SPR Assignment 3

Jayanth S (201081003), Praveen Kumar N (201082001), Rishabh Roy (201082002)

Contents

0.1 EM Algorithm

0.1.1 EM algorithm in general

- A limitation of maximum likelihood estimation is that it assumes that the dataset is complete, or fully observed. This does not mean that the model has access to all data; instead, it assumes that all variables that are relevant to the problem are present.
- This is not always the case. There may be datasets where only some of the relevant variables can be observed, and some cannot, and although they influence other random variables in the dataset, they remain hidden(latent variables).
- The Expectation-Maximization algorithm is an approach for performing maximum likelihood estimation in the presence of latent variables.
- The EM algorithm is an iterative approach that cycles between two steps. The first step attempts to estimate the missing or latent variables, called the Estimation-step or E-step. The second step attempts to optimize the parameters of the model to best explain the data, called the Maximization-step or M-step.
- Consider a probabilistic model in which $\mathbf{X}$ denotes observed variables and $\mathbf{Z}$ denotes hidden variables . The joint distribution. Our goal is to maximize the likelihood function given by

$$p(\mathbf{X}/\theta) = \sum_{\mathbf{Z}} p(\mathbf{X},\mathbf{Z}/\theta)$$

- If we consider $q(\mathbf{Z})$ to be distribution over the latent variables, then for any choice of $q(\mathbf{Z})$ following decomposition holds

$$lnp(\mathbf{X}/\theta) = \mathcal{L}(q,\theta) + KL(q||p)$$

where,

$$\begin{aligned}\mathcal{L}(q,\theta) &= \sum_{\mathbf{Z}} q(\mathbf{Z}) \frac{ln(p(\mathbf{X},\mathbf{Z}/\theta))}{ln(q(\mathbf{Z}))} \\ &= \sum_{\mathbf{Z}} q(\mathbf{Z}) \frac{(ln(p(\mathbf{Z}/\mathbf{X},\theta)) + ln(p(\mathbf{X}/\theta)))}{ln(q(\mathbf{Z}))} \\ KL(q||p) &= -\sum_{\mathbf{Z}} q(\mathbf{Z}) \frac{ln(p(\mathbf{Z}/\mathbf{X},\theta))}{ln(q(\mathbf{Z}))}\end{aligned}$$

- Here $KL(q||p)$ is the Kullback-Leibler divergence between $q(\mathbf{Z})$ and posterior distribution $p(\mathbf{Z}/\mathbf{X},\theta)$. Kullback-Leibler divergence satisfies $KL(q||p) \geq 0$ and here $KL(q||p) = 0$ only if $q(\mathbf{Z}) = p(\mathbf{Z}/\mathbf{X},\theta)$. Therefore from above equations it follows that $\mathcal{L}(q,\theta)$ is a lower bound on $ln(p(\mathbf{X}/\theta))$.

- EM Algorithm is a iterative optimization technique for finding ML estimates. We can use the above decomposition to demonstrate that EM Algorithm does maximize the log likelihood. Let the current value of the parameter vector is θ^{old}.
  - **E-step**: In the E-step, lower bound $\mathcal{L}(q, \theta^{old})$ is maximized with respect to $q(\mathbf{Z})$ while holding θ^{old} fixed. the value of $ln(p(\mathbf{X}|\theta^{old}))$ does not depend on $q(\mathbf{Z})$ so the largest value of $L(q, \theta^{old})$ will occur when the Kullback-Leibler divergence vanishes, in other words when $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X}, \theta^{old})$. In this case, the lower bound will equal the log likelihood i.e, $\mathcal{L}(q, \theta^{old}) = ln(p(\mathbf{X}|\theta^{old}))$
  - **M-step**: In the M-step, $q(\mathbf{Z})$ is held fixed and the lower bound $\mathcal{L}(q, \theta^{old})$ is maximized with respect to θ to give new value θ^{new}. This will cause the lower bound to increase, which will cause the corresponding log likelihood function to increase. Because the distribution q is determined using the old parameter values i.e, $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X}, \theta^{old})$, it will not be equal to new posterior distribution $p(\mathbf{Z}|\mathbf{X}, \theta^{new})$ and hence $KL(q||p)$ is non-zero. The increase in log likelihood is therefore greater than increase in the lower bound.

0.1.2 Algorithm

Given $p(\mathbf{X}, \mathbf{Z}/\theta)$, where $\mathbf{X}$ are observed variables and $\mathbf{Z}$ are latent variables governed by parameters θ, the goal is to maximize the likelihood function $p(\mathbf{X}/\theta)$.

1. Choose an initial setting for the parameters θ^{old}.
2. **E step** Evaluate $p(\mathbf{Z}/\mathbf{X}, \theta^{old})$
3. **E step** Evaluate θ^{new} given by

   $$\theta^{new} = argmax_\theta \mathcal{Q}(\theta, \theta^{old})$$

   where

   $$\mathcal{Q}(\theta, \theta^{old}) = \sum_{\mathbf{Z}} p(\mathbf{Z}/\mathbf{X}, \theta^{old}) ln(p(\mathbf{X}, \mathbf{Z}/\theta))$$

4. Check for convergence of either the log likelihood or the parameter values. If the convergence criterion is not satisfied, then let

   $$\theta^{old} = \theta^{new}$$

   and return to step 2.

0.1.3 Mixture of Bernoulli distributions

- Consider a set of D binary variables each of which is governed by a Bernoulli distribution with parameter μ_i. Let $\mathbf{x} = (x_1, \ldots, x_D)$ and

$\mu = (\mu_1, \ldots, \mu_D)$, then

$$p(\mathbf{x}/\mu) = \prod_{i=1}^{D} \mu_i^{x_i}(1-\mu_i)^{(1-x_i)}$$
$$E[\mathbf{x}] = \mu$$
$$cov[\mathbf{x}] = diag\{\mu_i(1-\mu_i)\}$$

- Now consider a finte mixture of these distributions given by

$$p(\mathbf{x}/\mu, \pi) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}/\mu_k)$$

where $\mu = \{\mu_1, \ldots, \mu_K\}$, $\pi = \{\pi_1, \ldots, \pi_K\}$, and

$$p(\mathbf{x}/\mu_k) = \prod_{i=1}^{D} \mu_{ki}^{x_i}(1-\mu_{ki})^{(1-x_i)}$$
$$E[\mathbf{x}] = \sum_{k=1}^{K} \pi_k \mu_k$$
$$cov[\mathbf{x}] = \sum_{k=1}^{K} \pi_k\{\Sigma_k + \mu_k\mu_k^T\} - E[\mathbf{x}]E[\mathbf{x}]^T$$

where $\Sigma_k = diag\{\mu_{ki}(1-\mu_{ki})\}$

- if we have dataset $\mathbf{X} = \{x_1, \ldots, x_N\}$, log likelihood function is given by

$$ln(p(\mathbf{X}/\mu, \pi)) = \sum_{n=1}^{N} ln\{\sum_{k=1}^{K} \pi_k p(\mathbf{x}_n/\mu_k)\}$$

- Let $\mathbf{z} = \{z_1, \ldots, z_K\}^T$ be a binary K-dimensional latent variable having a single component equal to 1, with all other components equal to 0. then,

$$p(\mathbf{x}/\mathbf{z}, \mu) = \prod p(\mathbf{x}/\mu_k)^{z_k}$$

the prior distribution of latent variables is $p(\mathbf{z}/\pi) = \prod_{k=1}^{K} \pi_k^{z_k}$

- Now for EM algorithm complete data log likelihood function is given by

$$lnp(\mathbf{X}, \mathbf{Z}/\mu, \pi) = \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma(z_{nk})\{ln\pi_k + \sum_{i=1}^{D}[x_{ni}ln\mu ki + (1-x_{ni}ln(1-\mu_{ki})]\}$$

where $\gamma(z_{nk}) = E[z_{nk}]$ is the posterior probability, or responsibility, of component k given data point $\mathbf{x}$.

