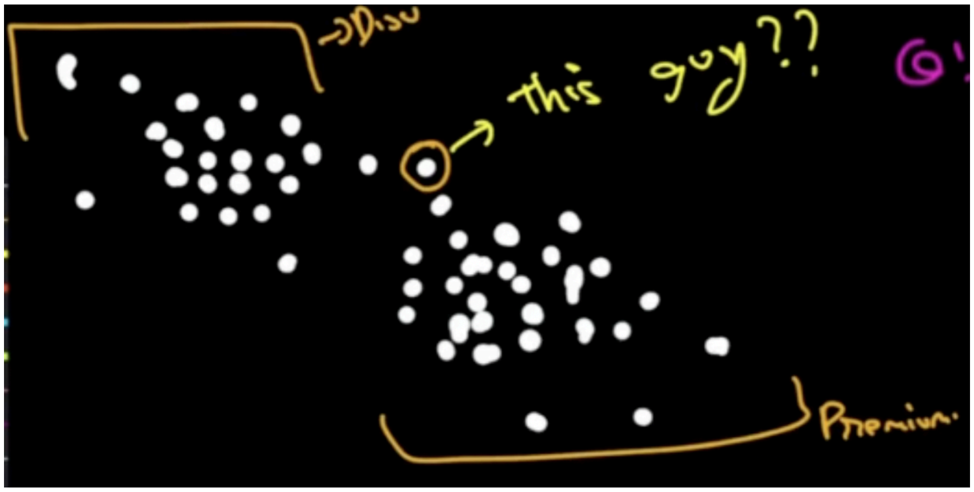
GMM

Scenario: We want to suggest deals to a new customer (“**this guy**”) for which we want to know if he’s a **premium** or **discount** customer.



Can we use hard clustering? (Kmeans) ?

It would assign this customer to the premium cluster (because it's closer to that cluster)

- but say in the future the deals that you have shown to the customer don’t get converted to purchases then this clustering won’t be the most appropriate **Why?**
- because we can see here that this customer is also somewhere near to discount customers, therefore it can be the conclusion that this customer is partially discount type and partially premium, and showing him only premium deals won’t be the best option.

Soft clustering with Gaussian mixture model

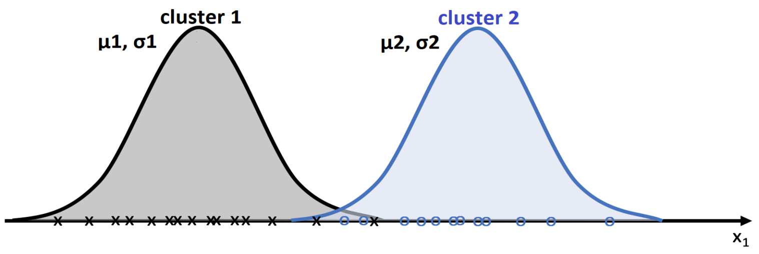
- A probability of each data point being in different clusters is assigned instead of putting each data point completely into a separate cluster. (probabilistic model)

- We can represent the clusters in our data using the **Gaussian distributions**.

- A Gaussian mixture comprises k Gaussians where each Gaussian has:
    - Mean μ
    - variance σ^2

- In GMM **k** (number of clusters) is a **hyperparameter**.
- In GMM mean (μ) and variance (σ^2) are the parameters.

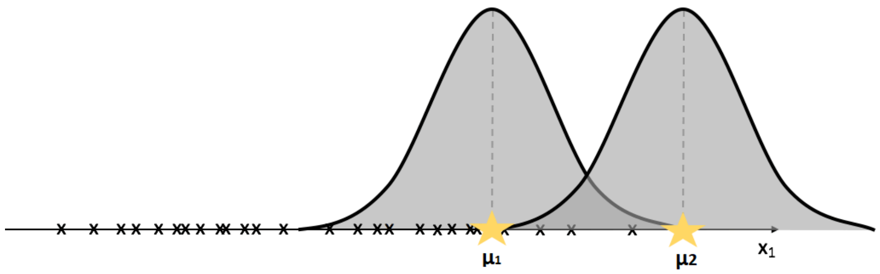
- For example, in the below given 1-dimensional data, each cluster is represented using a Gaussian distribution.



