Assignment 4

Devansh Jain, 190100044

30th Oct 2021

Contents

1 Clustering

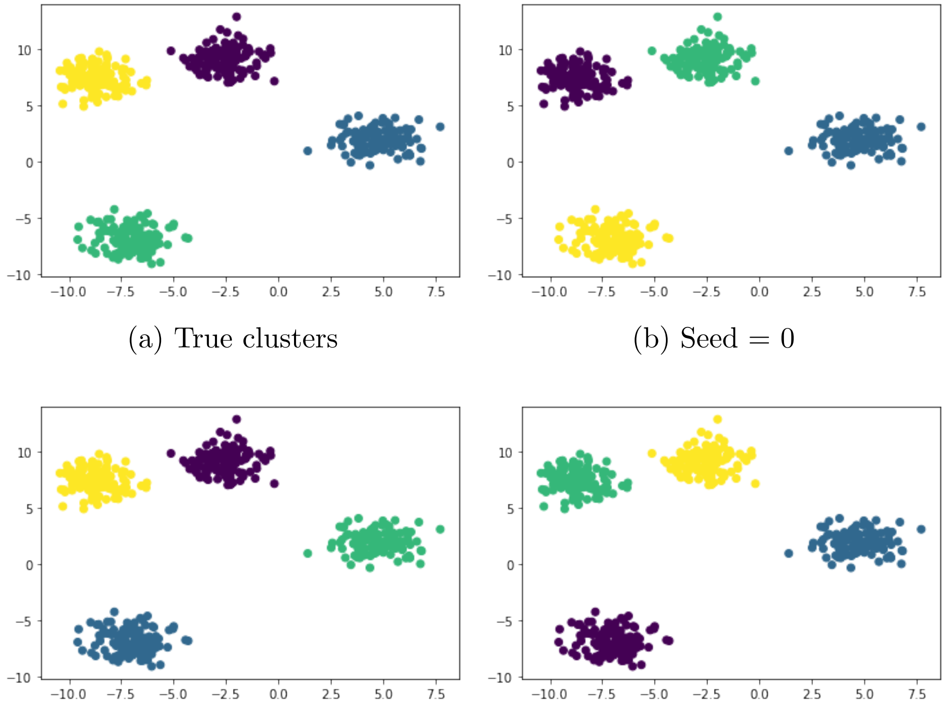
1.1 CS 335 KMeans Implementation

(i)

Code for the class `Kmeans` updated in notebook.

(ii)

For Dataset 1, we observe that the algorithm works well as the dataset is linearly separable.
For Dataset 2, we get algorithm converges to same centroids (except flipping of color) but these are not able to separate the clusters as the dataset isn't linearly separable.
For Dataset 2, we get different centroids for different initialization. This can be explained by the radial symmetry.



