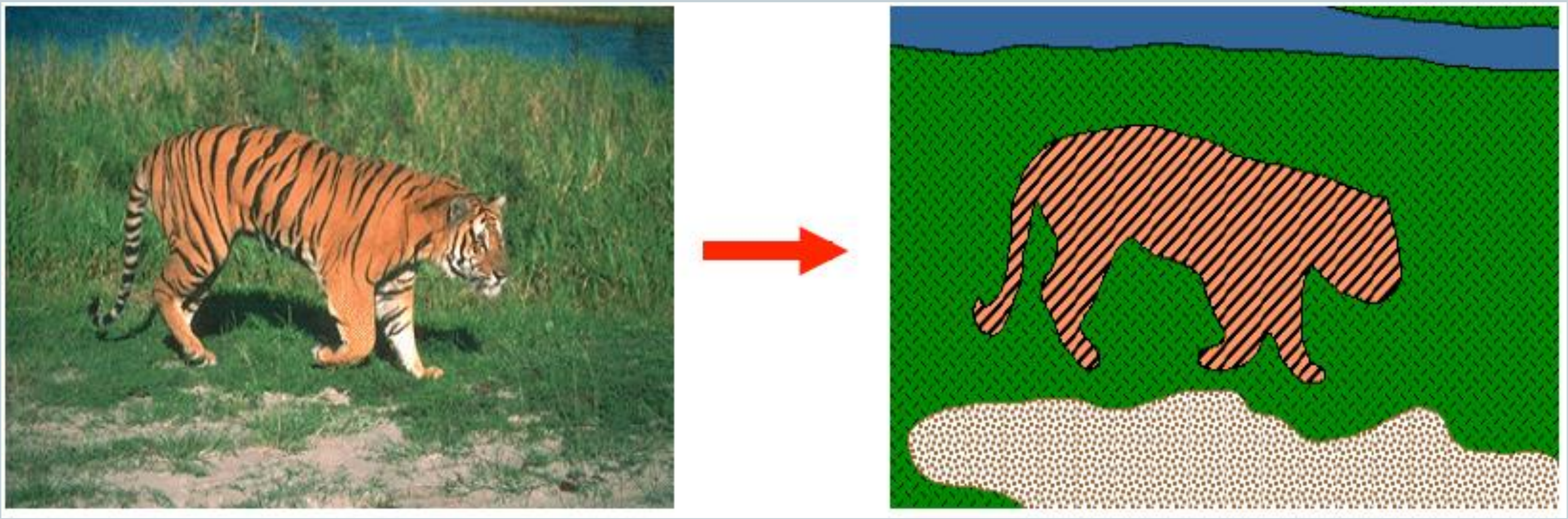
ELEC 474 Machine Vision

K-MEANS CLUSTERING

Edith Deretey, Zia Ud Din, Michael Greenspan

Review: Segmentation

- Identifying groups of pixels that go together



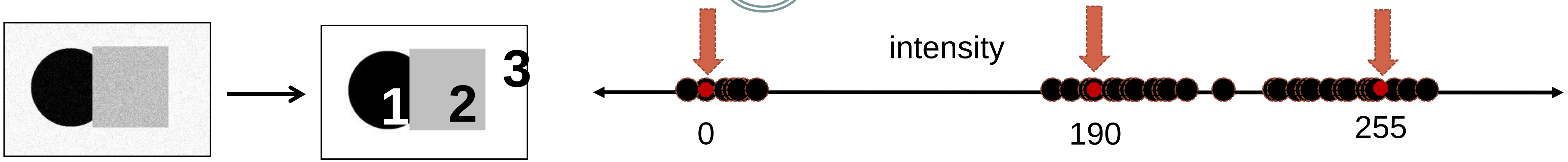
Clustering

- Clustering: grouping together similar connected points and representing them with a single token
- Criteria for forming a cluster depends on the application
- Some examples:
  - Intensity
  - Colour
  - Distance
  - Texture

K-Means Clustering

- K-means clustering is one of the simplest unsupervised learning algorithms.
- Supervised Learning
  - Knowledge of output
    - Learning in the presence of an expert
    - Training data is labelled
  - Goal: Predict labels of unseen data
  - Examples: Support Vector Machines (SVM), Decision Trees, Neural Networks,
- Unsupervised Learning
  - No knowledge of output
  - Goal: Discover patterns/groupings in data
  - Examples: Clustering, Genetic Algorithms, Autoencoders

K-Means Clustering

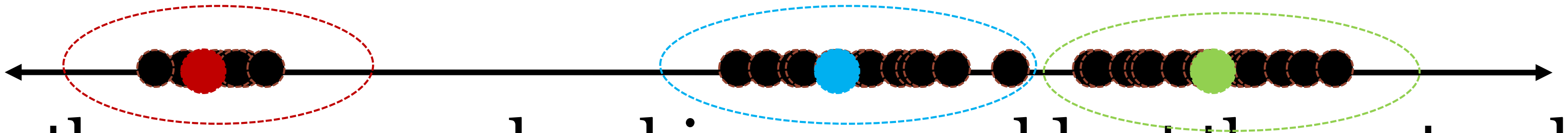


- Approach: choose three "centers" as the representative intensities
  - Label every pixel according to which of these centers it is nearest to
- Best cluster centers are those that minimize Sum of Square Distance (SSD) between all points x and their nearest cluster center c_i

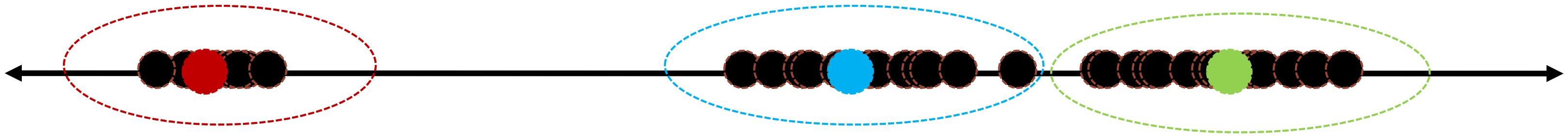
$$SSD = \sum_{clusters\ i} \sum_{x \in cluster\ i} (x - c_i)^2$$

K-Means Clustering

- "chicken and egg problem"
- If we knew the cluster centers, we could allocate points to groups by assigning each to its closest center



- If we knew the group memberships, we could get the centers by computing the mean per group

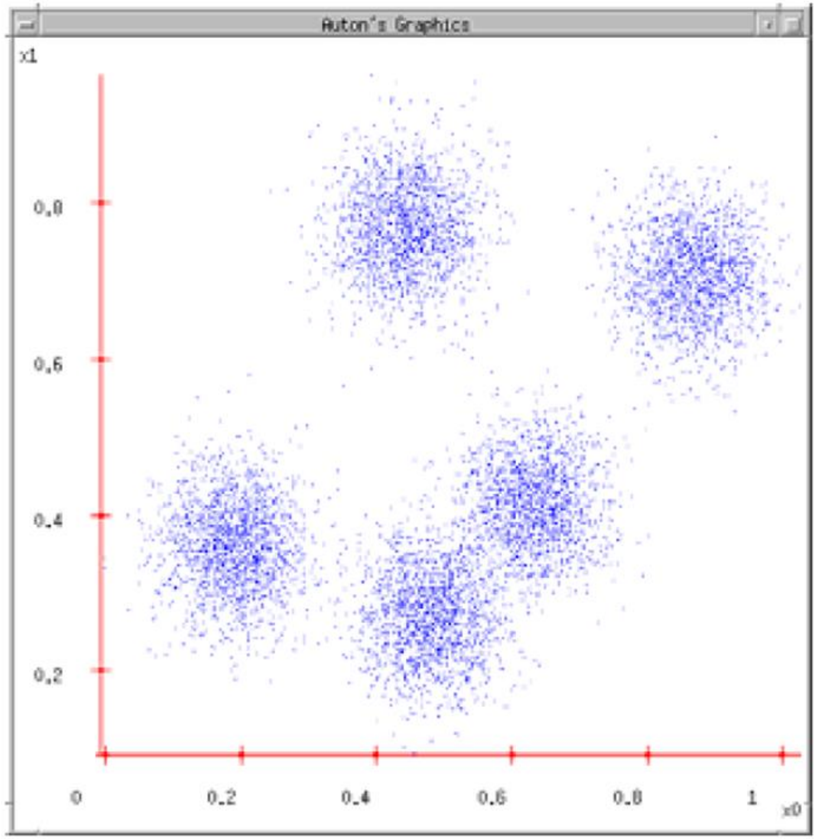


K-Means Clustering

Algorithm:

