CS 4700: Foundations of Artificial Intelligence

ONLINE

Spring 2020
Prof. Haym Hirsh

Lecture 34
May 4, 2020

Naïve Bayes: Laplace Smoothing

Laplace (or "Additive") Smoothing

Estimate $P\left(\bar{x}_{test,j} = v_a \,|c_l\right)$ using $\frac{n_{jal}+\alpha}{N_l+\alpha|V_j|}$

α > 0: "smoothing parameter"

(Often $\alpha \leq 1$)

Naïve Bayes: Laplace Smoothing

Attributes from discrete sets: Assign label c

$$c = \underset{c_l \in C}{\operatorname{argmax}} \left[\frac{N_l}{N} \prod_{j=1}^{n} \frac{n_{j a_j l} + \alpha}{N_l + \alpha |V_j|} \right]$$

Binary attributes case: Assign label c

$$c = \underset{c_l \in C}{\operatorname{argmax}} \left[\frac{N_l}{N} \prod_{\mathrm{j}=1}^{\mathrm{n}} \left(\frac{n_{j1l} + \alpha}{N_l + \alpha |V_j|} \right)^{\mathrm{x_j}} \left(1 - \frac{n_{j1l} + \alpha}{N_l + \alpha |V_j|} \right)^{1 - \mathrm{x}_j} \right]$$

Naïve Bayes: Laplace Smoothing

Attributes from discrete sets: Assign label c

$$c = \underset{c_l \in C}{\operatorname{argmax}} \left[\frac{N_l}{N} \prod_{j=1}^{n} \frac{n_{j a_j l} + \alpha}{N_l + \alpha |V_j|} \right]$$

Binary attributes case: Assign label c

$$c = \underset{c_l \in C}{\operatorname{argmax}} \left[\frac{N_l}{N} \prod_{\mathrm{j}=1}^{\mathrm{n}} \left(\frac{n_{j1l} + \alpha}{N_l + 2\alpha} \right)^{\mathrm{x_j}} \left(1 - \frac{n_{j1l} + \alpha}{N_l + 2\alpha} \right)^{1 - \mathrm{x}_j} \right]$$

$|V_j|=2$

Machine Learning

- Most common approaches:
  - Supervised learning
  - Unsupervising learning
  - Reinforcement learning

Machine Learning

- Most common approaches:
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning ✓ - Chapters 17 and 22

Machine Learning

- Most common approaches:
  - Supervised learning ✓ - Chapter 19
  - Unsupervised learning
  - Reinforcement learning ✓ - Chapters 17 and 22

Machine Learning

- Most common approaches:
  - Supervised learning ✓ - Chapter 19
    Chapter 21: Deep Learning
  - Unsupervised learning
  - Reinforcement learning ✓ - Chapters 17 and 22

Machine Learning

- Most common approaches:
  - Supervised learning ✓ - Chapter 19
    Chapter 21: Deep Learning
    https://en.wikipedia.org/wiki/Backpropagation
  - Unsupervised learning
  - Reinforcement learning ✓ - Chapters 17 and 22

Machine Learning

- Most common approaches:
  - Supervised learning ✓ - Chapter 19
    Chapter 21: Deep Learning
    https://en.wikipedia.org/wiki/Backpropagation
  - Unsupervised learning – Dimensionality reduction
  - Reinforcement learning ✓ - Chapters 17 and 22

Machine Learning

- Most common approaches:
  - Supervised learning ✓ - Chapter 19
    Chapter 21: Deep Learning
    https://en.wikipedia.org/wiki/Backpropagation
  - Unsupervised learning – ~~Dimensionality reduction~~
  - Reinforcement learning ✓ - Chapters 17 and 22

Machine Learning

- Most common approaches:
  - Supervised learning ✓ - Chapter 19
    Chapter 21: Deep Learning
    https://en.wikipedia.org/wiki/Backpropagation
  - Unsupervised learning – Clustering
  - Reinforcement learning ✓ - Chapters 17 and 22

Clustering

(Somewhat) more formally:

Given:

- a set D of N examples D=$\{\overline{x_1}, \overline{x_2}, \ldots, \bar{x}_N\}$
  where each $\overline{x_i}$ is in some $\mathcal{X}$ (often $\mathcal{X}$ = R^n)

Find:

