Gaussian Mixture Models (GMMs)

Dr. Virendra Singh Kushwah

Assistant Professor Grade-II

School of Computing Science and Engineering

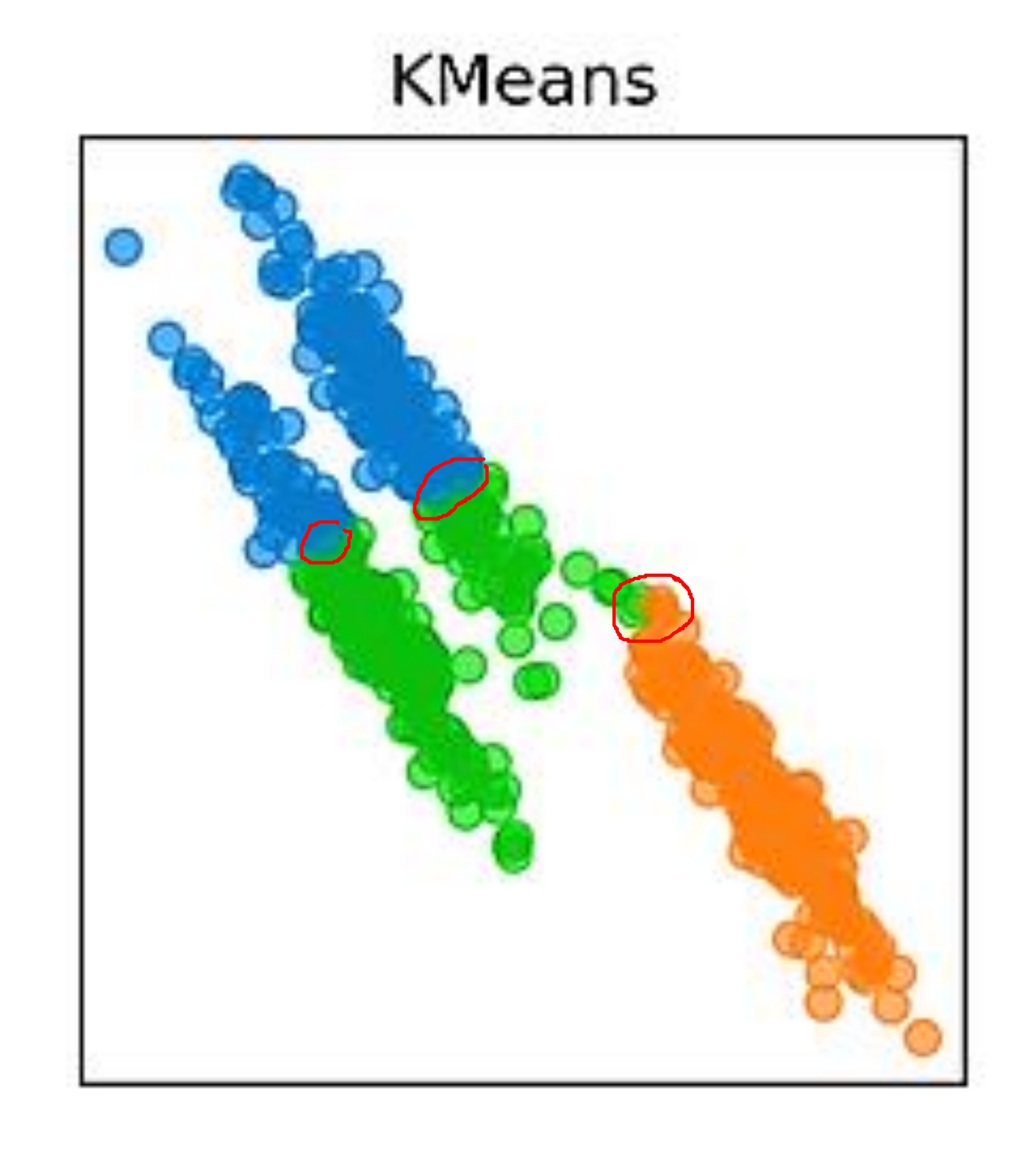
Virendra.Kushwah@vitbhopal.ac.in

7415869616

Drawbacks of k-means Clustering

- The k-means algorithm seems to be working pretty well, right? Hold on – if you look closely, you will notice that all the clusters created have a circular shape. This is because the centroids of the clusters are updated iteratively using the mean value.

- Now, consider the example (next slide) where the distribution of points is not in a circular form. What do you think will happen if we use k-means clustering on this data? It would still attempt to group the data points in a circular fashion. That’s not great! k-means fails to identify the right clusters.



