But what if the data distribution wasn't that simple, i.e. we couldn't fit any distributions of the forms Gaussian, Laplace, Categorical, Student t, etc.? In figure 1a you may see a simple example for that case. One may consider this distribution as two Gaussian distributions mixed with each other to create this form. This is shown in figure 1b.



(a) (b)

Figure 1

In order to create much more complex probability models, yet taking advantages of the common distributions, we introduce "Mixture Models". This notion is nothing but creating a convex combination of some distributions. A Mixture Model could be written as:

$$p_{\mathbf{Y}}(\mathbf{y};\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{y};\boldsymbol{\theta}_k). \tag{1}$$

Clearly in order to have a valid probability distribution, it should be the case that for all k, $0 \leq \pi_k \leq 1$ and also $\sum_{k=1}^{K} \pi_k = 1$. We call each p_k, a component of the mixture model. In the most general case, each $p_k(\mathbf{y})$ could be any distribution. But in most cases, we prefer them to be similar to each other, for example we choose all of them to be Normal distribution but with different μ and σ.

Another way to look at mixture models is using "latent variables". Let us define a new random variable, Z. This is what we call it "latent variable". This random variable can take any of the values $\{z_k\}_{k=1}^K$, each with probability π_k. This means: $z \sim \text{Cat}(\boldsymbol{\pi})$.

Now consider conditional random variable $\mathbf{Y}|Z$. We have:

$$p_{\mathbf{Y}|Z}(\mathbf{y}|z_k) = p_{\mathbf{Y}|Z}(\mathbf{y}|Z = z_k) = p_k(\mathbf{y}) = p(\mathbf{y}; \boldsymbol{\theta}_k). \tag{2}$$

So if we put the two aforementioned distributions together, we would have:

$$\begin{cases} p_Z(z; \boldsymbol{\theta}) = p_Z(z) = \text{Cat}(z; \boldsymbol{\pi}) \\ p_{\mathbf{Y}|Z}(\mathbf{y}|Z = z_k; \boldsymbol{\theta}) = p_k(\mathbf{y}; \boldsymbol{\theta}_k) \end{cases} \tag{3}$$

Thus we have:

$$p_{\mathbf{Y}}(\mathbf{y}; \boldsymbol{\theta}) = \sum_{k=1}^{K} p_Z(z_k; \boldsymbol{\theta}) p_{\mathbf{Y}|Z}(\mathbf{y}|Z = z_k; \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{y}; \boldsymbol{\theta}_k) \tag{4}$$

The later result is nothing but what we defined as "Mixture Model".

Let us explain this notion more, through an example. Suppose that we have a dice. Also we have 6 gaussian distributions, each with a zero mean and their variance matches the faces of that dice. First, we throw the dice and based on the result, we choose one of the gaussian distributions. Next, we generate a random number using that distribution. If we call the random variable that is the result of throwing a dice and the final random number, Z and X respectively, based on the law of total probability, we have:

$$p_X(x) = \sum_{k=1}^{6} p_Z(z = k) p_{X|Z}(x|Z = k) = \sum_{k=1}^{6} \pi_k \mathcal{N}(x; 0, \sigma_k^2) = \sum_{k=1}^{6} \pi_k p_k(x) \tag{5}$$

You may consider this model as a hierarchical model: We first choose Z, then based on that, we choose a distribution and then generate a random variable. This means that each X is actually generated only based on one normal distribution, if we know which of those normal distributions we have chosen. This also means that we have a Z corresponding to each X generated.

In the definition of mixture models, we may change the base distribution p_k, to create wide variety of mixture models. Now we will discuss two examples of mixture models:

- Gaussian Mixture Models

  A Gaussian mixture model or GMM, also called a mixture of Gaussians (MoG), is defined as follows:

  $$p_{\mathbf{Y}}(\mathbf{y}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{6}$$

- Categorical Mixture Models

  Categorical mixture model(CMM) is defined as:

  $$p_{\mathbf{Y}}(\mathbf{y}; \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \text{Cat}(\mathbf{y}; \boldsymbol{\theta}_k), \tag{7}$$

  where: $\boldsymbol{\theta} = \left\{ \boldsymbol{\pi} = \begin{bmatrix} \pi_1 \\ \vdots \\ \pi_K \end{bmatrix}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K \right\}$

- Mixtures of Bernoullis

  If the data is binary valued, we can use a mixture of Bernoullis, where each mixture component has the following form:

$$p_{\mathbf{Y}}(\mathbf{y}|Z=k;\boldsymbol{\theta}) = \prod_{d=1}^{D} \text{Ber}(y_d;\mu_{k,d}) = \prod_{d=1}^{D} \mu_{k,d}^{y_d}(1-\mu_{k,d})^{1-y_d}. \tag{8}$$

  Here $\mu_{k,d}$ is the probability that bit d of cluster k turns on.