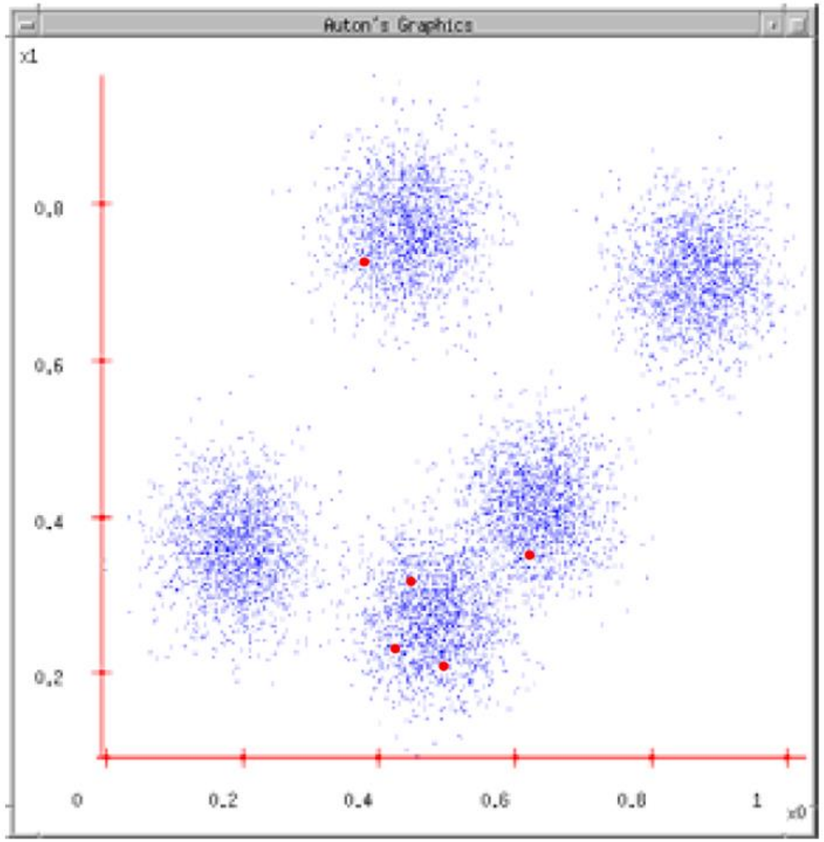
1. Randomly initialize k points to act as cluster centers (c_1, …, c_k)
2. Given cluster centers, determine points in each cluster
   - point x is in cluster i iff $|x-c_i| \leq |x-c_j|$ for all i, j in $\{1..k\}$
3. Given points in each cluster, solve for c_i
4. If c_i have changed, repeat from Step 2

Example: K-Means Clustering



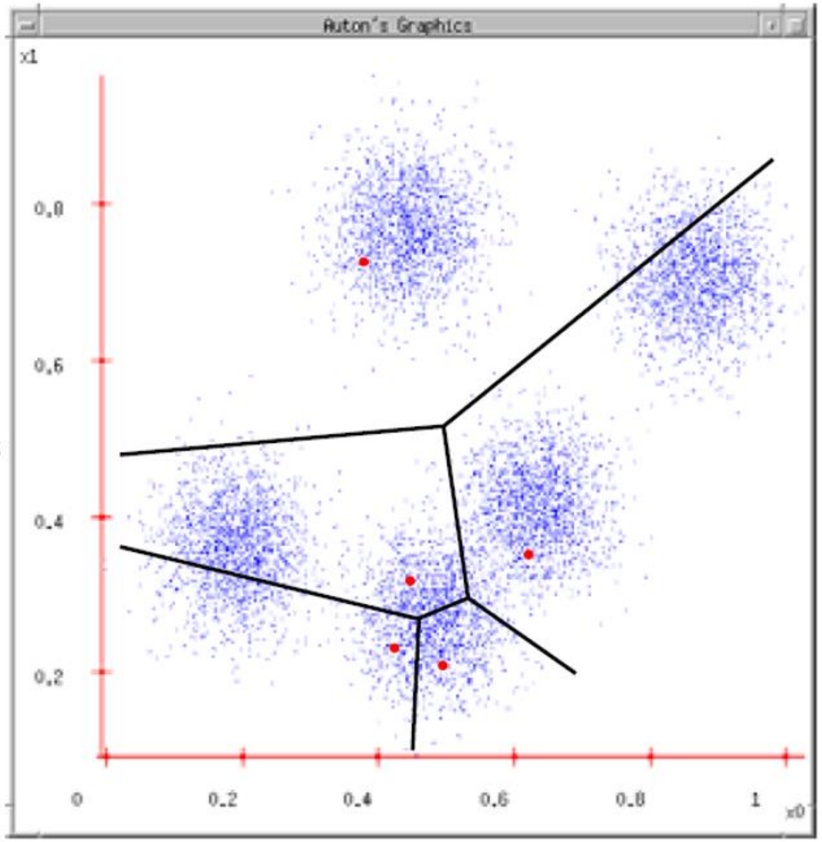
Example: K-Means Clustering

- Step 1: Randomly initialize *k* points to act as cluster centers



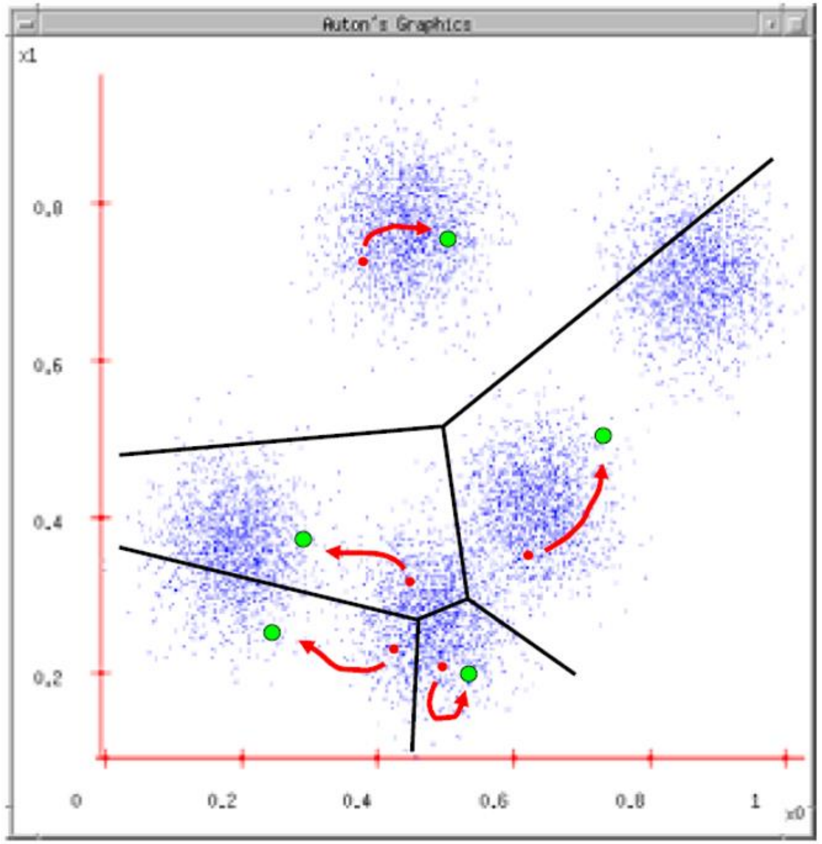
Example: K-Means Clustering

- Step 2: Given cluster centers, determine points in each cluster



Example: K-Means Clustering

- Step 3: Given points in each cluster, solve for c_i



Step 4: If c_i have changed, repeat from Step 2

K-Means Clustering

- Online demo:
  - https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

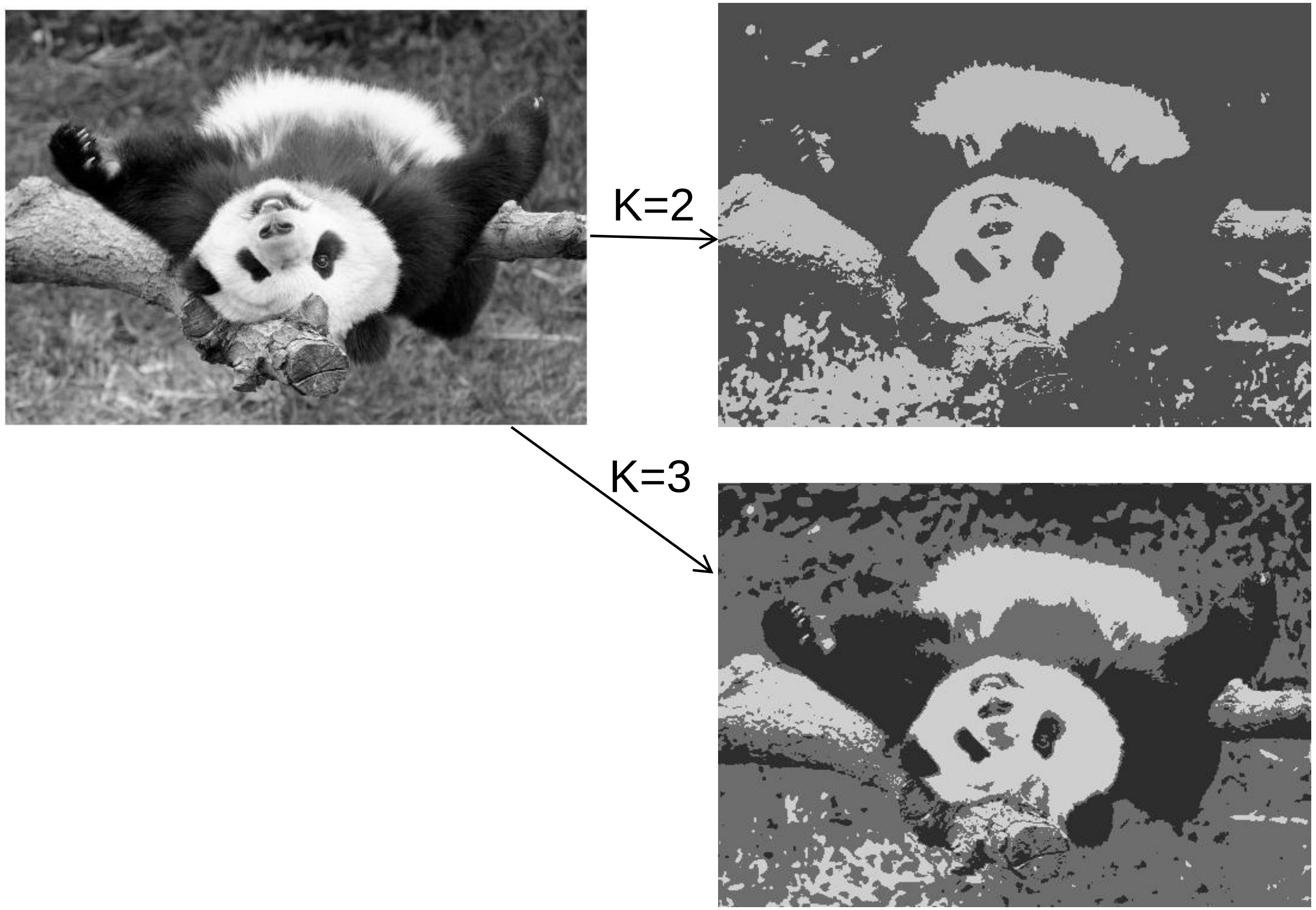
Edith Deretey, Zia Ud Din, Michael Greenspan

