MACHINE LEARNING

Kernel for Clustering
kernel K-Means

Outline of Video

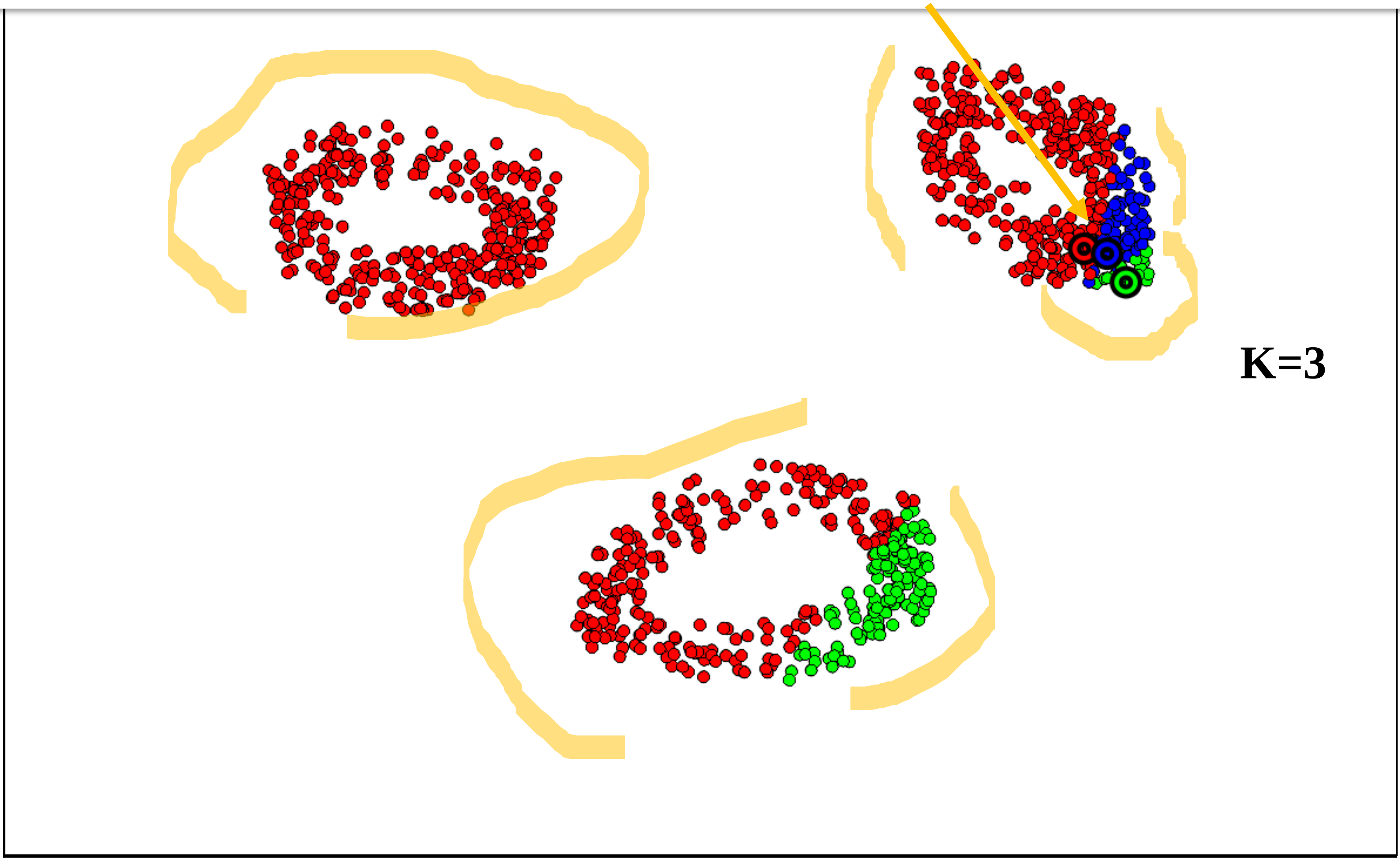
1. Review principle and steps of K-Means algorithm

2. Derive kernel version of K-means

K-means Clustering: Algorithm

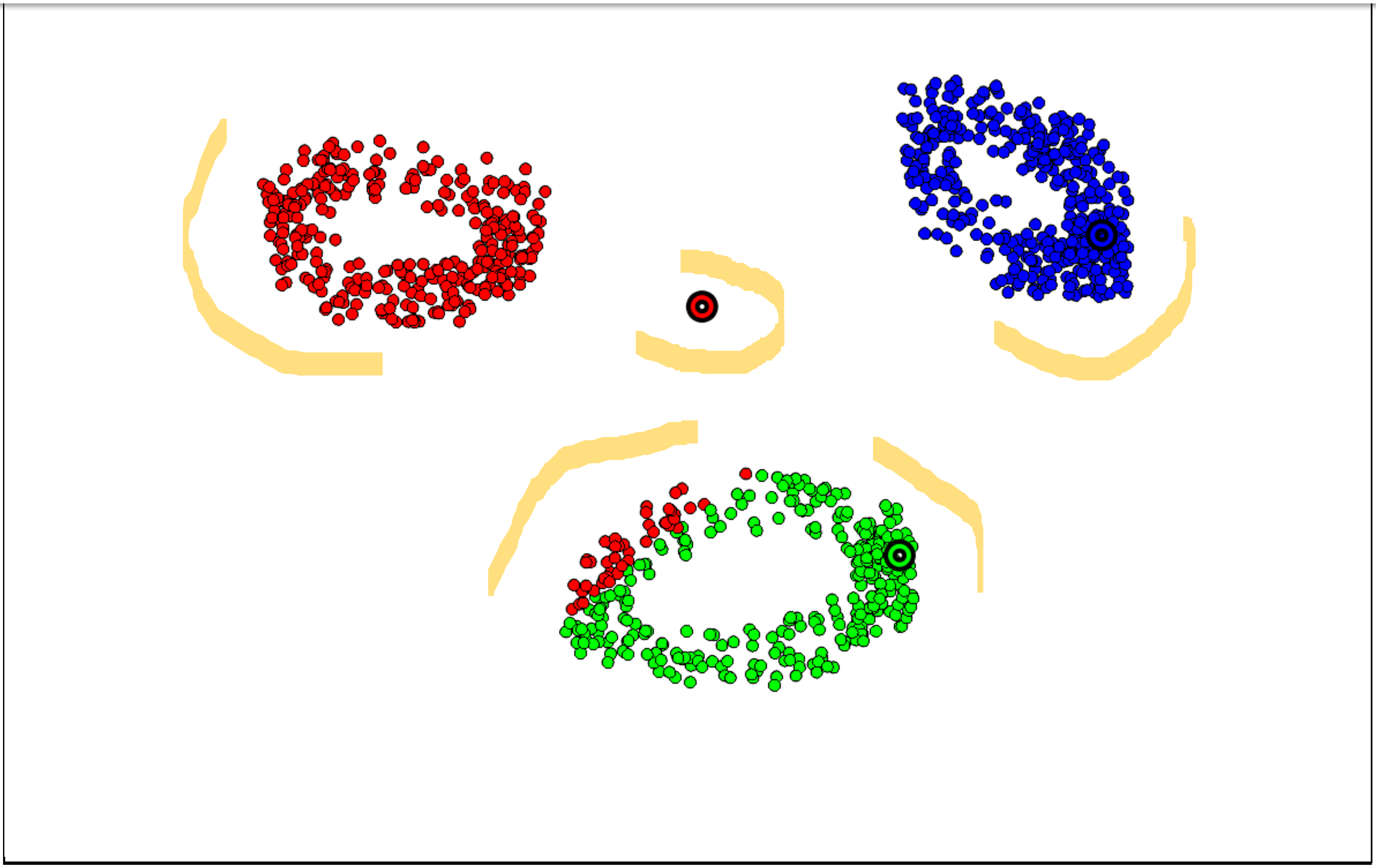
Step 1: Make a guess on what is the correct number of clusters K

Step 2: Initialize randomly the center of the clusters & assign points to closest clusters.