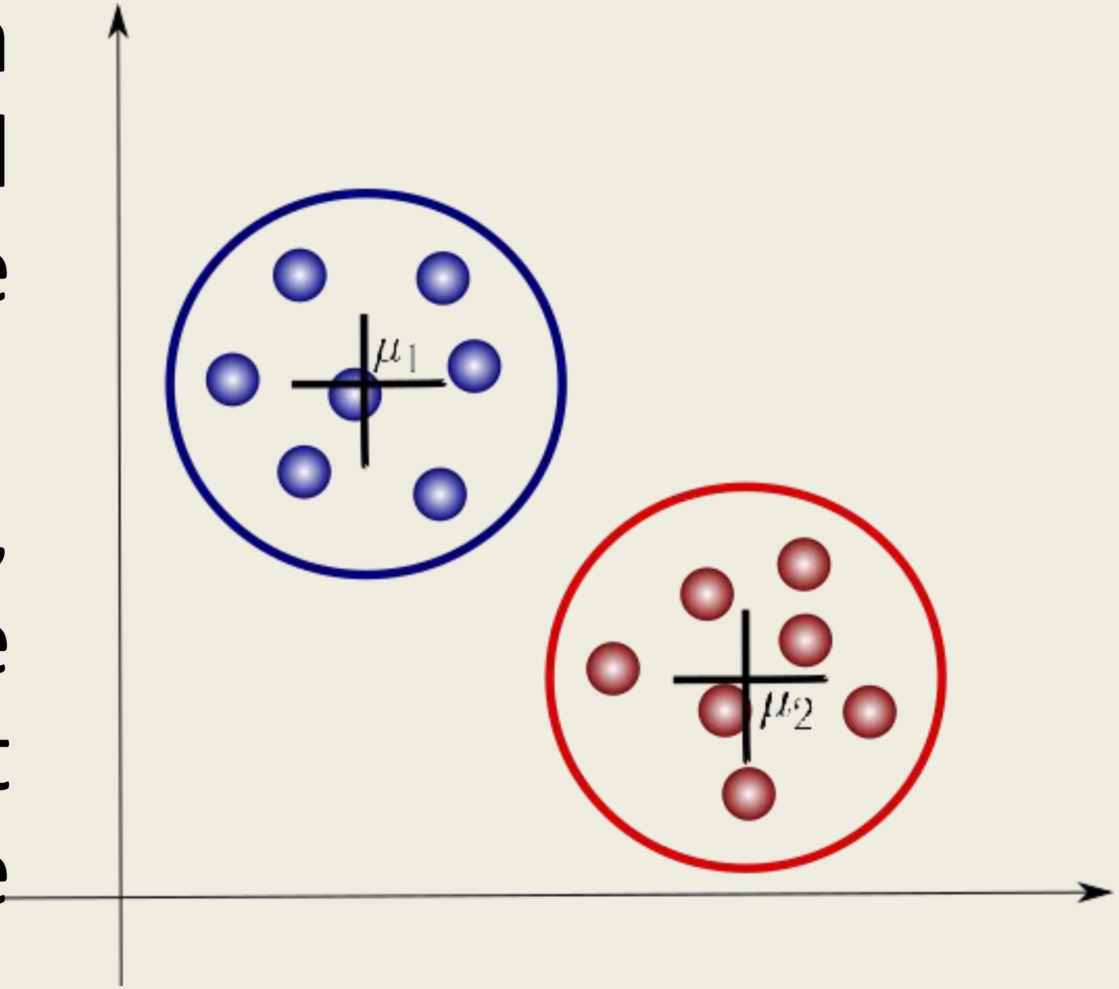
- Hence, we need a different way to assign clusters to the data points. So instead of using a distance-based model, we will now use a distribution-based model.

Introduction to Gaussian Mixture Models (GMMs)

- Gaussian Mixture Models (GMMs) assume that there are a certain number of Gaussian distributions, and each of these distributions represent a cluster. Hence, a Gaussian Mixture Model tends to group the data points belonging to a single distribution together.

