Clustering:
K-Means & Variants

Supervised vs Unsupervised Learning

- Supervised learning
  - Predict target value (“y”) given features (“x”)
  - Categorical *y* : classification
  - Continuous *y* : regression
- Unsupervised learning
  - Understand patterns of data (just “x”)
- One example: *clustering*

Clustering

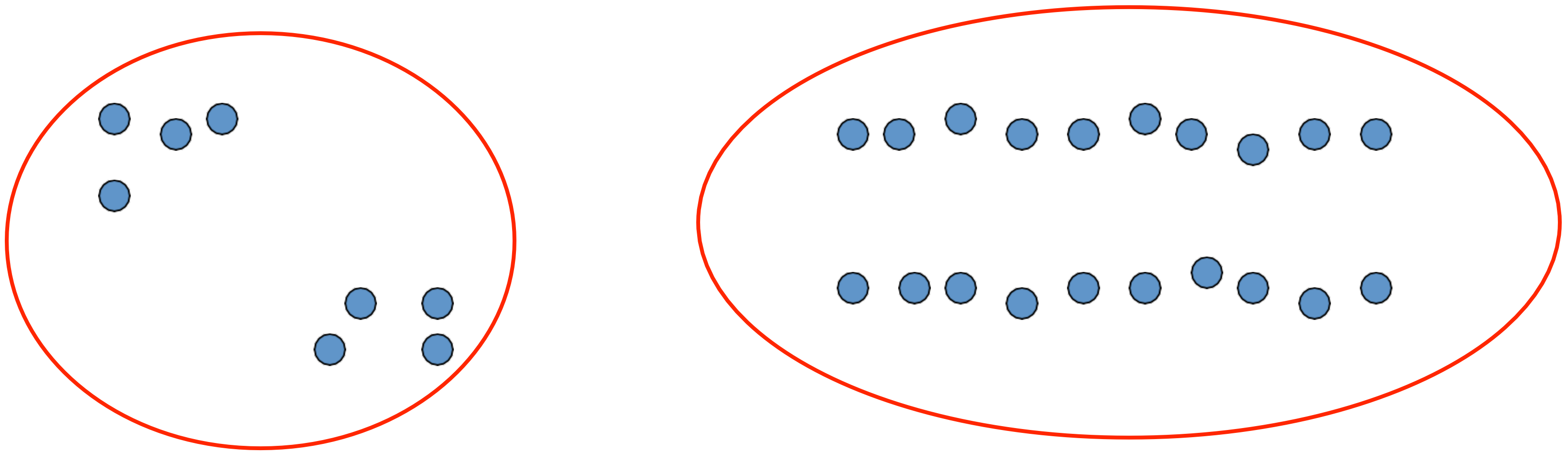
- **Goal**: Automatically cluster unlabeled data into groups
  - Data points within a group are similar

- **Useful**
  - Unlabeled data cheap, but labeled data expensive
  - Automatically organizing data
  - Understanding hidden structure in data
  - Preprocessing for further analysis
    - Data compression: Save memory/computation
    - Data visualization: Represent high–dim data in a low–dim space
    - For supervised learning

An Example

- Basic idea: group together similar data points
- Example: 2D point patterns

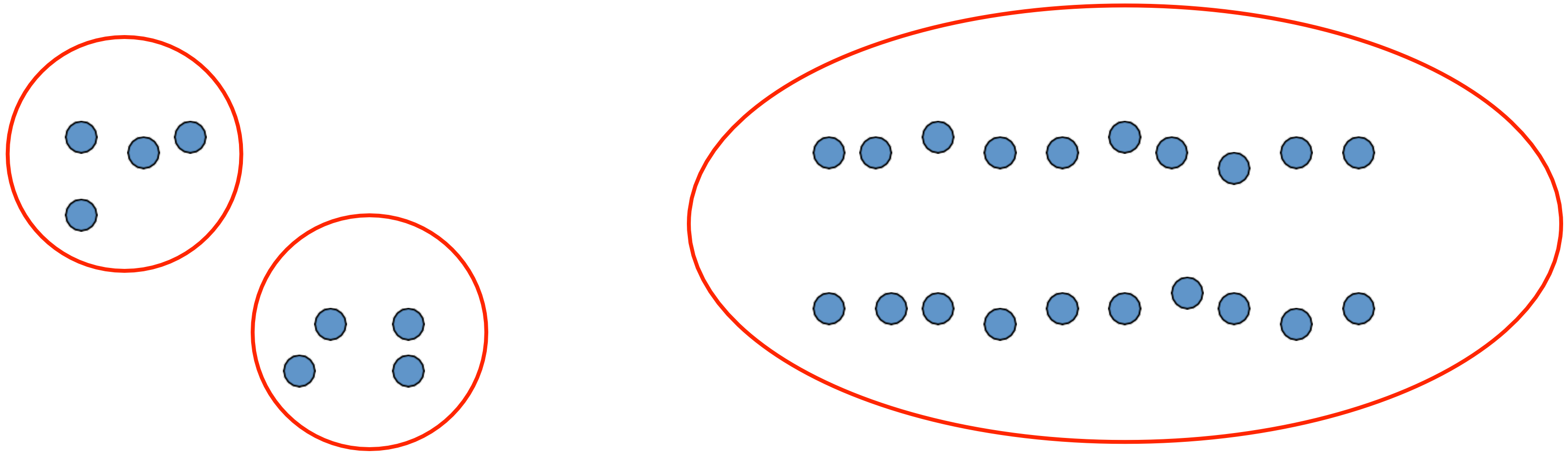
2 Clusters



An Example

- Basic idea: group together similar data points
- Example: 2D point patterns

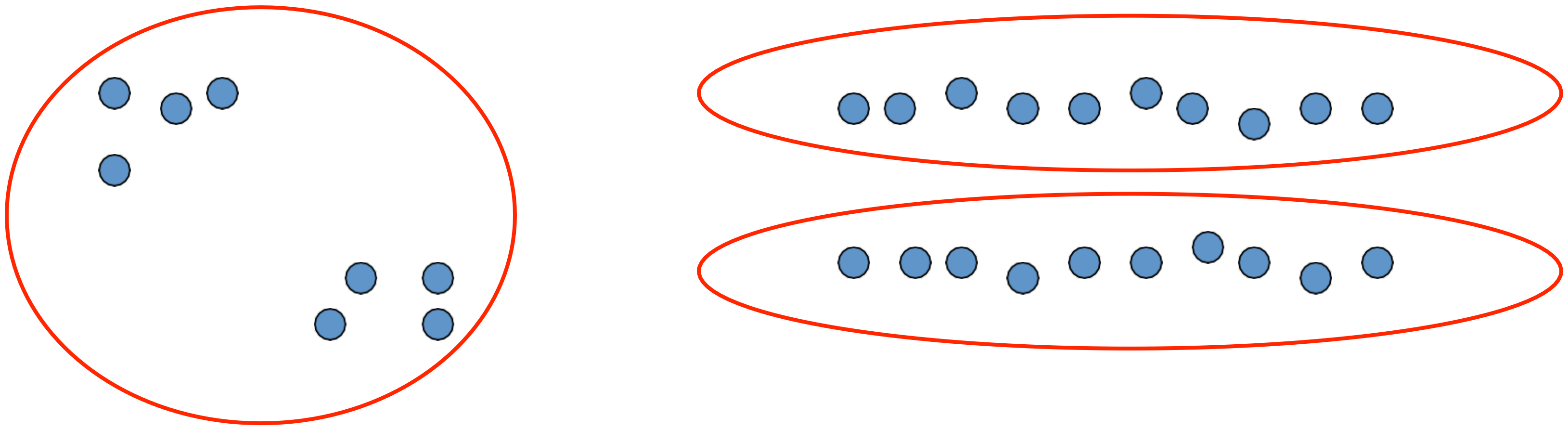
3 Clusters



An Example

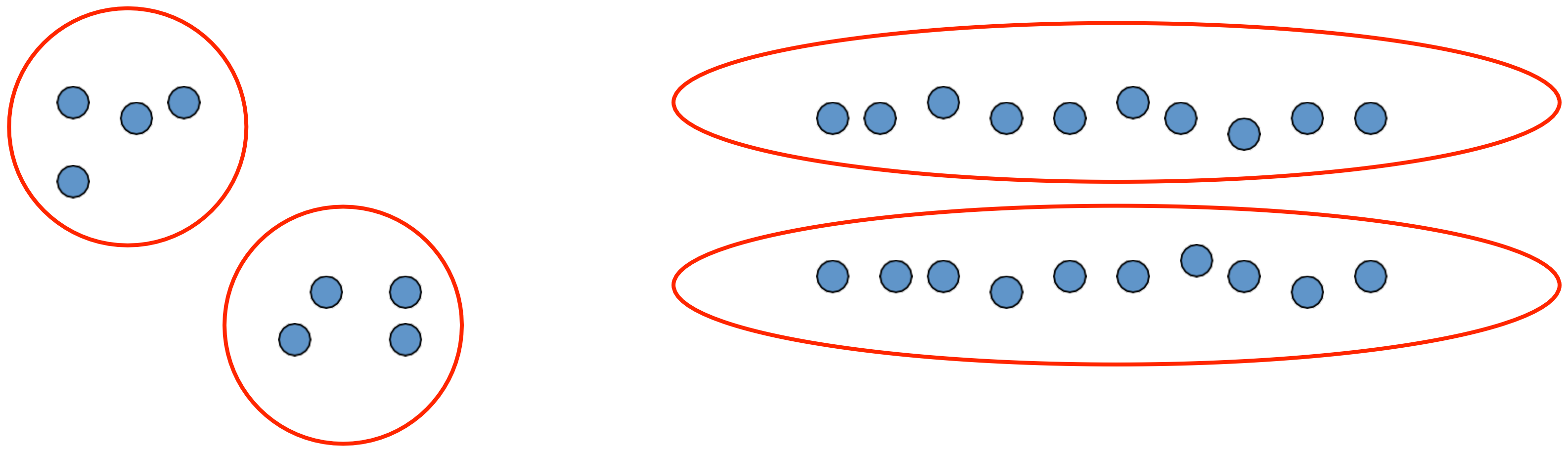
- Basic idea: group together similar data points
- Example: 2D point patterns