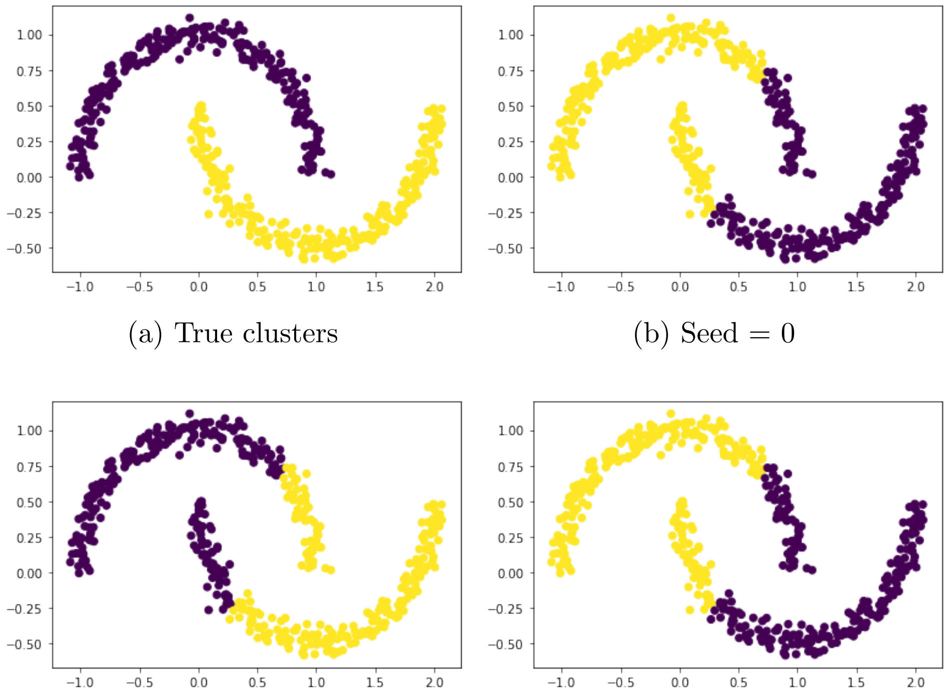
(a) True clusters

(b) Seed = 0

(c) Seed = 11

(d) Seed = 123

Figure 1: Dataset 1



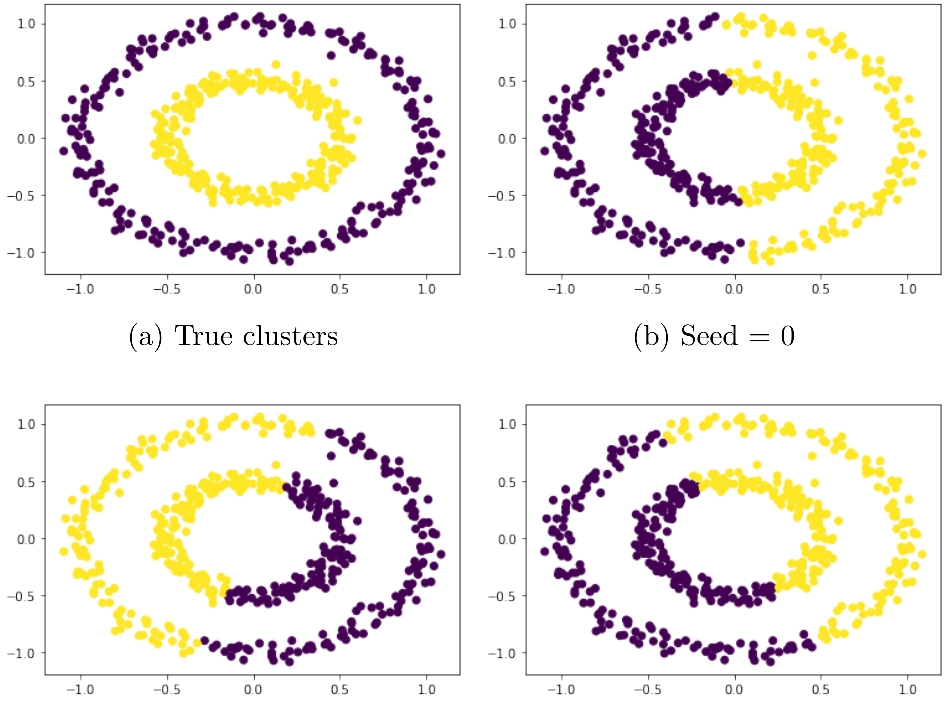
(a) True clusters

(b) Seed = 0

(c) Seed = 11

(d) Seed = 123

Figure 2: Dataset 2



(a) True clusters

(b) Seed = 0

(c) Seed = 11

(d) Seed = 123

Figure 3: Dataset 3

(iii)

We need to initialize the centroids at different locations.
One of the fastest method is to randomly choose `n_clusters` data points. This method is also known as Forgy Initialization.
I have implemented the algorithm with this initialization.
This initialization method makes sense because the clusters detected through KMeans are more probable to be near the modes present in data.
By randomly choosing points from data, we are making it more probable to get a point that lies close to the modes.