2 Expectation Maximization

Finding the MAP/MLE estimates for the parameters of a mixture model requires solving for two sets of unknowns simultaneously: the latent variables of the data (e.g. which cluster a datapoint belongs to) and the parameters of each mixture. Solving this problem directly is often hard. Suppose Z denotes the cluster each sample belongs to, and $\mathbf{X}$ denotes the observations. Our aim is to maximize the likelihood function:

$$\ln\left(p_{\mathbf{Y}}(\mathbf{y};\boldsymbol{\theta})\right) = \sum_{i=1}^{N} \ln\left(p_Y(y^{(i)};\boldsymbol{\theta})\right) = \sum_{i=1}^{N} \ln\left(\sum_{k=1}^{K} p_{Y,Z}(y^{(i)}, Z^{(i)}=k;\boldsymbol{\theta})\right) \tag{9}$$

The sum inside the logarithm is what makes this function hard to optimize. If we knew the value of Z beforehand, we wouldn't need the sum because we would know which density function the sample comes from.

The idea behind the EM algorithm is to estimate the hidden variables Z in the **E step** (expectation step), and then using the completed data to compute the MLE for model parameters during the **M step** (maximization step).

It is not clear whether this procedure will converge or not. We first discuss a class of optimization algorithms known as **bound optimization** or **MM** algorithms. MM stands for **majorize-minimize** and **minorize-maximize** in the contexts of minimization and maximization, respectively. Then, we describe the full EM algorithm and show that it is an MM algorithm, which implies that it converges to a local maximum of the log likelihood.

2.1 The MM Principle

We assume our goal is to maximize some function $l(\boldsymbol{\theta})$, such as the log likelihood, with respect to its parameters, $\boldsymbol{\theta}$. The basic approach in MM algorithms is to construct a **surrogate function** $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$ which is a tight lowerbound to $l(\boldsymbol{\theta})$ such that $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) \leq l(\boldsymbol{\theta})$ and $Q(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)}) = l(\boldsymbol{\theta}^{(t)})$. If these conditions are met, we say that Q minorizes l. We then perform the following update at each step:

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}). \tag{10}$$

This guarantees us monotonic increases in the original objective:

$$l(\boldsymbol{\theta}^{(t+1)}) \geq Q(\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\theta}^{(t)}) \geq Q(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)}) = l(\boldsymbol{\theta}^{(t)}). \tag{11}$$

.

This process is sketched in Fig.2 The dashed red curve is the original function (e.g., the log-likelihood of the observed data). The solid blue curve is the lower bound, evaluated at $\boldsymbol{\theta}^{(t)}$; this touches the objective function at $\boldsymbol{\theta}^{(t)}$. We then set $\boldsymbol{\theta}^{(t+1)}$ to the maximum of the lower bound (blue curve), and fit a new bound at that point (dotted green curve). The maximum of this new bound becomes $\boldsymbol{\theta}^{(t+2)}$, etc.

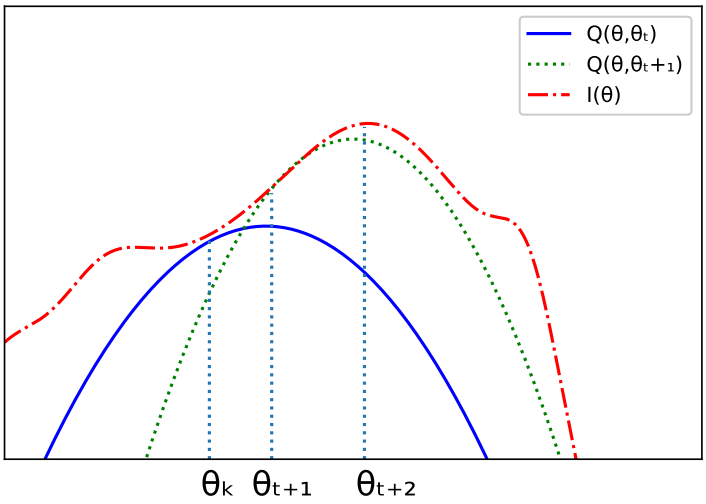


Figure 2: Illustration of a bound optimization algorithm.