3 Clusters



An Example

- Basic idea: group together similar data points
- Example: 2D point patterns

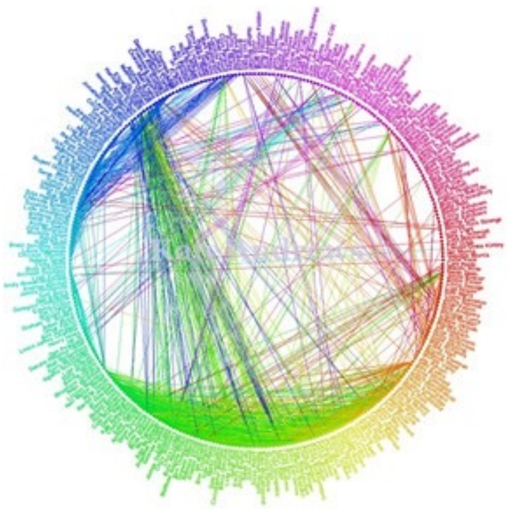


Ingredients of Clustering Analysis

- A (dis-)similarity function between data points
  - What could “(dis)similar” mean?
- A loss function to evaluate clusters
  - How to formalize such that similar data points form a cluster?
- An algorithm that optimizes the loss function
  - How to obtain the final clusters?

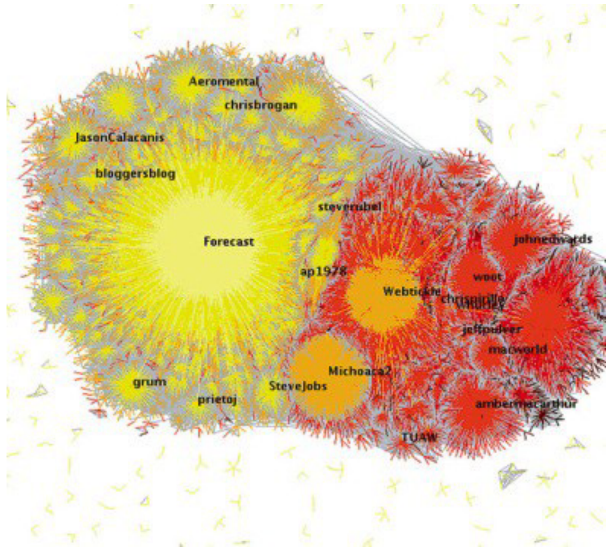
Applications: Image Segmentation

Applications: Community Detection



Facebook network

Cluster users of social networks by similar interests/professions



Twitter Network

K-Means Clustering
(Lloyd, 1982)

K-Means: Main Idea

- K clusters: each cluster k is summarized by a **centroid** μ_k

- Each $\boldsymbol{x}_i$ uses **one-hot encoding** to specify a **hard** assignment

$$\mathbf{r}_i = [0, 0, \cdots, 1, \cdots] \in \{0, 1\}^K$$

$$r_{ik} = 1, \text{if } \mathbf{x}_i \text{ is assigned to cluster k}$$

An example with 4 data points and 3 clusters $\begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \\ \mathbf{r}_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

- Loss function

$$J(\{r_{ik}\}, \{\mu_k\}) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \|\mathbf{x}_i - \mu_k\|_2^2$$

Total Euclidean distance of all data points to their corresponding centroid

K-Means: Objective Function

- Minimize loss J with respect to (r_{ik} ,μ_k)

$$\min J(\{r_{ik}\}, \{\mu_k\}) = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \|\mathbf{x}_i - \mu_k\|_2^2$$

- Chicken and egg problem
  - Obtaining the centroids $\{\mu_k\}$ needs to know the cluster assignment $\{r_{ik}\}$
  - Obtaining cluster assignment $\{r_{ik}\}$ needs to know the centroids $\{\mu_k\}$

- Combinatorial optimization problem
  - NP hard => Impossible to obtain the global minimum
  - Expect the local minimum

- Lloyd’s method: alternative minimization algorithm

Lloyd's Method: Update Cluster Assignments

- **Step I**: Fix cluster centroids $\{\mu_k\}$, minimize J w.r.t. r_{ik}

$$\min J(\{r_{ik}\}) = \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik}\|\mathbf{x}_i - \mu_k\|_2^2$$

- For each i, exactly one of the following terms is nonzero

$$r_{i1}\|\mathbf{x}_i - \mu_1\|_2^2, r_{i2}\|\mathbf{x}_i - \mu_2\|_2^2, \cdots, r_{iK}\|\mathbf{x}_i - \mu_K\|_2^2$$

- Take

$$r_{ik} = \mathbf{1}[k = \arg\min_j \|\mathbf{x}_i - \mu_j\|_2^2]$$

- That is, assign $\boldsymbol{x}_i$ to to its ***nearest*** cluster centroid

Lloyd's Method: Update Cluster Centroids

- **Step II**: Fix r_{ik} , minimize J w.r.t. μ_k

$$\min J(\{\mu_k\}) = \sum_{i=1}^{N}\sum_{k=1}^{K} r_{ik}\|\mathbf{x}_i - \mu_k\|_2^2 \quad = \sum_{k=1}^{K}\sum_{i=1}^{N} r_{ik}\|\mathbf{x}_i - \mu_k\|_2^2$$

$$= \sum_{k=1}^{K} J_k(\mu_k) \qquad\qquad J_k(\mu_k) = \sum_{\mathbf{x}_i : r_{ik}=1} \|\mathbf{x}_i - \mu_k\|_2^2$$

- J_k is minimized by

$$\mu_k = \mathrm{mean}(\{\mathbf{x}_i : r_{ik} = 1\})$$

- That is, each centroid μ_k is the ***mean*** of the data points in the cluster k

Lloyd's Method Summary

- An alternating minimization algorithm
- Initialization: ***randomly*** choose K data points as initial centroids $\{\mu_k\}$
- Repeat
  - For given cluster centroids, find optimal cluster assignments

$$r_{ik} = \mathbf{1}[k = \arg\min_j \|\mathbf{x}_i - \mu_j\|_2^2]$$

  - For given cluster assignments, find optimal cluster centroids

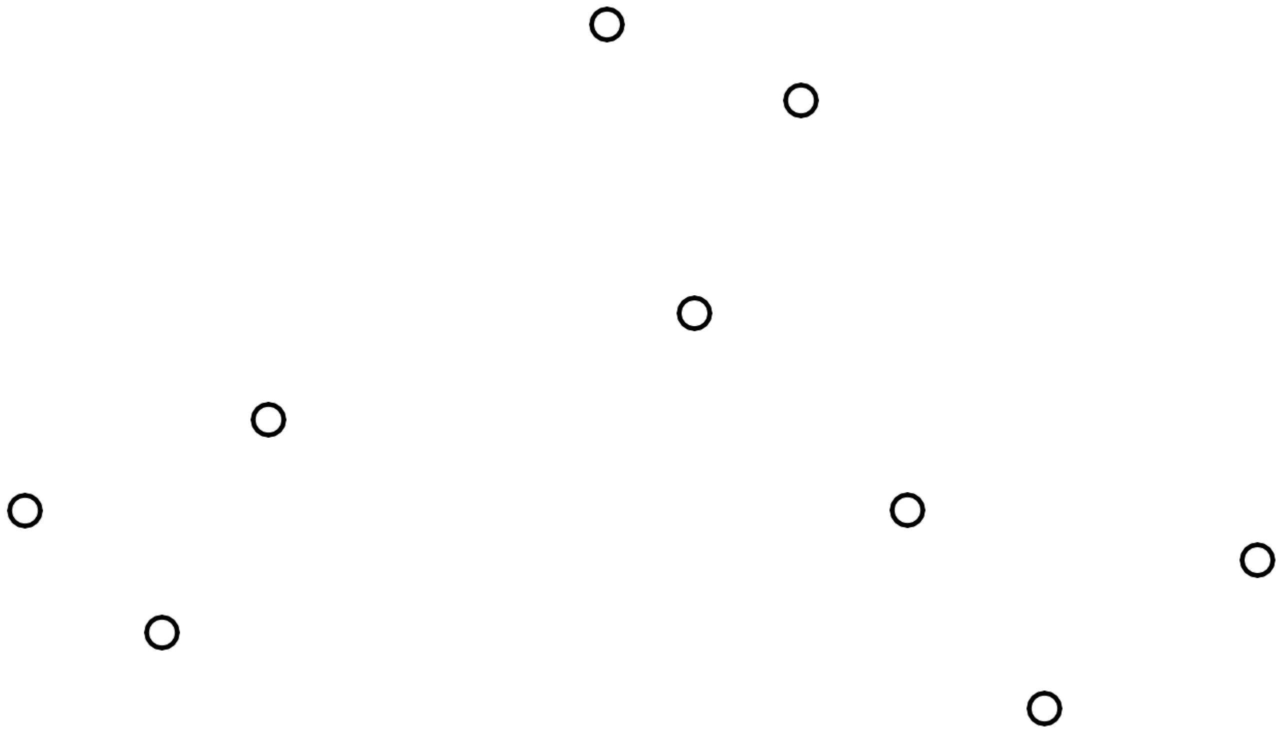
$$\mu_k = \frac{\sum_i r_{ik}\mathbf{x}_i}{\sum_i r_{ik}}$$

