Clustering Basic;
k-Means Clustering & EM Algorithm

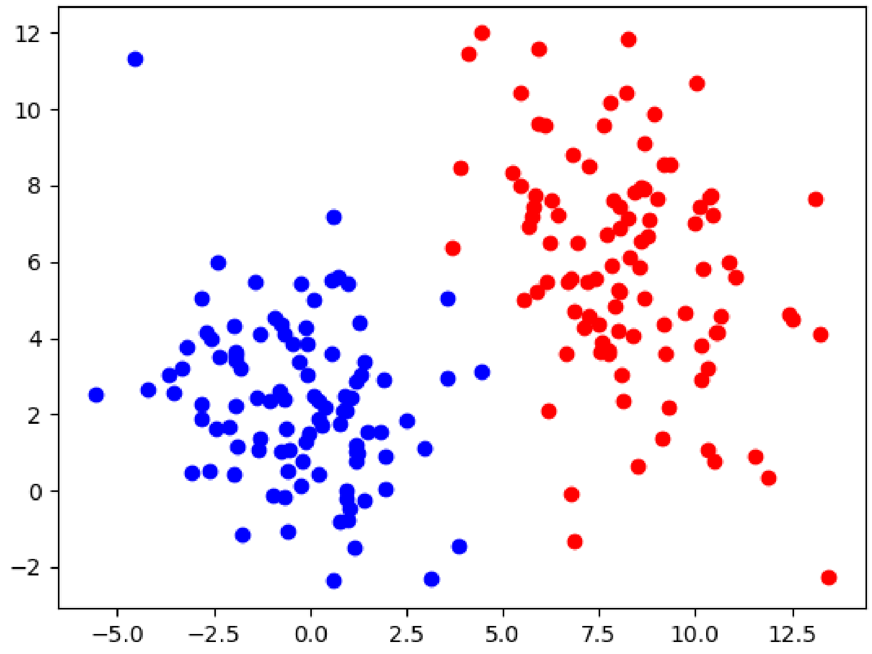
Hanwool Jeong

hwjeong@kw.ac.kr

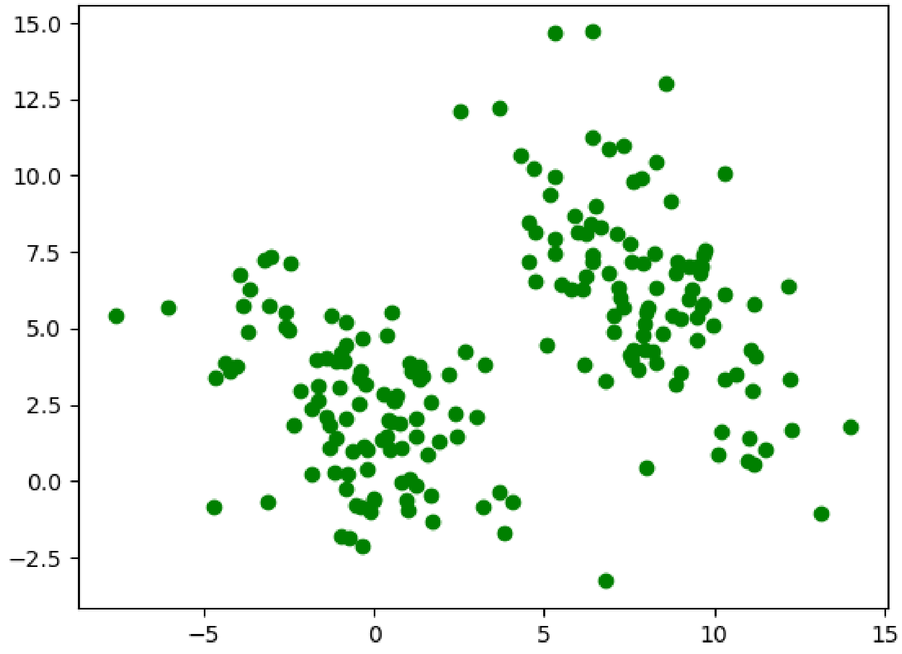
Revisit Logistic Regression

General Flow of Supervised Learning

- During the training, the output response (e.g., class) is known for each input.
- That is,