- In the **E step**, these responsibilities are evaluated using Bayes theorem, which takes the form

$$\begin{aligned}\gamma(z_{nk}) &= E[z_{nk}] \\ &= \frac{\pi_k p(\mathbf{x}_n/\mu_k)}{\sum_{j=1}^{K} \pi_j p(\mathbf{x}_n/\mu_j)} \\ N_k &= \sum_{n=1}^{N} \gamma(z_{nk}) \\ \bar{x}_k &= \frac{1}{N_k} \sum \gamma(z_{nk}) \mathbf{x}_n\end{aligned}$$

  where N_k is the effective number of data points associated with component k.

- In the **M step**, we maximize the expected complete-data log likelihood with respect to the parameters μ_k and π. we obtain

$$\begin{aligned}\mu_k &= \bar{x}_k \\ \pi_k &= \frac{N_k}{N}\end{aligned}$$

  as the parameter values which maximizes complete-data log likelihood.

0.1.4 Observations

- We took 200 images from each class (labels) 2,3 and 4 of MNIST data set. Total no. of images used are 600.
- We implemented the EM algorithm considering the distribution as mixture of Bernoulli distributions.

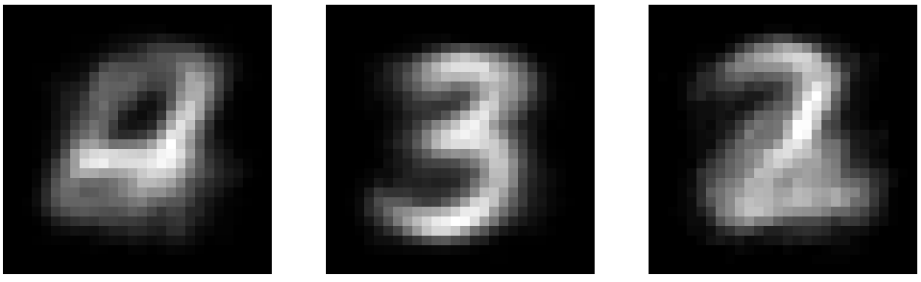

Figure 1: We used k-means with less no. of iterations(3) for initialising the parameters before running EM algorithm and it gave the following images when we used the parameters of each mixture component to plot the image

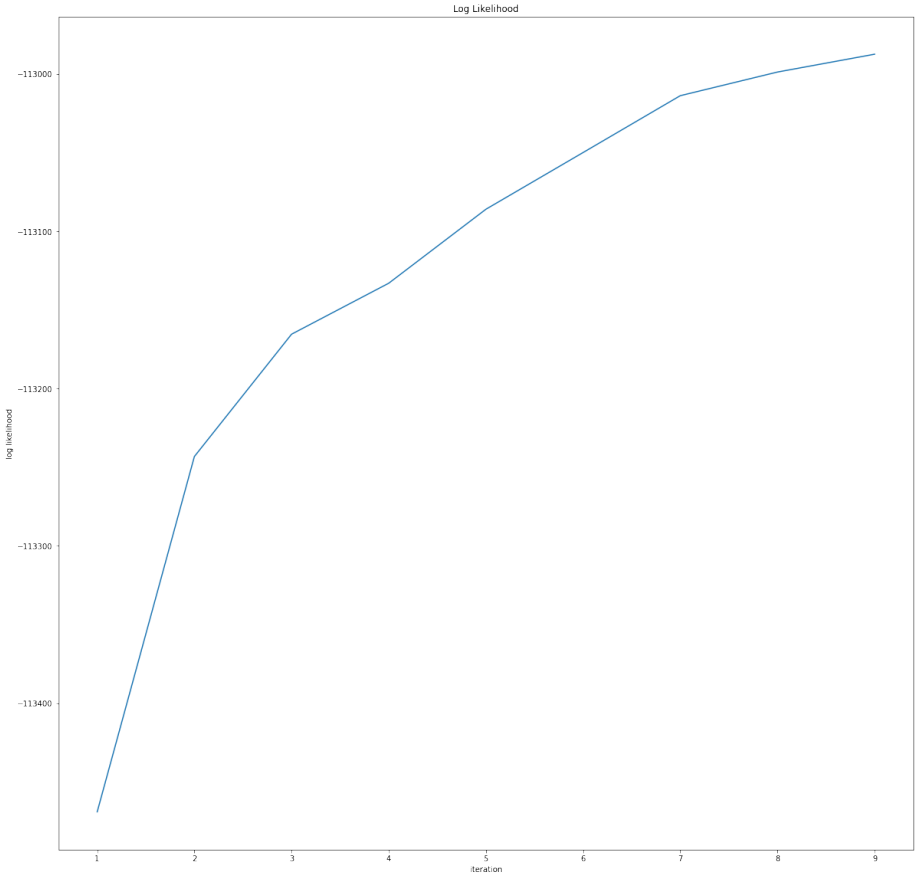

Figure 2: The log-likelihood plot with x-axis representing the iteration and y-axis representing the log-likelihood at that iteration

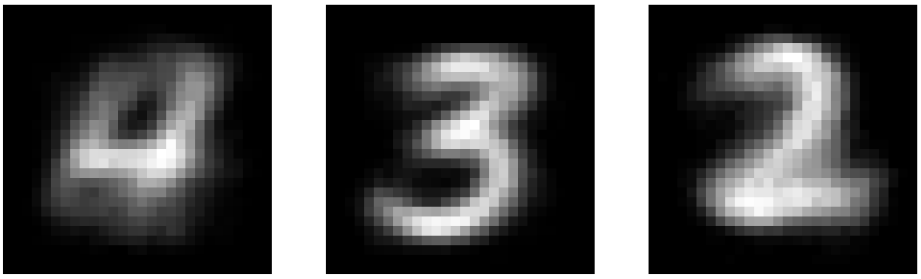

Figure 3: The final images that we obtained from estimated parameters using EM algorithm with parameters initialised using k-means algorithm

0.2 K-means

0.2.1 Introduction

- K-means clustering is one of the simplest and popular unsupervised machine learning algorithms. Typically, unsupervised algorithms make inferences from datasets using only input vectors without referring to known, or labelled, outcomes.
- The objective of K-means is simple: group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (k) of clusters in a dataset

0.2.2 Algorithm

The way k means algorithm works is as follows:

- Specify number of clusters K.
- Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement.
- Keep iterating until there is no change to the centroids. i.e., assignment of data points to clusters isn't changing.
- Compute the sum of the squared distance between data points and all centroids.
- Assign each data point to the closest cluster (centroid).
- Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

The approach k means follows to solve the problem is called Expectation-Maximization. The E-step is assigning the data points to the closest cluster. The M-step is computing the centroid of each cluster.