Segmentation as Clustering

- Depending on what we choose as the feature space, the pixels can be grouped in different ways
- Grouping pixels based on **intensity** similarity:

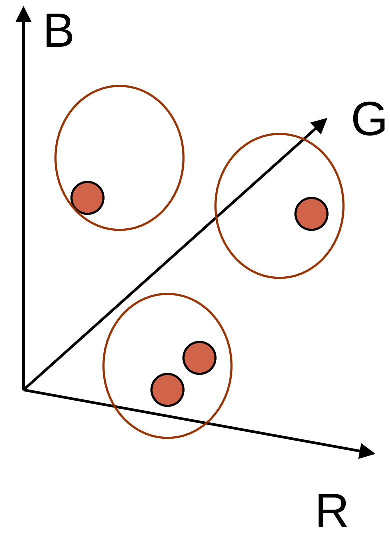
Feature space: intensity value (1-D)

Segmentation as Clustering

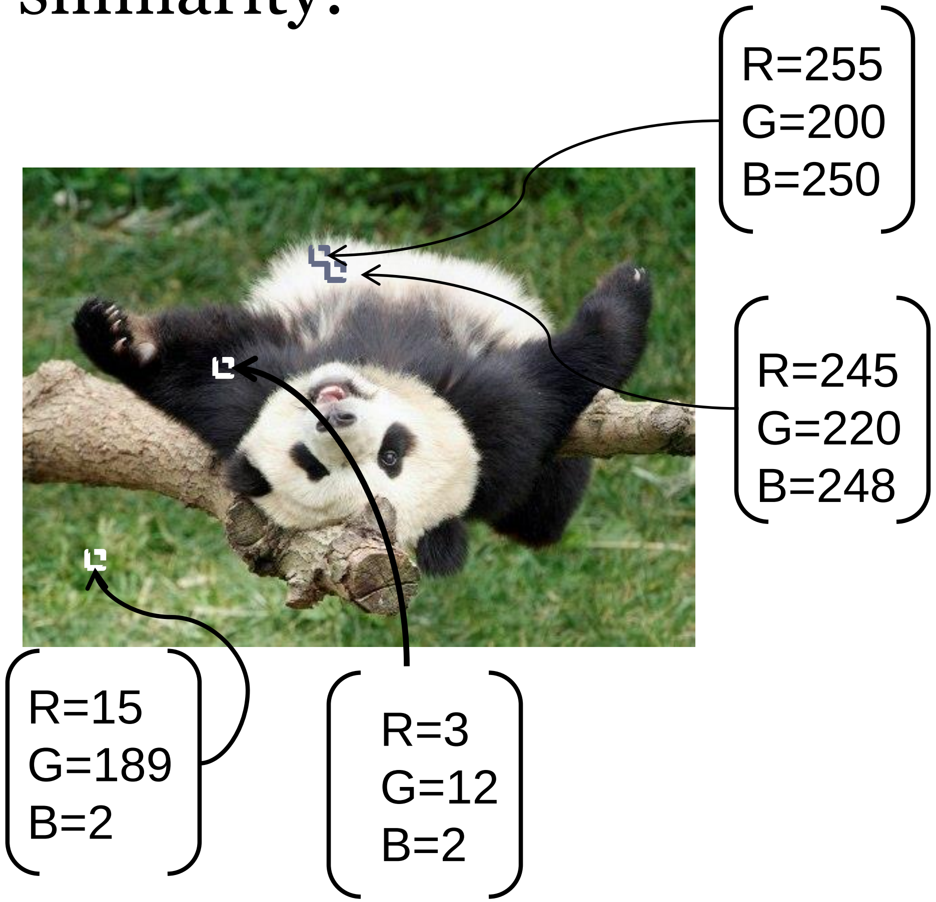


Segmentation as Clustering

- Grouping pixels based on **colour** similarity:

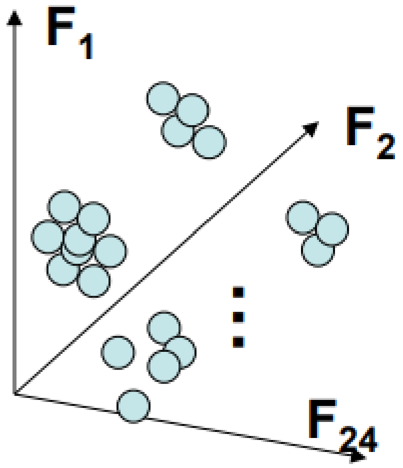


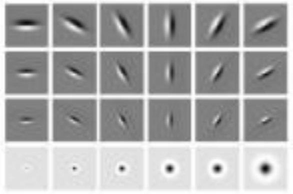
Feature space: intensity value (3-D)



Segmentation as Clustering

- Grouping pixels based on **texture** similarity





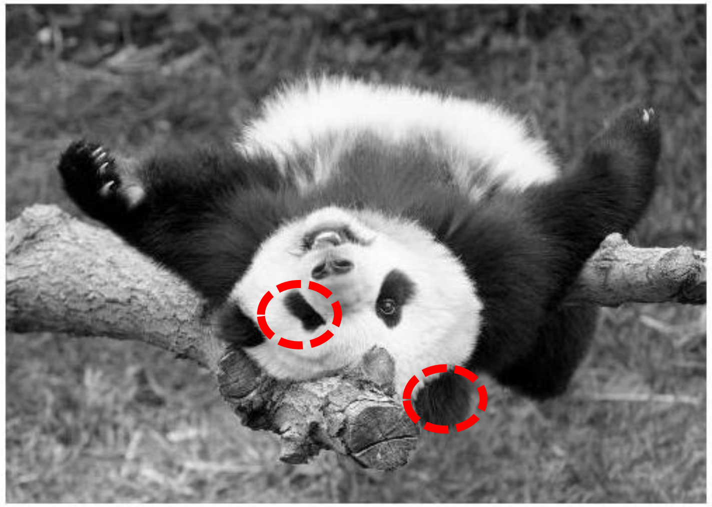
Filter bank of
24 filters

Feature space: filter bank responses (e.g., 24-D)

Segmentation as Clustering

- Returning to grouping pixels based on **intensity** similarity
- In what case is intensity not enough for segmentation?

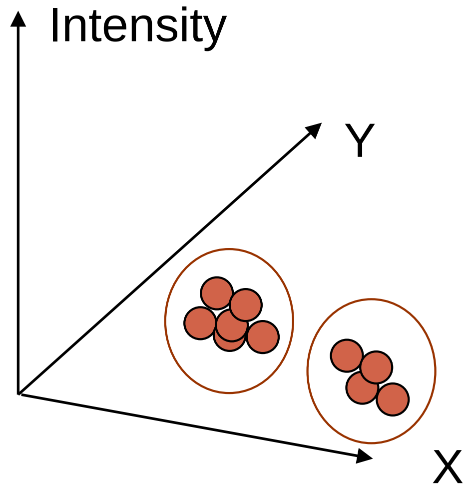
Feature space: intensity value (1-D)

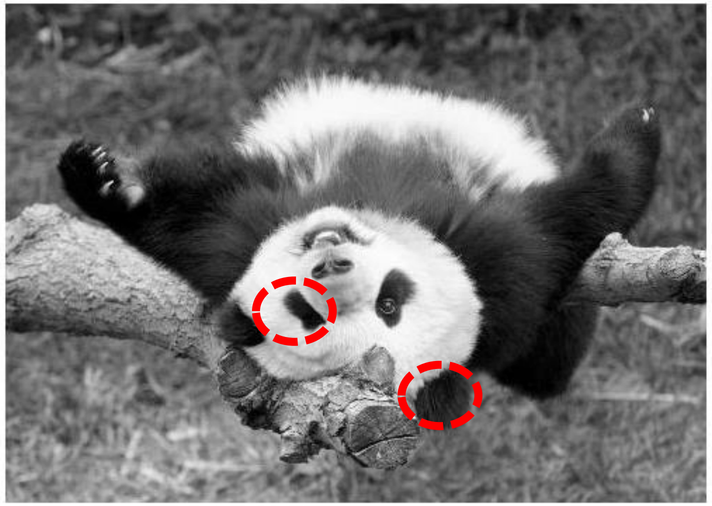


- Clusters based on intensity similarity don’t have to be spatially coherent

Segmentation as Clustering

- Grouping pixels based on **intensity & position** similarity:





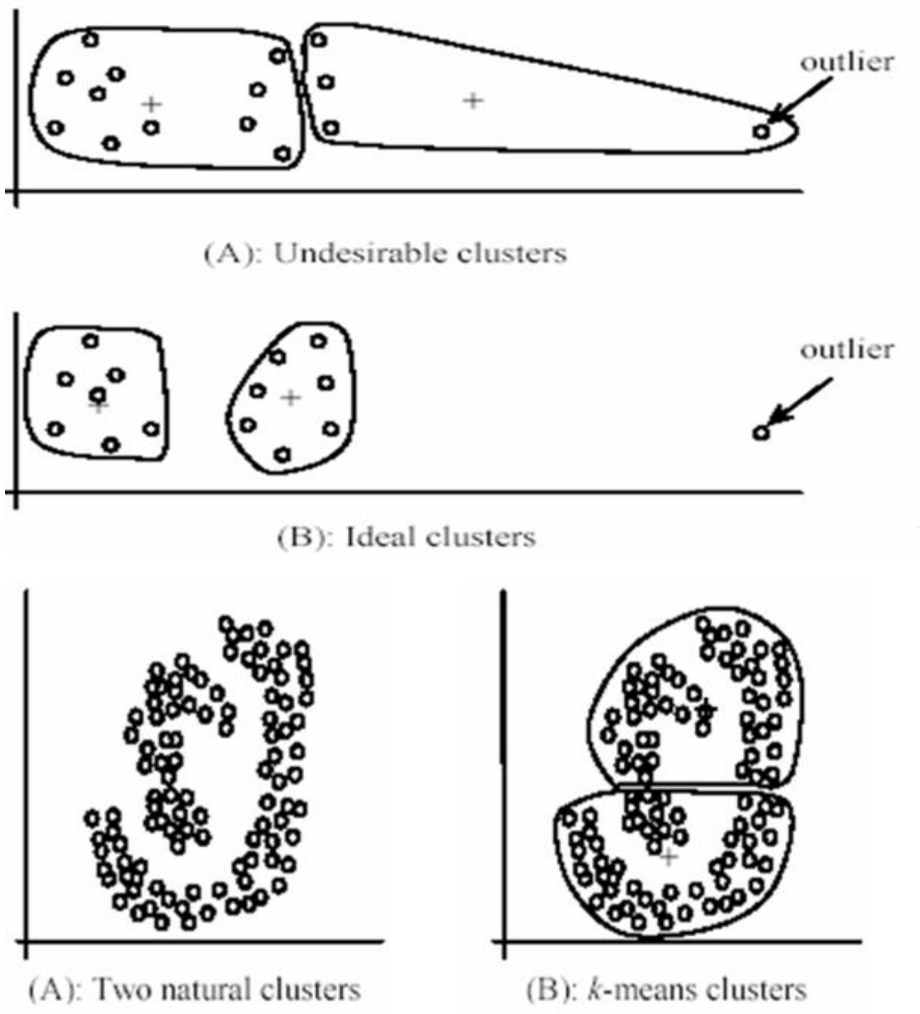
- Both regions are black
- Including **position (*x*,*y*)** information allows us to group the regions into two distinct segments

Edith Deretey, Zia Ud Din, Michael Greenspan

K-Means Clustering

- **Pros:**
  - Simple and fast
  - Easy to implement
  - Converges to local minimum of within-cluster squared error

- **Cons:**
  - Specifying *k*
  - Sensitive to initial centers
  - Sensitive to outliers
  - Detects spherical clusters only



K-Means: Choosing K

- The most common approach is to use elbow method
  - Try different values of K
  - Plot the K-means objective versus K
  - Look at the "elbow-point" in the plot
    - K value below which SSD explodes
- For the above plot, K = 6 is the elbow point

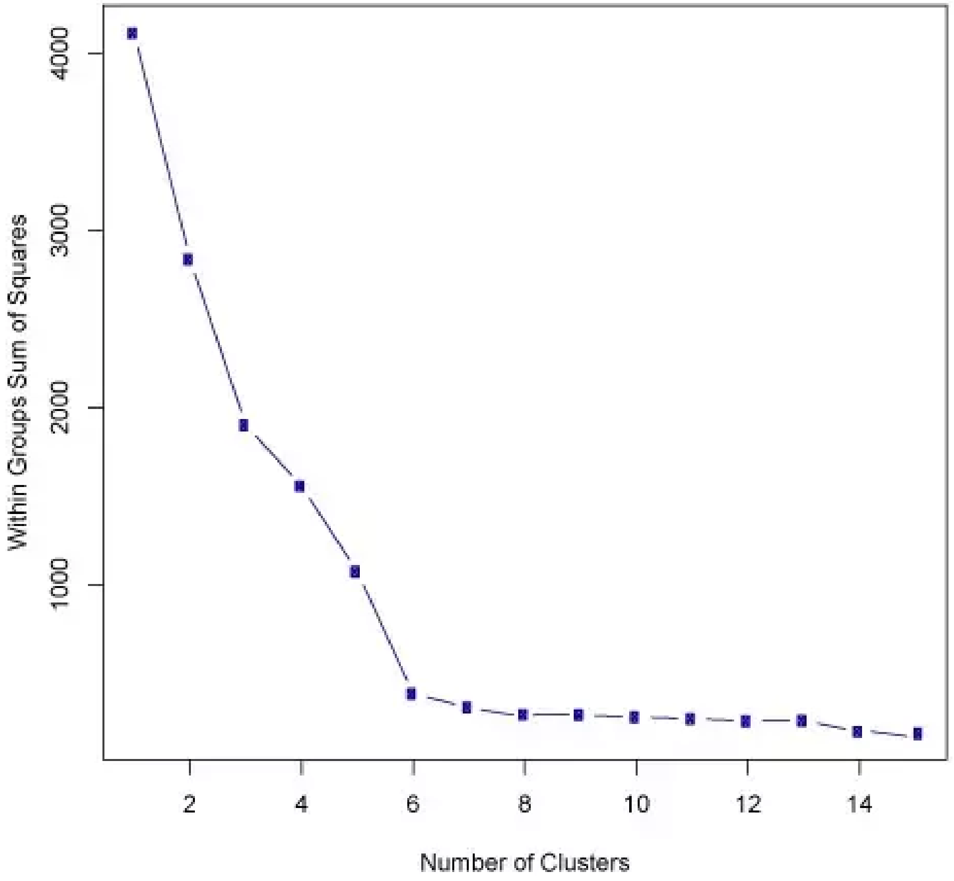


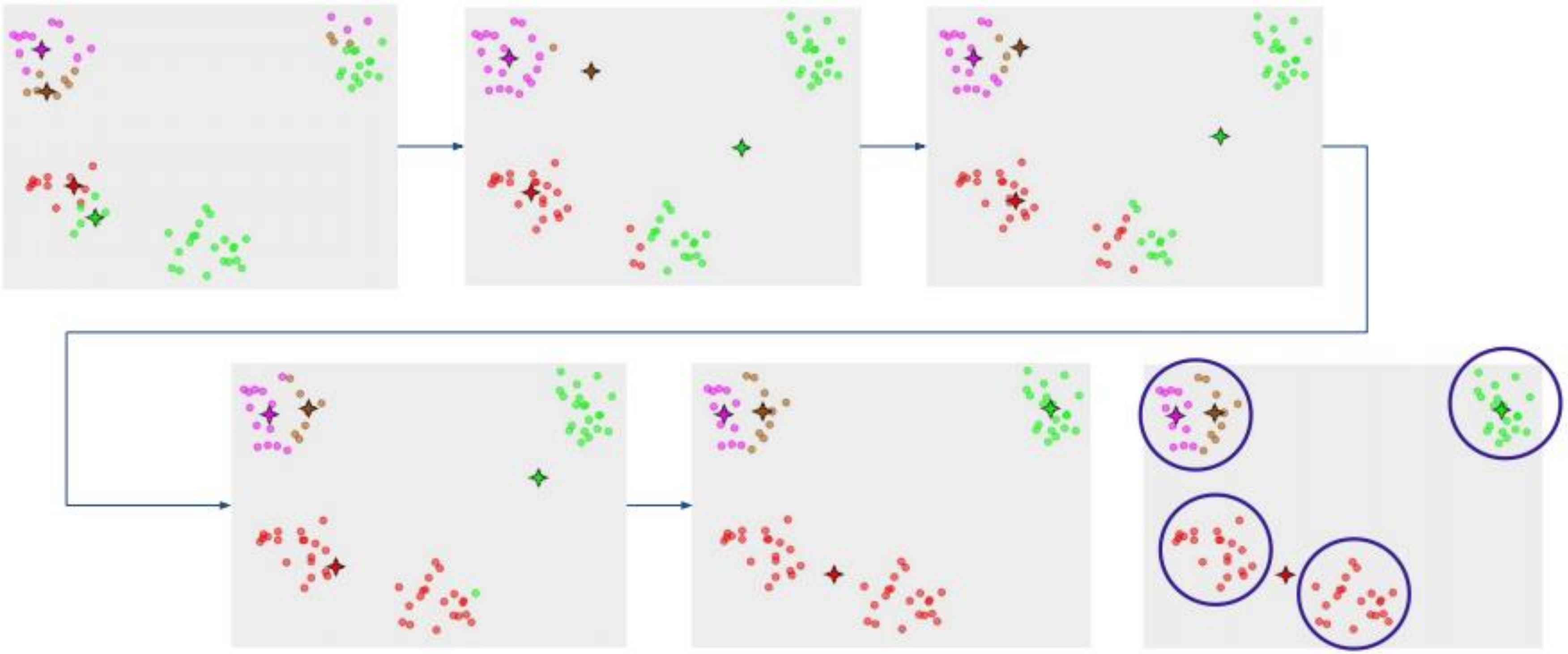
Image source: https://www.quora.com/How-can-we-choose-a-good-K-for-K-means-clustering

K-Means: Initialization

- Initialization = where to distribute the initial c_i
- K-means is extremely sensitive to initialization
- Bad initialization can lead to:
  - Poor convergence speed
  - Bad overall clustering
- How to initialize?
  - Randomly from data
  - Try to find K “spread-out” points
    - Furthest point algorithm
    - K-means ++
- Safeguarding measure:
  - Try multiple initializations and choose the best