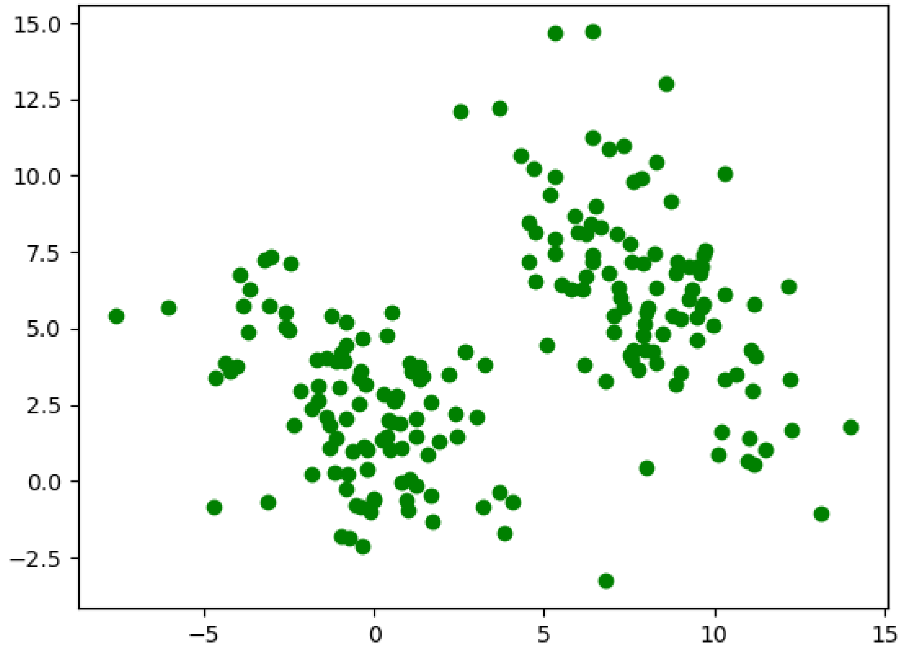


But, what if?

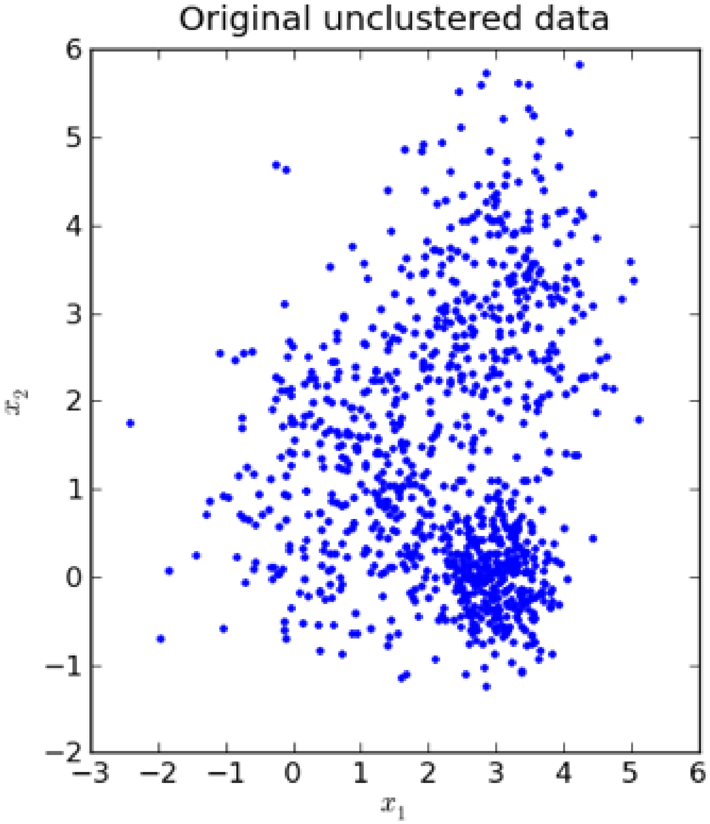


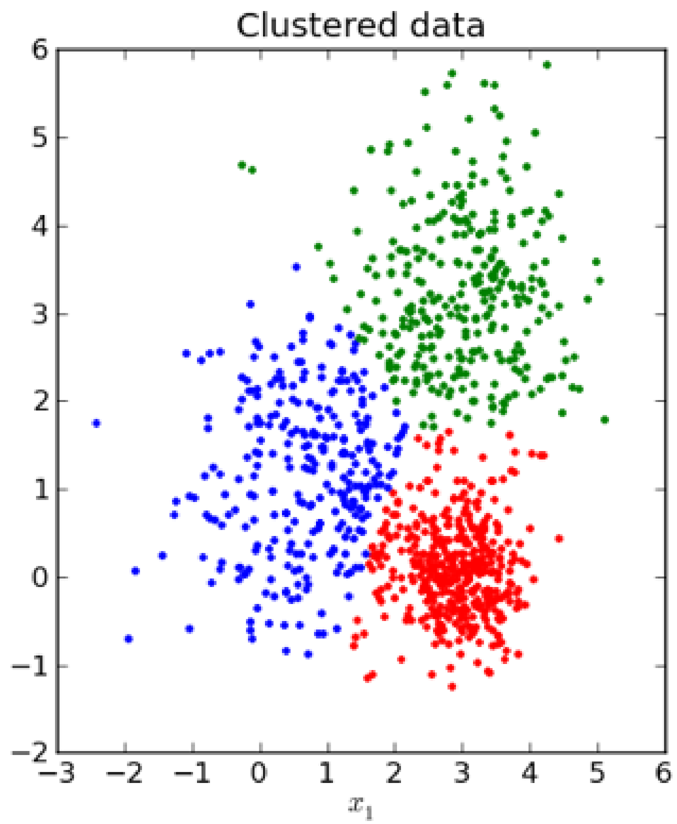
Revisit Supervised vs. Unsupervised

- Supervised : Driven by input-output data training set
- Unsupervised : Only inputs are given.
- **Clustering** is the representative unsupervised learning algorithm.