- k sets $\{S_1, \ldots, S_k)$ such that each $S_i \subseteq D$ and $\bigcup_1^k S_i = D$
  (often: and for $1 \leq i \neq j \leq k$, $S_i \cap S_j = \emptyset$)
  where $\sum_{i=1}^{k} \sum_{\overline{x_1}, \overline{x_2} \epsilon S_i} D(\overline{x_1}, \overline{x_2})$ is "small" and
  $\sum_{1 \leq i \neq j \leq k} \sum_{\overline{x_1} \epsilon S_i, \overline{x_2} \epsilon S_j} D(\overline{x_1}, \overline{x_2})$ is "big"

Assign items in D to k (possibly disjoint) subsets of D so that items in the same set are "close" and items in two different sets are "far"

Machine Learning

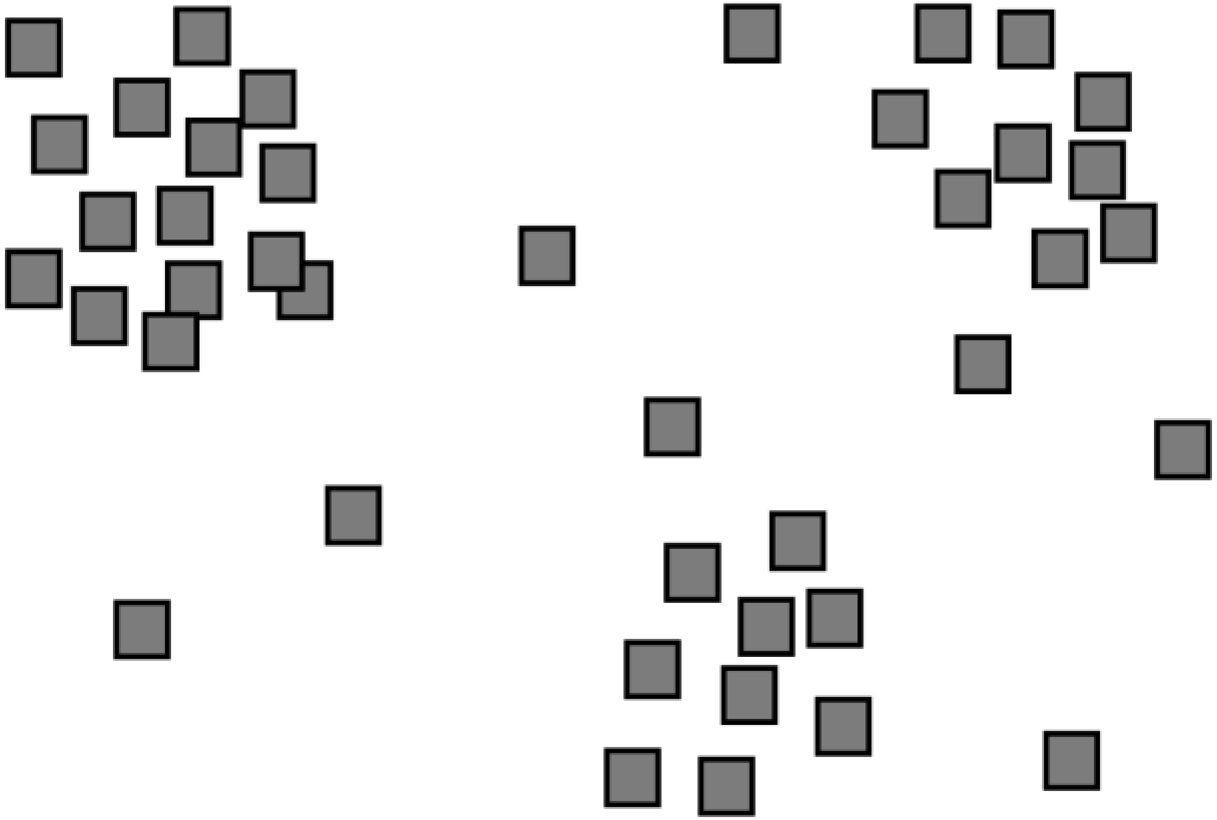
- Most common approaches:
  - Supervised learning ✓ - Chapter 19
    Chapter 21: Deep Learning
    https://en.wikipedia.org/wiki/Backpropagation
  - Unsupervised learning – Clustering
    https://en.wikipedia.org/wiki/K-means_clustering
  - Reinforcement learning ✓ - Chapters 17 and 22