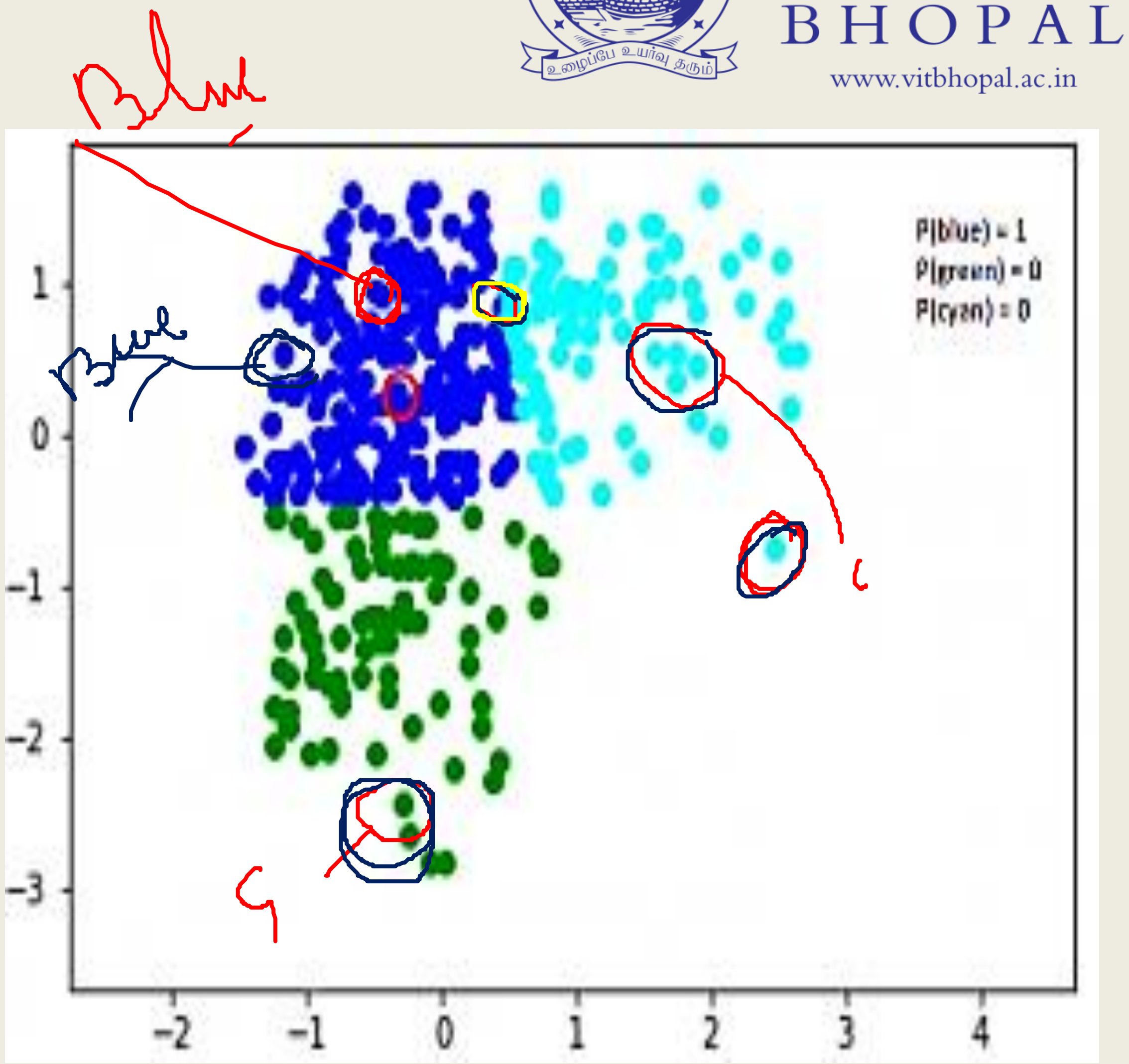
- **The normal distribution is a probability distribution. It is also called Gaussian distribution because it was first discovered by Carl Friedrich Gauss. The normal distribution is a continuous probability distribution that is very important in many fields of science.**

- Let’s say we have three Gaussian distributions – GD1, GD2, and GD3. These have a certain mean (μ1, μ2, μ3) and variance (σ1, σ2, σ3) value respectively.
- For a given set of data points, our GMM would identify the probability of each data point belonging to each of these distributions.



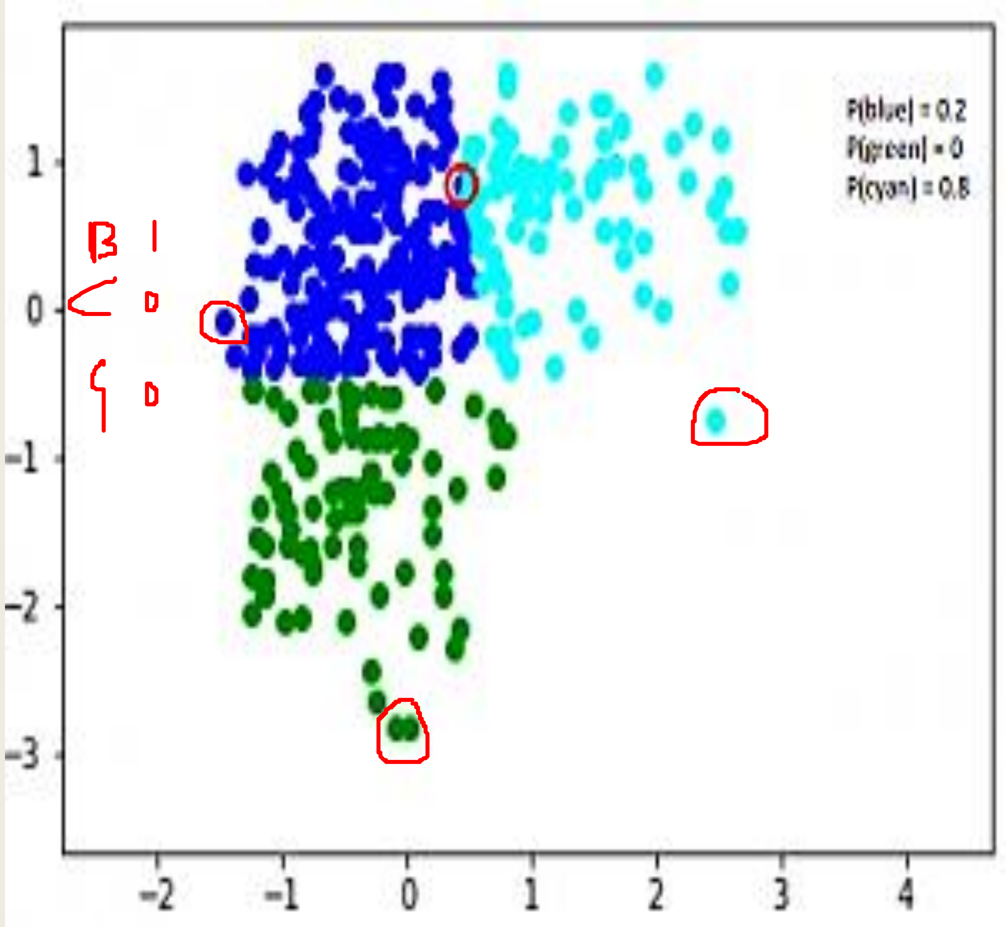
- Gaussian Mixture Models are probabilistic models and use the soft clustering approach for distributing the points in different clusters. I'll take another example that will make it easier to understand.