Lloyd's Method: Convergence

- Consider the sequence of loss values: J_1, J_2, …
  - Monotonically decreases
  - Bounded below by 0
- Hence, K-means' loss value converges to a local minimum
- In practice, to obtain the minimum loss
  - Best to repeat K-means several times, with different starting points

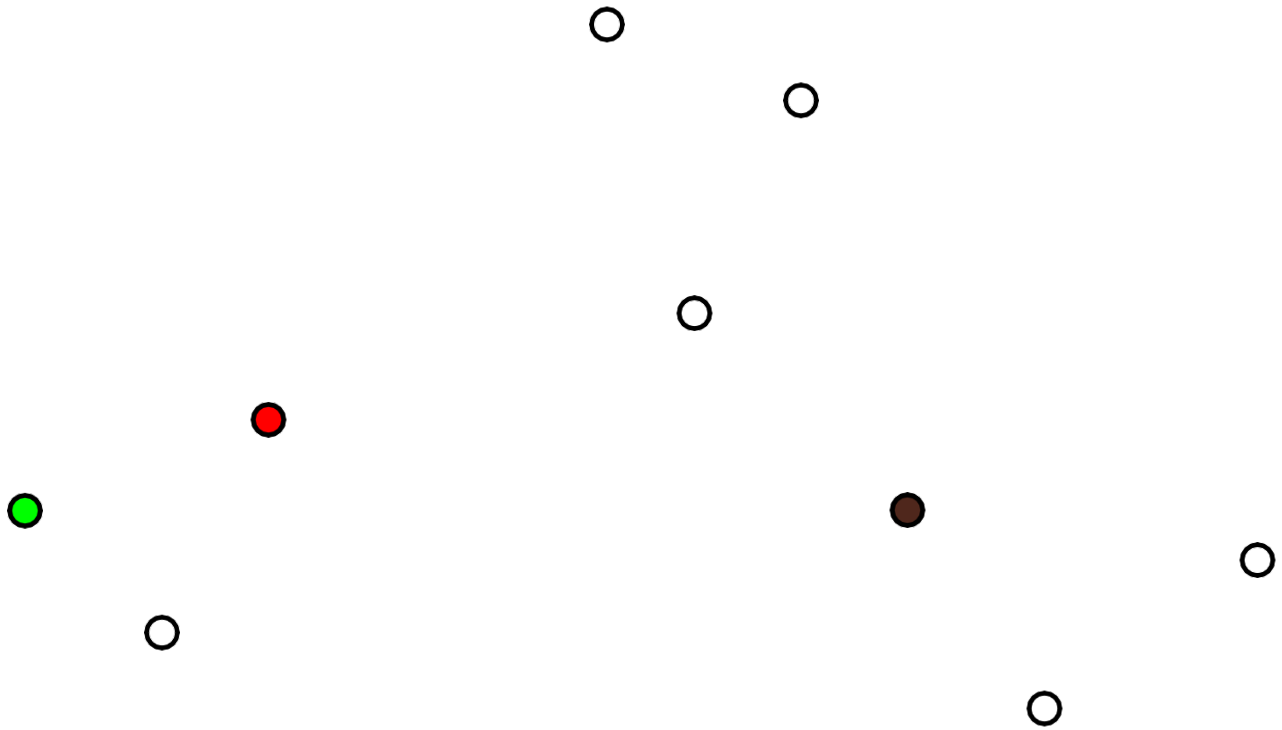
An Example

Given a set of data points



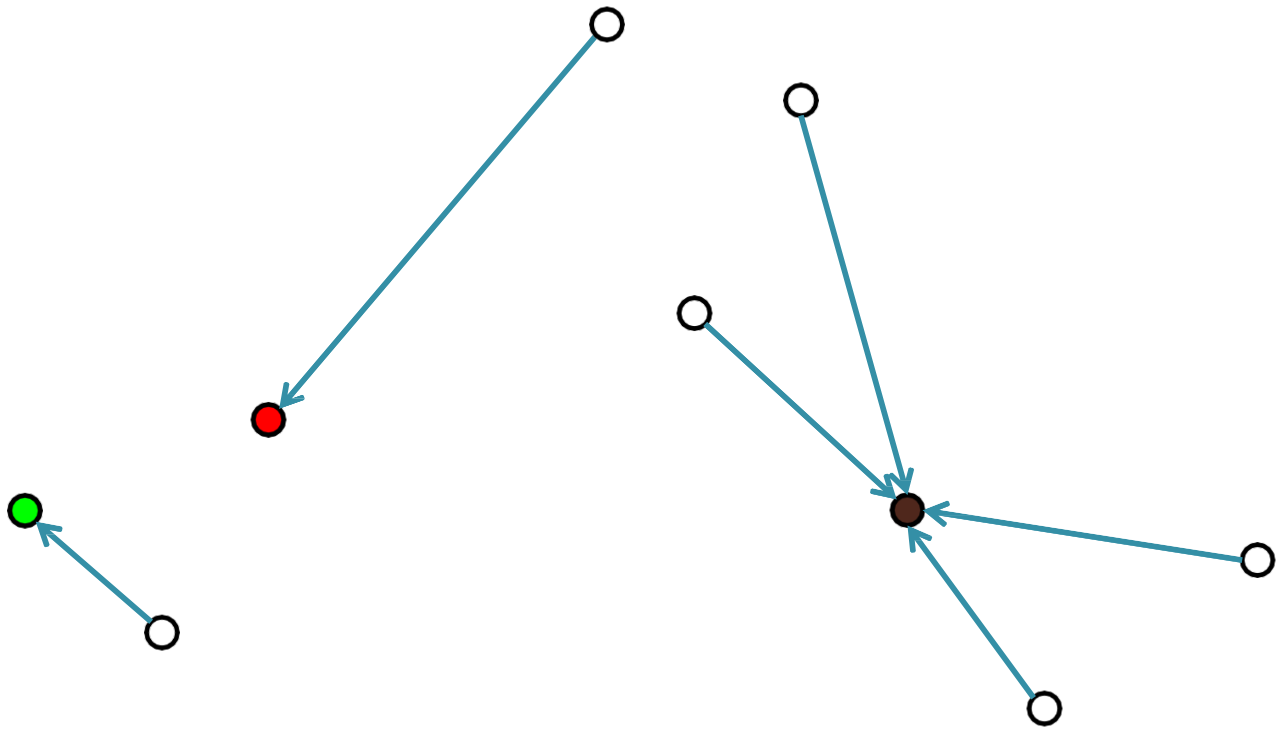
An Example

Select initial centroids **at random**



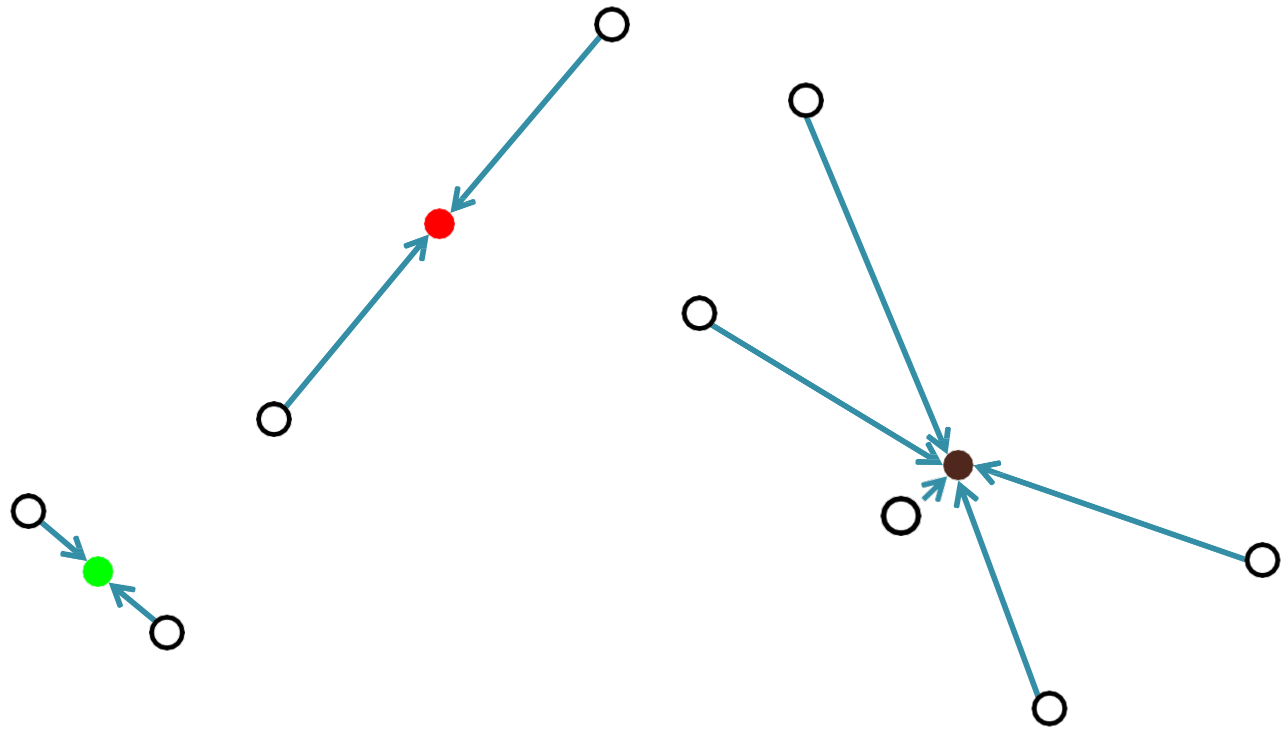
An Example