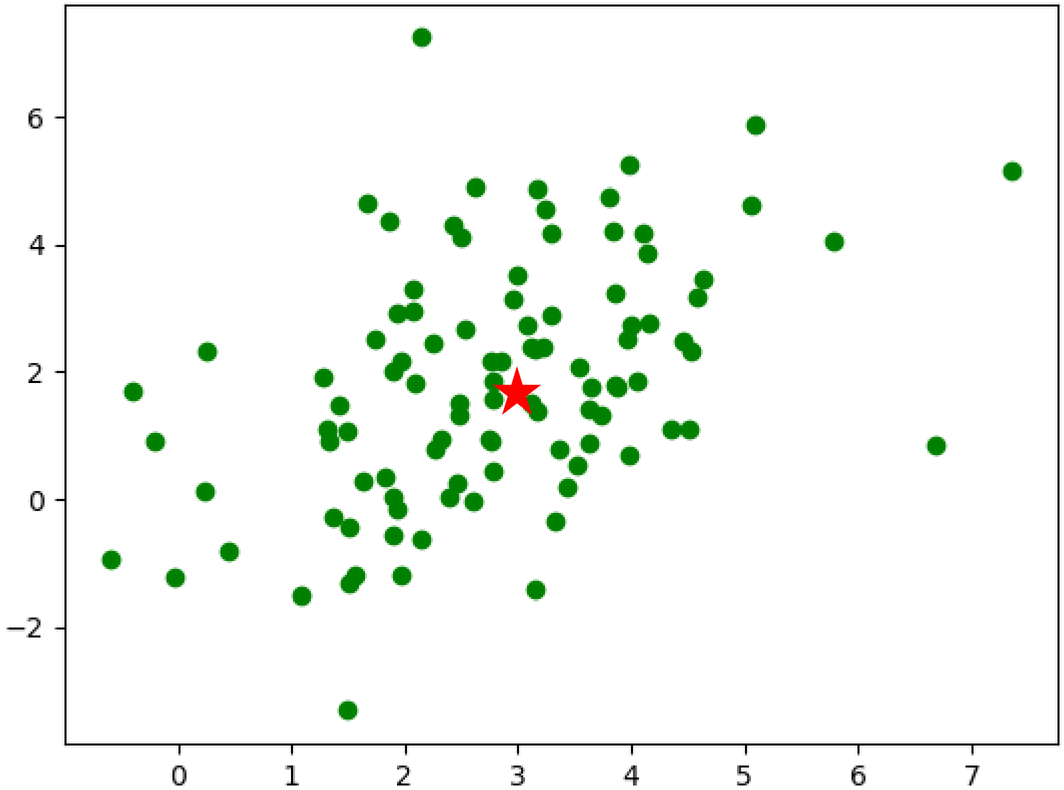
Result of Clustering





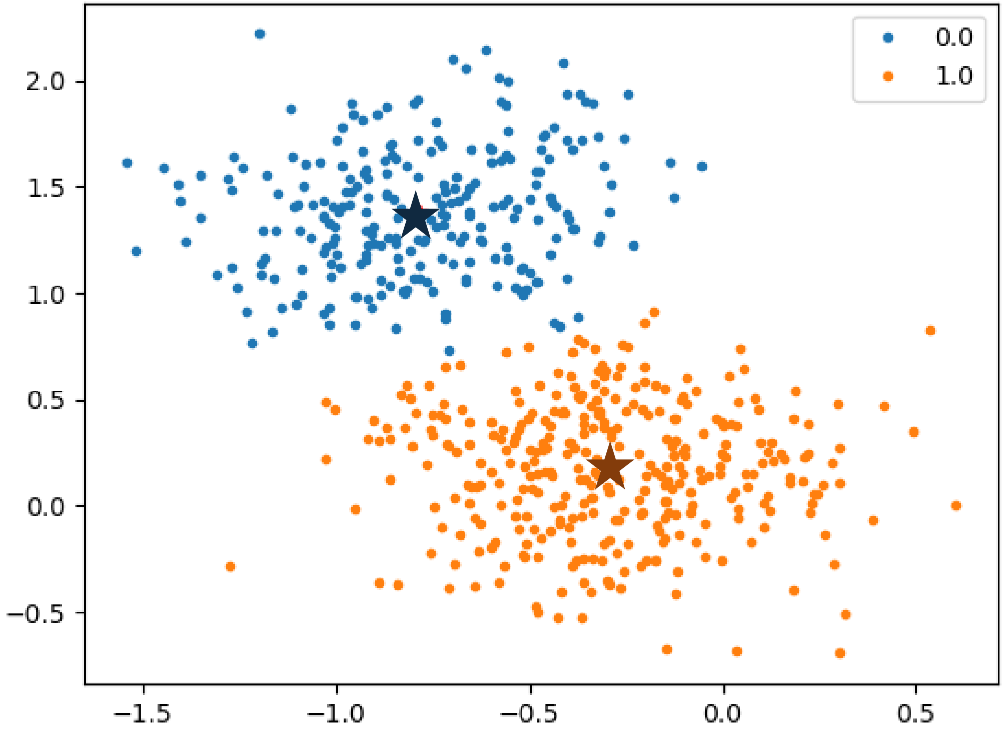
Mean of Data?