2 Kernel design and Kernelized clustering

2.1 CS 337: Proving Kernel Validity

We are going to use the following property of kernels from Lecture slides (Lecture 11):

$$K(x, y) = \sum_{d=1}^{r} \alpha_d (x^T y)^d \text{ where } \alpha_d \geq 0 \text{ is a kernel } (r \text{ can be } \infty)$$

Another property of kernels which we are going to use is:

$$K(x, y) \text{ is a kernel } \implies K'(x, y) = f(x)f(y)K(x, y) \text{ is also a kernel}$$
$$\text{Corresponding feature map, } \phi'(x) := f(x)\phi(x)$$

An important property of exponential function which we are going to exploit is:

$$\exp(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

Using the above properties, we can conclude that $\exp(\alpha x^T y)$ where $\alpha \geq 0$ is a kernel.
Now, if take $\alpha = \dfrac{1}{\sigma^2}$ and $f(x) = \exp\left(-\dfrac{x^T x}{2\sigma^2}\right)$, we get:
$K_\alpha(x, y) = \exp\left(-\dfrac{||x-y||^2}{2\sigma^2}\right) = \exp\left(-\dfrac{x^T x}{2\sigma^2}\right) \exp\left(-\dfrac{y^T y}{2\sigma^2}\right) \exp\left(\dfrac{x^T y}{\sigma^2}\right)$ is a kernel.
Hence, proved. □

2.2 CS 337: Simple Kernel Design

(i)

We proved in Lecture slides (Lecture 13, Part 2) that the vanilla KMeans algorithm would converge in finite number of iterations (say, N).
Without loss of generality, let $r_1 \geq r_2$.
For the sake of contradiction, let's assume that after N iterations we get μ_1, μ_2 which can correctly classify the clusters. Obviously, $\mu_1 \neq \mu_2$.
So, blue points have $Pr_1 = 1$ and $Pr_2 = 0$ while red points have $Pr_1 = 0$ and $Pr_2 = 1$.
As the algorithm has converged, the Prs for all points and μs won't change any further.

The decision boundary would be $f(x) = \left(x - \dfrac{\mu_1 + \mu_2}{2}\right) \cdot (\mu_2 - \mu_1) = (x \cdot (\mu_2 - \mu_1)) - \dfrac{||\mu_1 - \mu_2||^2}{2}$.
This decision boundary is linear in x, therefore, due to radial symmetry some of blue and red points would be on same side.
However, we know that $Pr_1 = 0$ for red points and $Pr_1 = 1$ for blue points. So in $(N+1)^{th}$ iteration, the values of Prs would change for some points.

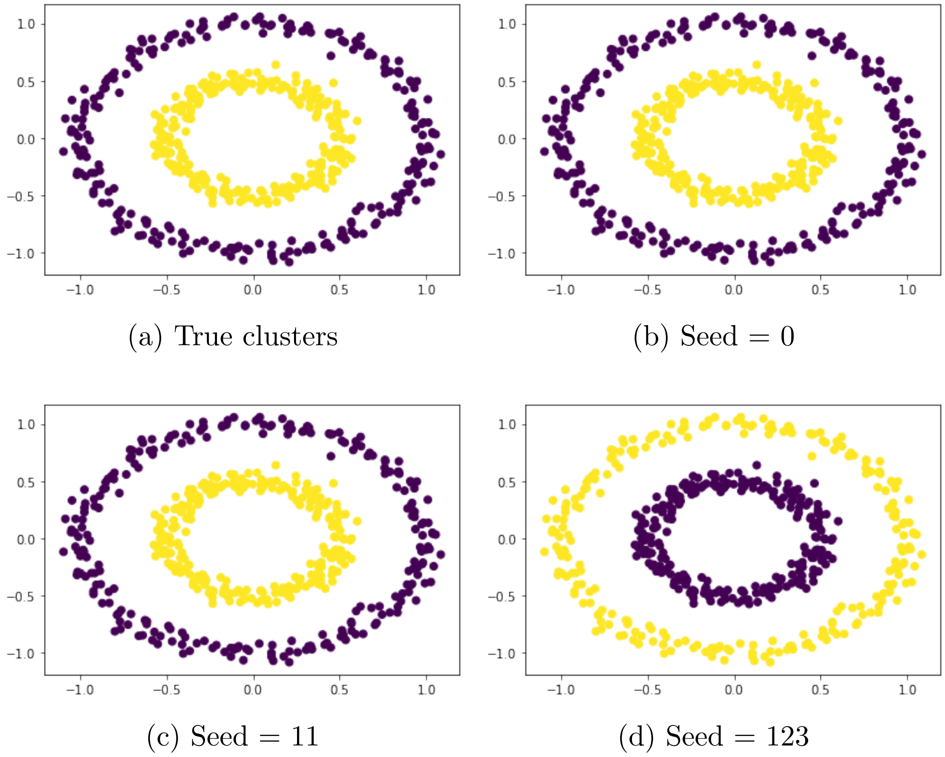
This leads to contradiction.
Thus, even after convergence, the vanilla KMeans algorithms won't be able to correctly classify the clusters for any value of r_1 and r_2.