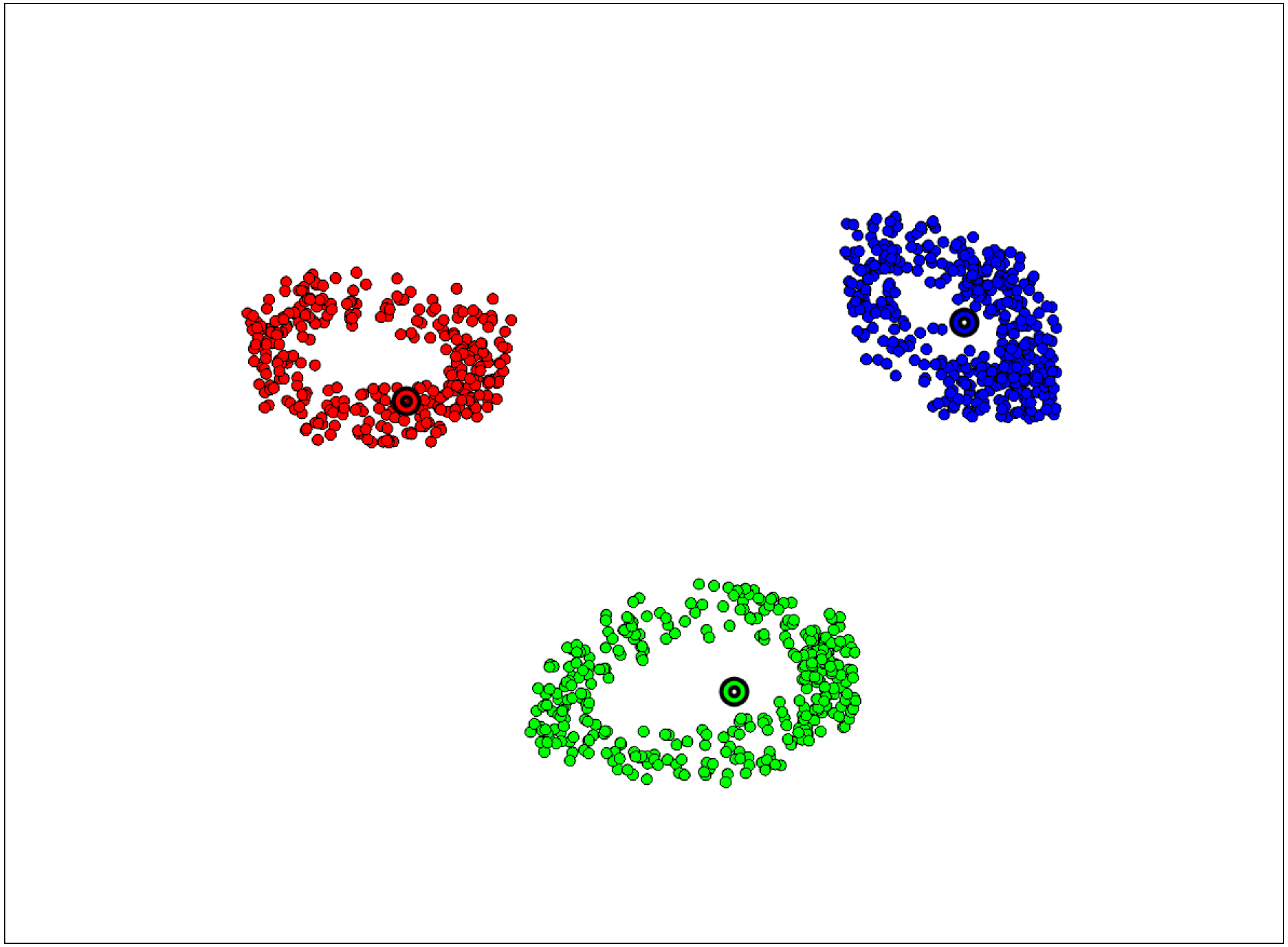


K-means Clustering: Algorithm

Step 3: recompute the location of the centers of the clusters by taking the centroid of the datapoints assigned to each center.