K-Means: Initialization

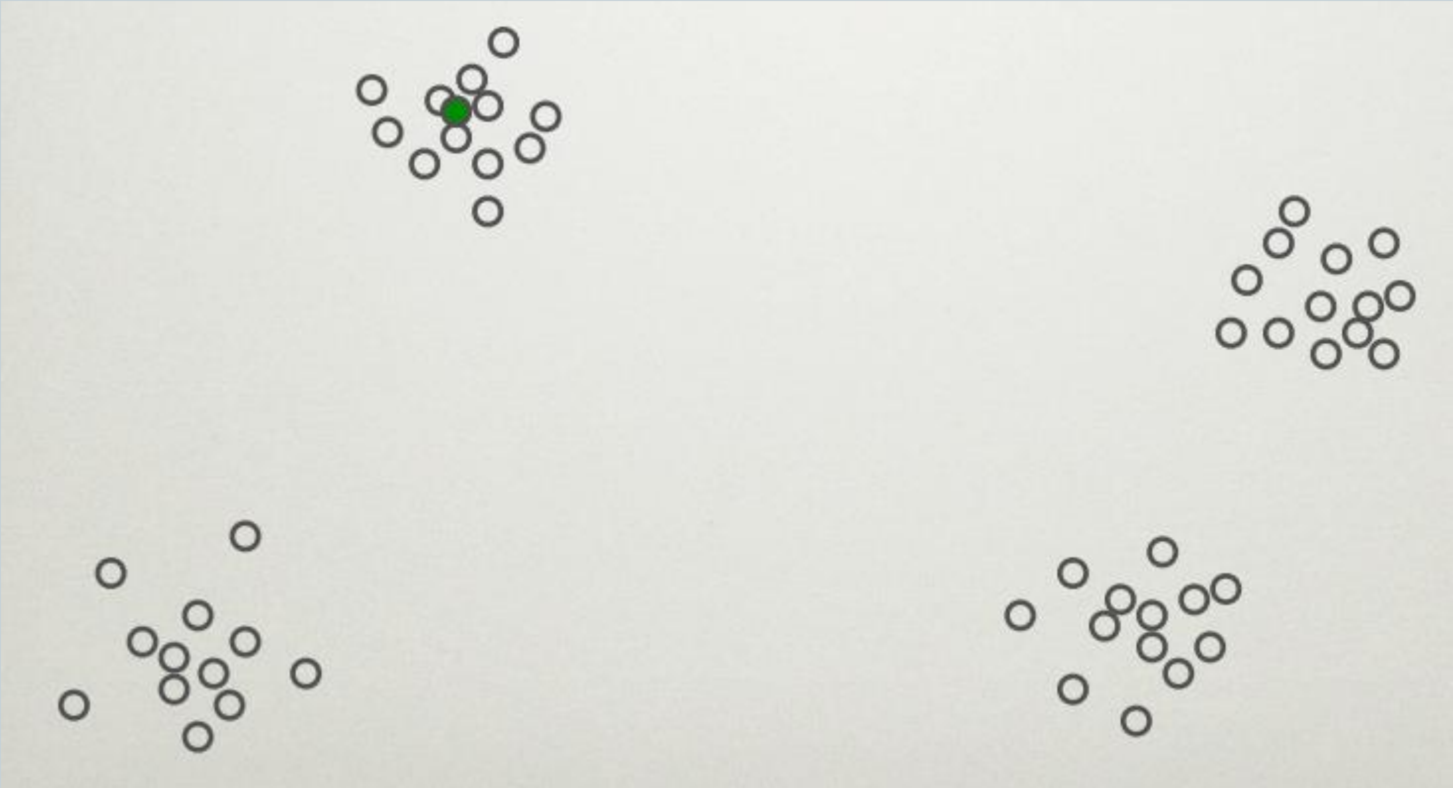
- Bad initialization can lead to:
  - Poor convergence speed
  - Bad overall clustering



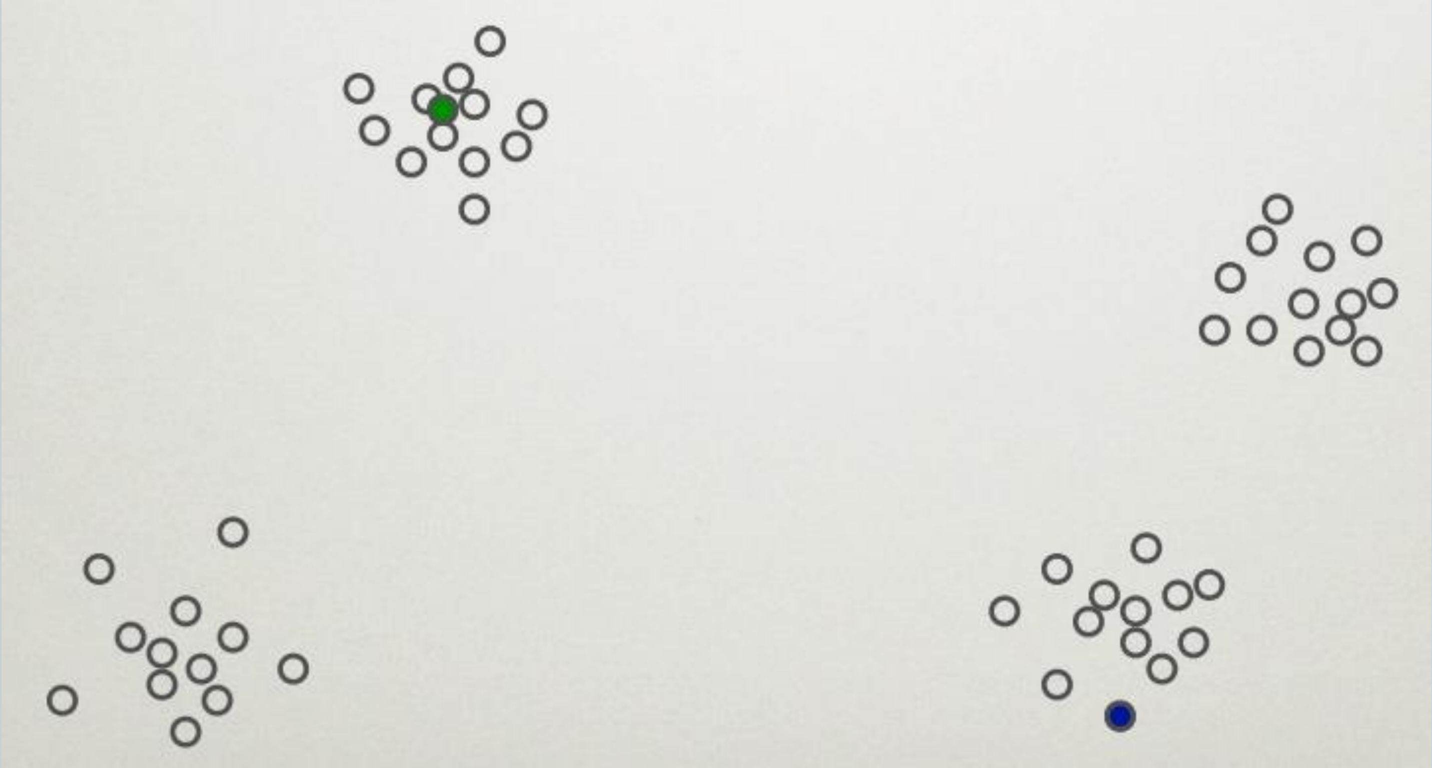
Furthest Point Algorithm

- For better initialization of cluster centers
- Algorithm:
  - Pick first center randomly
  - Next center is the point furthest from the first center
  - Next center is the point furthest from both previous centers
  - In general: next center is $\arg\max_x \min_c d(x,c)$