- You aware of mean? What is mean of [3, 4, 8, 9, 12]?
- How about the mean of the following 2D data? $\mu = (1/N)\Sigma\mathbf{x}$

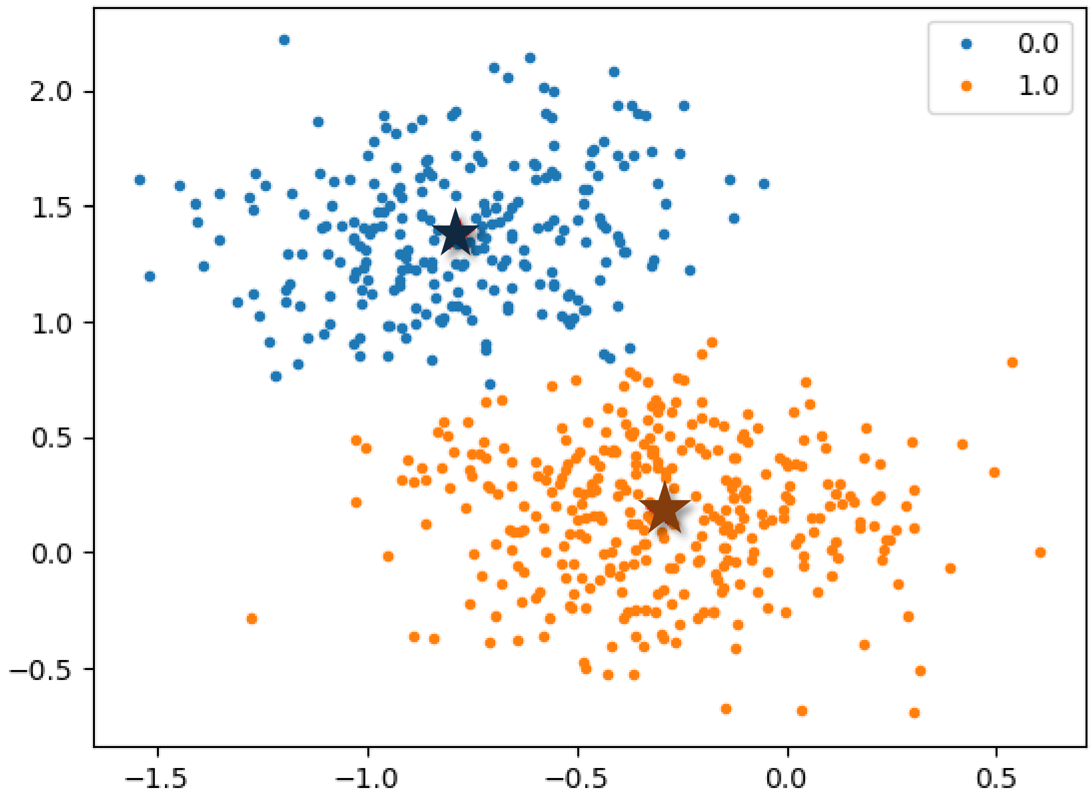


We Can Use Mean For Clustering!

- Assume that the means of two clusters k_1 and k_2 (=z_1 and z_2) are known
- You can map arbitrary **x** to k_1 or k_2 by comparing the distance to z_1 and z_2
- Key is **"How can we decide z_1 and z_2?"**



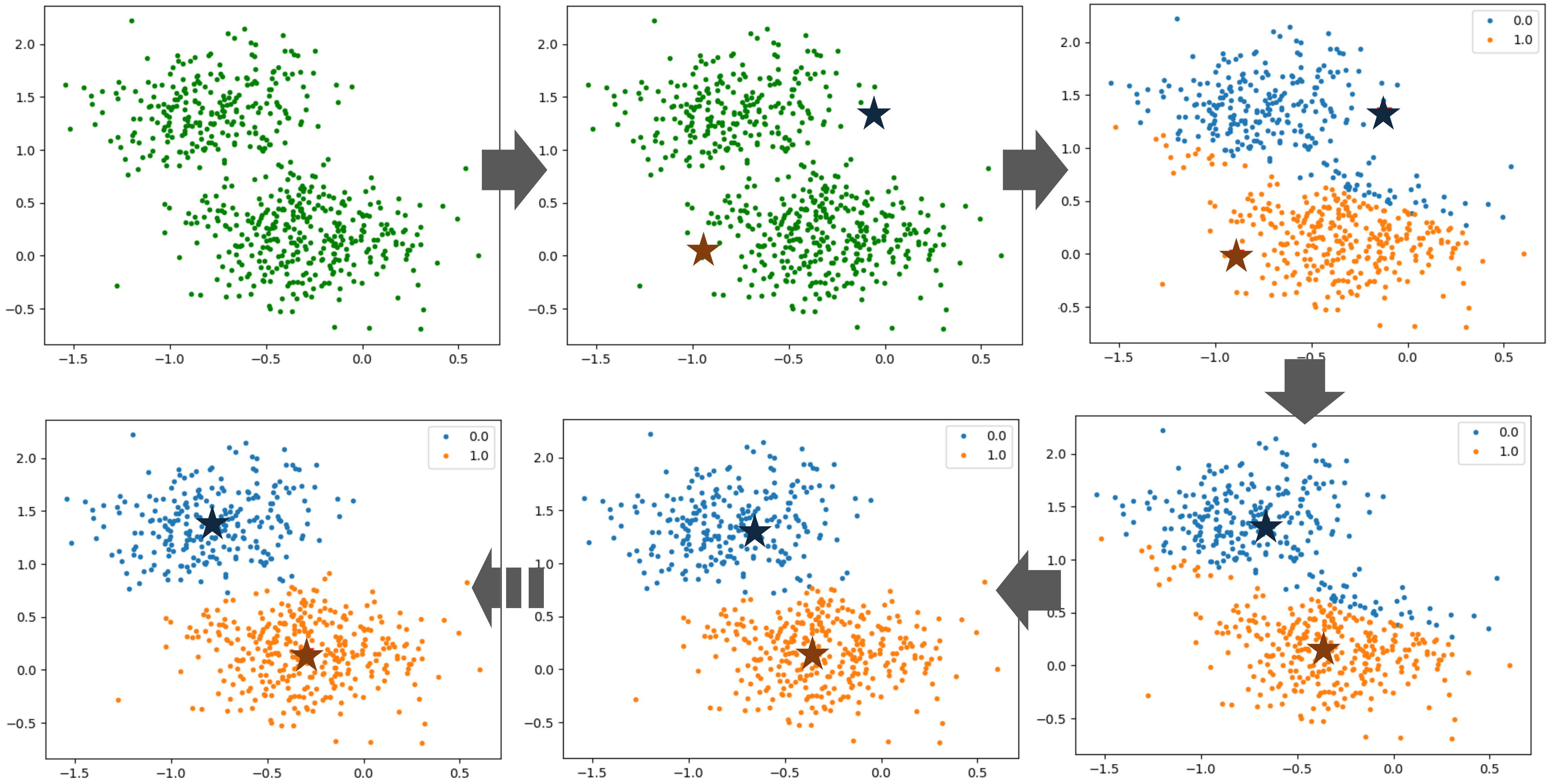
k-Means Clustering!