- Here, we have three clusters that are denoted by three colors – Blue, Green, and Cyan. Let's take the data point highlighted in red. The probability of this point being a part of the blue cluster is 1, while the probability of it being a part of the green or cyan clusters is 0.

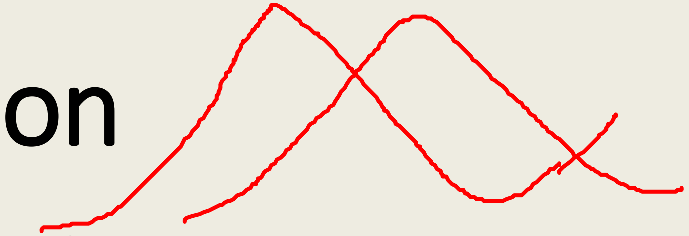


- Now, consider another point – somewhere in between the blue and cyan (highlighted in the below figure). The probability that this point is a part of cluster green is 0, right? And the probability that this belongs to blue and cyan is 0.2 and 0.8 respectively.

- Gaussian Mixture Models use the soft clustering technique for assigning data points to Gaussian distributions.

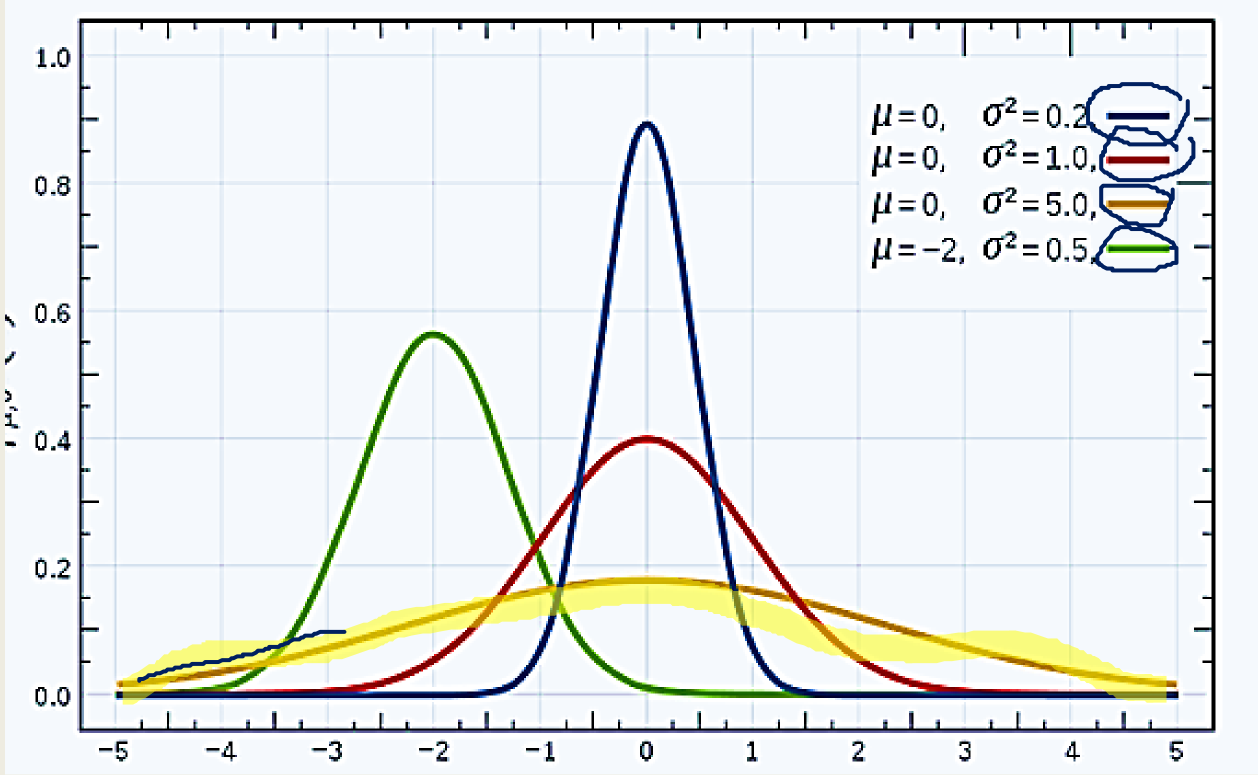


The Gaussian Distribution

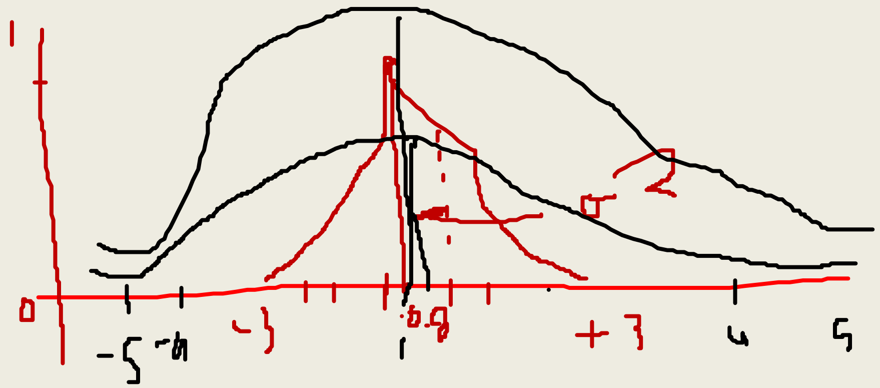


- I’m sure you’re familiar with Gaussian Distributions (or the Normal Distribution). It has a bell-shaped curve, with the data points symmetrically distributed around the mean value.

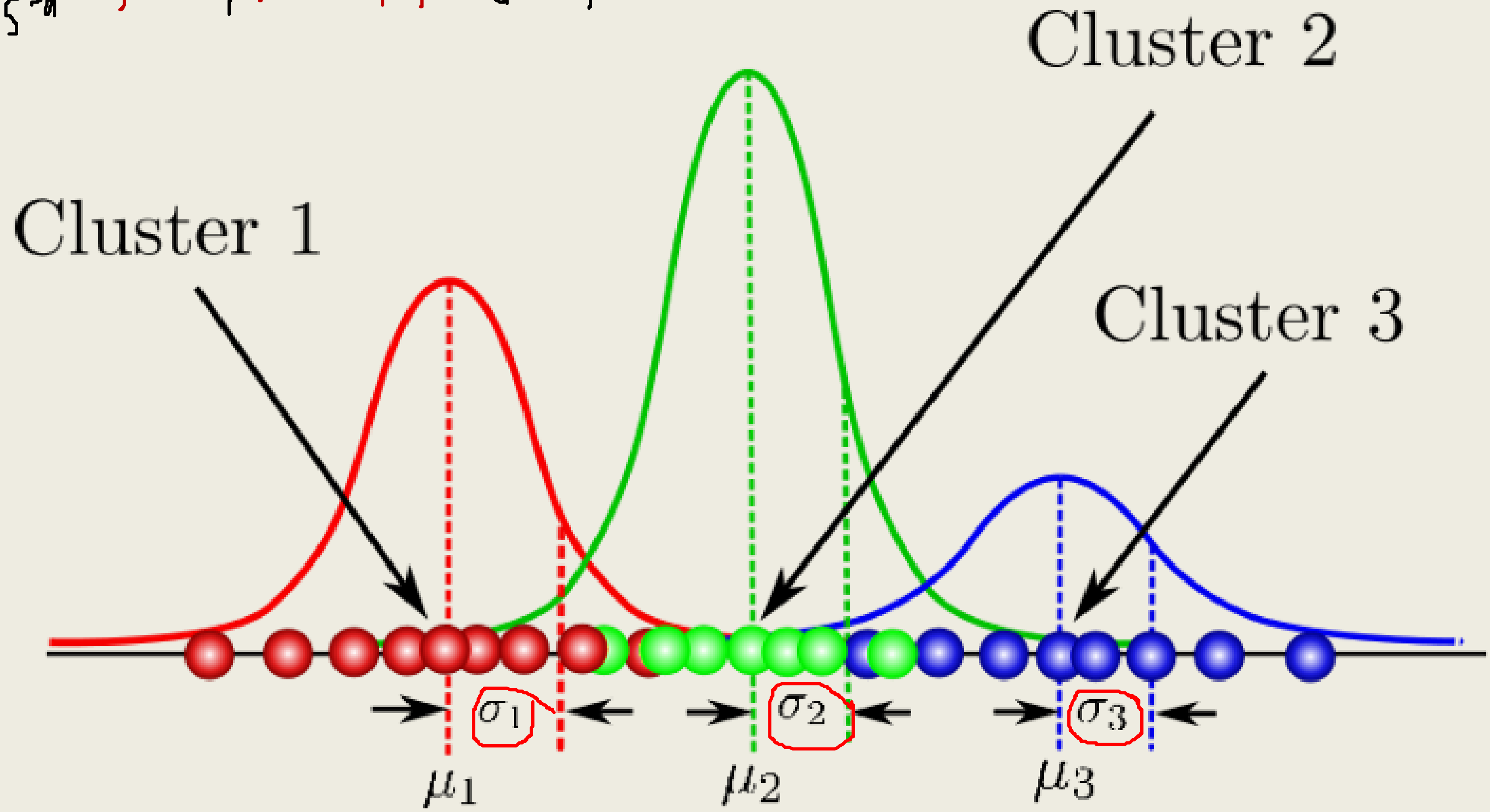
- The image has a few Gaussian distributions with a difference in mean (μ) and variance (σ2). Remember that the higher the σ value more would be the spread