Assign each point to its nearest centroid



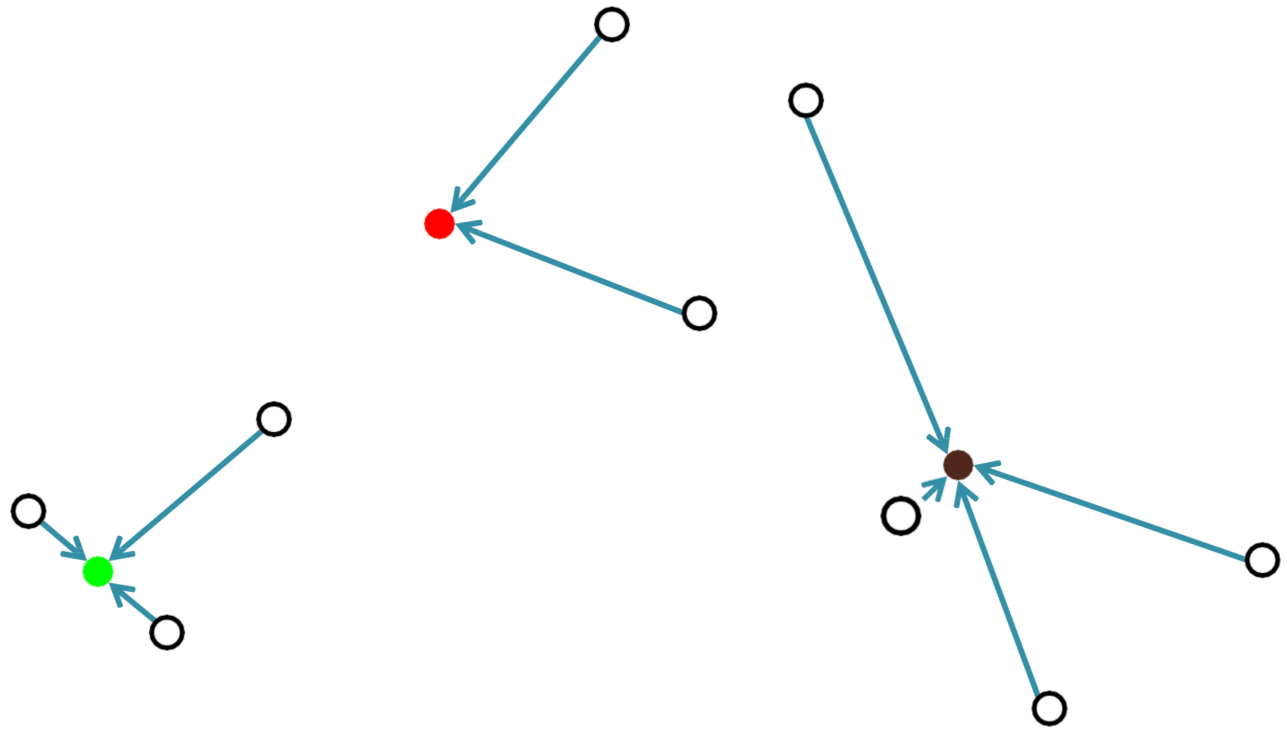
An Example

Recompute centroids as the ***mean*** of the points in that cluster



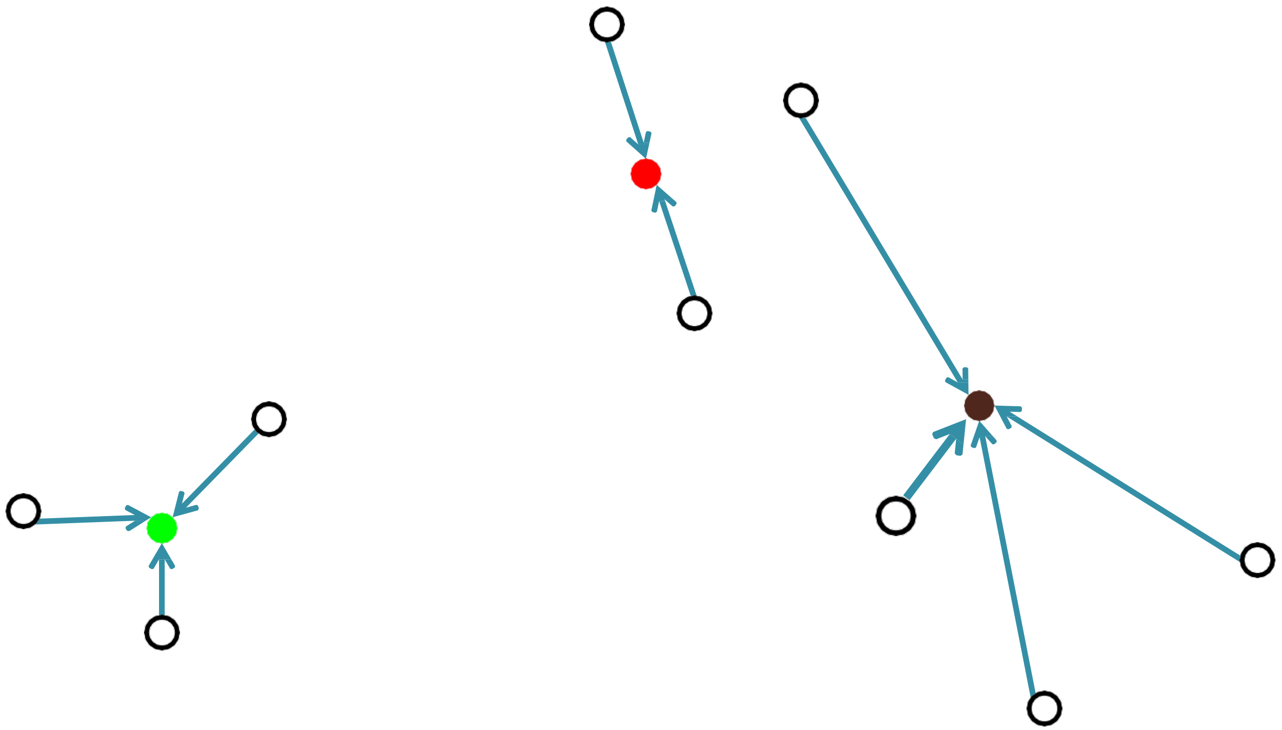
An Example

Assign each point to its nearest centroid



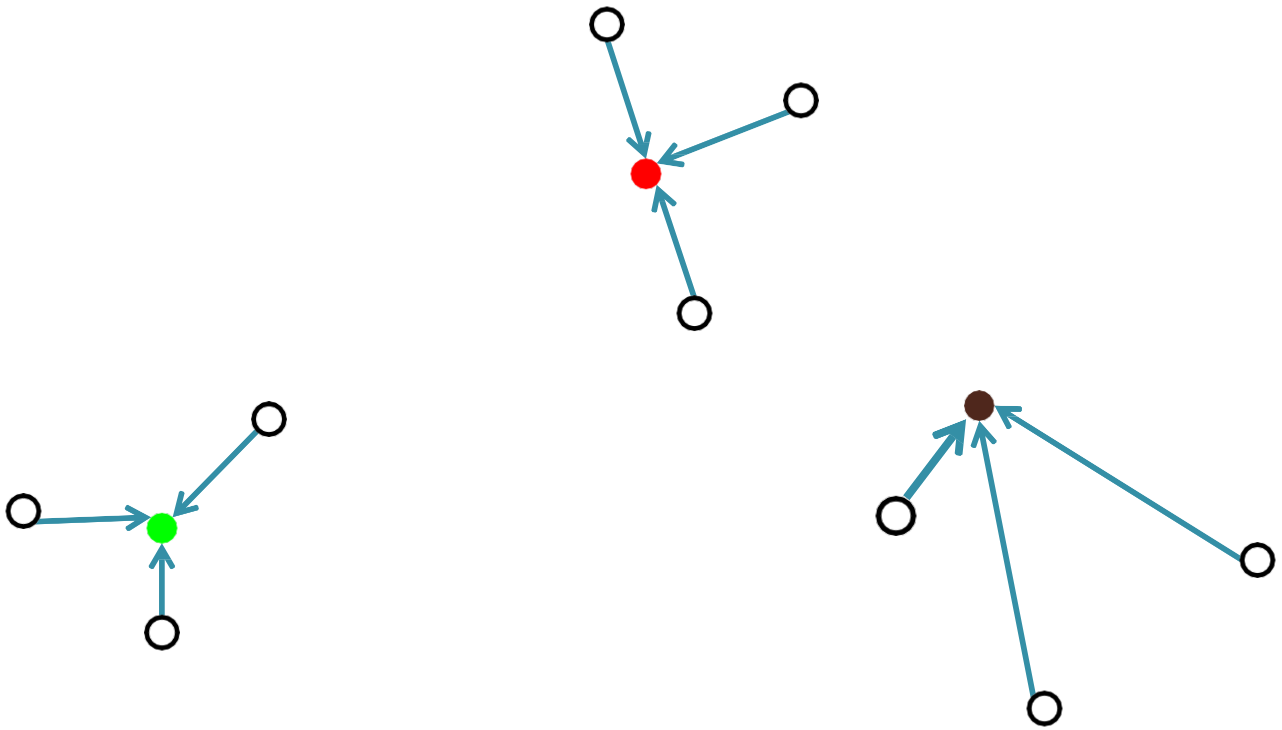
An Example

Recompute centroids as the ***mean*** of the points in that cluster



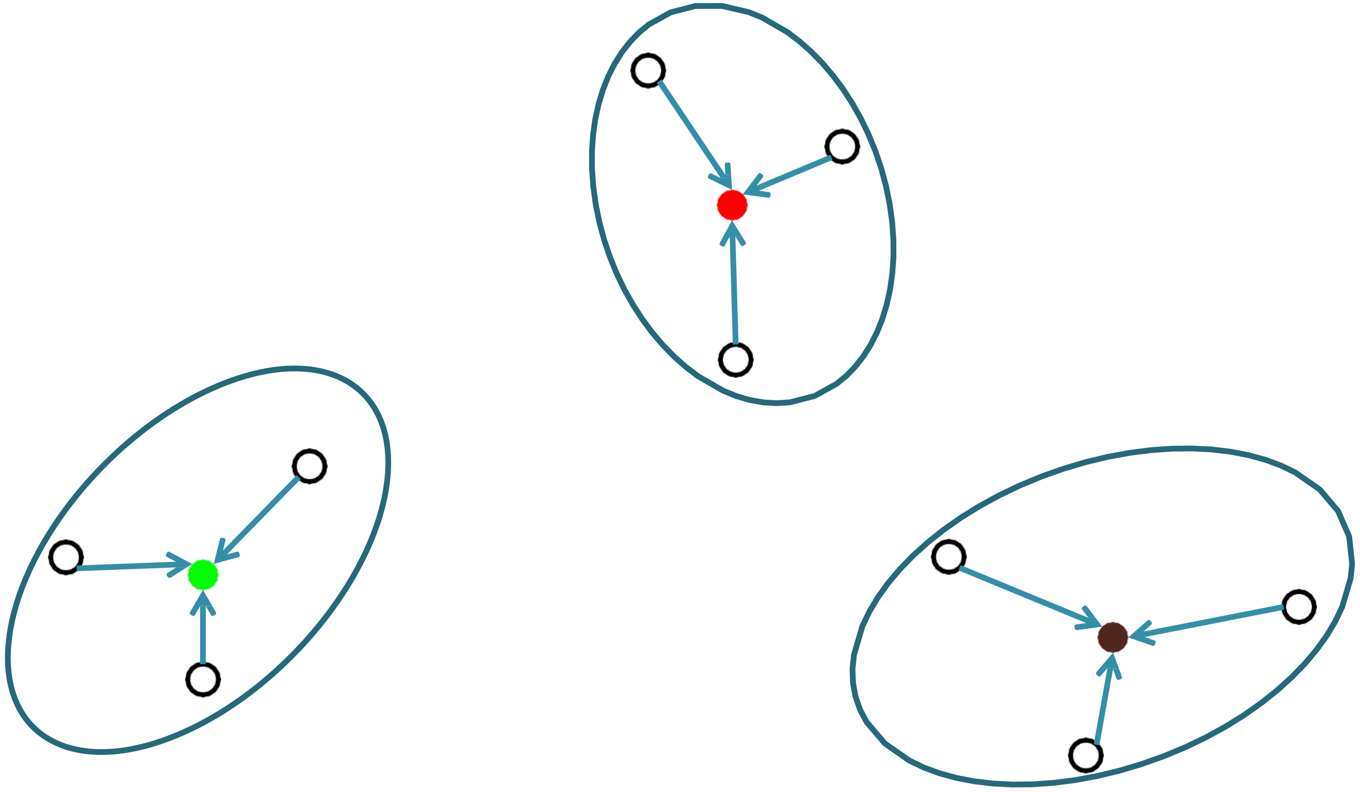
An Example

Assign each point to its nearest centroid



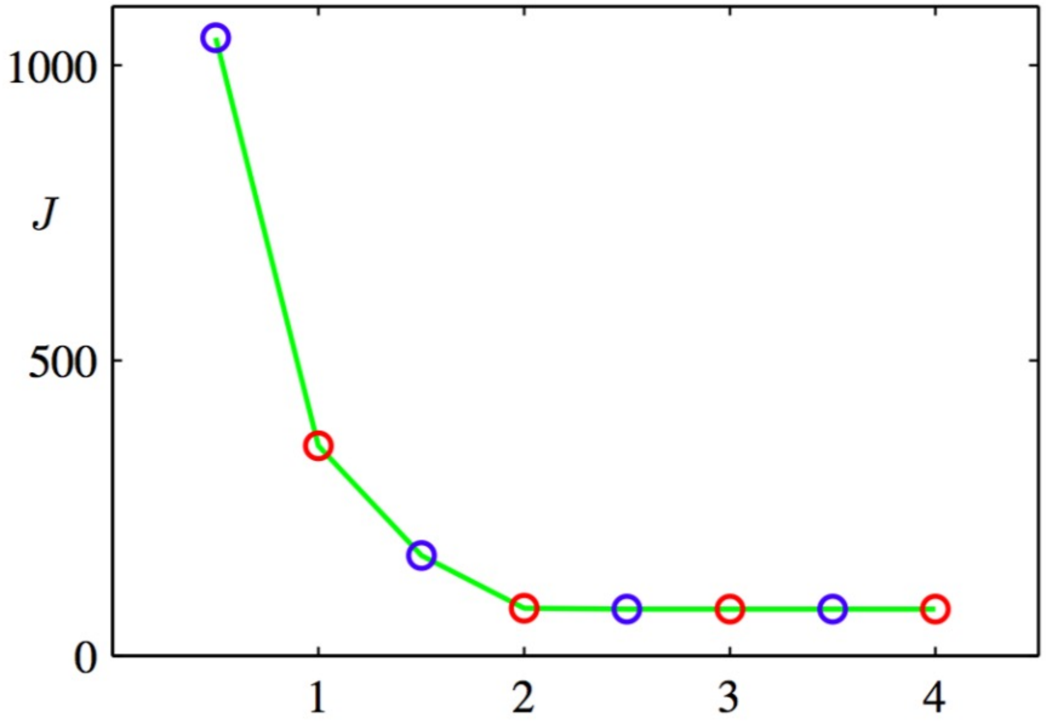