[illegible]

Until the clustered result is unchanged

Steps for k-Means



k-Means Clustering

- Given inputs **X** and the number of clusters of **k**, K
- Defining latent variable matrix **Z**, algorithm is like as follows:

```
Initialize Z = {z_1, z_2, z_3, … z_K}
while(true)
    for(i=1 to N)
        Map x_i into the nearest z_j
    if(No change of mapping from the prev. loop) break
    for(j=1 to K)
        replace z_j with the mean of the x_i mapped to z_j
for (j=1 to K)
    allocate the samples mapped to z_j to k_j
```

- Output : $\mathbf{k} = \{\mathbf{k}_1, \mathbf{k}_2, \dots, \mathbf{k}_K\}$

k-Means vs. k-Mediods

- k-means clustering is weak for outliers.
- What else for the weakness of k-means algorithm?

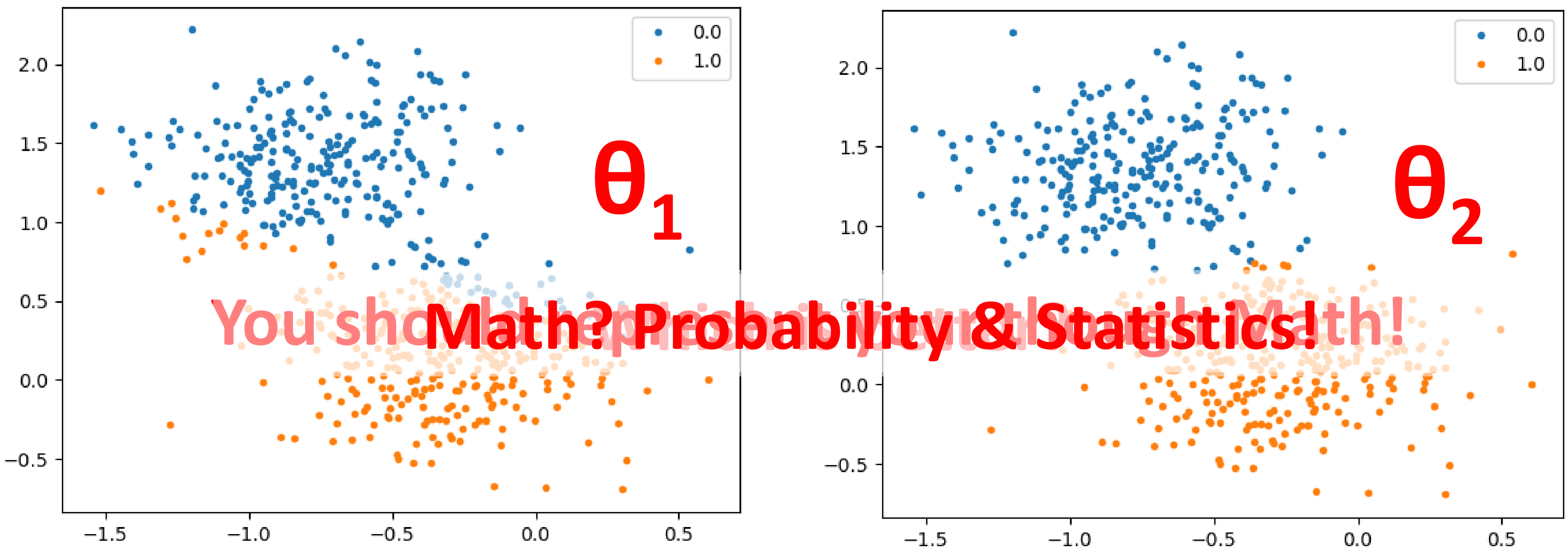
Generalizing k-Means Algorithm

- Output prediction update is nothing but MLE during training phase. That is, finding **θ** that maximizes p(D/**θ**).