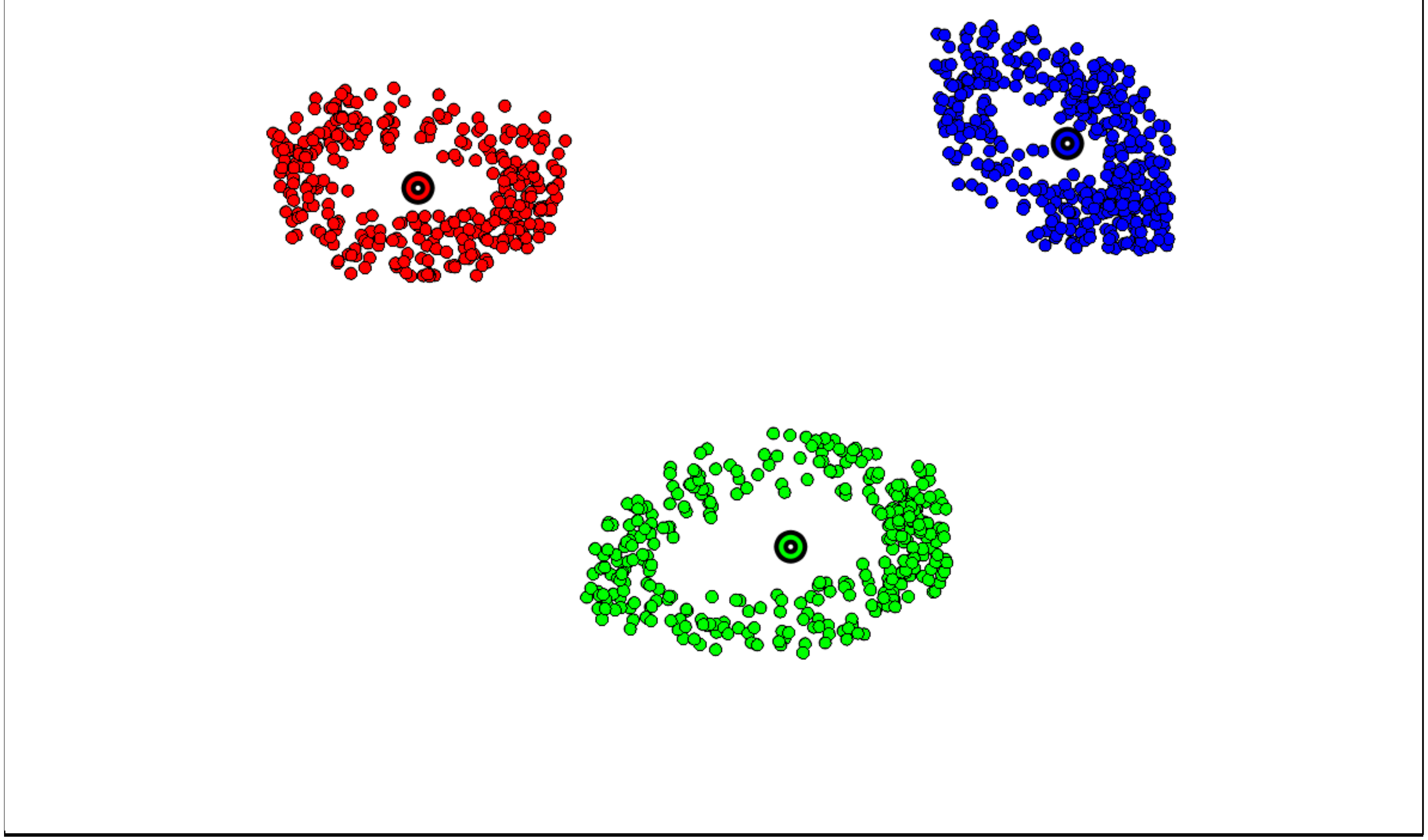
K-means Clustering: Algorithm



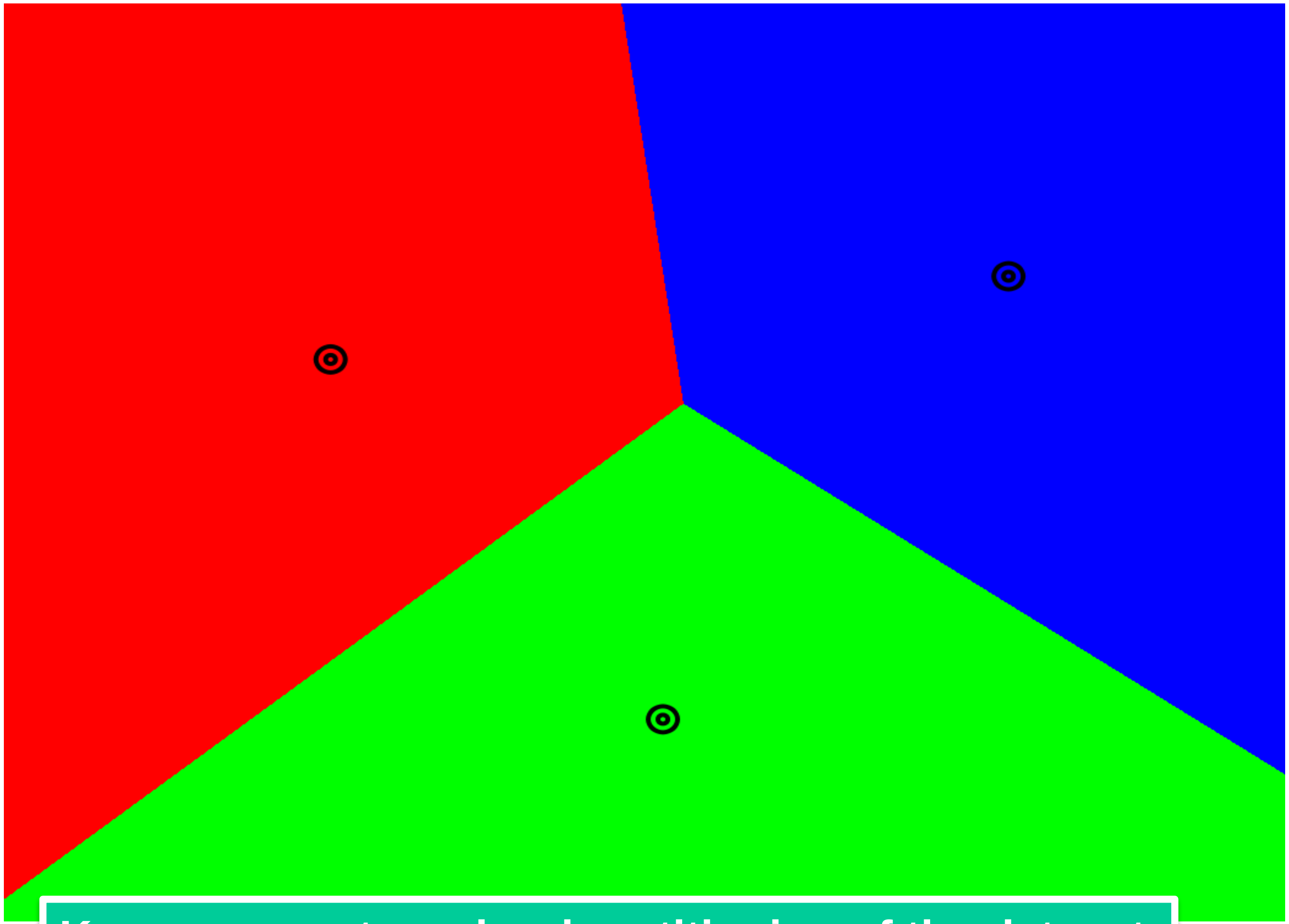
K-means Clustering: Algorithm

Iterate until convergence

K-means is guaranteed to converge in a finite number of steps.



K-means Clustering: Algorithm



K-means creates a hard partitioning of the dataset

K-means Clustering: Objective function

K-Means clustering minimizes a quadratic cost function

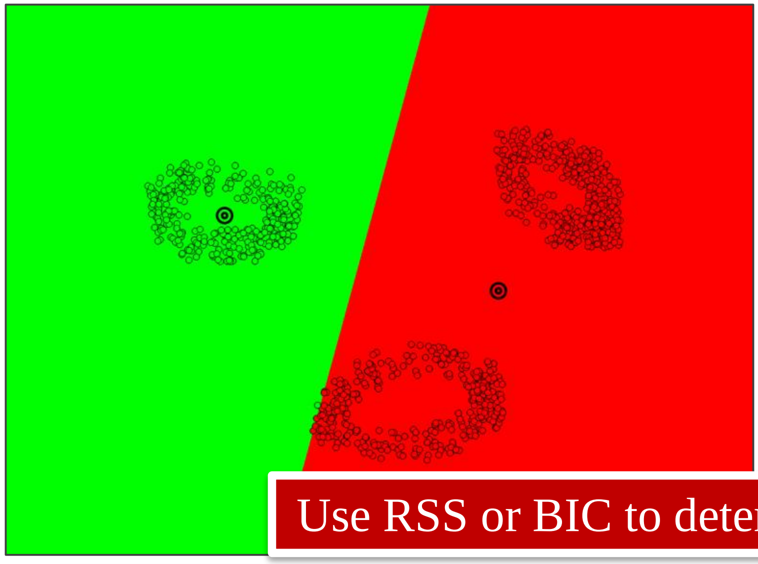
$$J\left(\mu^1,\dots,\mu^K\right)=\sum_{k=1}^{K}\sum_{x^j\in C^k}\sqrt{\left(x-\mu^k\right)^T\left(x-\mu^k\right)}$$

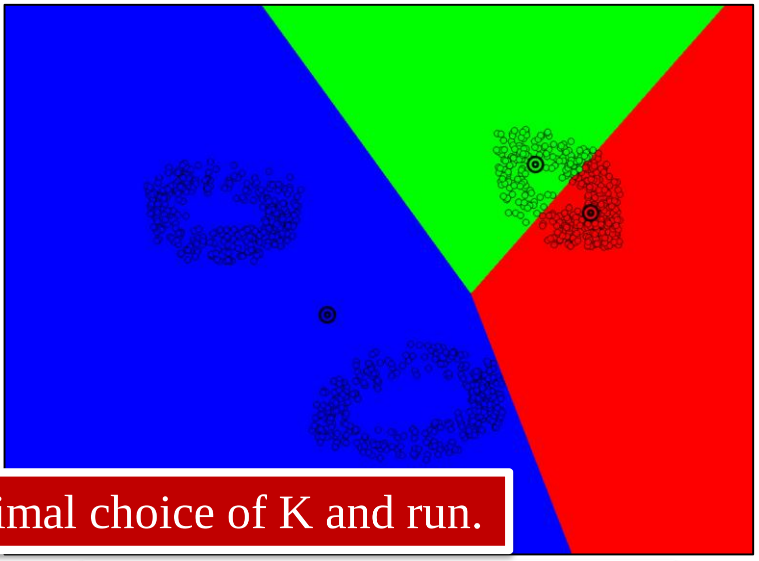
$\mu^1,\dots,\mu^K$: centers of the K clusters

K-means Clustering: Advantages

- The algorithm is guaranteed to converge in a finite number of iterations (but it converges to a local optimum!)
- It is computationally cheap and faster than other clustering techniques - update step is ~O(N).

K-means Clustering: Disadvantages





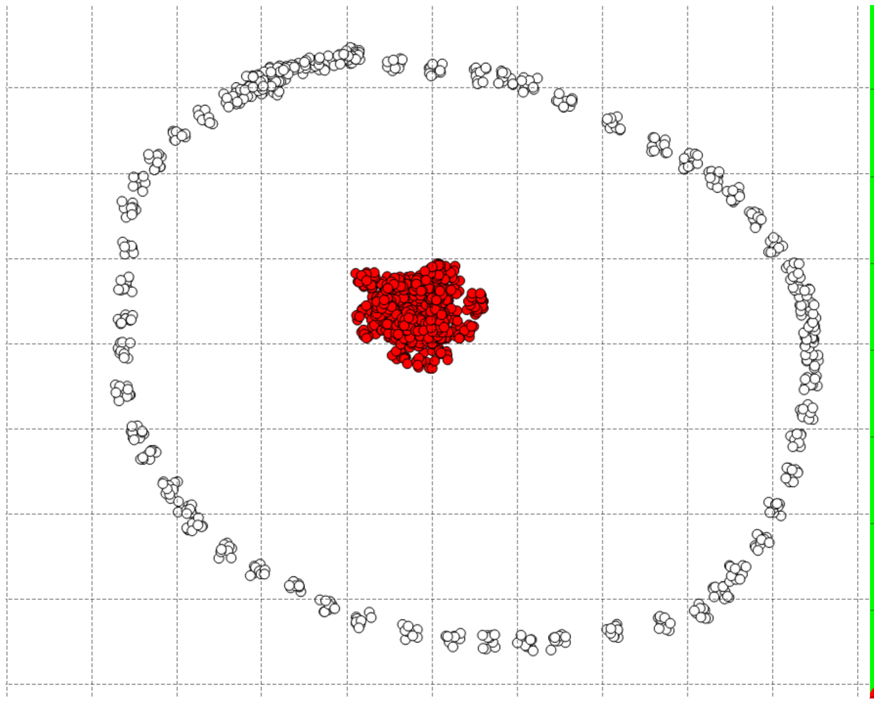
Use RSS or BIC to determine optimal choice of K and run.