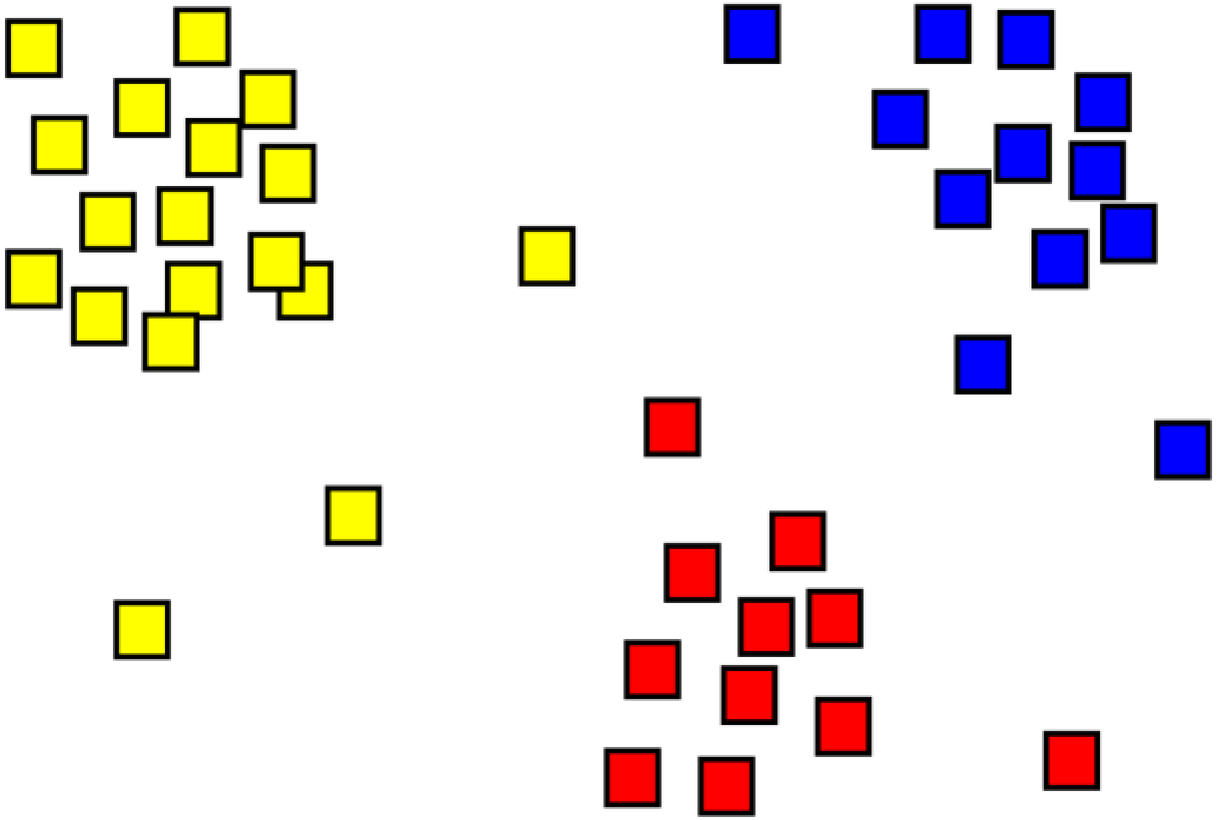
Looking Ahead

- Wednesday, May 6: Natural Language Processing
  Prof. Claire Cardie

- Friday, May 8: Computer Vision
  Grant Van Horn, Ph.D. (Lab of O)