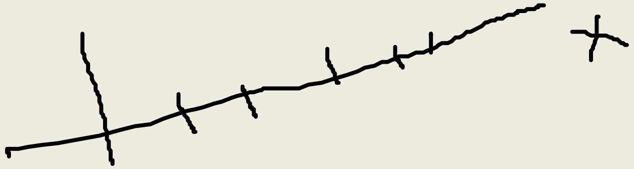


σ^2


1
0
-5
-4
-3
0.9
1
+3
4
5
σ-2

$\mu = 0.4$


Cluster 1
Cluster 2
Cluster 3
σ_1
σ_2
σ_3
μ_1
μ_2
μ_3



- In a one dimensional space, the probability density function of a Gaussian distribution is given by:

PDF

data point

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

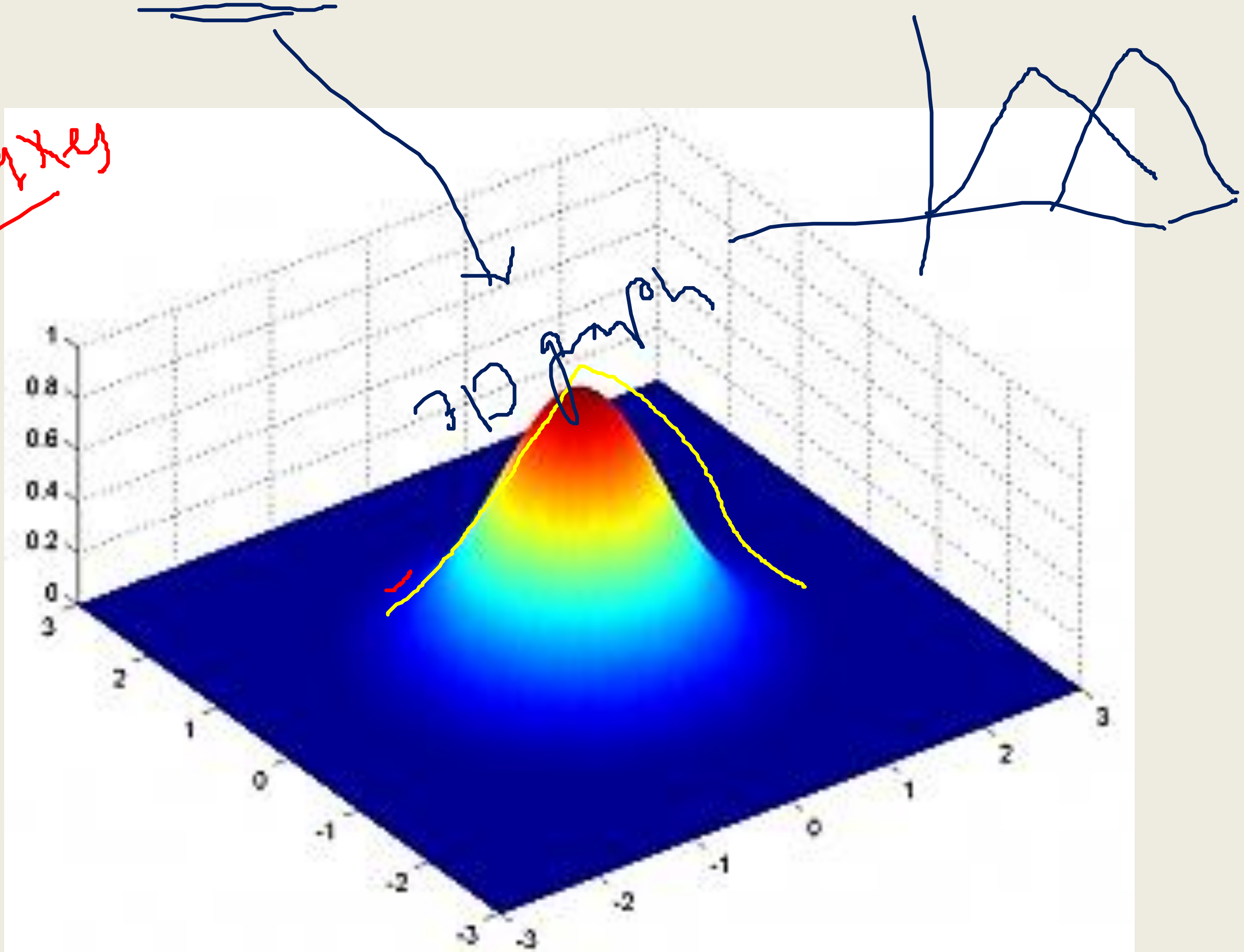
✓

- where μ is the mean and σ2 is the variance.