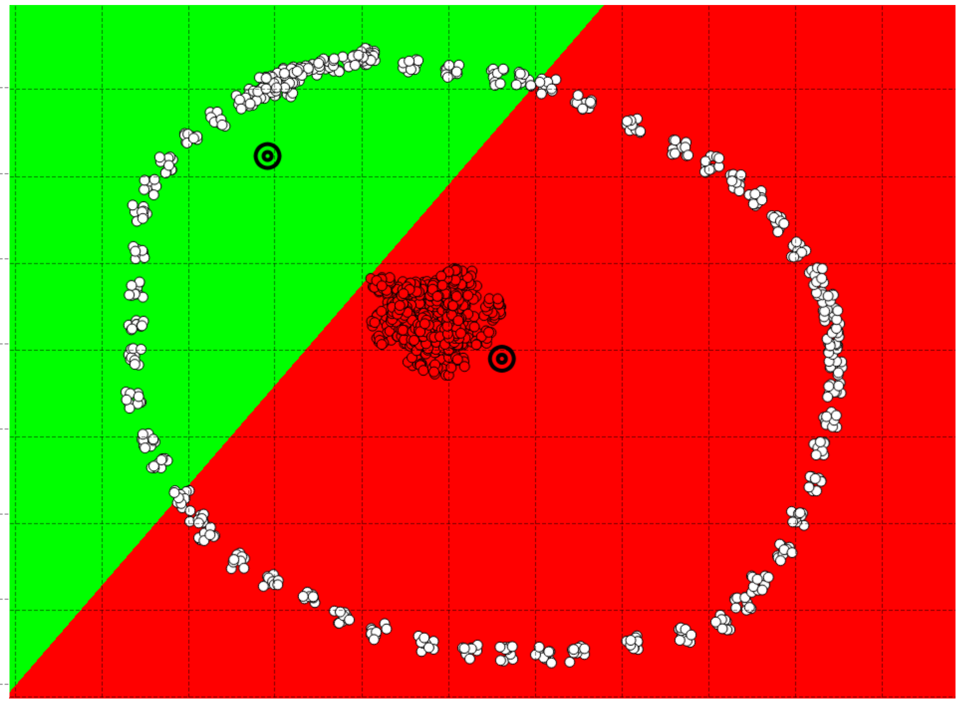
Good performance depends on properly choosing K.

Good performance depends on initialization.

K-means Clustering: Limitations

K-means can separate clusters linearly, or quasi-linearly (with norm-p) only.





Kernel K-means can generate non-linear separations of groups of points.

Kernel K-means: Principle

Idea:

- ❑ Exploit the principle of the kernel to perform classical K-means clustering with norm-2 in feature space:
  → This yields non-linear boundaries.
  → This retains simplicity of computation of linear K-means.

- ❑ The objective function of K-means is composed of an inner product across datapoints.
  → One can replace the inner product with a kernel to perform inner product in feature space.

Kernel K-means: Derivation

K – Means algorithm minimizes the objective function :

$$J\left(\mu^1,....,\mu^K\right)=\sum_{k=1}^{K}\sum_{x^j\in C^k}\left\|x^j-\mu^k\right\|^2 \qquad \text{with } \mu^k=\frac{\sum_{x^j\in C^k} x^j}{m_k}$$

m_k : number of datapoints in cluster C^k

Project the points into feature space

$$\left\{x^i\right\}_{i=1}^{M} \longrightarrow \left\{\phi\left(x^i\right)\right\}_{i=1}^{M}$$

Express the same problem in feature space:

$$J\left(\mu^1,....,\mu^K\right)=\sum_{k=1}^{K}\sum_{x^j\in C^k}\left\|\phi\left(x^j\right)-\phi\left(\mu^k\right)\right\|^2$$

$$\frac{\sum_{x^j\in C^k}\phi\left(x^j\right)}{m_k}$$

We cannot observe the image of μ^k in feature space.

$\rightarrow$ Construct $\phi\left(\mu^k\right)$ using images of points in same cluster.

Kernel K-means: Solution

$$J\left(\mu^1,\ldots,\mu^K\right)=\sum_{k=1}^{K}\sum_{x^j\in C^k}\left\|\phi\left(x^j\right)-\frac{\sum_{x^l\in C^k}\phi\left(x^l\right)}{m_k}\right\|^2$$

$k\left(x^l,x^j\right)$ $k\left(x^j,x^l\right)$

$$=\sum_{k=1}^{K}\sum_{x^j\in C^k}\left(\phi\left(x^j\right)^T\phi\left(x^j\right)-\frac{2\sum_{x^l\in C^k}\phi\left(x^l\right)^T\phi\left(x^j\right)}{m_k}+\frac{\sum_{x^t,x^l\in C^k}\phi\left(x^t\right)^T\phi\left(x^l\right)}{\left(m_k\right)^2}\right)$$

$k\left(x^j,x^j\right)$

$$\boxed{=\sum_{k=1}^{K}\sum_{x^j\in C^k}\left(\mathrm{k}\left(x^j,x^j\right)-\frac{2\sum_{x^t\in C^k}\mathrm{k}\left(x^i,x^t\right)}{m_k}+\frac{\sum_{x^t,x^l\in C^k}\mathrm{k}\left(x^t,x^l\right)}{\left(m_k\right)^2}\right)}$$

Objective function in feature space

Kernel K-means: Algorithm

Kernel K-means algorithm is also an iterative procedure:

1. **Initialization:** pick K clusters (random assignment of points to a cluster, or use K-means at initialization)

2. **Assignment Step:** Assign each data point to its "closest" centroid (E-step).

$$\arg\min_k d\left(x, C^k\right) = \min_k \left(k\left(x,x\right) - \frac{2\sum_{x^j \in C^k} k\left(x, x^j\right)}{m_k} + \frac{\sum_{x^j, x^l \in C^k} k\left(x^j, x^l\right)}{\left(m_k\right)^2} \right)$$

3. **Update Step:** Update the list of points belonging to each centroid (M-step)

4. Go back to step 2 and repeat the process until the clusters are stable.

Kernel K-means: interpreting the solution

Consider a RBF kernel. Let us interpret the different terms.

Cst of value 1

$$\arg\min_k d\left(x, C^k\right) = \min_k \left(\boxed{k(x,x)} - \frac{\underbrace{2\sum_{x^j \in C^k} k\left(x, x^j\right)}}{\boxed{m_k}} + \frac{\underbrace{\sum_{x^j, x^l \in C^k} k\left(x^j, x^l\right)}}{\boxed{\left(m_k\right)^2}} \right)$$