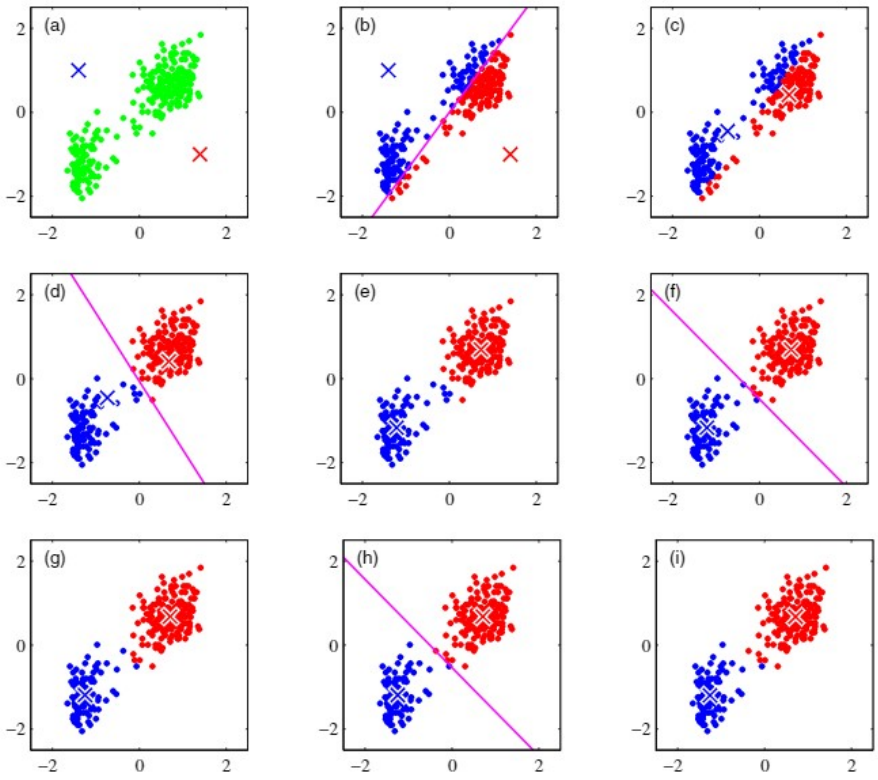

Figure 4: Illustration of performing k-means on scaled Old-faithful dataset

0.2.3 Observations

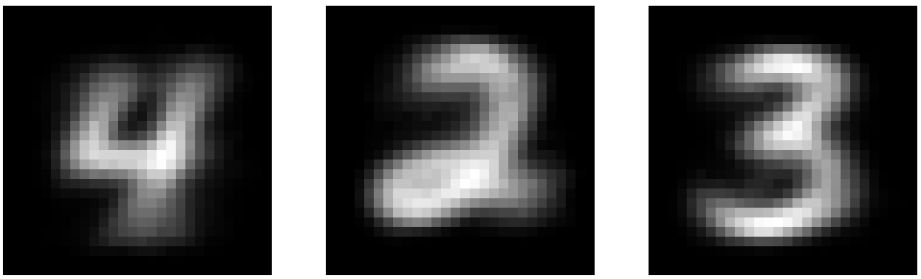

Figure 5: We ran k-means algorithm for question 1 and obtained the following images when we used the parameters of each mixture component to plot the image

0.3 Observations for Q2

- This section contains the observations for Q2 i.e., Sample data from univariate and multivariate Gaussian Mixture Models and verify whether EM algorithm is able to retrieve the mixture densities in each of the case.
- The observations when k-means was use to retrieve the mixture densities is also included in this section.
- We also ran gradient ascent algorithm for 1-dimensional data and we able to retrieve the mean values correctly.
- The plots when the parzen window is used for Q2 is in the **parzen window section** of this report.

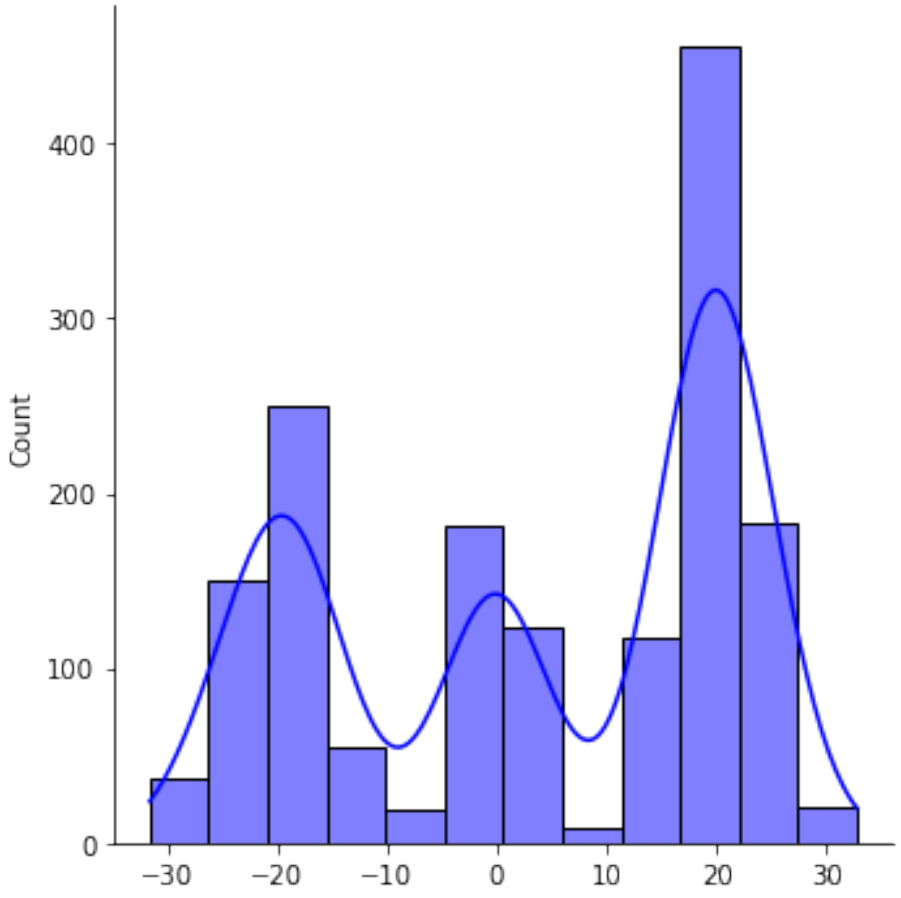

Figure 6: Original 1-dim Gaussian Mixture distribution (with histogram) that needs to be estimated