1-D Example of GMM

GMM algorithm Steps

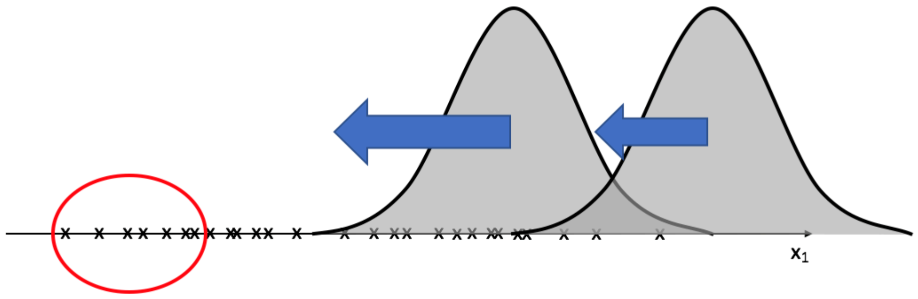
➔ Randomly initialize μ and σ^2.



➔ Update μ and σ^2.

- Calculate the probability of each point belonging to each Gaussian using the formula: $N(x \mid \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.

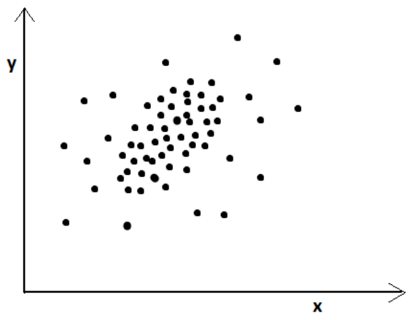
- Assign all the points which have the highest probability to their respective Gaussians.
- Now, compute new μ for each Gaussian as the mean of the points belonging to that Gaussian
- Similarly, update the variance.
- Here μ is updated as the **weighted average** of the points belonging to a Gaussian against a simple average which we followed in K-Means.
  $\boldsymbol{\mu_j = 1/n \sum x_i. (P_i(x_i))}$, where **n** is the number of points with a non-zero probability of belonging to a cluster and $\boldsymbol{P_i(x_i)}$ represents the probability of ith point belonging to the jth cluster.
- σ^2 of the ith cluster is similarly updated in a weighted manner.



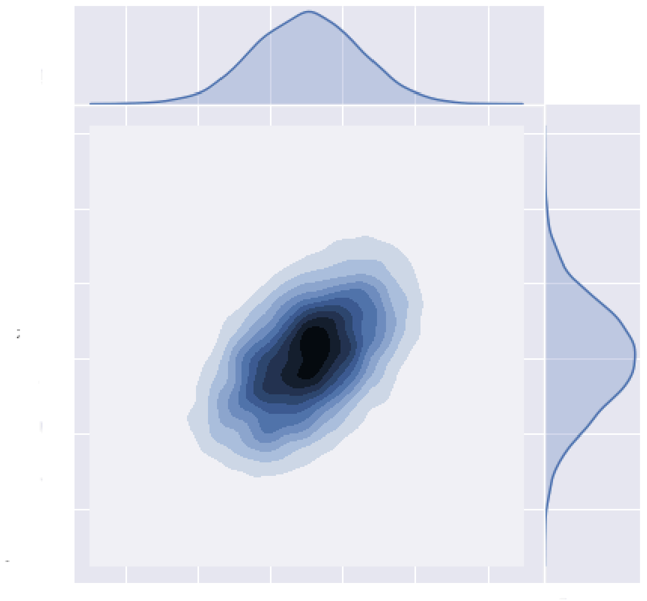
➔ After multiple updates, we will have tightly fitting Gaussian distributions representing different clusters.

2-Dimensional Gaussians

- Let's consider two variables **x** and **y** with the following 2D scatter plot.



- The corresponding Multivariate Gaussian distribution would be:



- The part where the color is dark has more density (probability) of points and vice versa.
- Animation for modifying parameters of a 2d Gaussian distribution:
  - **Bivariate Gaussian**

- The above multivariate Gaussian distribution is defined by five parameters:

1. **μ_x** - the mean of the x-coordinate.
2. **μ_y** - mean of y-coordinate
3. **σ^2_x** - variance in x direction.
4. **σ^2_y** - variance in the y direction.
5. **Cov(x,y)** - Covariance between x and y.

- The above ideas can be applied to 2D and N-D Gaussians, and this is how the GMM algorithm works. We will update the parameters in more dimensions using the formulas we used in 1-D and will get the results.

- Animation for GMM algorithm:
  - **GMM Animation**