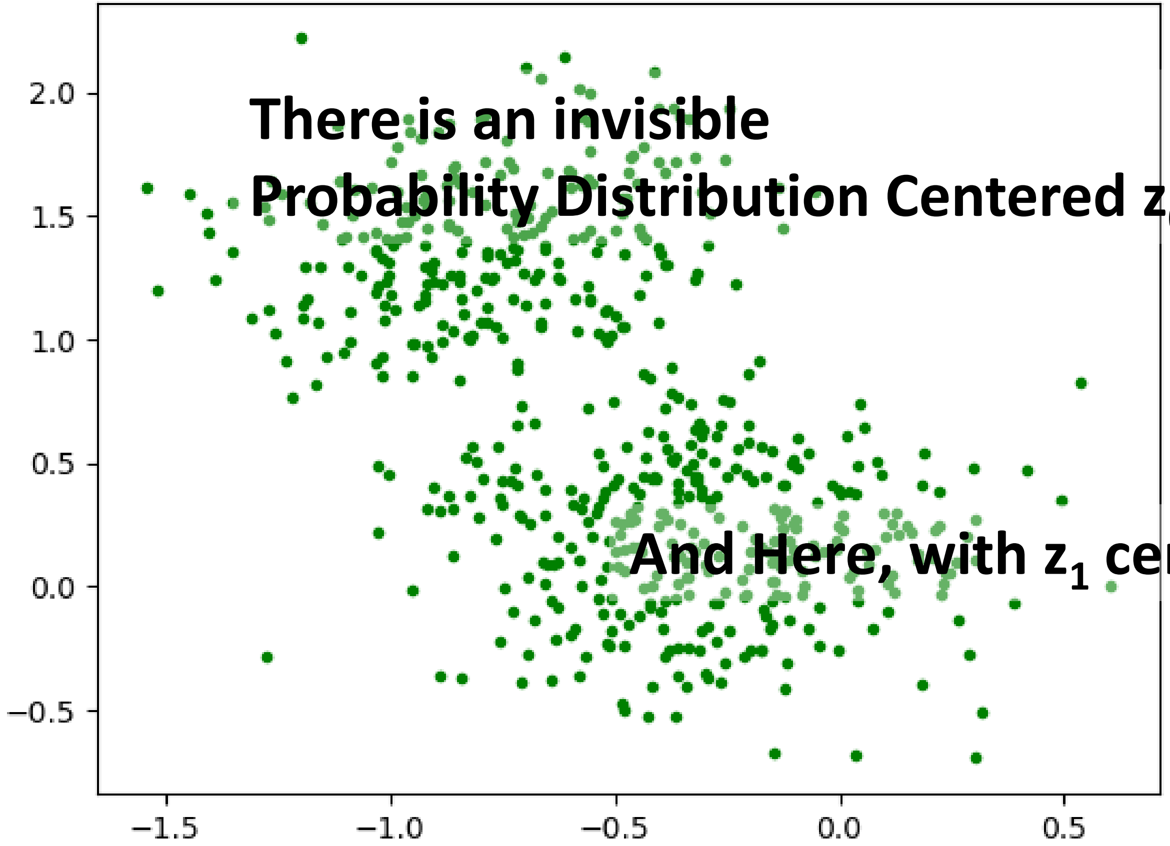
= Deciding θ

- When Z is given, allocating X to nearest z_i to decide the clusters.

➔ **Is it MLE? What is θ in the clustering? What is p(D/θ)?**



How To "Evaluate" Clustering with Number (=Quantify)