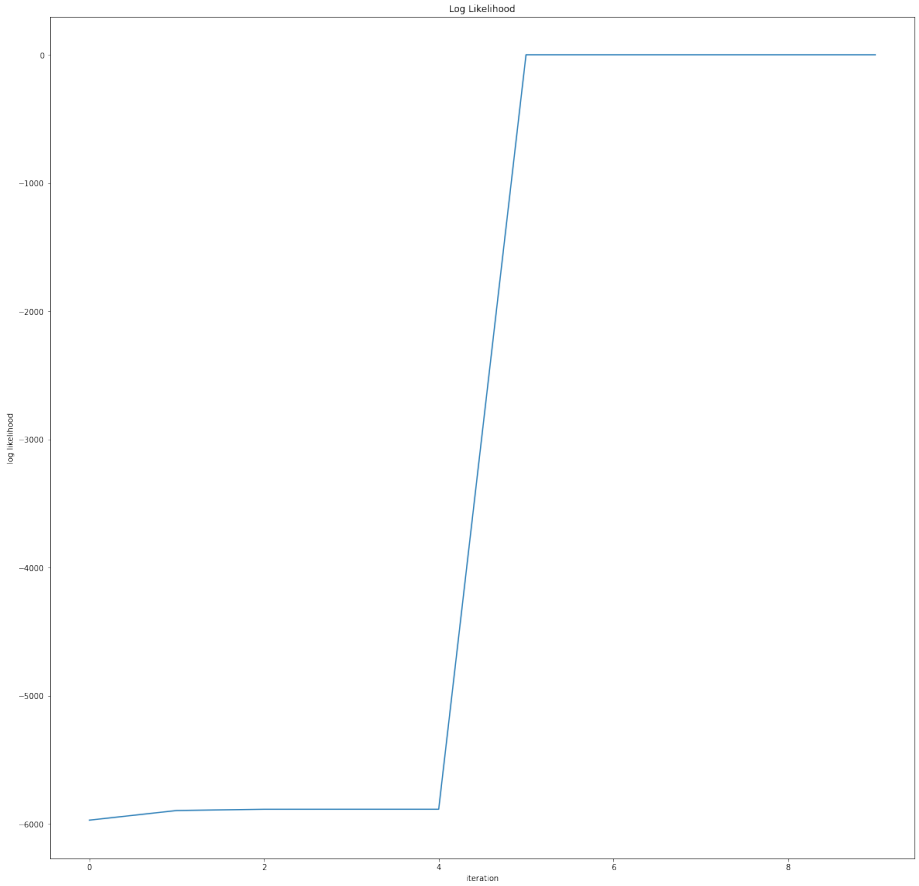

Figure 7: The log-likelihood plot obtained when we ran Gaussian mixture model to estimate the parameters of the mixture densities. *Ignore the plot after 4th iteration. It is showing zero because we the EM algorithm was ended at 4th iteration because there was no significant difference in log-likelihood obtained in iteration 3 and 4.

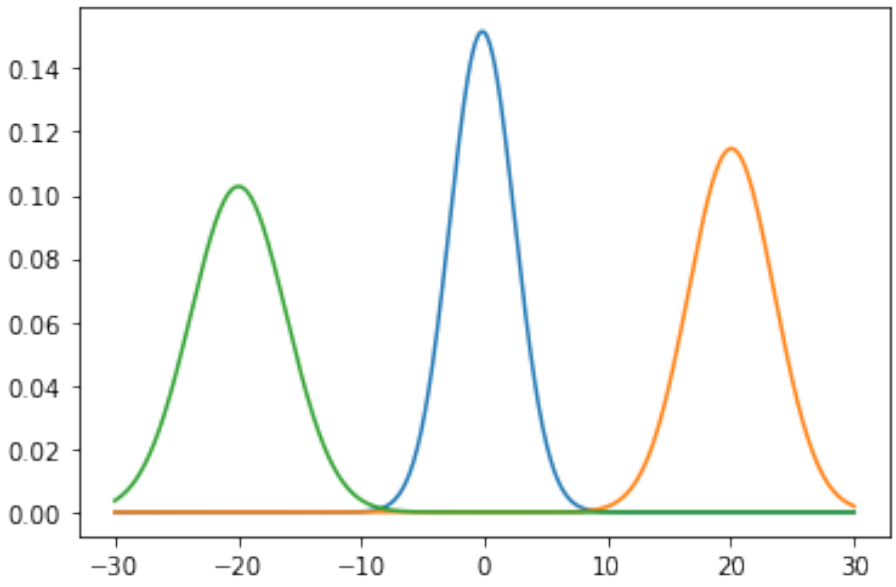

Figure 8: Plot of 1-dim Gaussian Mixture distribution obtained using the estimated parameters

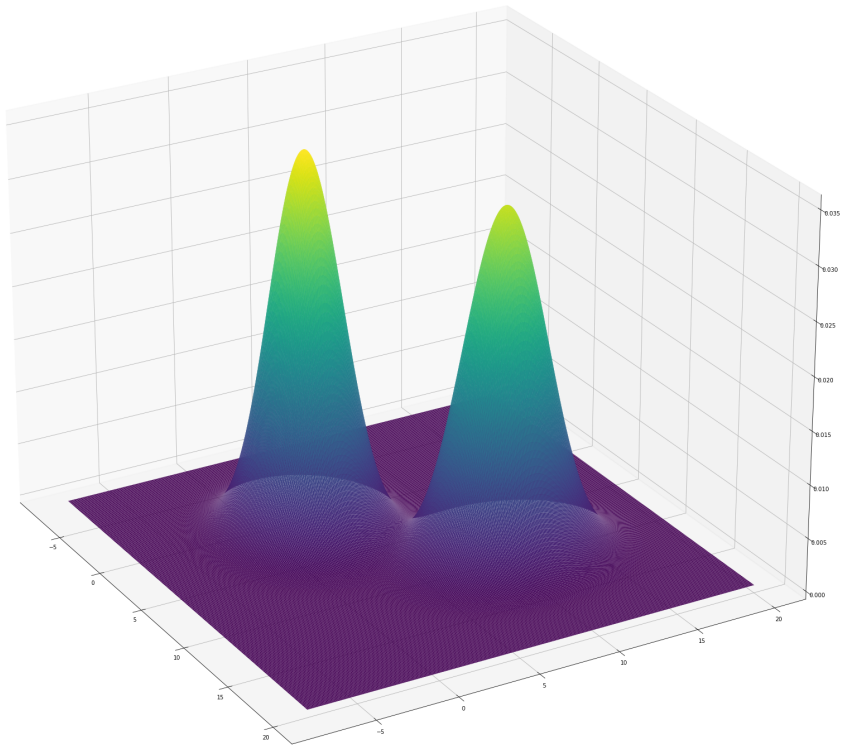

Figure 9: Original 2-dim Gaussian Mixture distribution for which the parameters needs to be estimated