- Monday, May 11: Societal implications of AI
  - What should you worry about



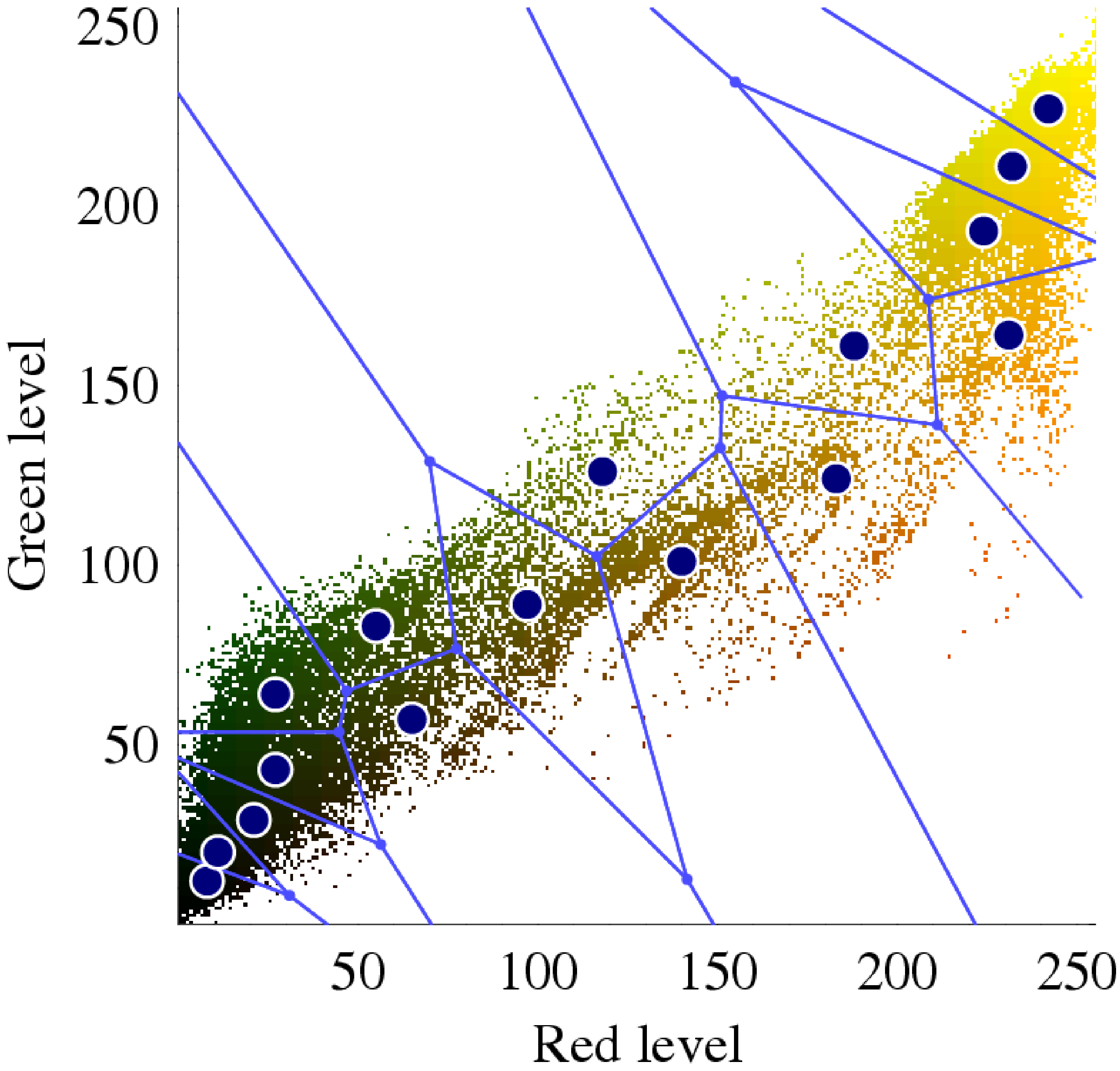


Clustering

- Reflects a human cognitive skill that we might like machines to emulate

Clustering

- Reflects a human cognitive skill that we might like machines to emulate
- Useful tool in image processing, machine learning, data analytics (“cluster analysis”), …



250
200
150
100
50
Green level
50
100
150
200
250
Red level

Sample Applications

- Image processing – color encodings, image representations
- Market analysis – partition customers into coherent segments
- E-commerce – product categories, user categories
- Social network analysis – identify “communities”
- Natural language processing – lexical ambiguity
- Evolutionary biology – inferring phylogenetic trees
- Linguistics – inferring language families

Facebook stops letting advertisers target people interested in 'pseudoscience'

The move comes after an investigation found that Facebook was allowing advertisers to target ads at 78 million users interested in the topic.