2.2 The EM Algorithm

2.2.1 Introduction

The goal of EM is to maximize the log likelihood of the observed data:

$$l(\boldsymbol{\theta}) = \ln\left(p_{\mathbf{Y}}(\mathbf{y};\boldsymbol{\theta})\right) = \ln\left(\prod_{n=1}^{N} p_{\mathbf{Y}}\left(\mathbf{y}_n;\boldsymbol{\theta}\right)\right) = \sum_{n=1}^{N} \ln\left(p\left(\mathbf{y}_n;\boldsymbol{\theta}\right)\right) = \sum_{n=1}^{N} \ln\left(\sum_{\mathbf{z}_n} p_{\mathbf{Y},\mathbf{Z}}\left(\mathbf{y}_n,\mathbf{z}_n;\boldsymbol{\theta}\right)\right) \tag{12}$$

where $\mathbf{y}_n$ are the our observations (samples in the case of mixture models) and $\mathbf{z}_n$ is the vector of hidden variables (cluster of each sample in the case of mixture models). Note that we sum over all possible values of $\mathbf{z}_n$ because we assume that it is easier to estimate $p_{\mathbf{Y},\mathbf{Z}}\left(\mathbf{y}_n,\mathbf{z}_n \mid \boldsymbol{\theta}\right)$ than $p_{\mathbf{Y}}\left(\mathbf{y}_n \mid \boldsymbol{\theta}\right)$ directly. Unfortunately, this is still hard to optimize, since the log cannot be pushed inside the sum.

EM gets around this problem by maximizing a tight lower bound given by the Jensen's inequality:

$$f(\sum_{i=1}^{n} \lambda_i x_i) \geq \sum_{i=1}^{n} \lambda_i f(x_i). \tag{13}$$

Where f is a concave function, $\lambda_i \geq 0$, and $\sum_{i=1}^{n} \lambda_i = 1$. Because ln is a concave function, pushing it inside the sum gives a lower bound for the likelihood function.

Working with probabilities containing $\mathbf{z}_n$ requires knowledge of $\mathbf{Z}_n$'s probability distribution. Thus, we need to estimate the probability distribution for $\mathbf{Z}_n$ as well. First consider a set of arbitrary distributions $q_n(\mathbf{z}_n)$ over each hidden variable $\mathbf{z}_n$. The observed data log likelihood can be written as follows:

$$l(\boldsymbol{\theta}) = \sum_{n=1}^{N} \ln\left(\sum_{\mathbf{z}_n} q_n\left(\mathbf{z}_n\right) \frac{p_{\mathbf{Y},\mathbf{Z}}\left(\mathbf{y}_n,\mathbf{z}_n;\boldsymbol{\theta}\right)}{q_n\left(\mathbf{z}_n\right)}\right). \tag{14}$$

Using Jensen's inequality:

$$\begin{aligned} l(\boldsymbol{\theta}) &\geq \sum_n \sum_{\mathbf{z}_n} q_n(\mathbf{z}_n) \ln\left(\frac{p_{\mathbf{Y},\mathbf{Z}}(\mathbf{y}_n, \mathbf{z}_n; \boldsymbol{\theta})}{q_n(\mathbf{z}_n)}\right) \\ &= \sum_n \underbrace{\mathbb{E}_{q_n}\left[\ln\left(p_{\mathbf{Y},\mathbf{Z}}(\mathbf{y}_n, \mathbf{z}_n; \boldsymbol{\theta})\right)\right] + \mathbb{H}(q_n)}_{\mathcal{E}(\boldsymbol{\theta}, q_n | \mathbf{y}_n)} \\ &= \sum_n \mathcal{E}(\boldsymbol{\theta}, q_n \mid \mathbf{y}_n) \triangleq \mathcal{E}(\boldsymbol{\theta}, \{q_n\} \mid \mathcal{D}) \end{aligned} \tag{15}$$

where $\mathbb{H}(q)$ is the entropy of probability distribution q, and $\mathcal{E}(\boldsymbol{\theta}, \{q_n\} \mid \mathcal{D})$ is called the **evidence lower bound** or **ELBO**, since it is a lower bound on the log marginal likelihood, $\ln(p(\mathbf{y}|\boldsymbol{\theta}))$, also called the evidence.

The EM algorithm alternates between maximizing **ELBO** with respect to the distributions q_n in teh E step, and the model parameters $\boldsymbol{\theta}$ in the M step.