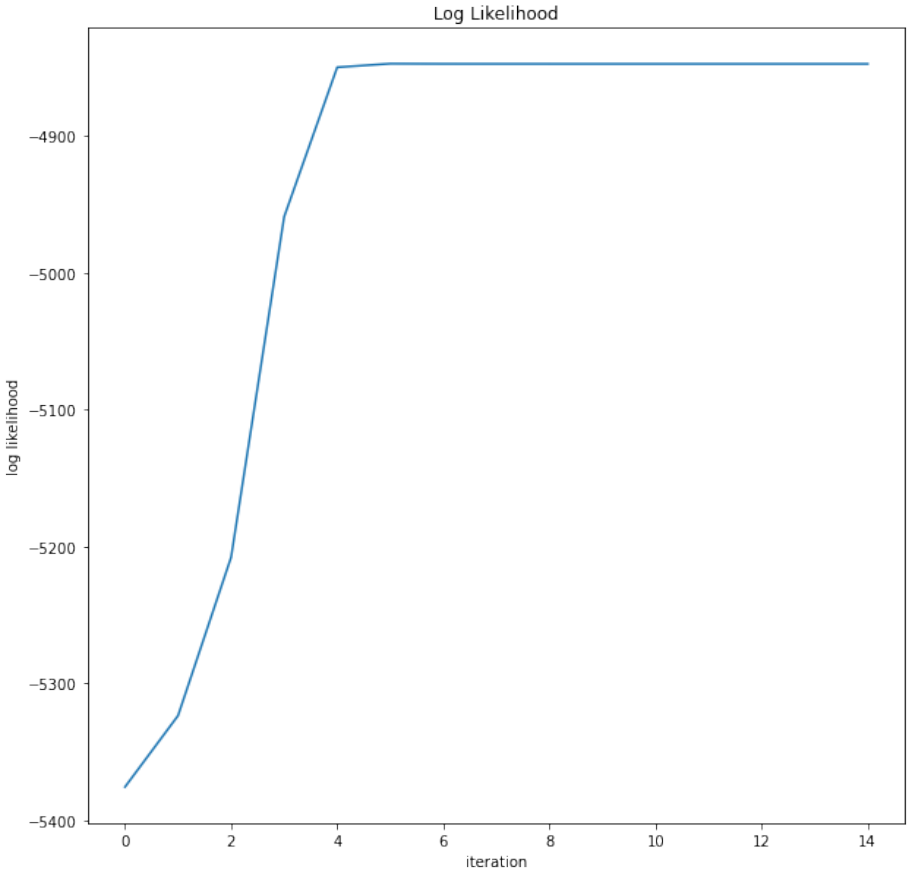

Figure 10: The log-likelihood plot obtained when we ran Gaussian mixture model to estimate the parameters of the mixture densities.

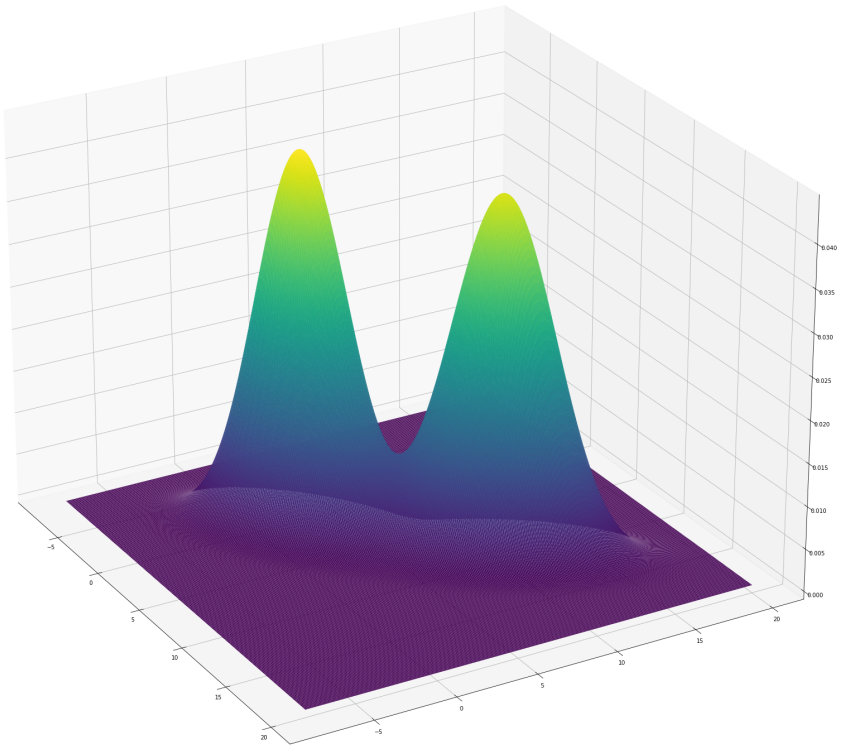

Figure 11: Plot of 2-dim Gaussian Mixture distribution obtained using the estimated parameters

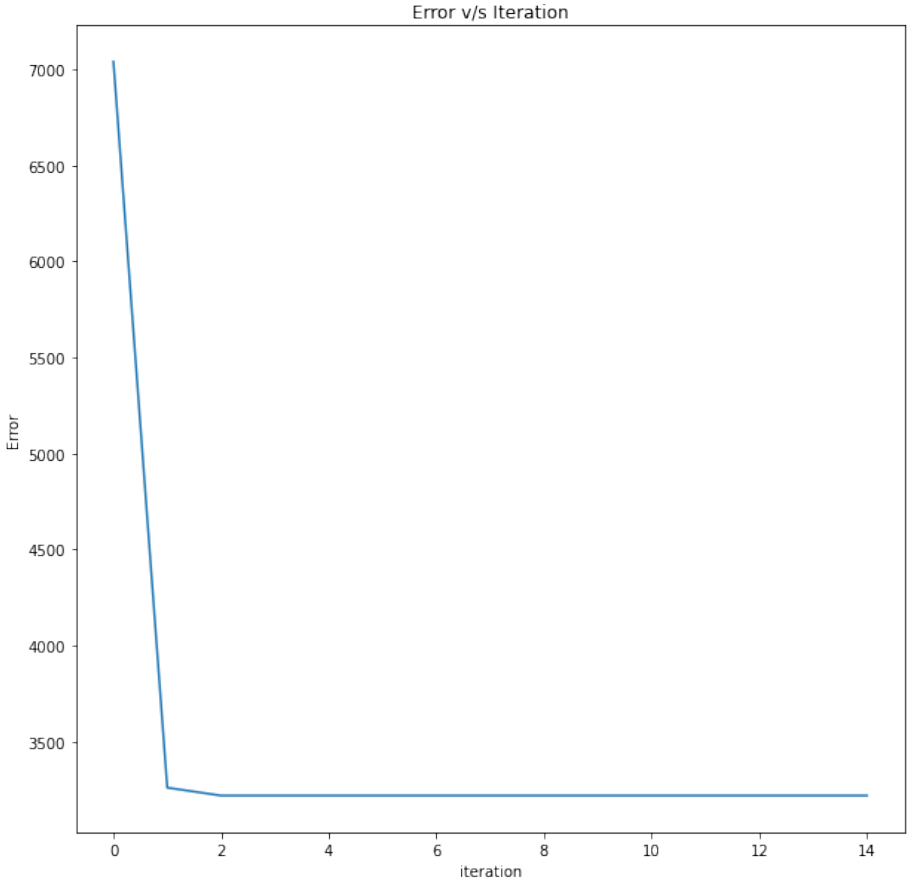

Figure 12: Plot of error v/s iteration. The error drops down drastically in starting few iterations and the error decreases for the successive iterations