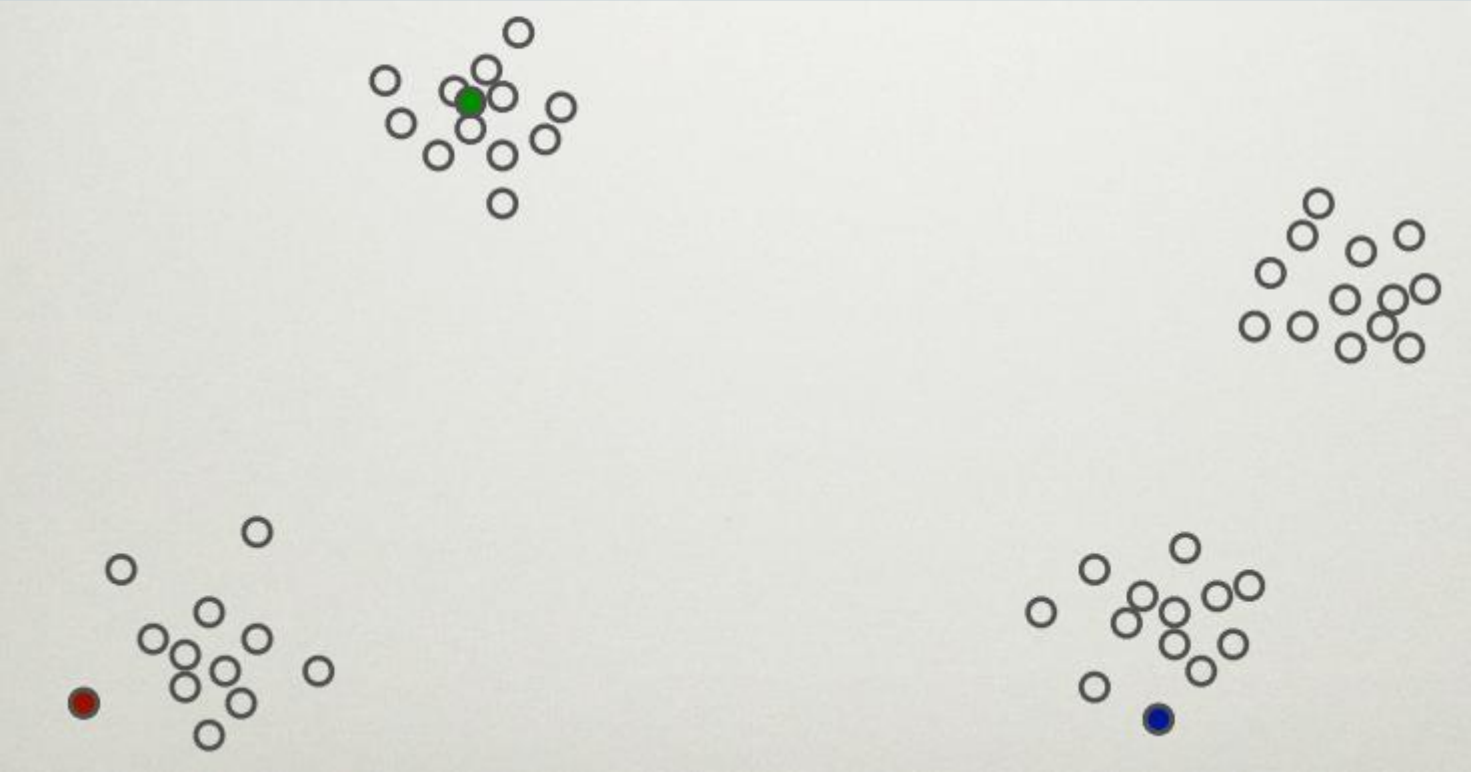
Furthest Point Algorithm



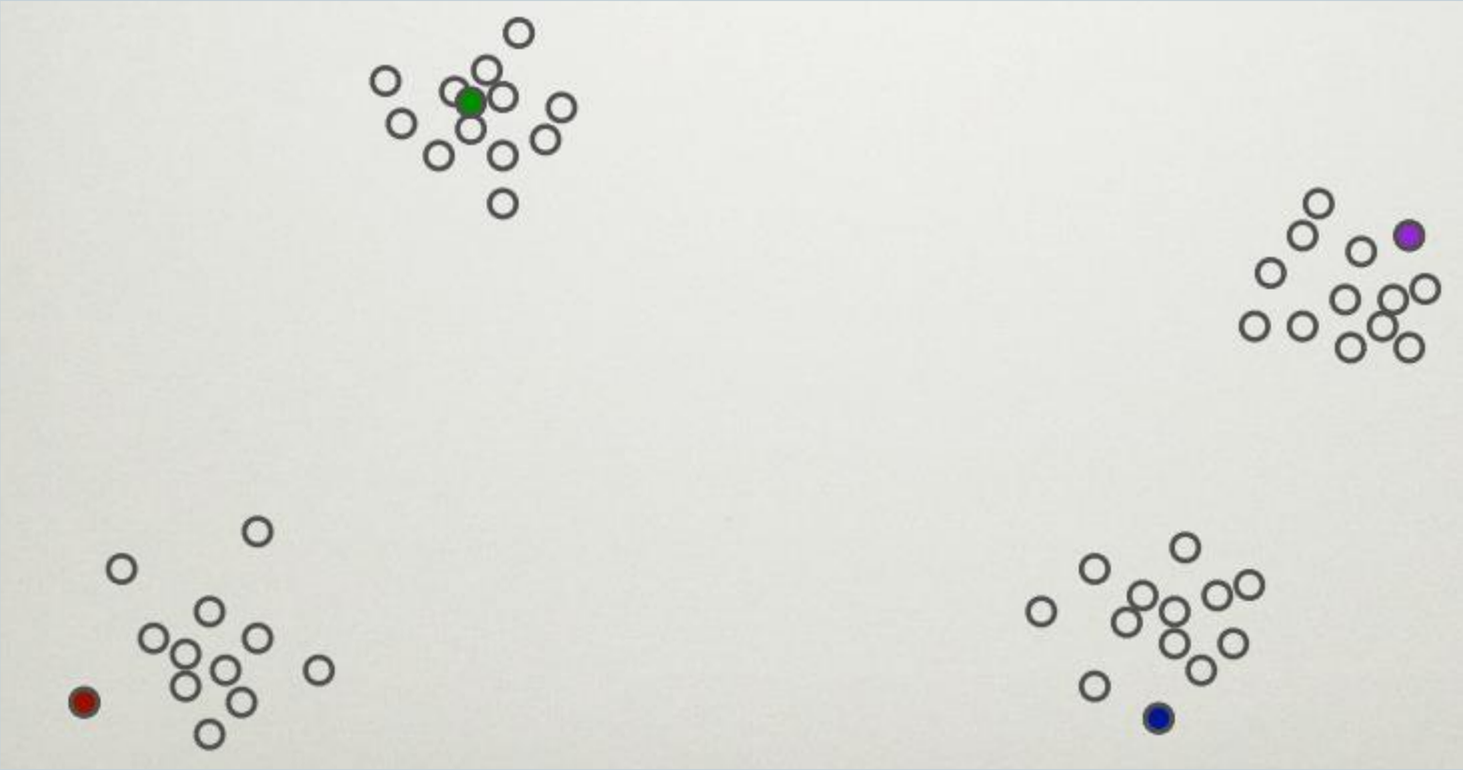
Furthest Point Algorithm



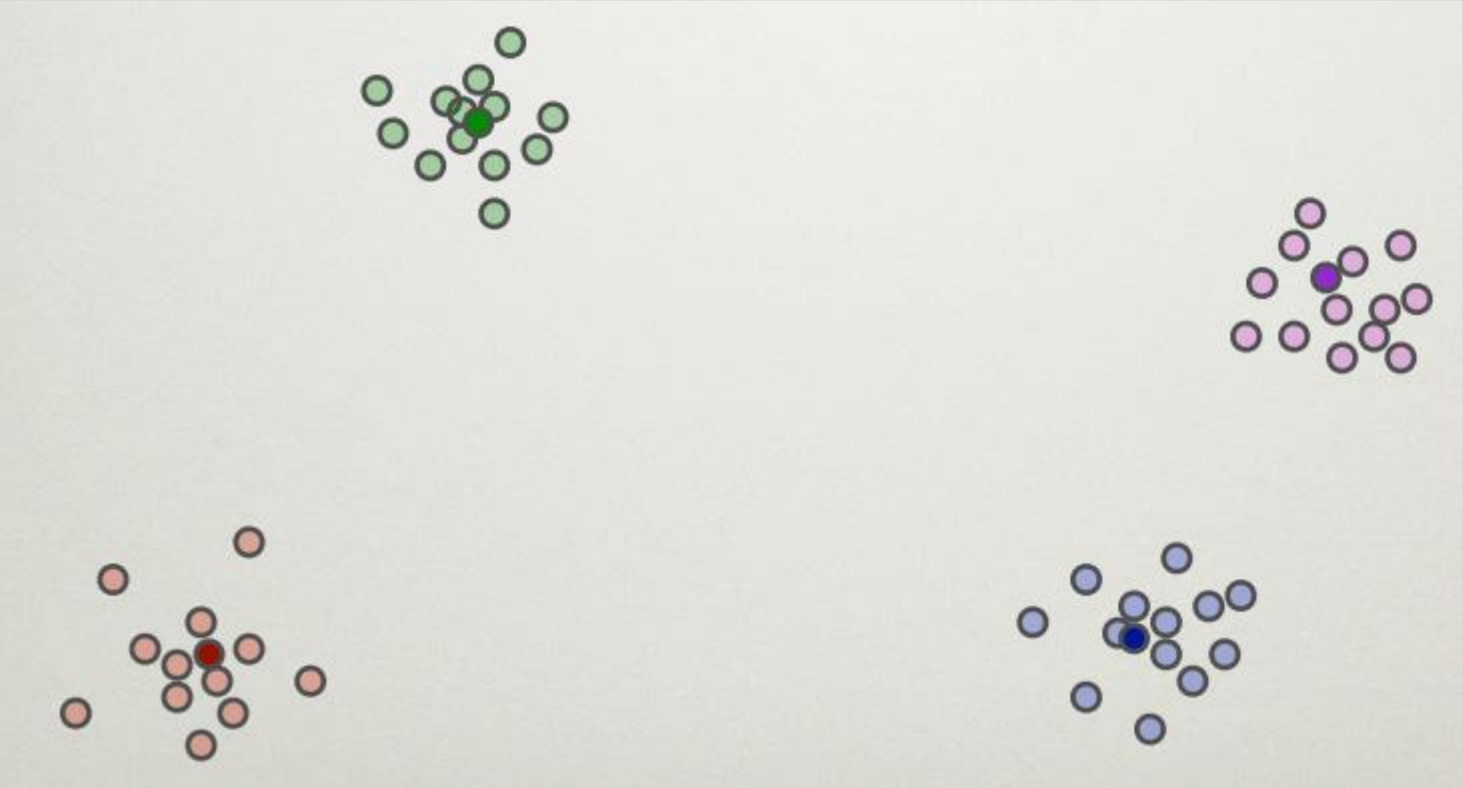
Furthest Point Algorithm



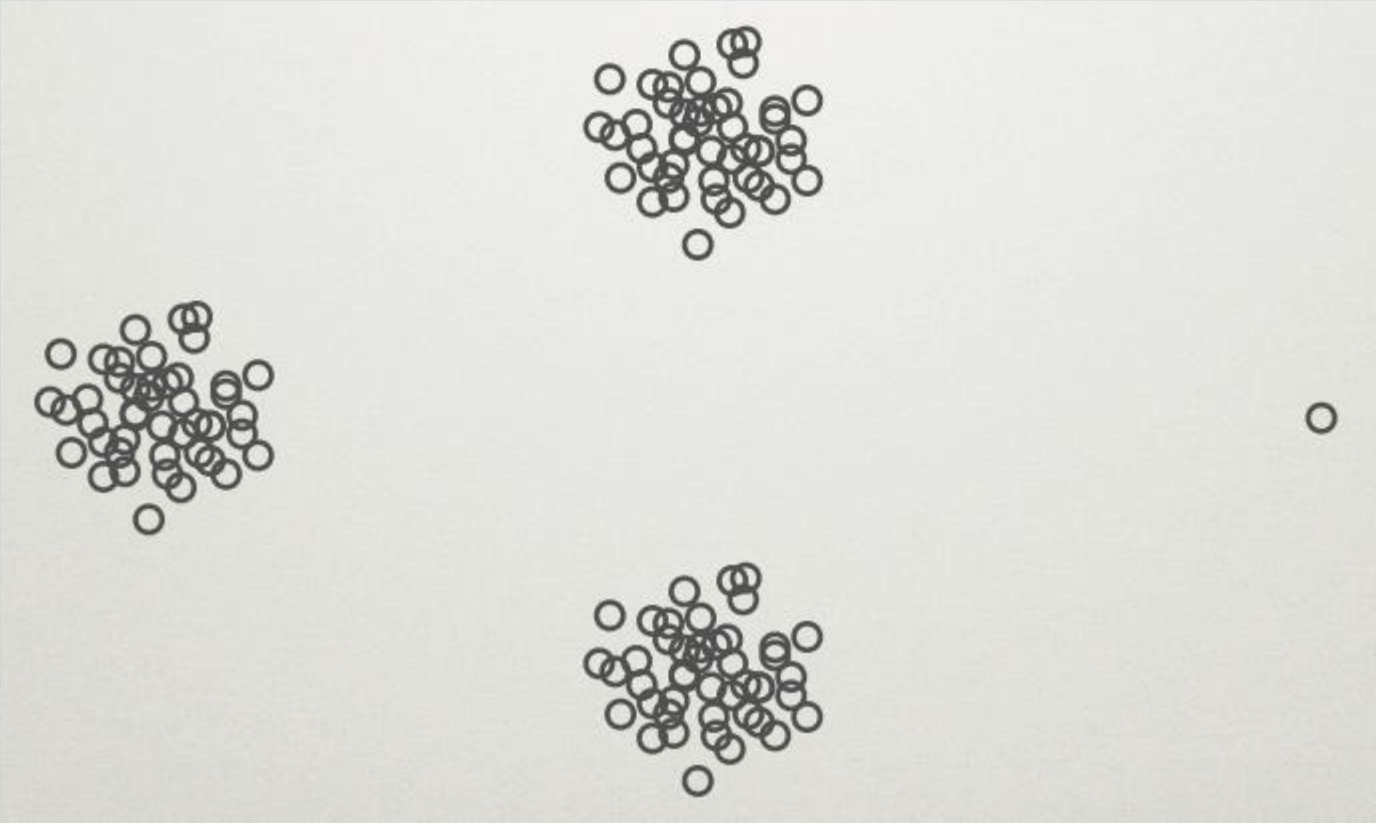
Furthest Point Algorithm



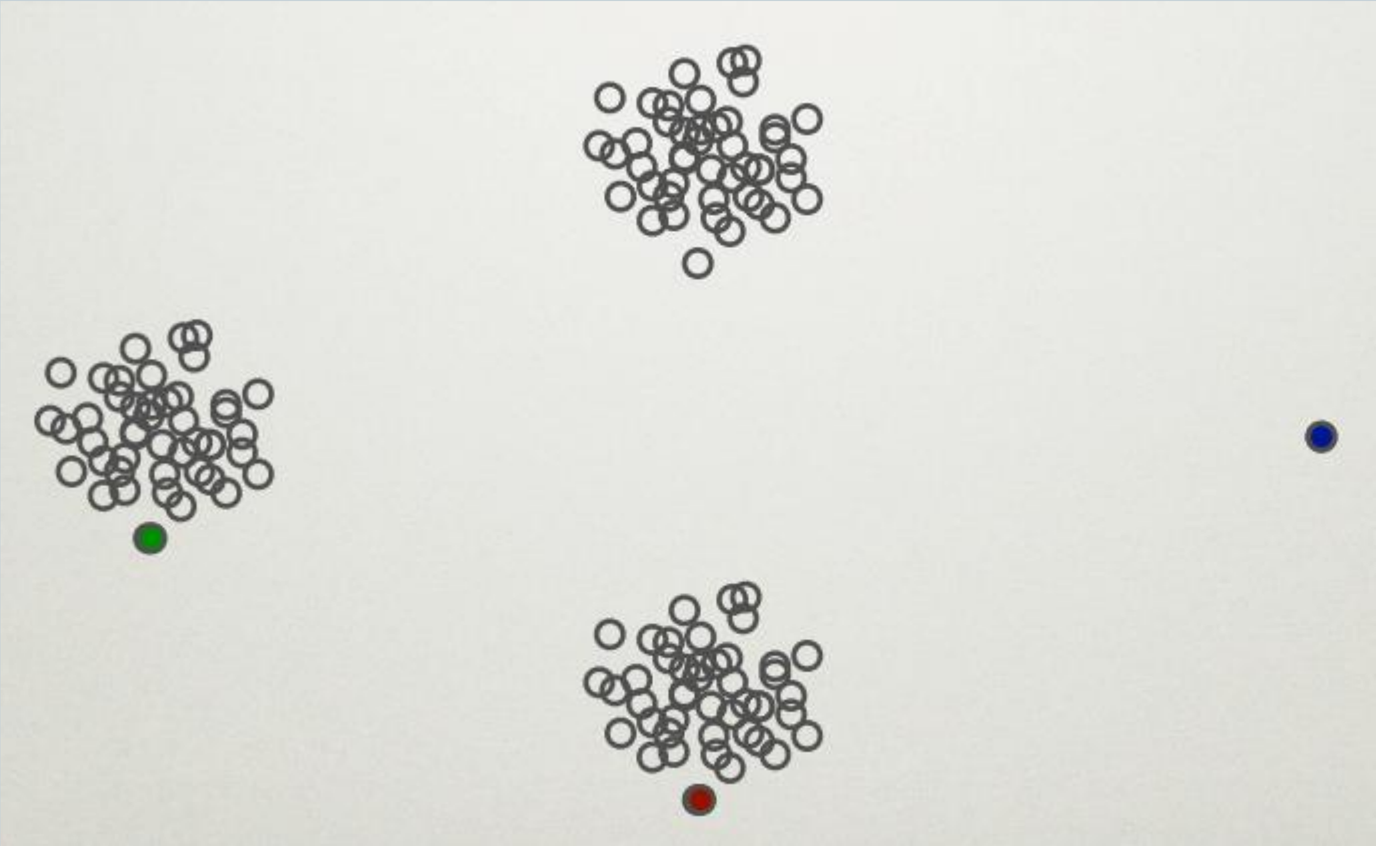
Furthest Point Algorithm



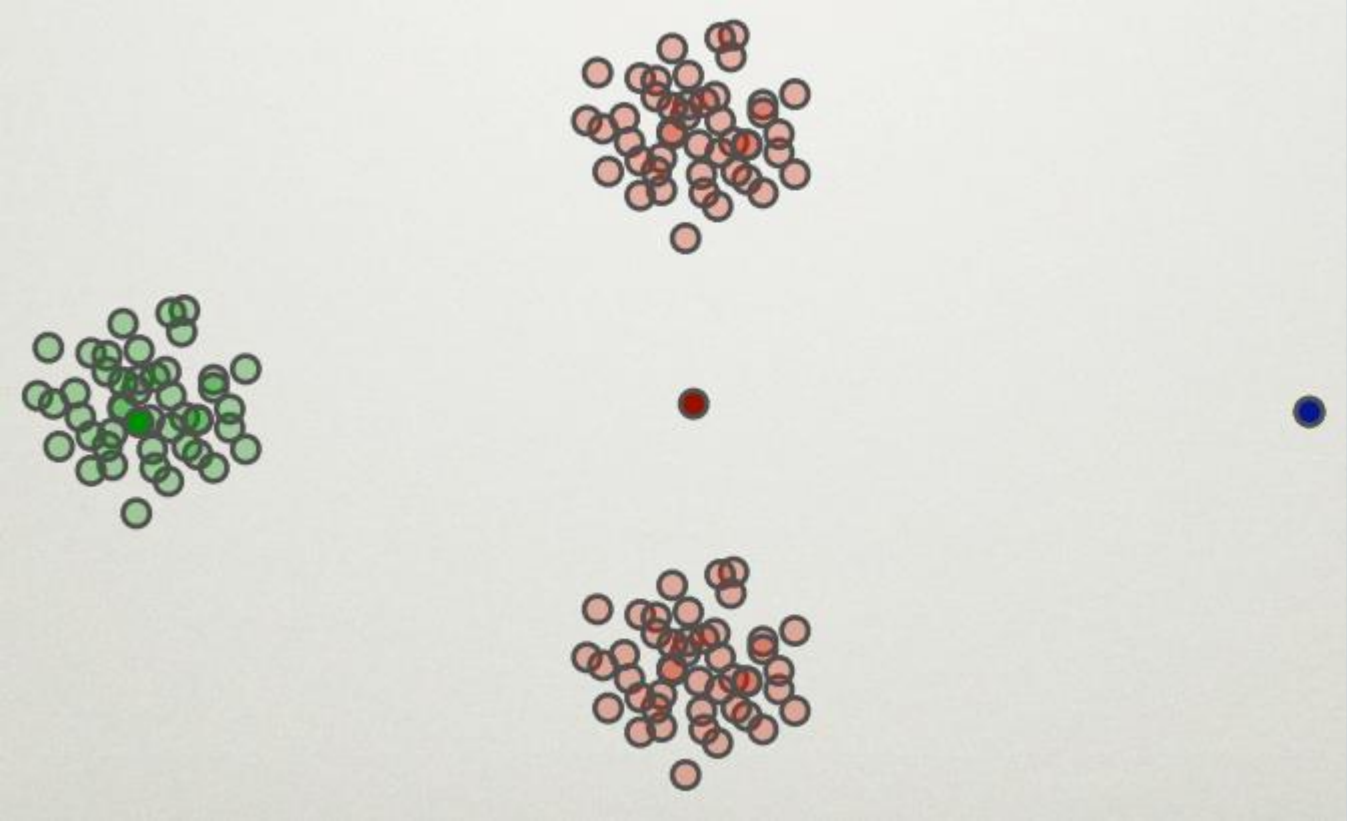
Sensitivity to Outliers



Sensitivity to Outliers



Sensitivity to Outliers

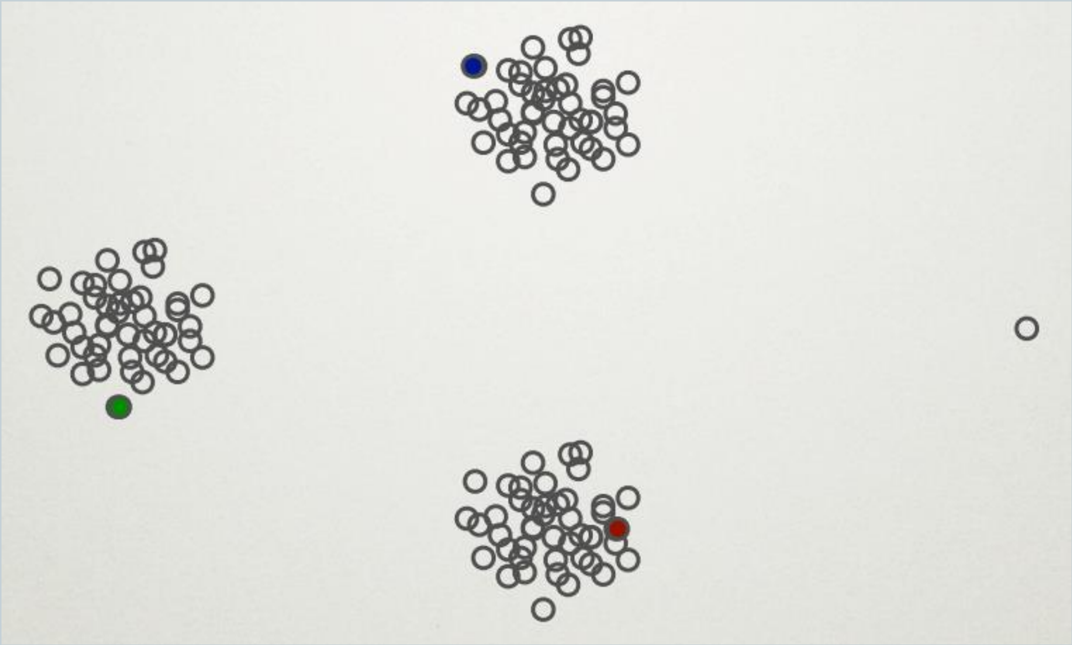


Edith Deretey, Zia Ud Din, Michael Greenspan