An Example

Recompute centroids as the ***mean*** of the points in that cluster



A good clustering result in this example

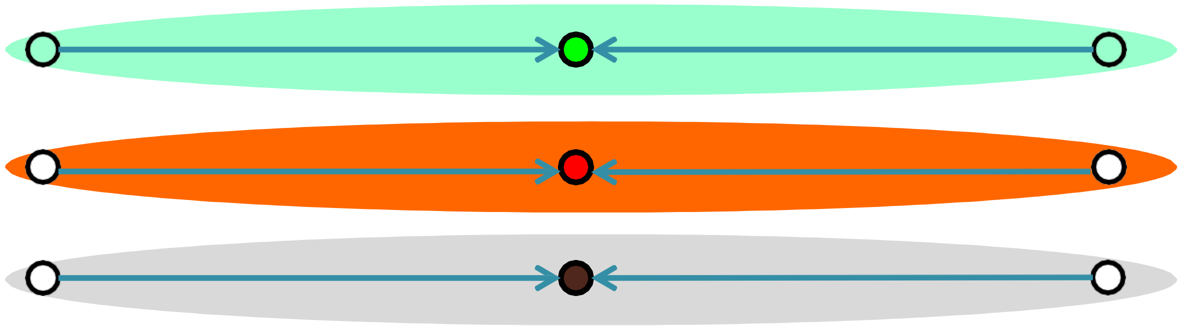
Loss Converges



Blue circles: Assigning each data point to a cluster

Red circles : Recomputing the cluster centroids

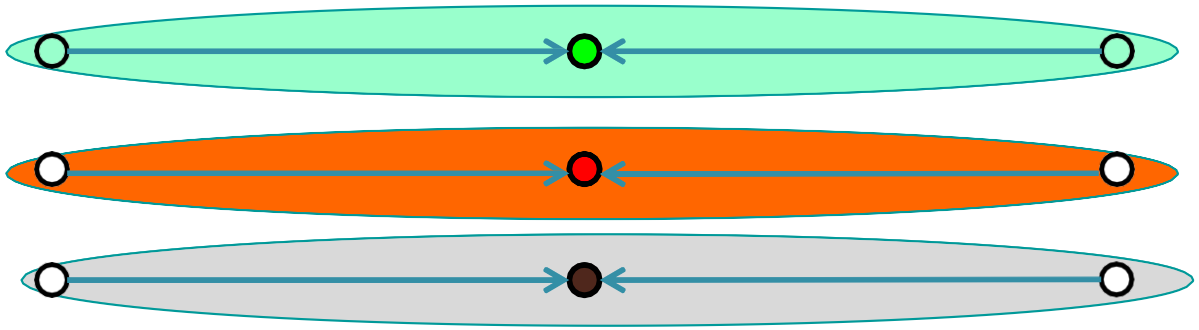
Another Example



Local optimum: every point is assigned to its nearest centroid and every centroid is the mean value of points belonging to this centroid

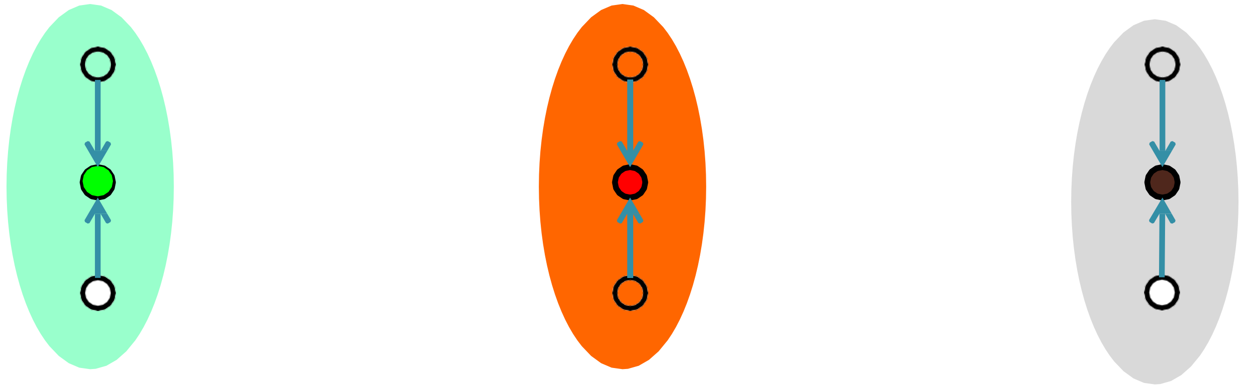
Converge!!!