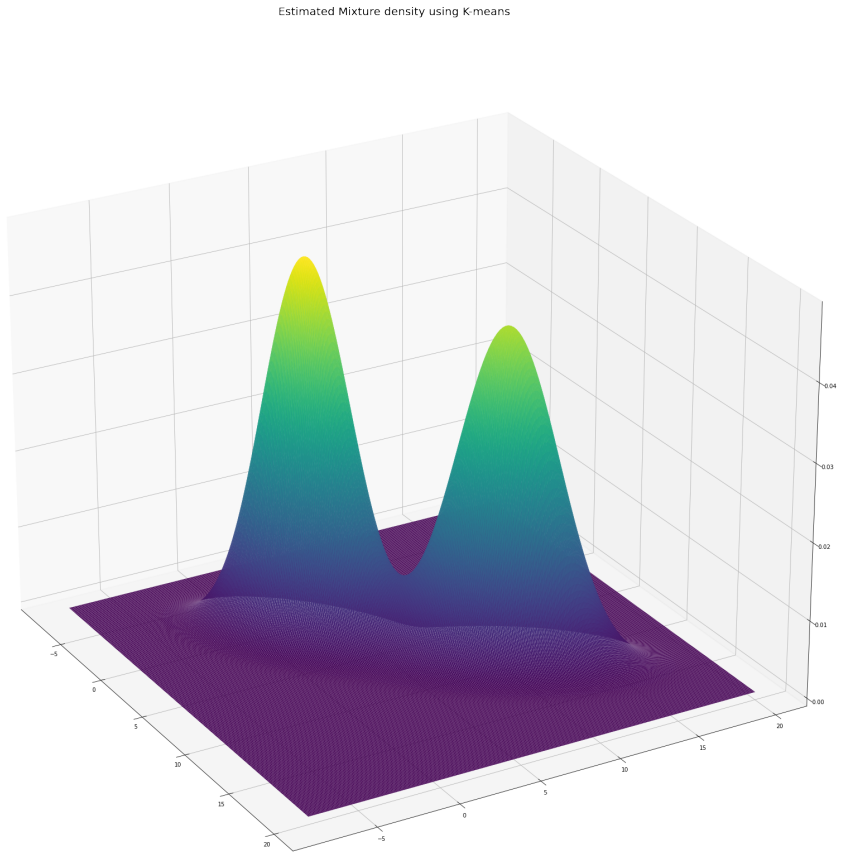

Figure 13: Plot of 2-dim Gaussian Mixture distribution using the estimated parameters obtained with the help of k-means algorithm

0.4 Parzen Window

0.4.1 Introduction

- Parzen-window method is a non-parametric approach to estimate a probability density function $f(x)$ at a specific point x from the samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ that doesn't require any knowledge or assumption about the distribution from which the samples are taken.
- Parzen window is a kernel density estimation technique in which we fix the volume and observe the number of points k that fall into the region.(Other method is we fix the number of points and then adjust the volume.)

0.4.2 Parzen Window

- Let $\mathcal{R}_n$ be the d-dimensional hypercube. The volume of this region is given by,
$$V_n = h_n^d$$
where h_n : Length of an edge of the hypercube.
- We use a window function to count the number of points that falls in the hypercube.
- We use two different window function to implement Parzen window. They are,
  - Rectangular window function
  - Gaussian window function

Rectangular Window Function

- The rectangular window function is given by the following expression,
$$\Phi(u) = \begin{cases} 1 & |u_j| < \frac{1}{2} \ , \ j = 1, 2, ..., d \\ 0 & \text{otherwise} \end{cases}$$
where $u = [u_1, u_2, ..., u_d]$
- Let $\mathbf{x}$ be the point at which we need to estimate the density. Let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ be the given data point using which we estimate the density at $\mathbf{x}$.
- We would count the no. of samples (k) that fall in the hypercube of volume V_n centered at $\mathbf{x}$. It is given by,
$$k = \sum_{i=1}^{n} \Phi(\mathbf{x} - \mathbf{x}_i/h)$$
- $\Phi(\mathbf{x} - \mathbf{x}_i/h)$ will be 1 when $\mathbf{x}_i$ fall within the hypercube of volume V_n centered at $\mathbf{x}_i$.
- The corresponding density estimate at $\mathbf{x}$ is,
$$f(\mathbf{x}) = (1/(nh)) * \sum_{i=1}^{n} \frac{1}{V_n} \Phi((\mathbf{x} - \mathbf{x}_i)/h_n$$

Gaussian Window Function

- Wkt, for $f(\mathbf{x})$ should be non-negative and integrate to one. Hence we can use any window function that satisfies the following properties 1.
$$\Phi(\mathbf{x} \geq 0)$$
2.
$$\int \Phi(\mathbf{x} d\mathbf{x} = 0$$

- Hence, we consider normal density function as the window function i.e.,

$$\Phi(\mathbf{x}) = \frac{1}{\sqrt{2\pi}} e^{-(\|\mathbf{x}\|_2^2/2)}$$

0.4.3 Observations

- We have generated the synthetic data for 1-dim and 2-dim Gaussian mixture models and tried to estimate them using the rectangular and gaussian window functions with different length of edges of the hypercube.

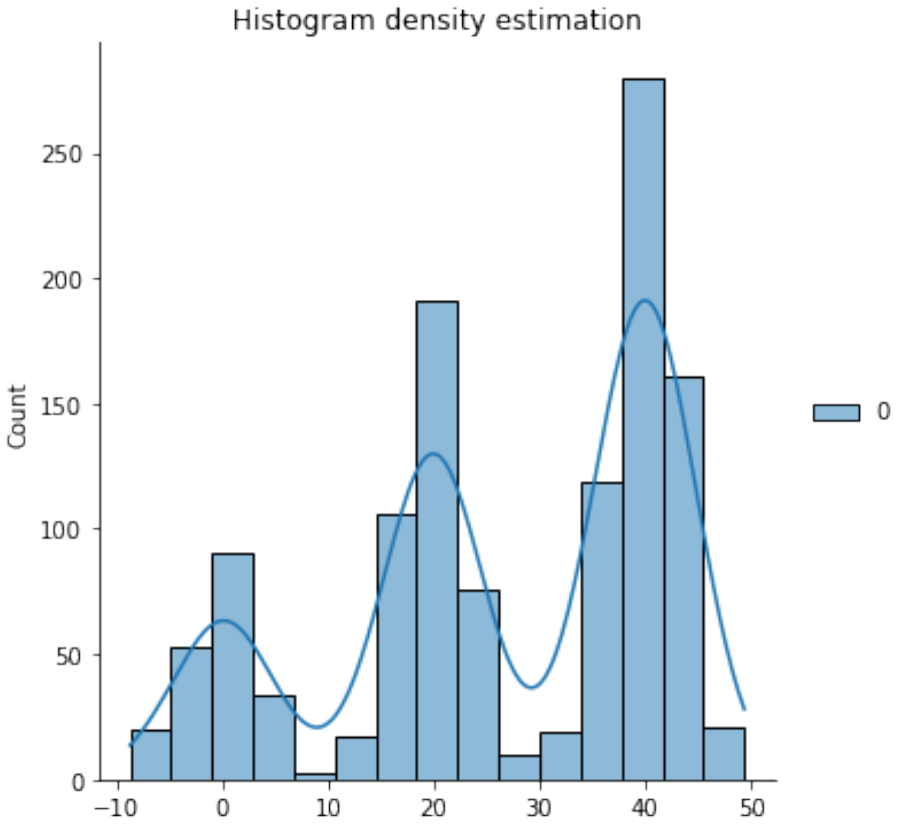

Figure 14: Original 1-dim Gaussian distribution (with histogram) that needs to be estimated