σ^2

- But this would only be true for a single variable. In the case of two variables, instead of a 2D bell-shaped curve, we will have a 3D bell curve as shown below:

3- axes



- The probability density function would be given by:

$$f(x \mid \mu, \Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left[-\tfrac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\right]$$

2×2

- where x is the input vector, μ is the 2D mean vector, and Σ is the 2×2 covariance matrix. The covariance would now define the shape of this curve. We can generalize the same for d-dimensions.

$\underline{x}$ ✓

- Thus, this multivariate Gaussian model would have x and μ as vectors of length d, and Σ would be a d x d covariance matrix.

EM Gmm

M σ^2

- Hence, for a dataset with d features, we would have a mixture of k Gaussian distributions (where k is equivalent to the number of clusters), each having a certain mean vector and variance matrix. But wait – how is the mean and variance value for each Gaussian assigned?

- **These values are determined using a technique called Expectation-Maximization (EM). We need to understand this technique before we dive deeper into the working of Gaussian Mixture Models.**

Expectation-Maximization in Gaussian Mixture Models

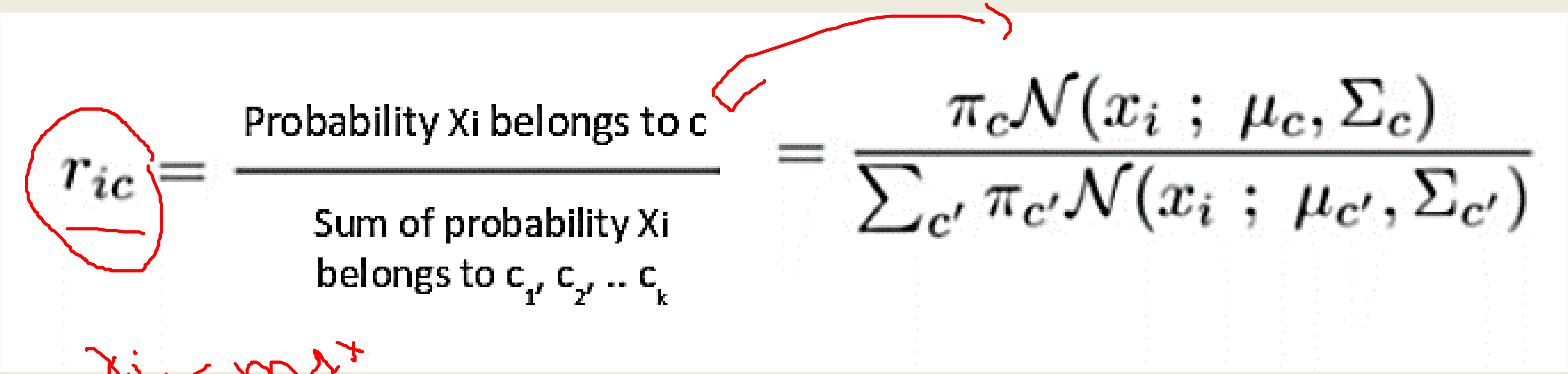
$K \quad \mu_i - \mu_k$

- Let’s say we need to assign *k* number of clusters. This means that there are k Gaussian distributions, with the mean and covariance values to be $\mu_1, \mu_{2, \dots} \mu k$ and $\Sigma_1, \Sigma_2, \dots \Sigma k$. Additionally, there is another parameter for the distribution that defines the number of points for the distribution. Or in other words, the density of the distribution is represented with Π_i.

- Now, we need to find the values for these parameters to define the Gaussian distributions. We already decided the number of clusters, and randomly assigned the values for the mean, covariance, and density. Next, we’ll perform the E-step and the M-step!

E-step

- For each point x_i, calculate the probability that it belongs to cluster/distribution $c_1, c_2, \ldots c_k$. This is done using the below formula:



$$r_{ic} = \frac{\text{Probability Xi belongs to c}}{\text{Sum of probability Xi belongs to } c_1, c_2, .. c_k} = \frac{\pi_c \mathcal{N}(x_i\ ;\ \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i\ ;\ \mu_{c'}, \Sigma_{c'})}$$

$r_{ic} = \max$

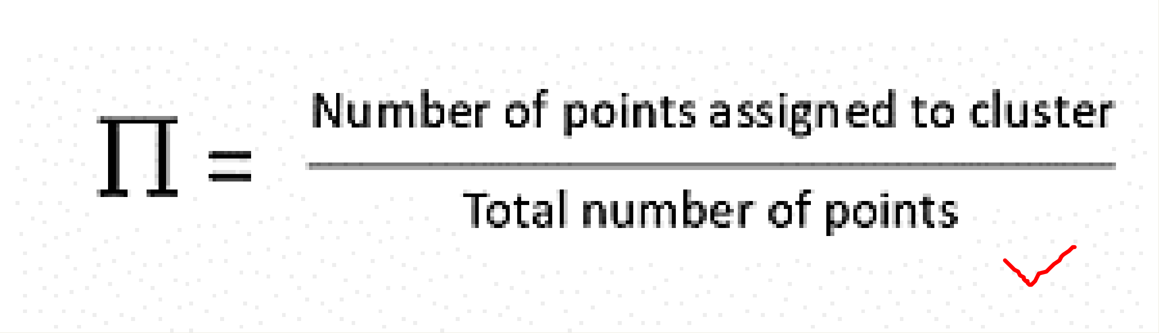
- This value will be high when the point is assigned to the right cluster and lower otherwise.