There is an invisible Probability Distribution Centered z_0 here

And Here, with z_1 center

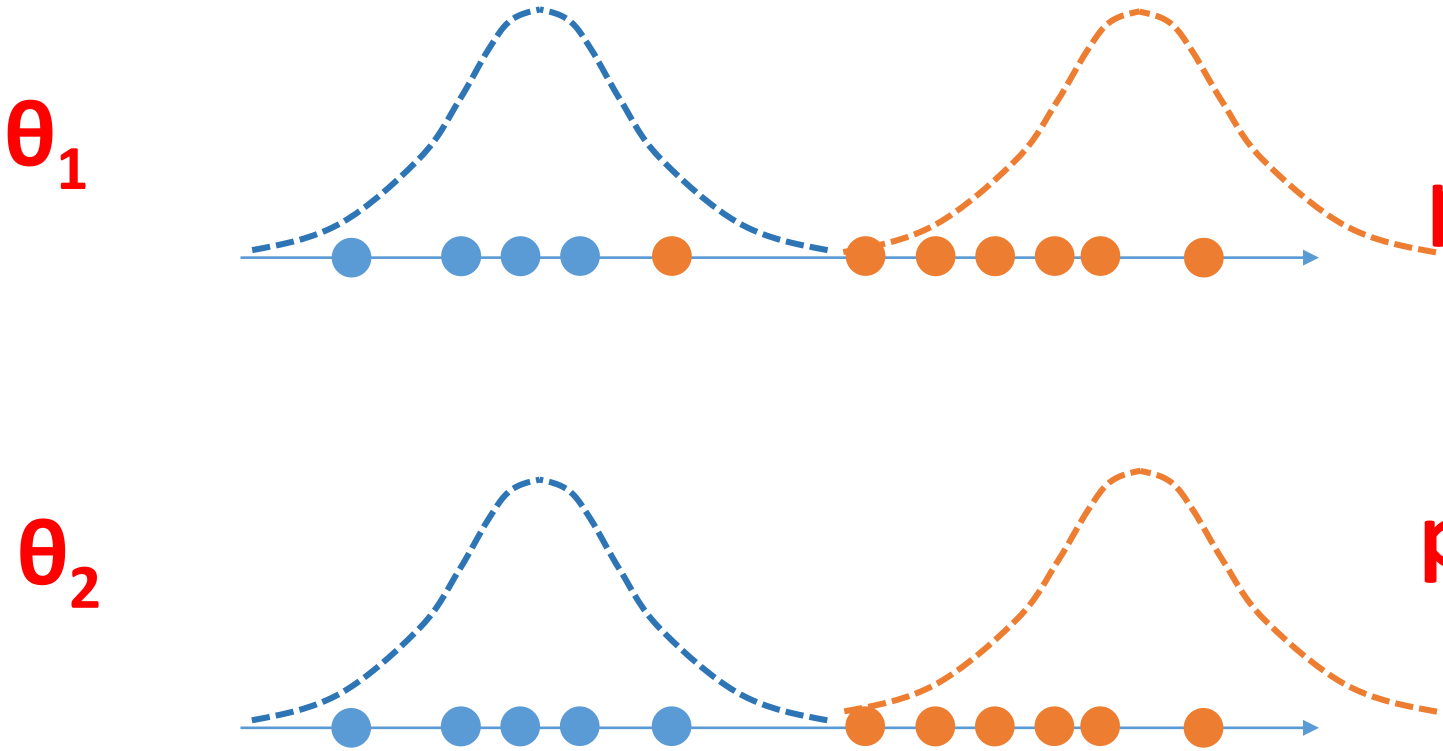
Now We Can Quantify Our Thought "Which is Better?"

- 1D is easier.

✓ Note that, **for now**, θ is not PDF parameter **z**= (z_0, z_1)

✓ θ is related to how the mapping is performed

For now, **z** is given, which is determined somehow else



$p(X/\theta_1, z)$

vs.

$p(X/\theta_2, z)$

➔ **How can we decide z then?**

MLE Formulation of Clustering

- MLE in clustering is nothing but maximizing the following

$$\mathbf{p(D/\theta) = p(X/\theta)}$$

- Based on the MLE results, **θ** is determined then Z is determined properly, (but usually based on the averaging out method)

Revisit Latent Variable vs. Output in k-Means Clustering → How are they related?

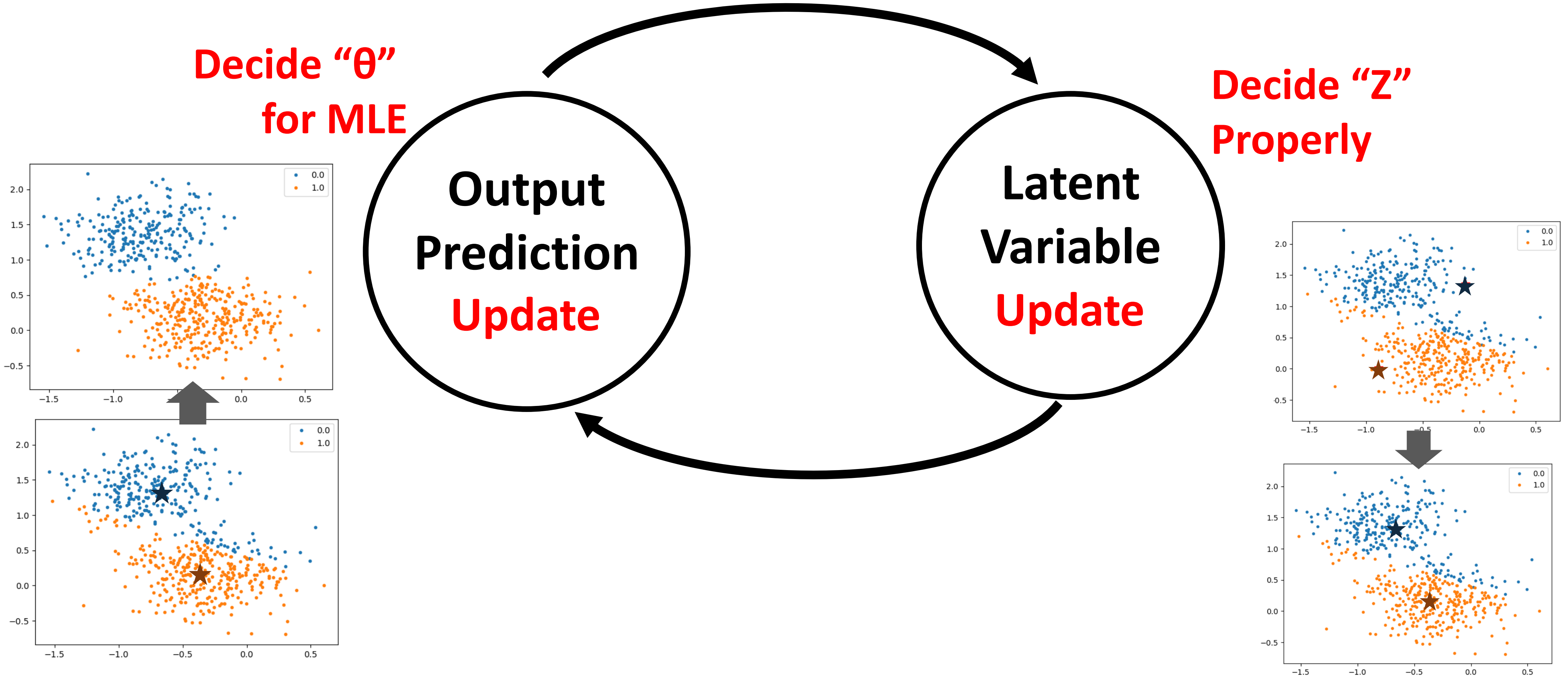
- Given inputs **X** and the number of clusters of **C**, K
- Defining **latent variable** matrix **Z**, algorithm is like as follows:

```
Initialize Z = {z_1, z_2, z_3, … z_K}
while(true)
    for(i=1 to N)
        Map x_i into the nearest z_j
    if(No change of mapping from the prev. loop) break
    for(j=1 to K)
        replace z_j with the mean of the x_i mapped to z_j
for (j=1 to K)
    allocate the samples mapped to z_j to c_j
```