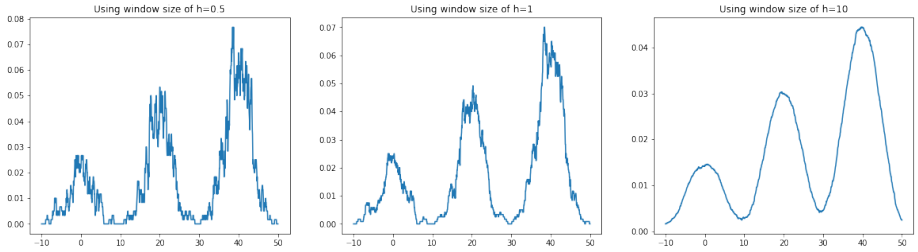

Figure 15: 1-dim Gaussian distribution estimated using rectangular windows with different values of h. As we can see from the plot for h=10 given a smoother estimate compared to h=0.5 or 1

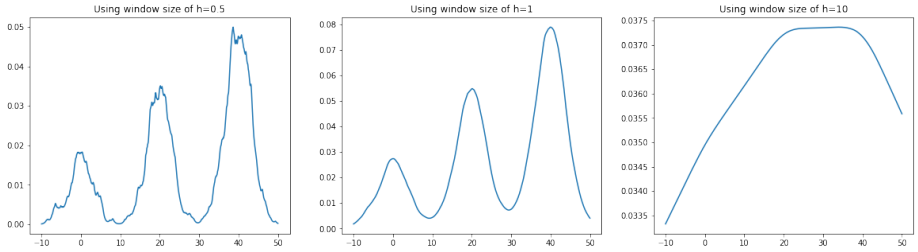

Figure 16: 1-dim Gaussian distribution estimated using gaussian window with different values of h. As we can see from the plot for h=1 given a smoother estimate compared to h=0.5 and for h=10 the estimate says there is no mixture density only which is wrong.

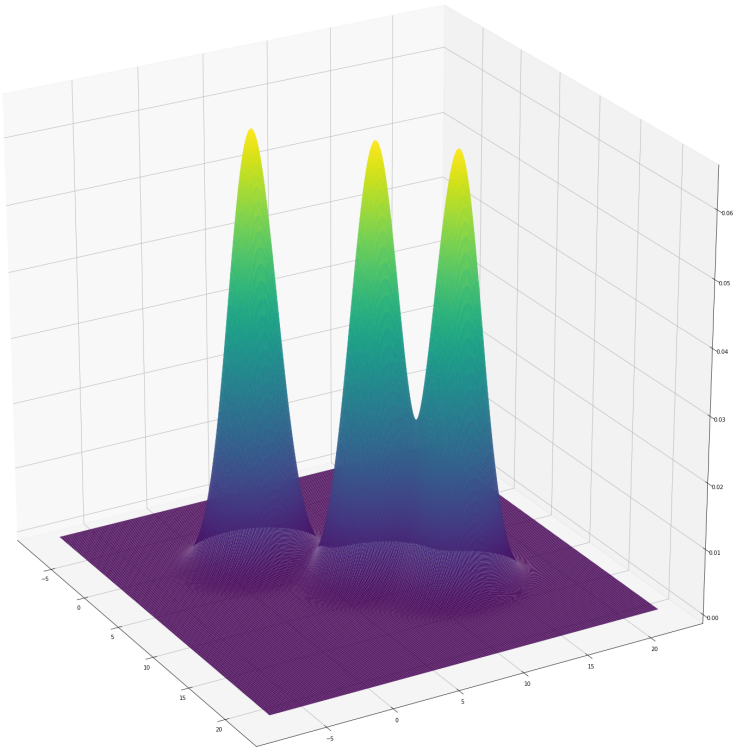

Figure 17: Original 2-dim Gaussian mixture(3) distribution that needs to be estimated

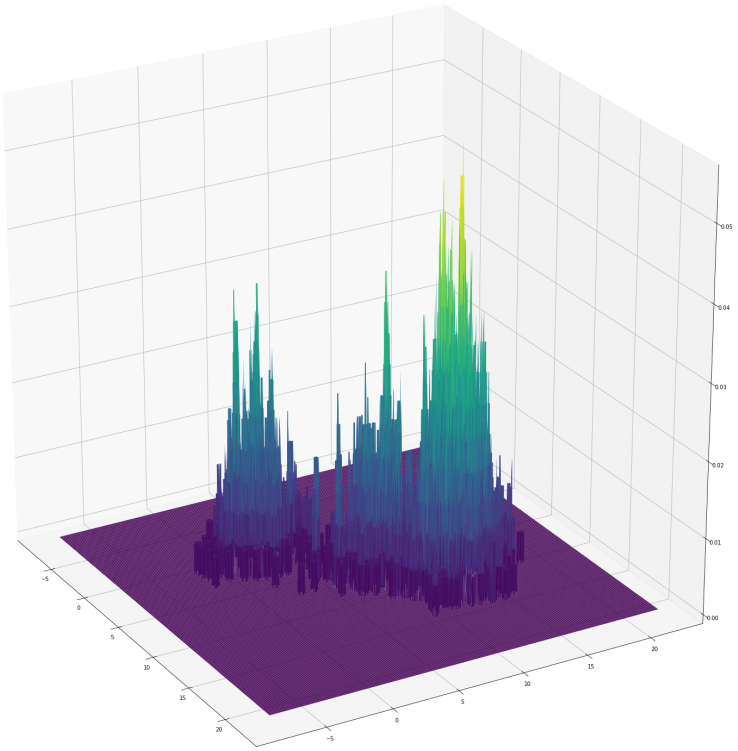

Figure 18: 2-dim Gaussian distribution estimated using rectangular windows with different values of h=0.5