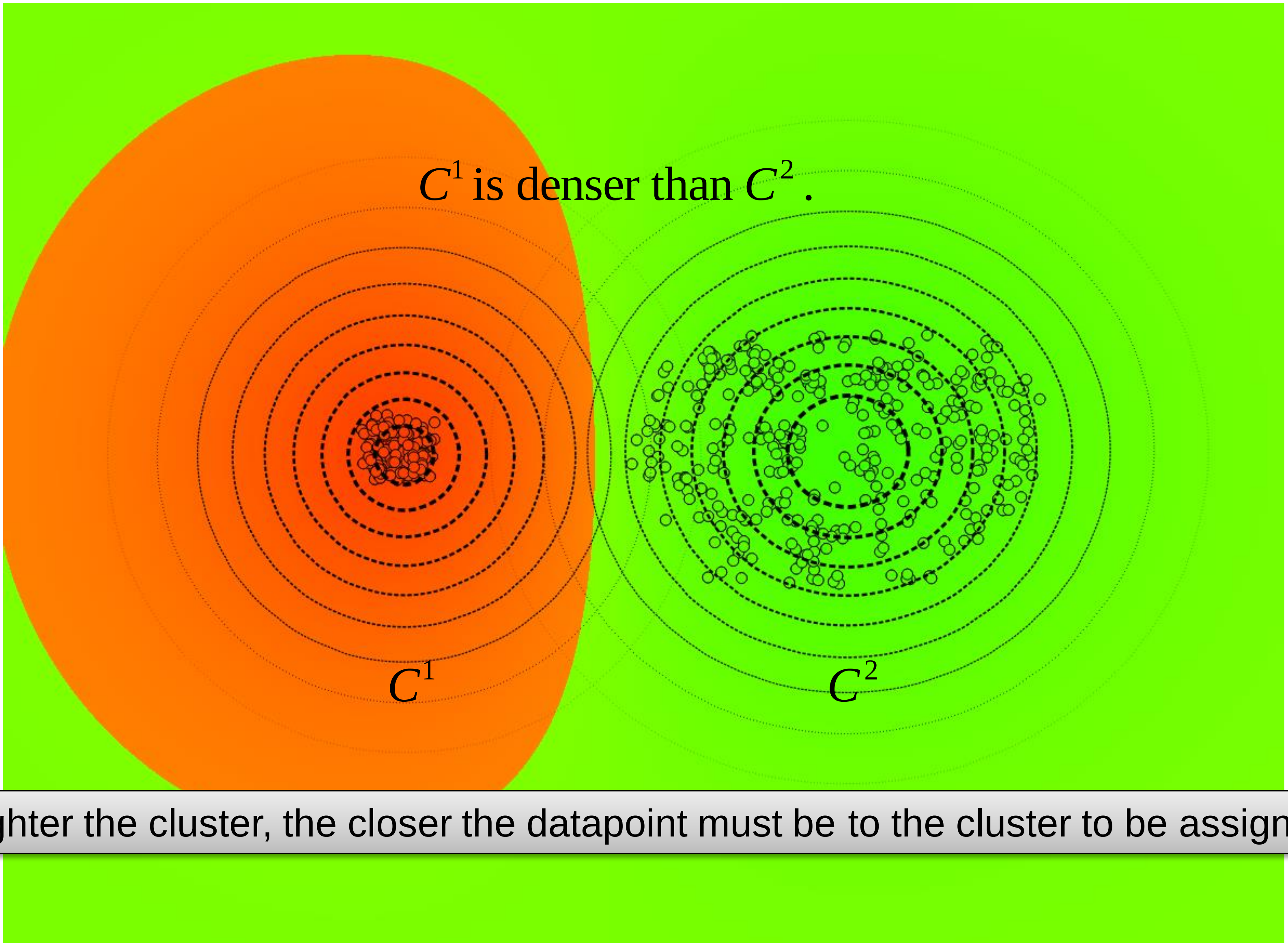
If x is close to all points in cluster k, this sum is close to 2.

Normalization factor

If the points are tightly grouped in the cluster, this sum is close to 1.

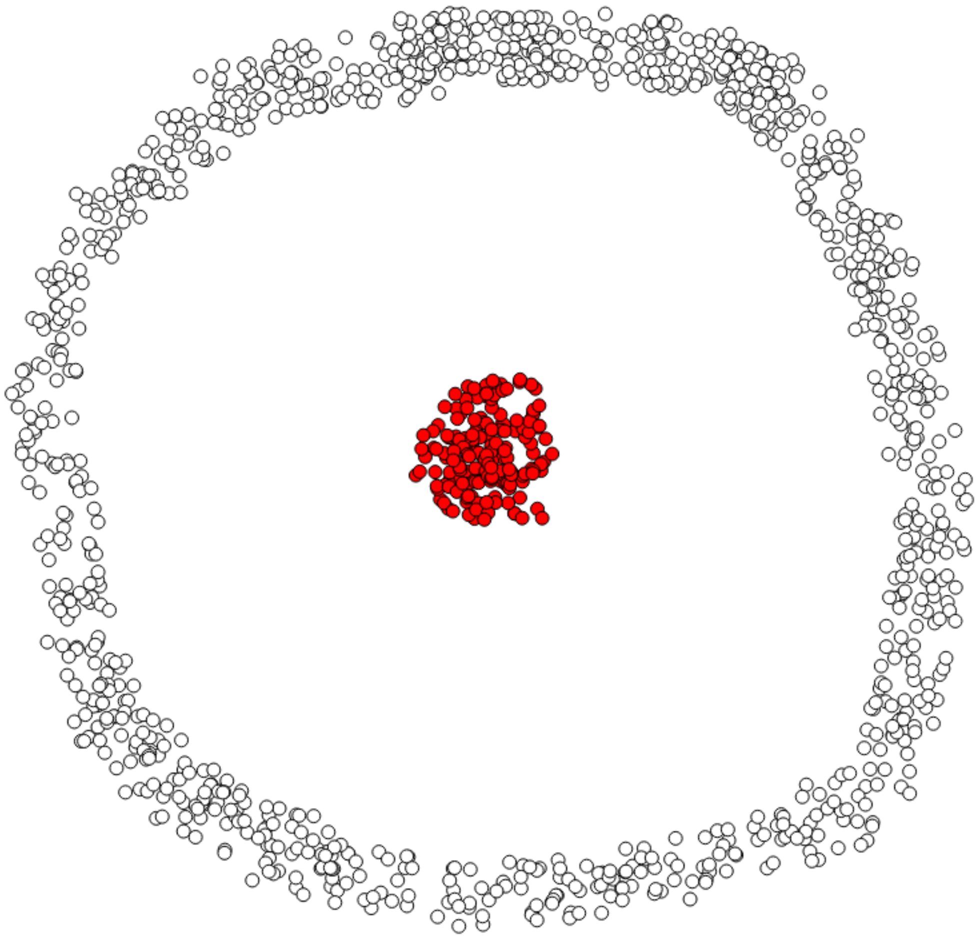
The tighter the cluster, the closer the datapoint must be to the cluster to be assigned.

Kernel K-means: interpreting the solution



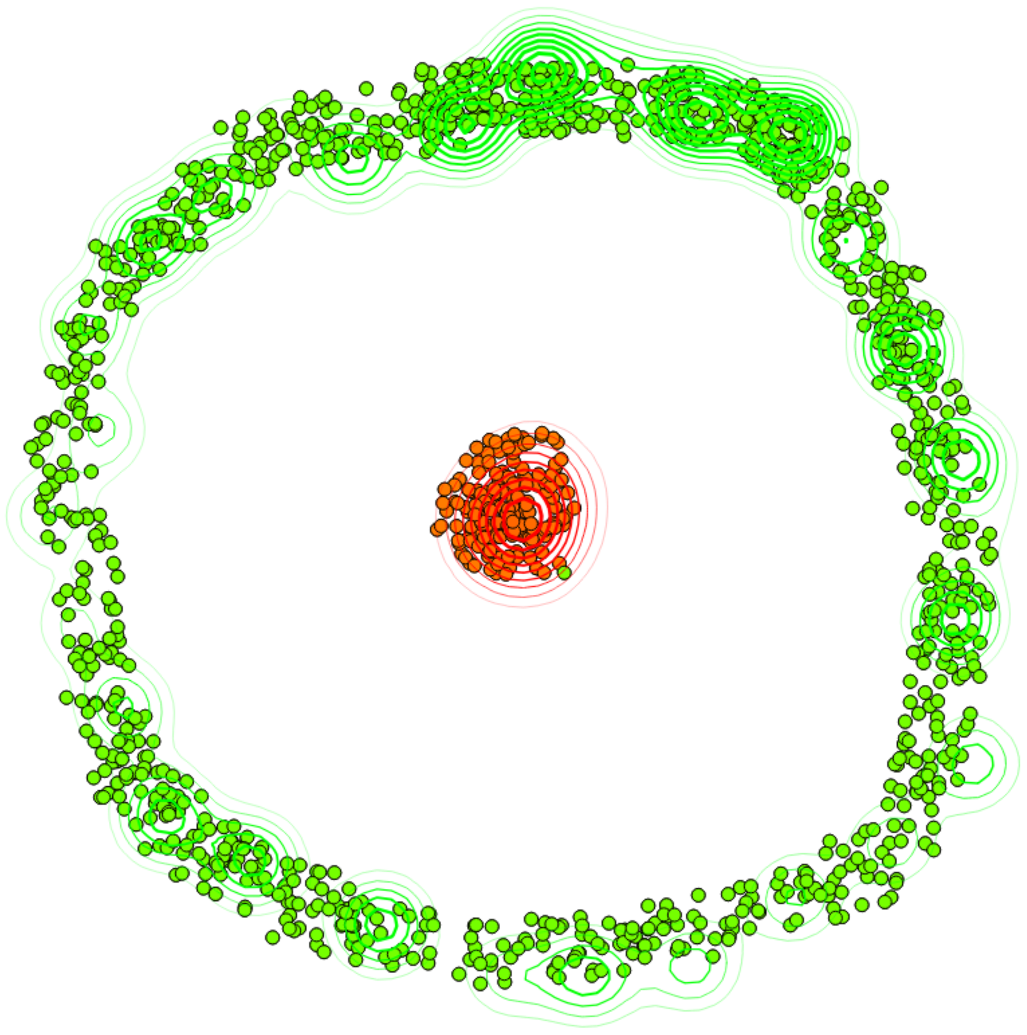
The tighter the cluster, the closer the datapoint must be to the cluster to be assigned.