Bad Local Minimum

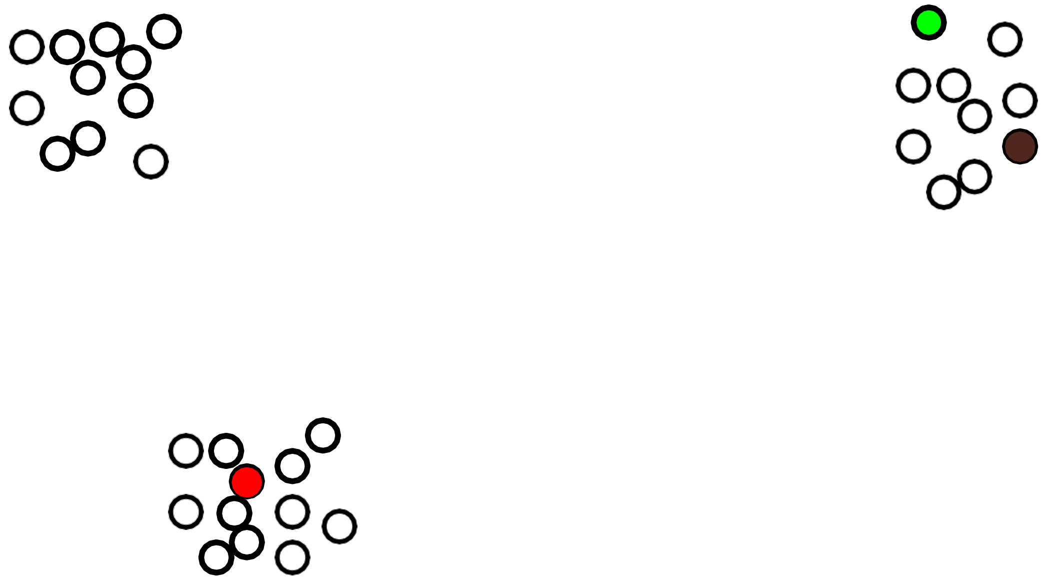


Much worse than the optimum solution

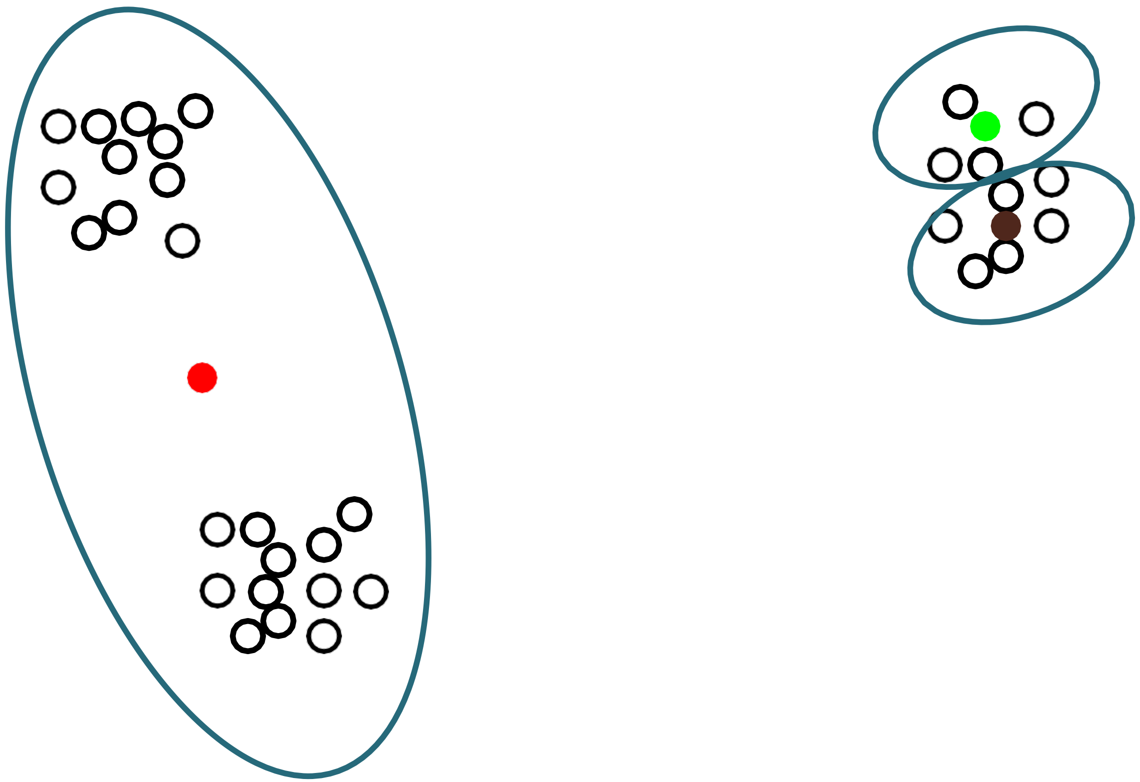
Optimal solution



Bad Local Minimum:
Even in Well Separated Clusters



Bad Local Minimum: Even in Well Separated Clusters



Analysis

- **Random initialization**: as K increases, ***less likely*** to perfectly pick one centroid per cluster

- **Analysis**: For K equal-sized clusters, the probability that each initial centroid is in a different cluster is

$$\frac{K!}{K^K} \approx \frac{1}{e^K}$$

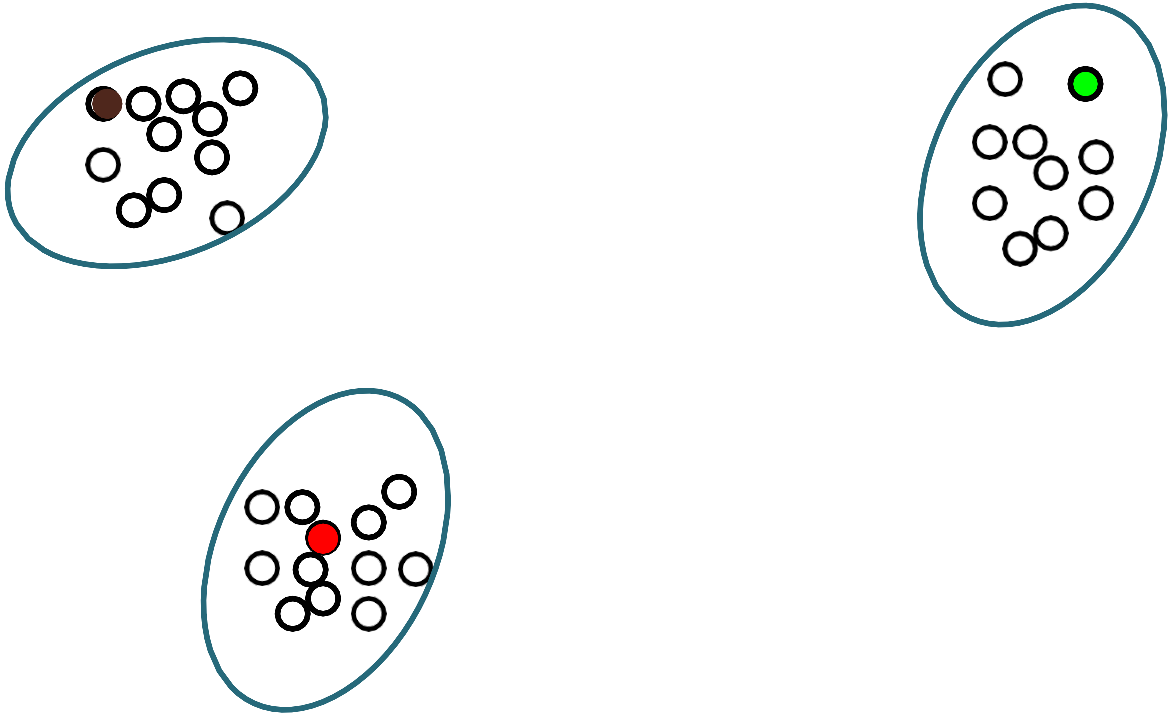
Becomes unlikely as K becomes large

Furthest Point Initialization

- Choose μ_1 at random

- For $k = 2, \ldots, K$
  - Pick μ_k among data points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ that is ***farthest*** from previously chosen $\mu_1\ \mu_2, \ldots, \mu_{k-1}$