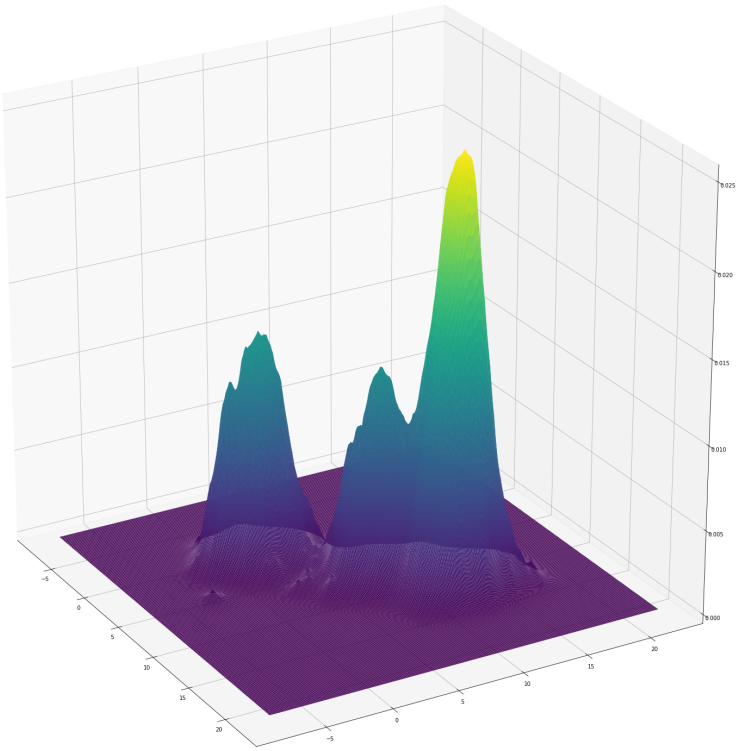

Figure 19: 2-dim Gaussian distribution estimated using gaussian window with different values of h=0.5. For same h gaussian window gives a smoother estimate than rectangular window

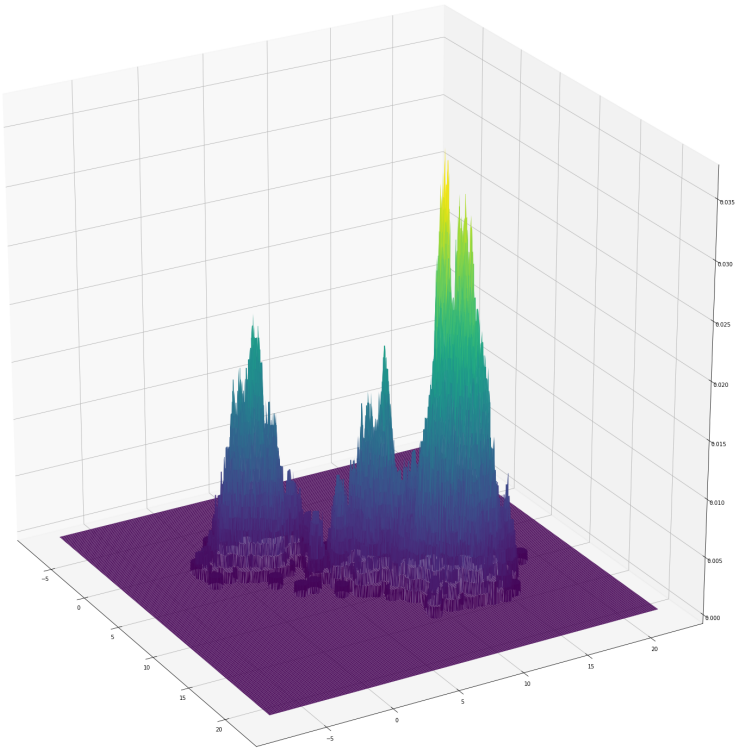

Figure 20: 2-dim Gaussian distribution estimated using rectangular windows with different values of h=1

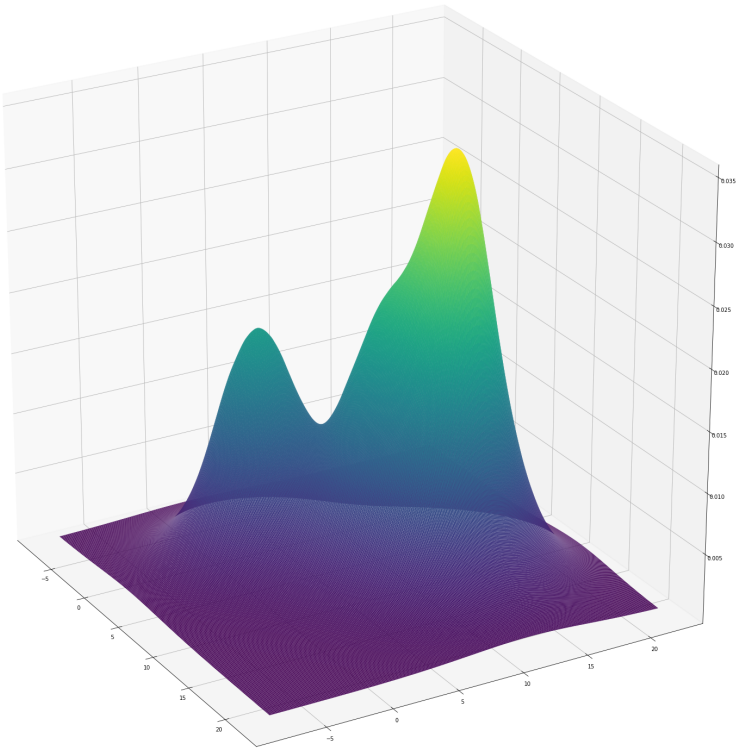

Figure 21: 2-dim Gaussian distribution estimated using gaussian window with different values of h=1. As we can see one of the mixture density is not clear as in original plot. So if h is increased further the mixture estimate will get worse.

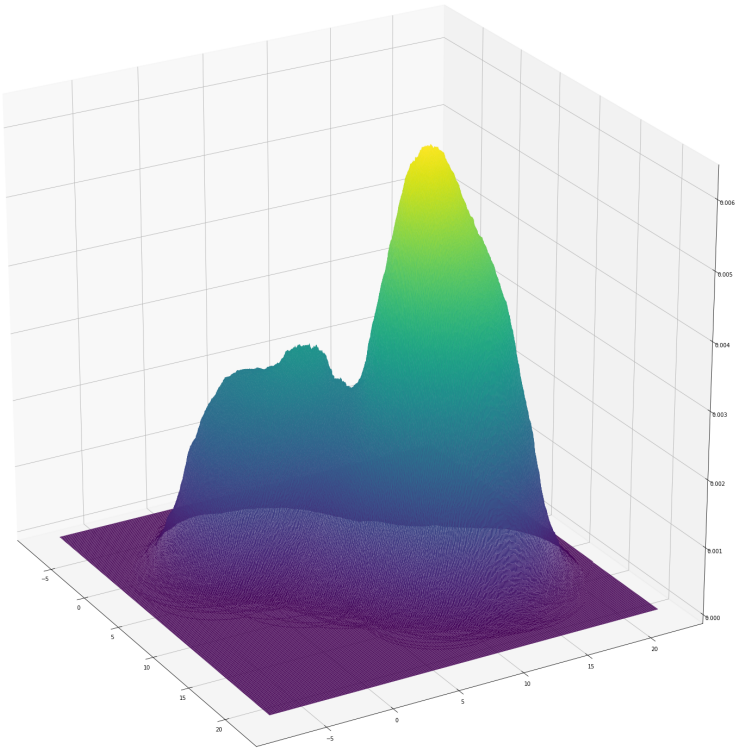

Figure 22: 2-dim Gaussian distribution estimated using rectangular windows with different values of h=10. As we can see from the plot one of the gaussian distribution in the mixture density is not estimated correctly

0.4.4 Drawbacks of Parzen window

- As we saw from the plots, we are not sure what should be the value of h for density estimation.
- If gaussian window function is used then, it increases the computation time as the dimension(d) of the sample points increases.