(ii)

We see that the two clusters differ in distance from origin.
We propose $\phi(x) = ||x||_2$. Thus, $K(x, x') = ||x||_2||x'||_2$.
$\int_x \int_{x'} K(x,x')g(x)g(x')dxdx' = \bigg(\int_x ||x||_2 g(x)dx\bigg)\bigg(\int_{x'} ||x'||_2 g(x')dx'\bigg) = \bigg(\int_x ||x||_2 g(x)dx\bigg)^2 \geq 0$
Therefore, chosen kernel function is a valid kernel.

Code for the class `Kmeans_Kernel` updated in notebook.

For Dataset 3, we observe that the algorithm works well as the dataset is linearly separable in the above defined ϕ-space.
For Dataset 3 shifted by $(5, 5)$, we observe that the algorithm works well after zero-centering the data.



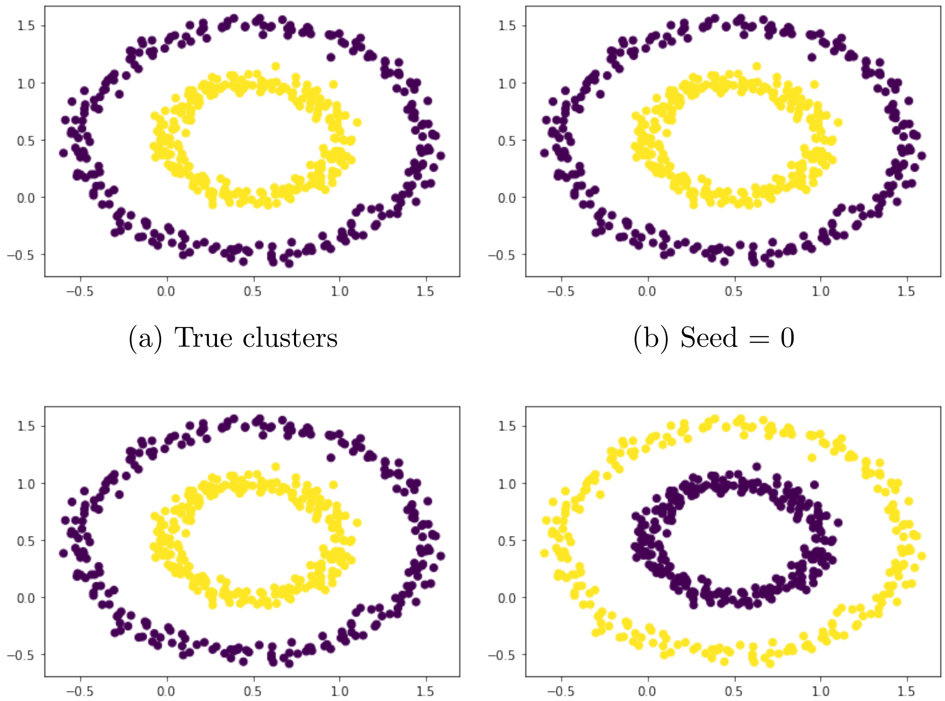
(a) True clusters

(b) Seed = 0

(c) Seed = 11

(d) Seed = 123

Figure 4: Dataset 3



(a) True clusters

(b) Seed = 0

(c) Seed = 11

(d) Seed = 123

Figure 5: Dataset 3 shifted by $(5, 5)$