2.2.2 The E Step

$$\begin{aligned} \mathcal{E}(\boldsymbol{\theta}, q_n \mid \mathbf{y}_n) &= \sum_{\mathbf{z}_n} q_n(\mathbf{z}_n) \ln\left(\frac{p_{\mathbf{Y},\mathbf{Z}}(\mathbf{y}_n, \mathbf{z}_n; \boldsymbol{\theta})}{q_n(\mathbf{z}_n)}\right) \\ &= \sum_{\mathbf{z}_n} q_n(\mathbf{z}_n) \ln\left(\frac{p_{\mathbf{Z}|\mathbf{Y}}(\mathbf{z}_n \mid \mathbf{y}_n; \boldsymbol{\theta}) p_{\mathbf{Y}}(\mathbf{y}_n; \boldsymbol{\theta})}{q_n(\mathbf{z}_n)}\right) \\ &= \sum_{\mathbf{z}_n} q_n(\mathbf{z}_n) \ln\left(\frac{p_{\mathbf{Z}|\mathbf{Y}}(\mathbf{z}_n \mid \mathbf{y}_n; \boldsymbol{\theta})}{q_n(\mathbf{z}_n)} + \sum_{\mathbf{z}_n} q_n(\mathbf{z}_n) \ln\left(p_{\mathbf{Y}}(\mathbf{y}_n; \boldsymbol{\theta})\right)\right) \\ &= -D_{\mathrm{KL}}\left(q_n(\mathbf{z}_n) \| p_{\mathbf{Z}|\mathbf{Y}}(\mathbf{z}_n \mid \mathbf{y}_n; \boldsymbol{\theta})\right) + \ln\left(p_{\mathbf{Y}}(\mathbf{y}_n; \boldsymbol{\theta})\right) \end{aligned} \tag{16}$$

where $D_{\mathrm{KL}}(q\|p) = \sum_z q(z) \log(\frac{q(z)}{p(z)})$ is the KL divergence between probability distributions q and p. We know two properties of the KL divergence between distributions q and p:

$$D_{\mathrm{KL}}(q\|p) \geq 0 \tag{17}$$

$$D_{\mathrm{KL}}(q\|p) = 0 \Leftrightarrow q = p. \tag{18}$$

Hence we can maximize the lower bound $\mathcal{E}(\boldsymbol{\theta}, q_n \mid \mathbf{y}_n)$ with respect to q_n by setting each one to $q_n^* = p_{\mathbf{Z}|\mathbf{Y}}(\mathbf{z}_n \mid \mathbf{y}_n; \boldsymbol{\theta})$. This is called the **E step**. This ensures the ELBO is a tight lower bound:

$$\mathcal{E}(\boldsymbol{\theta}, \{q_n^*\} \mid \mathcal{D}) = \sum_n \ln\left(p_{\mathbf{Y}}(\mathbf{y}_n; \boldsymbol{\theta})\right) = l(\boldsymbol{\theta} \mid \mathcal{D}) \tag{19}$$

To see how this connects to bound optimization, let us define

$$Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}\right) = \mathcal{E}\left(\boldsymbol{\theta}, \left\{p_{\mathbf{Z}|\mathbf{Y}}\left(\mathbf{z}_n \mid \mathbf{y}_n, \boldsymbol{\theta}^{(t)}\right)\right\}\right). \tag{20}$$

Then we have $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) \leq l(\boldsymbol{\theta})$ and $Q(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)}) = l(\boldsymbol{\theta}^{(t)})$, as required.

However, if we cannot compute the posteriors $p_{\mathbf{Z}|\mathbf{Y}}\left(\mathbf{z}_n \mid \mathbf{y}_n; \boldsymbol{\theta}^{(t)}\right)$ exactly, we can still use an approximate distribution $q_{\mathbf{Z}|\mathbf{Y}}\left(\mathbf{z}_n \mid \mathbf{y}_n; \boldsymbol{\theta}^{(t)}\right)$; this will yield a non-tight lower-bound on the log-likelihood. As we will see, in the case of Gaussian mixture models we assume q has the categorical distribution, and then find the categorical distribution that has the smallest KL divergence from the true $p_{\mathbf{Z}|\mathbf{Y}}\left(\mathbf{z}_n \mid \mathbf{y}_n; \boldsymbol{\theta}^{(t)}\right)$.