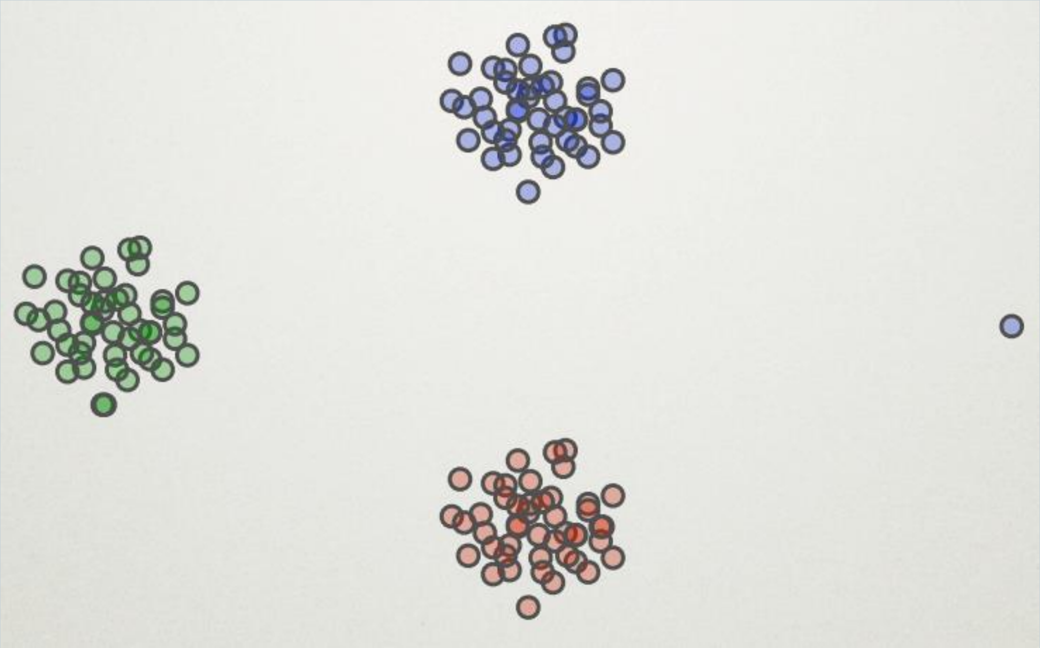
K-Means ++

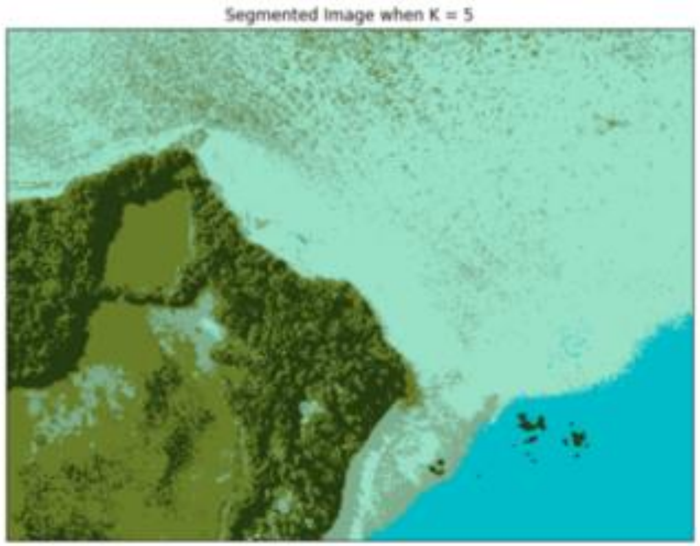
- Provides better initialization as well as better sensitivity to outliers
- Algorithm:
  - Randomly choose first center
  - Pick new center with probability proportional to $(x - c_i)^2$
  - Repeat for K centers

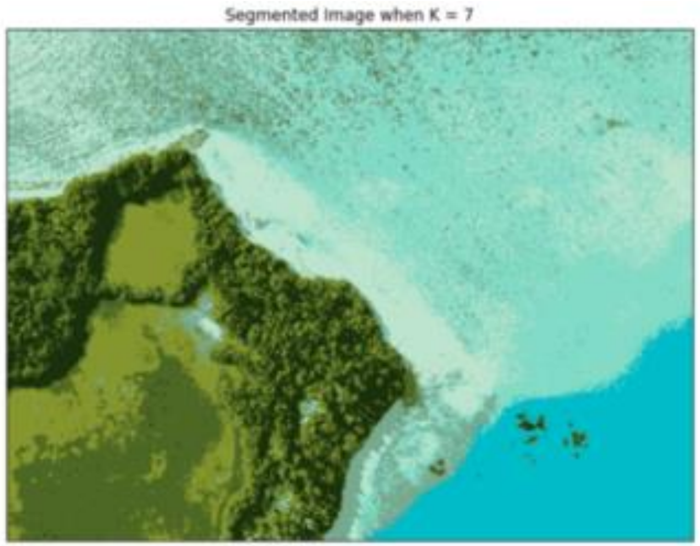
K-Means ++



K-Means ++







https://towardsdatascience.com/introduction-to-image-segmentation-with-k-means-clustering-83fd0a9e2fc3

https://towardsdatascience.com/introduction-to-image-segmentation-with-k-means-clustering-83fd0a9e2fc3

References and Resources

- Stuart P. Lloyd. Least squares quantization in pcm. IEEE Transactions on Information Theory, 28(2):129–136, 1982
- D. Arthur and S. Vassilvitskii, "K-Means++: The advantages of careful seeding," in Proc. Symp. Discrete Algorithms, 2007, pp. 1027–1035.
- Slide sources
  - Fei-Fei Li (http://vision.stanford.edu/teaching/cs131_fall1617/lectures/lecture13_kmeans_mean_shift_cs131_2016)
  - Sergei Vassilvitskii and David Arthur (http://theory.stanford.edu/~sergei/slides/BATS-Means.pdf)
  - http://shabal.in/visuals/kmeans/KMeansPlusPlus.pdf