- **Output** : $\mathbf{C = \{c_1, c_2, \ldots, c_K\}}$

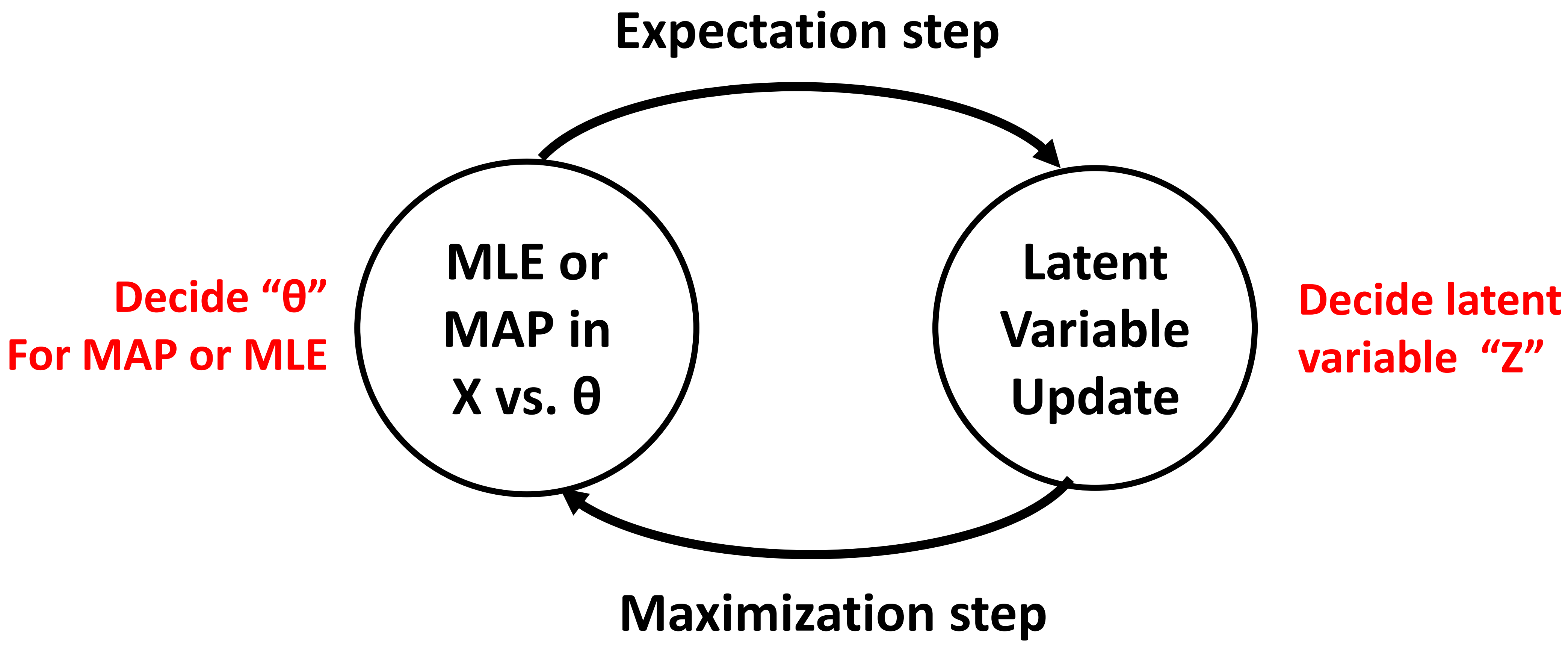
Latent Variable vs. Output Prediction in K-Means

- There is a cycle in the clustering!



Expectation Maximization (EM) Algorithm

- You should feel and deeply understand it
- And generalize it into other clustering algorithms than k-means



What Do You Think about This?