Nonlinear boundaries with kernel K-means

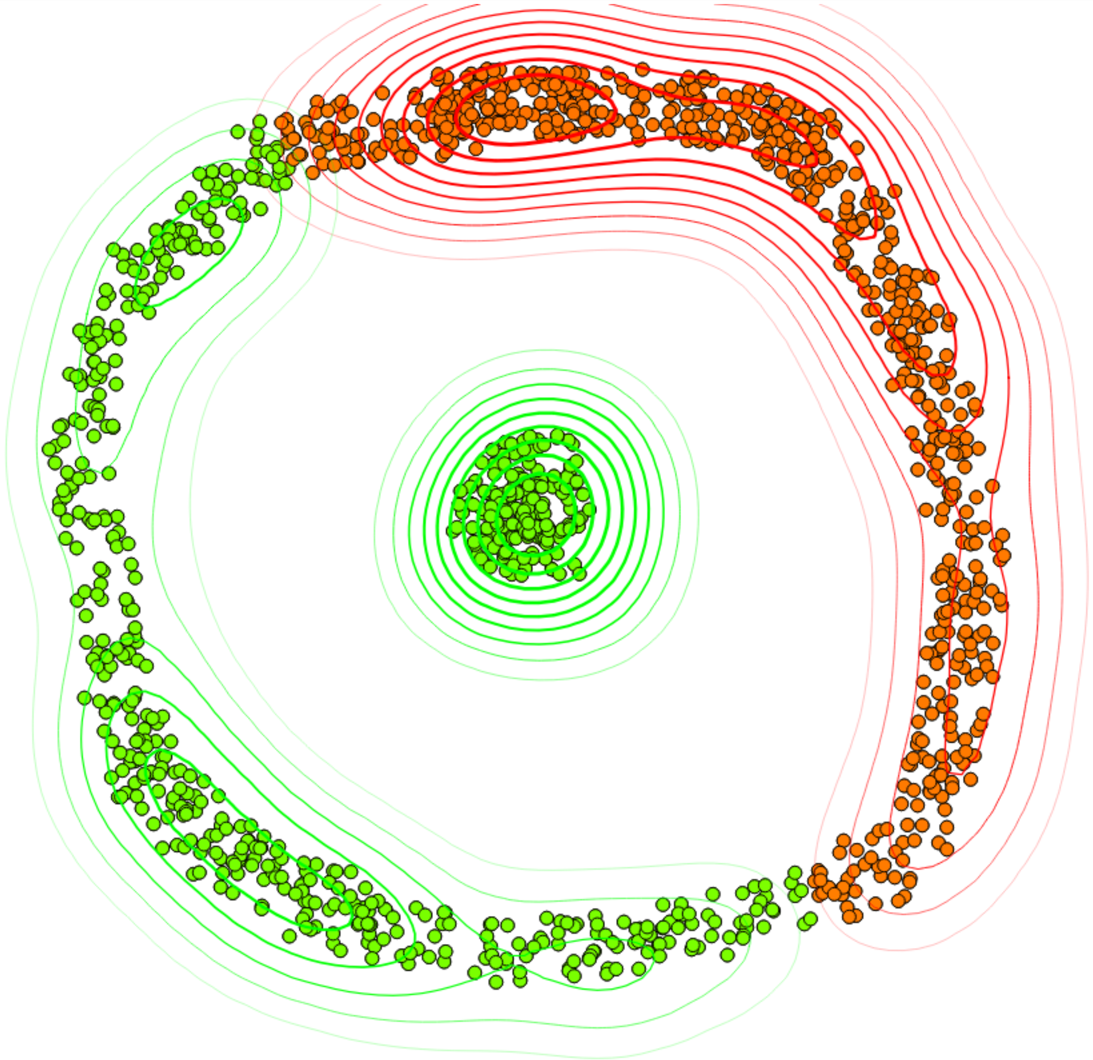


2 Clusters

Nonlinear boundaries with kernel K-means

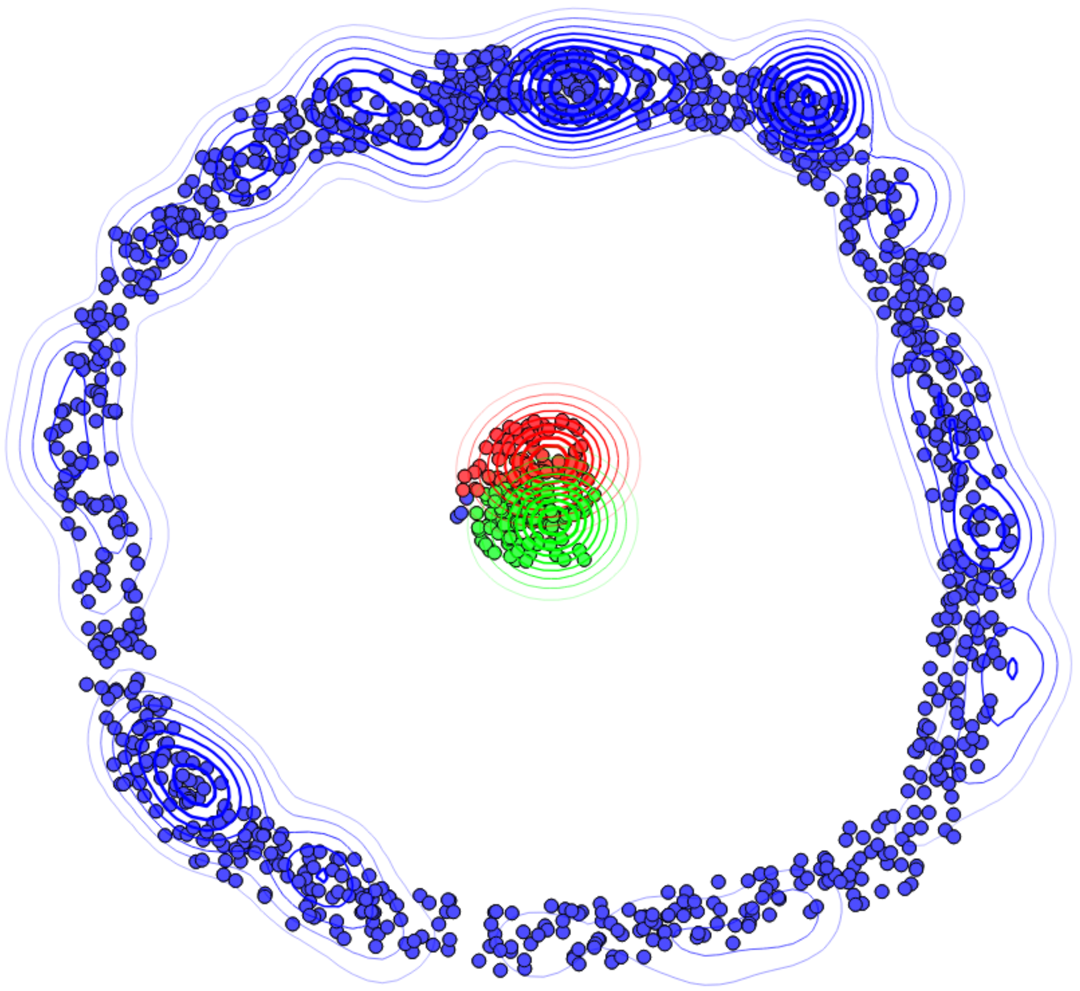


Sensitivity to kernel width



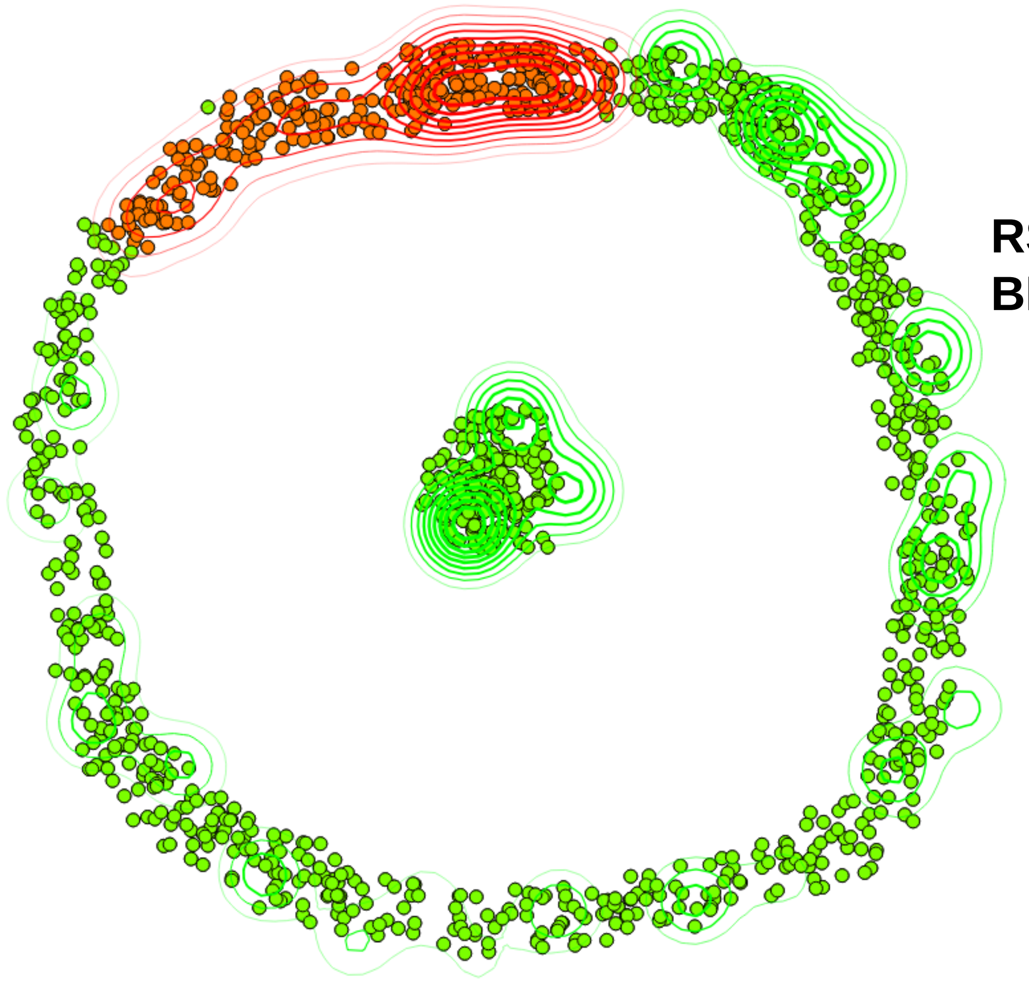
Kernel width too large

Sensitivity to choice of K



Still sensitive to choice of K. Here K=3.

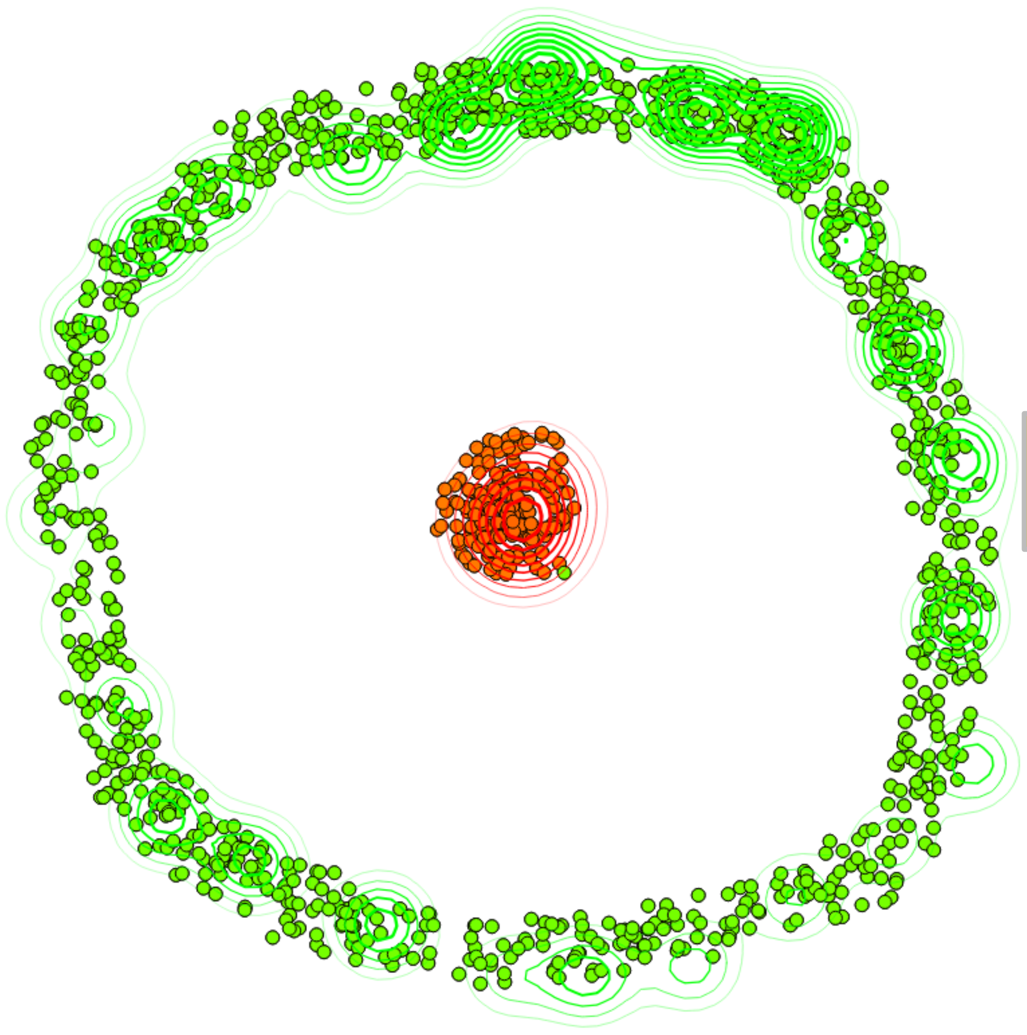
Sensitivity to initialization



RSS measure: 2773
BIC: -5500

But still sensitive to initialization

Finding the right kernel width



RSS measure: 2625
BIC: -5522

Better values on both RSS and BIC

With a correct Kernel width

Kernel K-means: interpreting the solution

With a polynomial kernel

B: Affected by the relative angle across the points.