Furthest Point Heuristic DOES Well



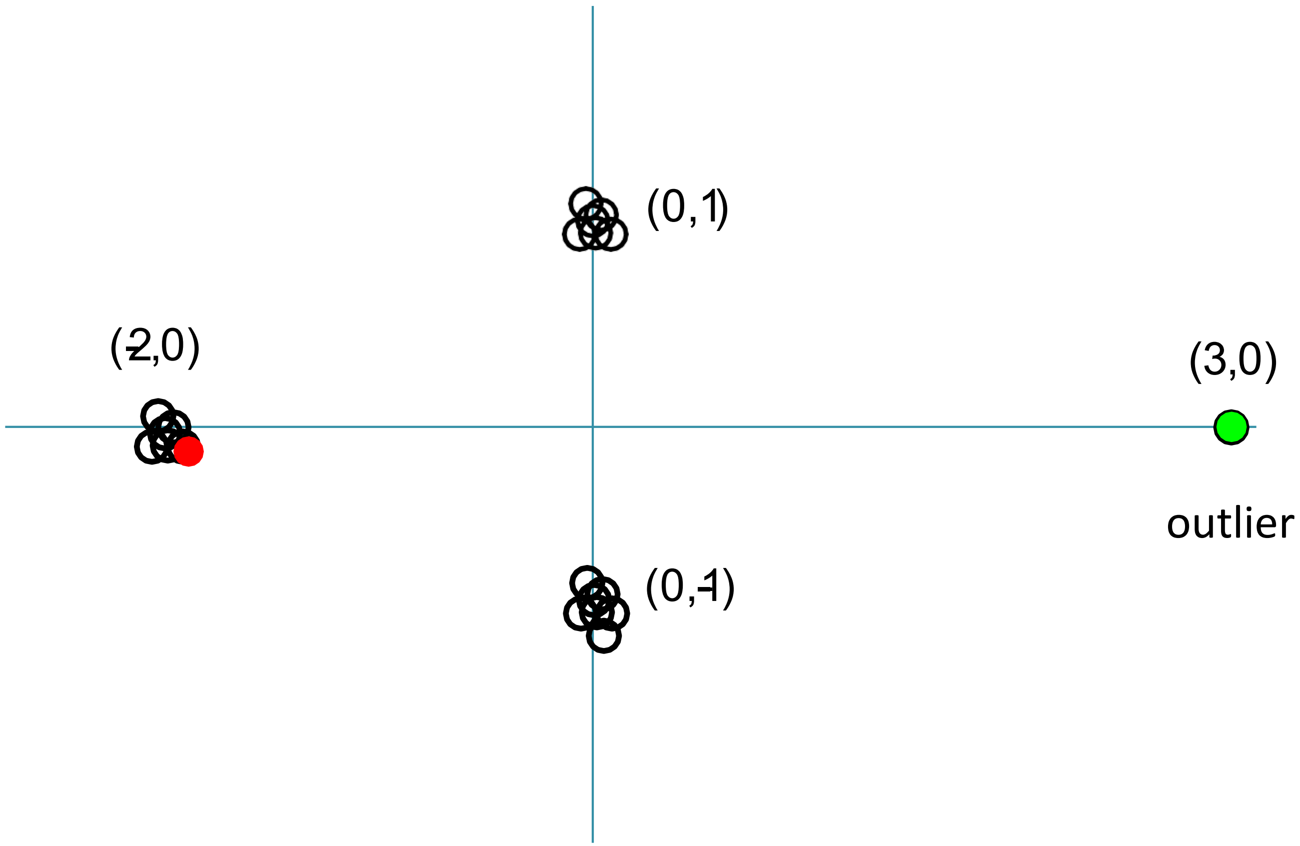
Fixes the issue in the previous example, but…

Furthest Point Heuristic is Sensitive to Outliers

Assume K=3

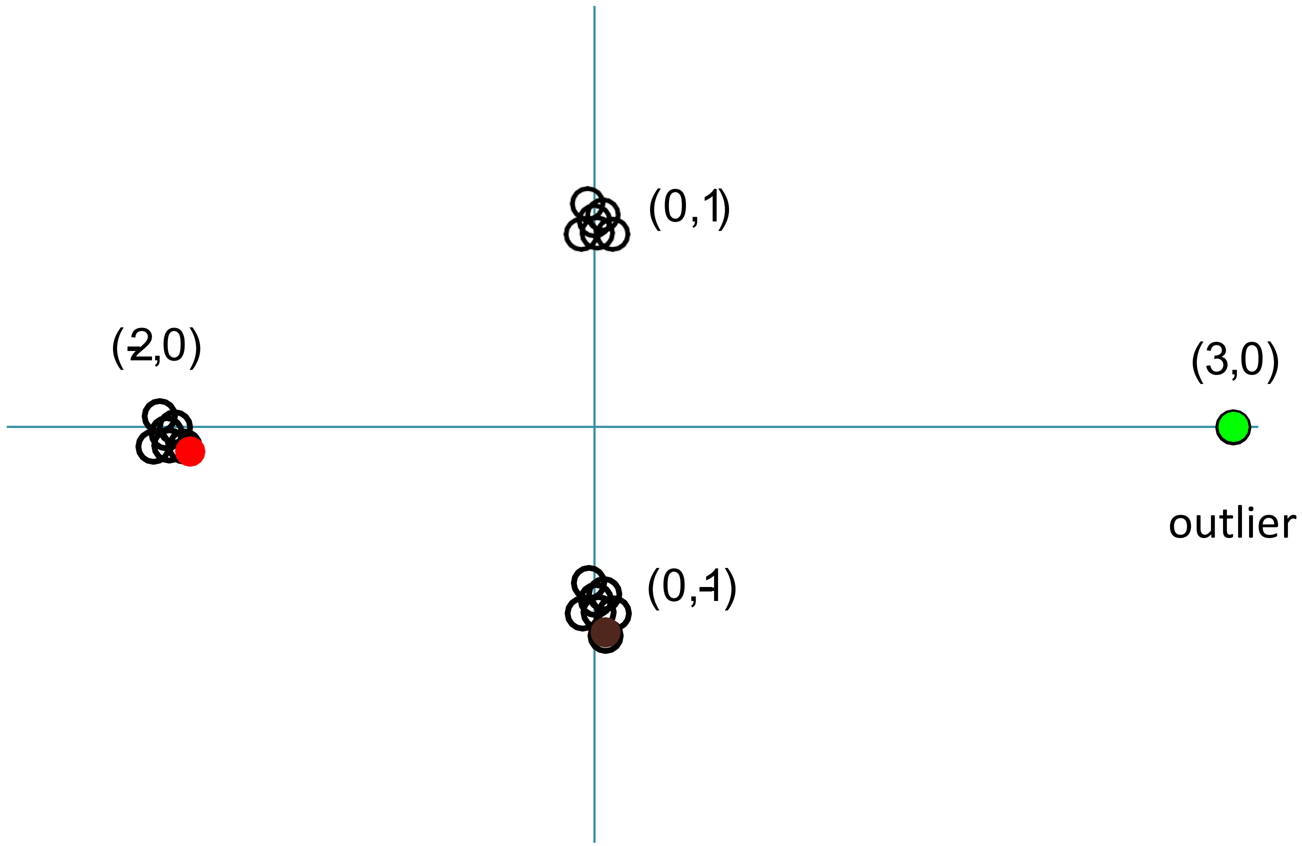
Furthest Point Heuristic is Sensitive to Outliers

Assume K=3



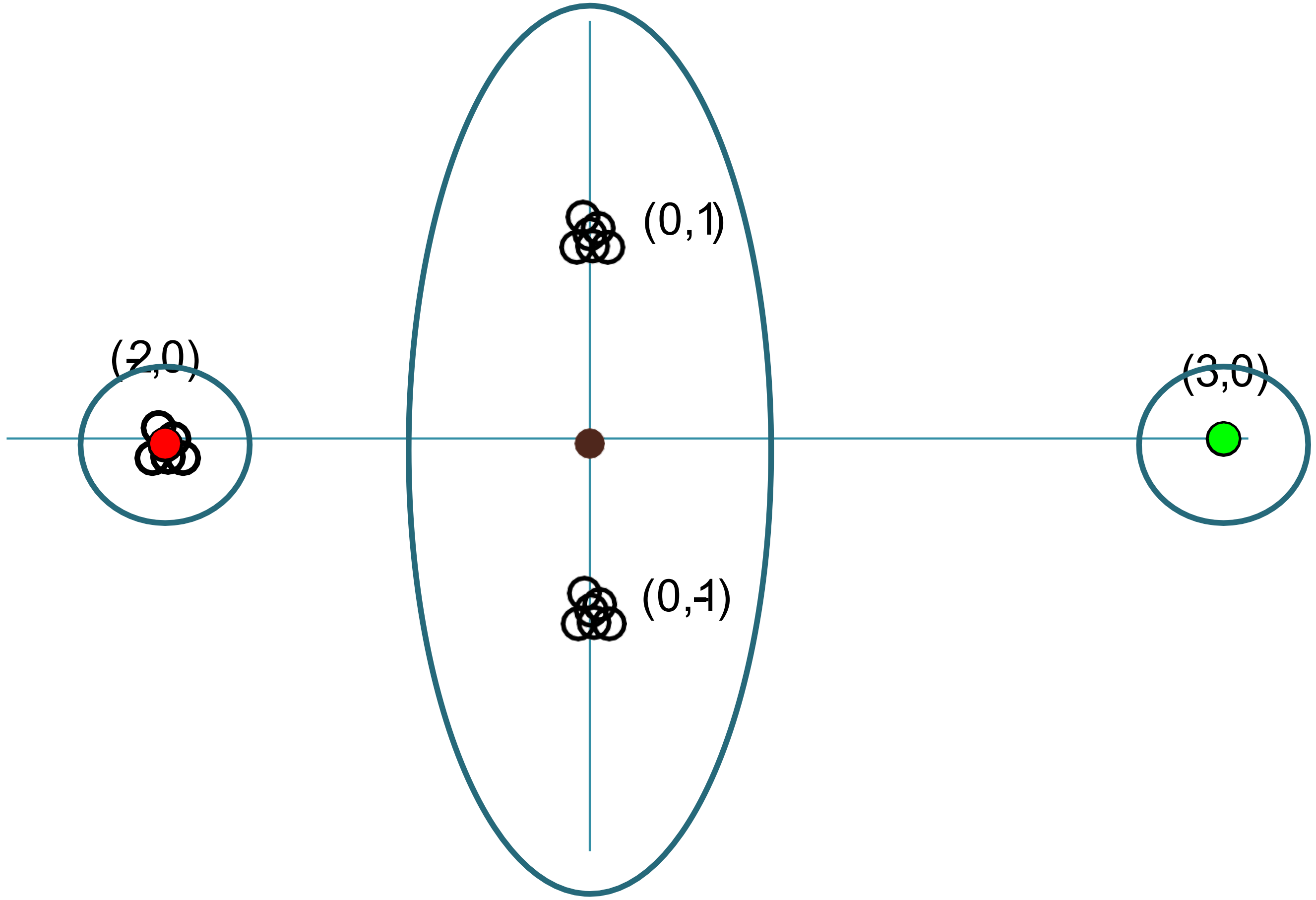
Furthest Point Heuristic is Sensitive to Outliers