M-step

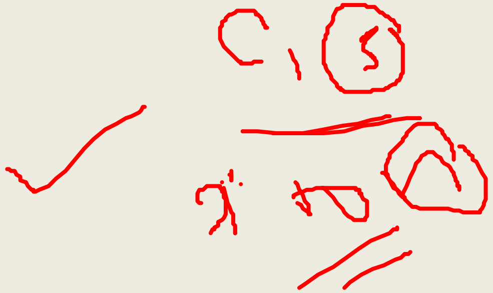
- Post the E-step, we go back and update the Π, μ and Σ values. These are updated in the following manner:

- 1. The new density is defined by the ratio of the number of points in the cluster and the total number of points:

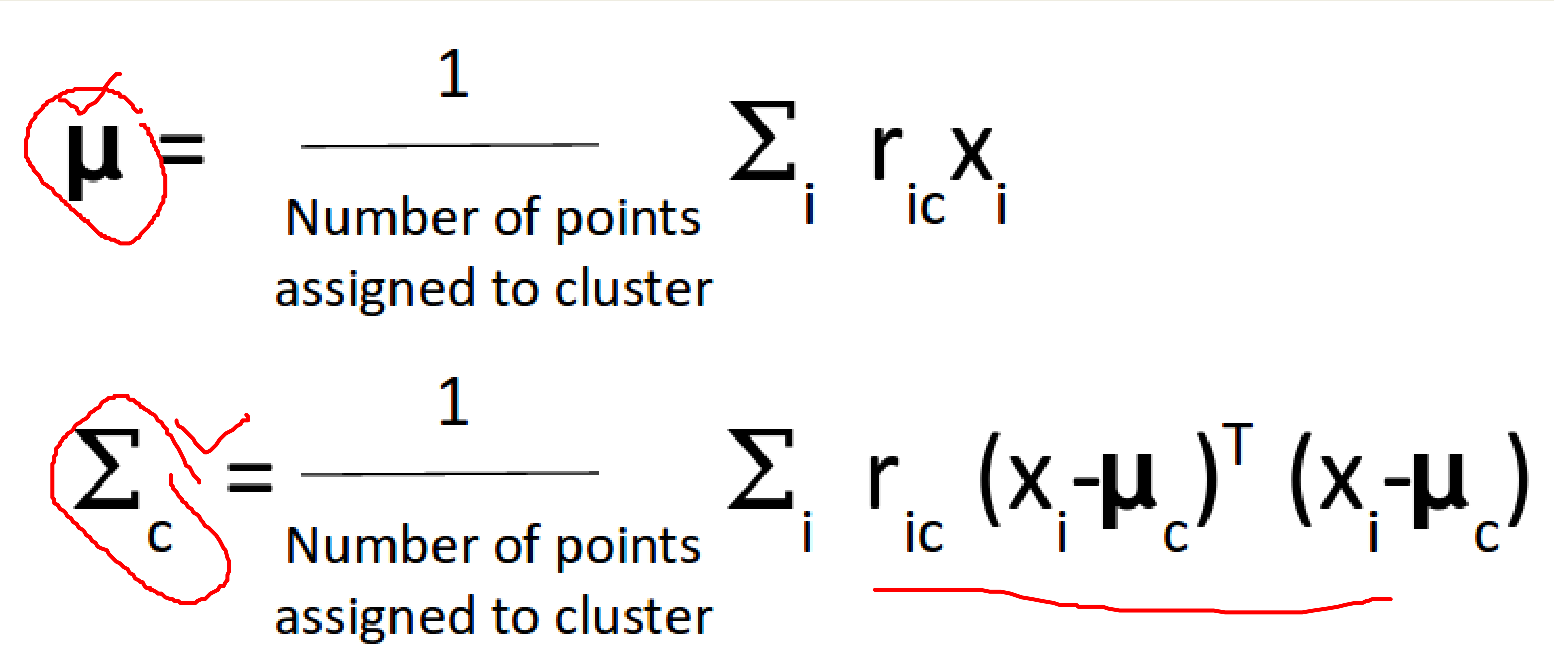
$\frac{3}{5}$



$$\Pi = \frac{\text{Number of points assigned to cluster}}{\text{Total number of points}}$$



- 2. The mean and the covariance matrix are updated based on the values assigned to the distribution, in proportion with the probability values for the data point. Hence, a data point that has a higher probability of being a part of that distribution will contribute a larger portion:



$$\mu = \frac{1}{\text{Number of points assigned to cluster}} \sum_i r_{ic} x_i$$

$$\Sigma_c = \frac{1}{\text{Number of points assigned to cluster}} \sum_i \underline{r_{ic} (x_i - \mu_c)^T (x_i} - \mu_c)$$

- Based on the updated values generated from this step, we calculate the new probabilities for each data point and update the values iteratively. This process is repeated in order to maximize the log-likelihood function. Effectively we can say that the

- *k-means only considers the mean to update the centroid while GMM takes into account the mean as well as the variance of the data!*