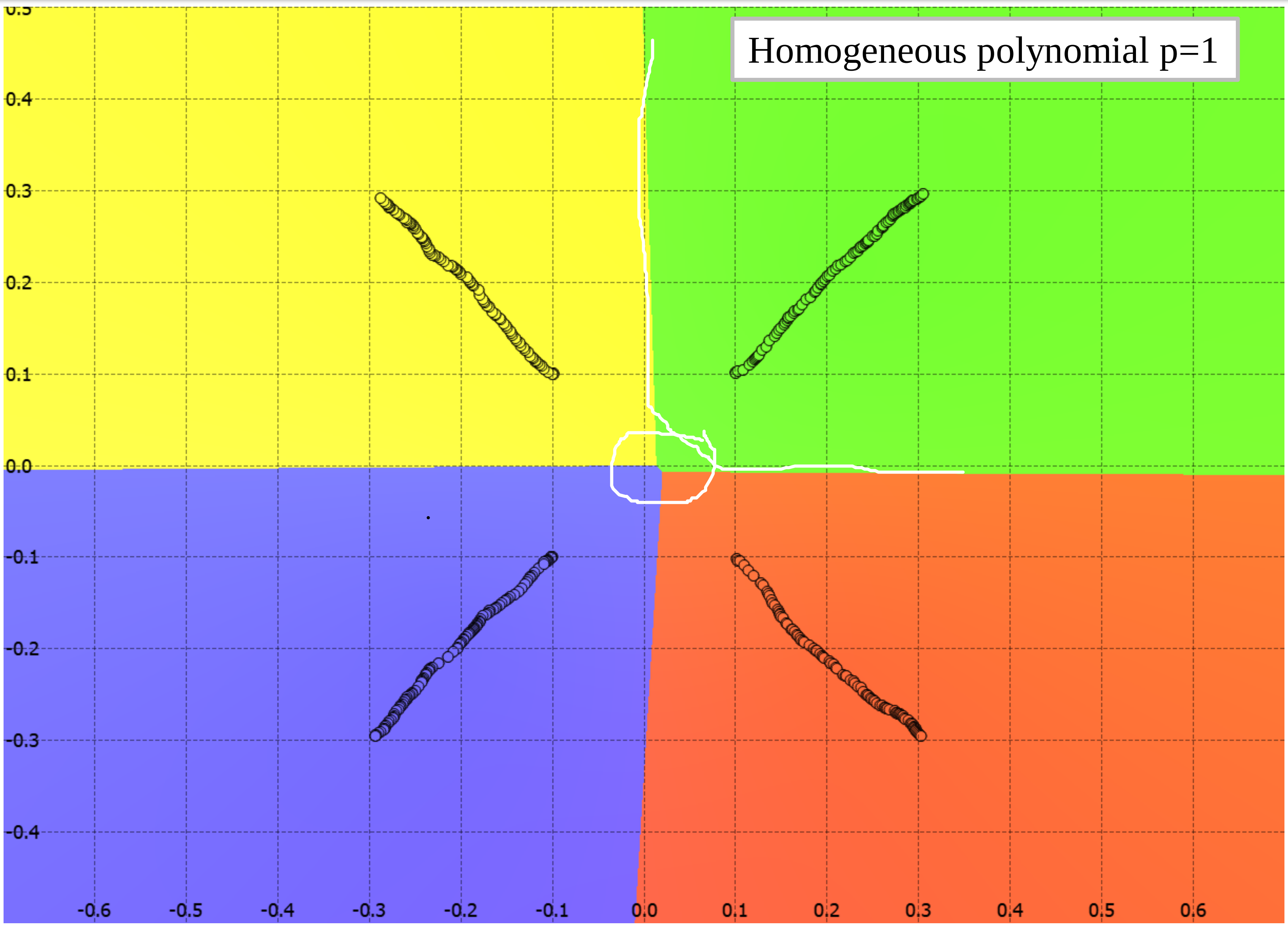
A: Affected by the position of the points from the origin (norm).

Norm - Positive value

$$\arg\min_k d\left(x, C^k\right) = \min_k \left(k(x,x) - \frac{2\sum_{x^j \in C^k} k\left(x, x^j\right)}{m_k} + \frac{\sum_{x^j, x^l \in C^k} k\left(x^j, x^l\right)}{\left(m_k\right)^2} \right)$$

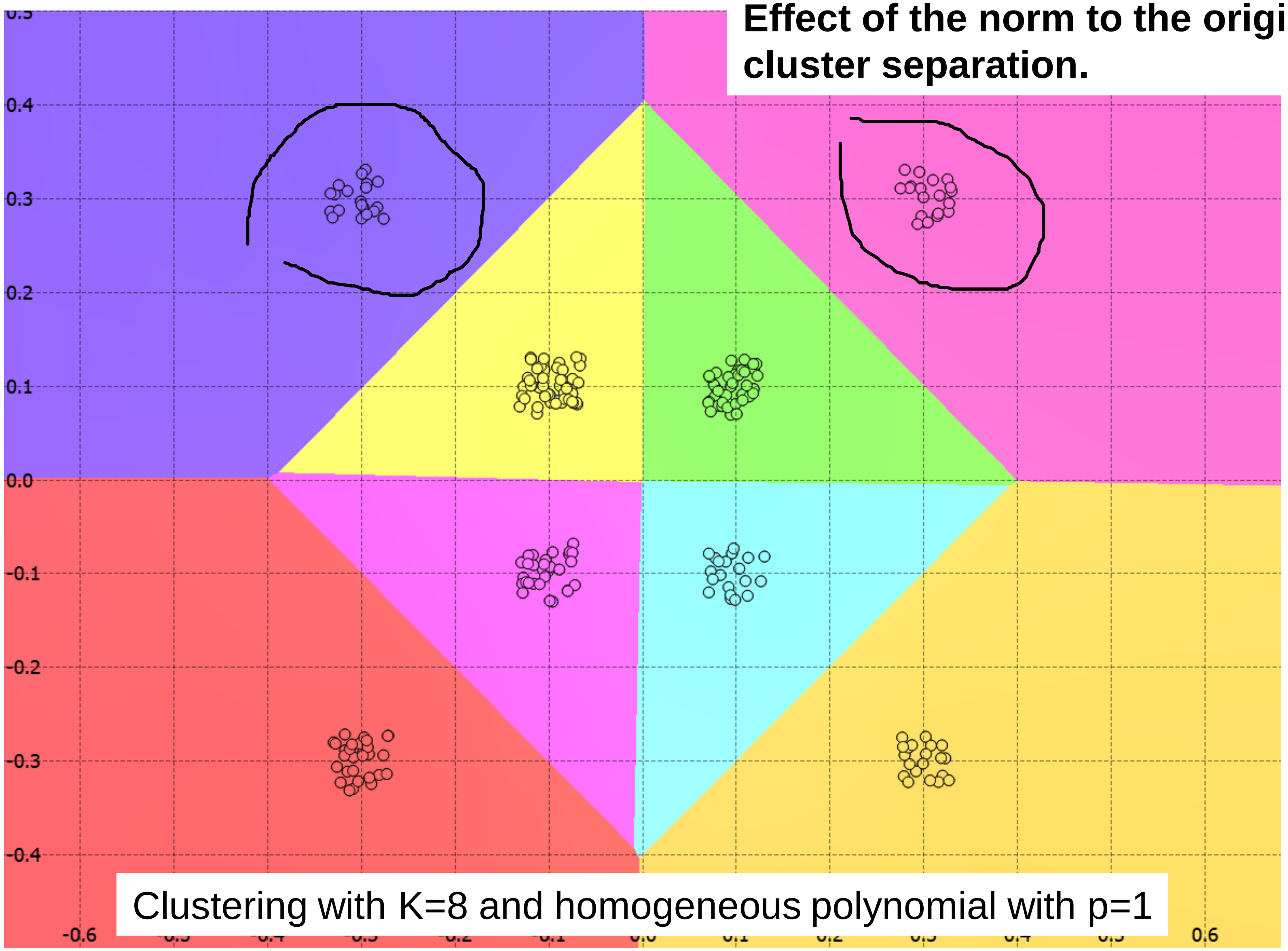
A datapoint will be assigned to the closest cluster in the closest partition.

Quadran partitioning



Quadran partitioning

Effect of the norm to the origin on cluster separation.



Clustering with K=8 and homogeneous polynomial with p=1

Summary

- Kernel K-means follows the same principle as K-means. It is an iterative procedure, akin to Expectation-Maximization.
- It allows to build highly non-linear boundaries when using the RBF kernel.
- When using the polynomial kernel, it allows to separate groups of datapoints exploiting the geometrical distribution of the points.
- As K-means, it depends on initialization of the centers of the clusters which is random.
- As K-means, the solution depends on choosing well the number of clusters *K.* To choose *K*, one can use the AIC, BIC or RSS criteria.
- It is computationally more expensive: K-means is O(N). Kernel K-means requires to store a Gram matrix which is $O(M^2)$.