Assume K=3



Furthest Point Heuristic is Sensitive to Outliers

Assume K=3



K-Means++ Clustering
(Arthur & Vassilvitskii, 2006)

K-Means++ Initialization: D^2 Sampling

- Interpolate between random and furthest point initialization

- Let $D(x)$ be the distance between a point $\boldsymbol{x}$ and its nearest centroid
  - **D^2 sampling**: chose the next centroid proportional to $D^2(x)$

- Choose μ_1 at random
- For $k = 2, \ldots, K$
  - Pick μ_k among $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ according to the distribution

$$p(\mu_k = \mathbf{x}_n) \propto \min_{j<k} \|\mathbf{x}_n - \mu_j\|_2^2 \qquad D^2(x_n)$$

D^d Sampling

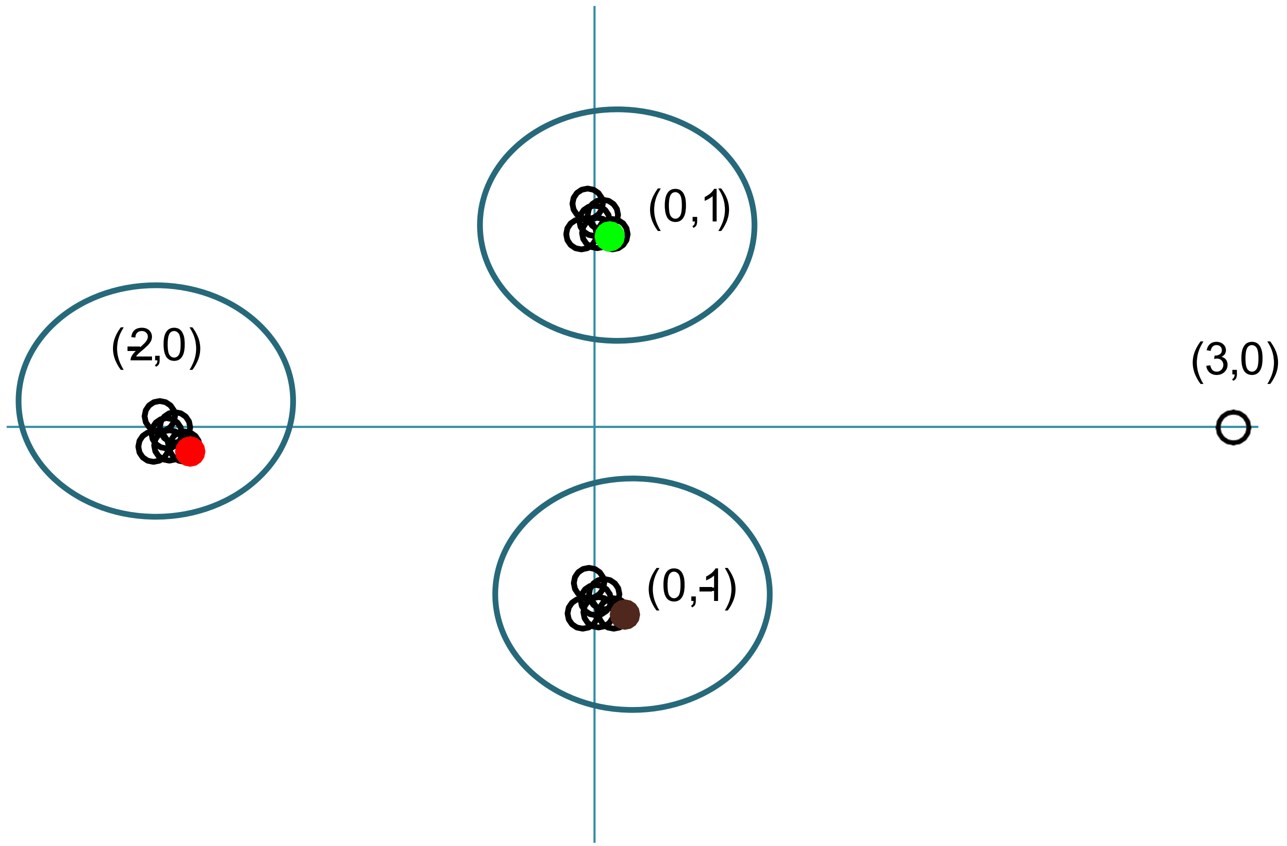
- Let $D(x)$ be the distance between a point ***x*** and its nearest centroid
    - Chose the next centroid proportional to $D^d(x)$

$$p(\mu_k = \mathbf{x}_n) \propto \min_{j<k} \|\mathbf{x}_n - \mu_j\|_2^d$$

- d=0, random sampling
- d=∞, furthest point initialization
- d=2, K-means++
- d=1, K-median

K-Means++ Fixes the Outliers

K-Means++ Fixes the Outliers



Theorem: K-means++ always attains an O(log K) approximation to optimal K-means solution in expectation

How to Choose K?

- **Gap statistics**: Find a large gap between the (K-1)-means loss and K-means loss

- **Cross-validation**: Partition data into two sets. Estimate centroids on one and use these to compute the loss on the other

- **Stability of clusters**: Measure the change in the clusters obtained by resampling or splitting the data
    - Choose K that leads to the most stable clustering result

- **Hierarchical clustering**

Summary

- Clustering
  - Unsupervised learning method

- K-Means clustering (Lloyd method)
  - Sensitive to initialized centroids

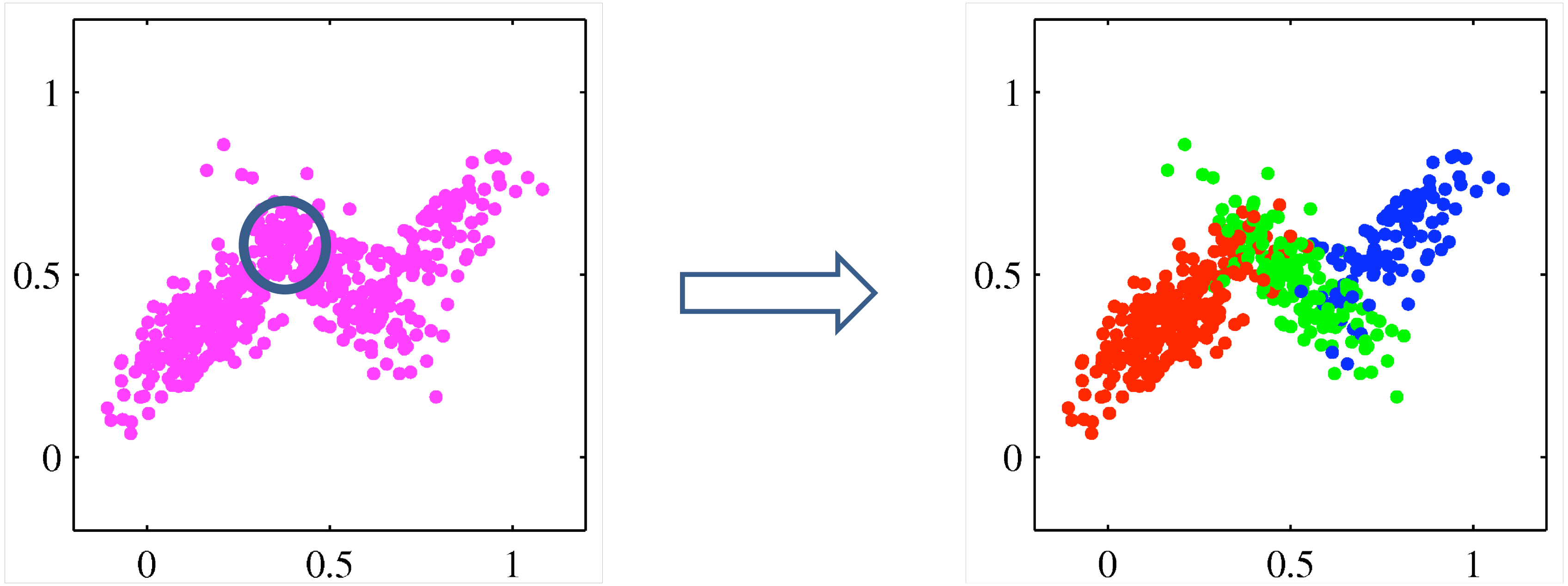
- Furthest point initialization
  - Sensitive to outliers

- K-Means++
  - Fix outlier issues
  - Theoretically guaranteed solution

Limitations of K-Means

Hard assignment

- Each data point is assigned to a cluster with 100% probability



Soft assignment

- Probability distribution over the cluster assignment

Next Lecture:
Gaussian Mixture Model (GMM)

Acknowledgement

Some slides are from

Matt Gormley (CMU)

https://www.cs.cmu.edu/~mgormley/courses/10601-s17/slides/lecture15-cluster.pdf