2.2.3 The M Step

In the M step, we need to maximize $\mathcal{E}\left(\boldsymbol{\theta}, \left\{q_n^{(t)}\right\} \mid \mathcal{D}\right)$ with respect to $\boldsymbol{\theta}$, where the $q_n^{(t)}$ are the distributions computed in the E step at the iteration t. The entropy terms $\mathbb{H}(q_n)$ are constant with respect to $\boldsymbol{\theta}$, so we can drop them in the M step. We are left with

$$l^{(t)}(\boldsymbol{\theta}) = \sum_n \mathbb{E}_{q_n^{(t)}}\left[\ln\left(p_{\mathbf{Y},\mathbf{Z}}\left(\mathbf{y}_n, \mathbf{z}_n; \boldsymbol{\theta}\right)\right)\right] \tag{21}$$

This is called the **expected complete data log likelihood**.

In the M step, we maximize the expected complete data log likelihood to get

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \sum_n \mathbb{E}_{q_n^{(t)}}\left[\log\left(p_{\mathbf{Y},\mathbf{Z}}\left(\mathbf{y}_n, \mathbf{z}_n; \boldsymbol{\theta}\right)\right)\right] \tag{22}$$

Theory Question 1. In your own words, explain how the MM algorithm can deal with non-convex optimization objective functions by considering simpler convex objective functions.

Theory Question 2. Briefly explain how the formula for mixture models:

$$p(\mathbf{y}; \boldsymbol{\theta}) = \sum_{k=1}^{K} p_Z(z_k; \boldsymbol{\theta}) p_{\mathbf{Y}|\mathbf{Z}}(\mathbf{y}|Z = z_k; \boldsymbol{\theta}),$$

is the same as the sum over all possible values of $Z^{(i)}$ in equation (9). Explain why it's easier to optimize $p_{\mathbf{Y},\mathbf{Z}}(\mathbf{y}_n, \mathbf{z}_n; \theta)$ than $p_{\mathbf{Y}}(\mathbf{y}_n; \theta)$ in the context of mixture models.

Theory Question 3. Read about variational inference (or variational bayesian methods) and compare it with the procedure we used for the EM algorithm (You might want to check Wikipedia for this!).

---

3 EM Algorithm for GMM and CMM

In this section, we want you to apply EM algorithm to learn(estimate) parameters for two different mixture model and find closed-form solution of their parameters.

3.1 EM for Gaussian Mixture Model

Theory Question 4. Compute estimate of parameters for Gaussian Mixture Models for N observed data $\{\mathbf{x}_i\}_{i=1}^N$.

1. Determine model parameters and initialize them.
2. Compute complete dataset likelihood[1].
3. Find closed-form solution for parameters using EM algorithm.

3.2 EM for Categorical Mixture Model

Theory Question 5. Compute estimate of parameters for Categorical Mixture Models for N observed data $\{\mathbf{x}_i\}_{i=1}^N$.

[1] $p(\mathcal{D}; \boldsymbol{\theta}) = p(\{\mathbf{x}_i\}_{i=1}^N, \{\mathbf{z}_i\}_{i=1}^N; \boldsymbol{\theta})$

1. Determine model parameters and initialize them.
2. Compute complete dataset likelihood.
3. Find closed-form solution for parameters using EM algorithm.

4 EM Algorithm in Real Applications

Suppose you have a set of MRI images of patients with brain tumors. You want to estimate the volume and location of the tumors from the images. You assume that each image is generated by a mixture of three Gaussian distributions: one for the background, one for the normal brain tissue and one for the tumor tissue. However, some of the images are corrupted by noise or artifacts (missing data). We have given you the three Gaussian distributions and need to estimate their means and variances. Use EM algorithm to estimate the parameters of the mixture model to help us to fill in the missing data.

Answer all simulation questions for both `Image1.csv` and `Image2.csv` datasets and compare the parameters obtained from these two.

Simulation Question 1. Each distribution has 200 data points that are concatenated in a two-dimensional array and given to you. Plot the data with three different colors in a graph.

Simulation Question 2. Write a function that performs the E-step. This means assigning each data to a distribution based on the Euclidean distance. Return as output a 3×600 array specifying which distribution each data belongs to. If the R_{ij} is one, it means that the i–th data is assigned to the j–th distribution. Run this function for one iteration and report the result.

Simulation Question 3. Write a function that performs M-step. This means updating the mean and variance of each distribution. Run this function for one iteration and report the new variances and means of each distribution.

Simulation Question 4. Using the functions you have written, run the EM algorithm until a convergence is reached or the maximum number of steps is passed. Replot the three new distributions and compare with the correct labels.

Simulation Question 5. Compare the parameters obtained from each of the images and explain the reason for their difference.