Ry Crist April 23, 2020 10:49 a.m. PT

2

LISTEN - 00:59

Clustering

- Partition points into k sets such that
  - Any two points in the same set are similar
  - Any two points in different sets are dissimilar

Clustering

- Partition points into k sets such that
  - Any two points in the same set are similar
  - Any two points in different sets are dissimilar
- k can be an external parameter or a value determined by the algorithm

Clustering

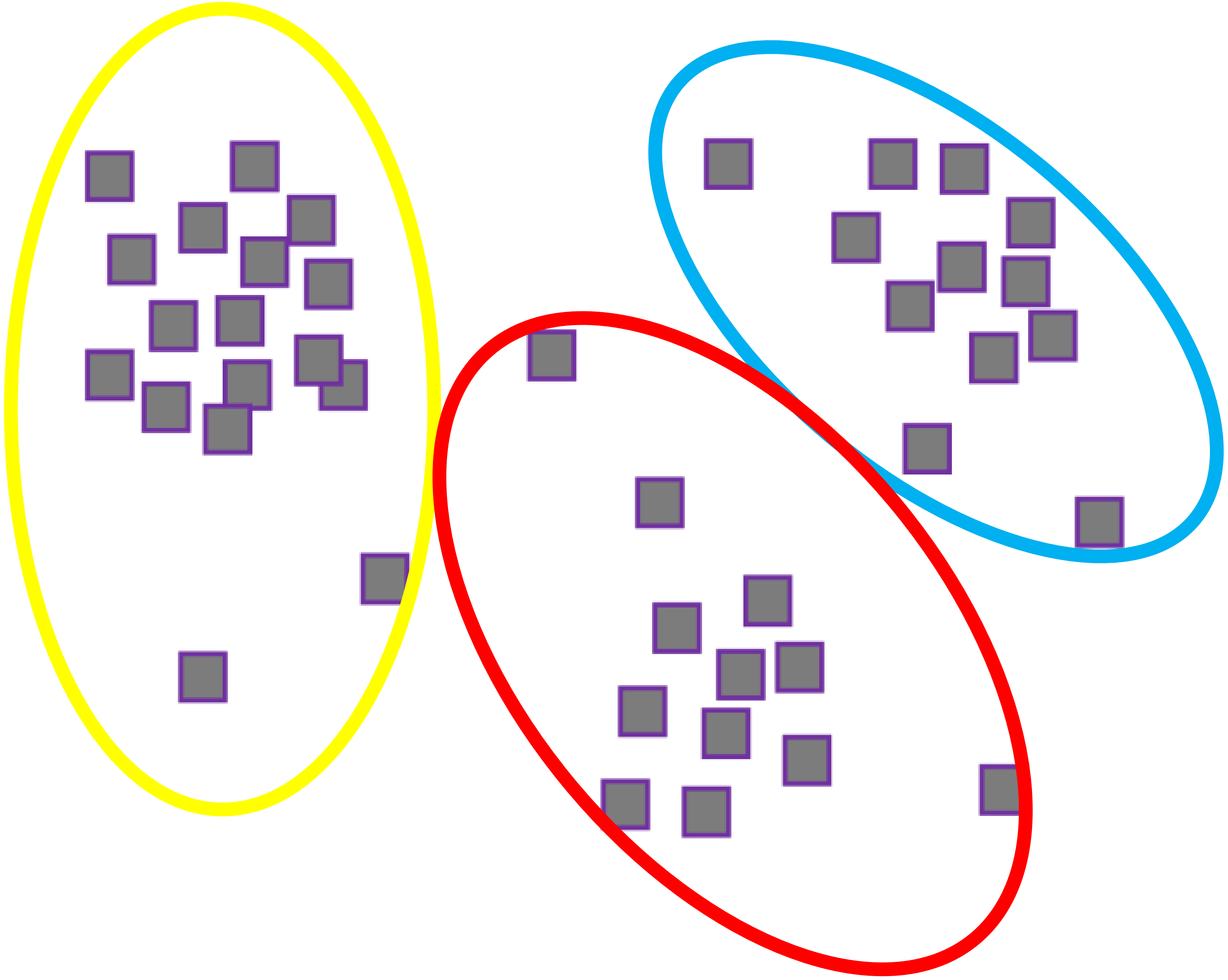
- Partition points into k sets such that
  - Any two points in the same set are similar
  - Any two points in different sets are dissimilar
- k can be an external parameter or a value determined by the algorithm
- Partitions can be categorical (in/out) (“hard clustering”) or probabilistic (each point is in each cluster with some weight, with the weights summing to 1) (“soft clustering”)

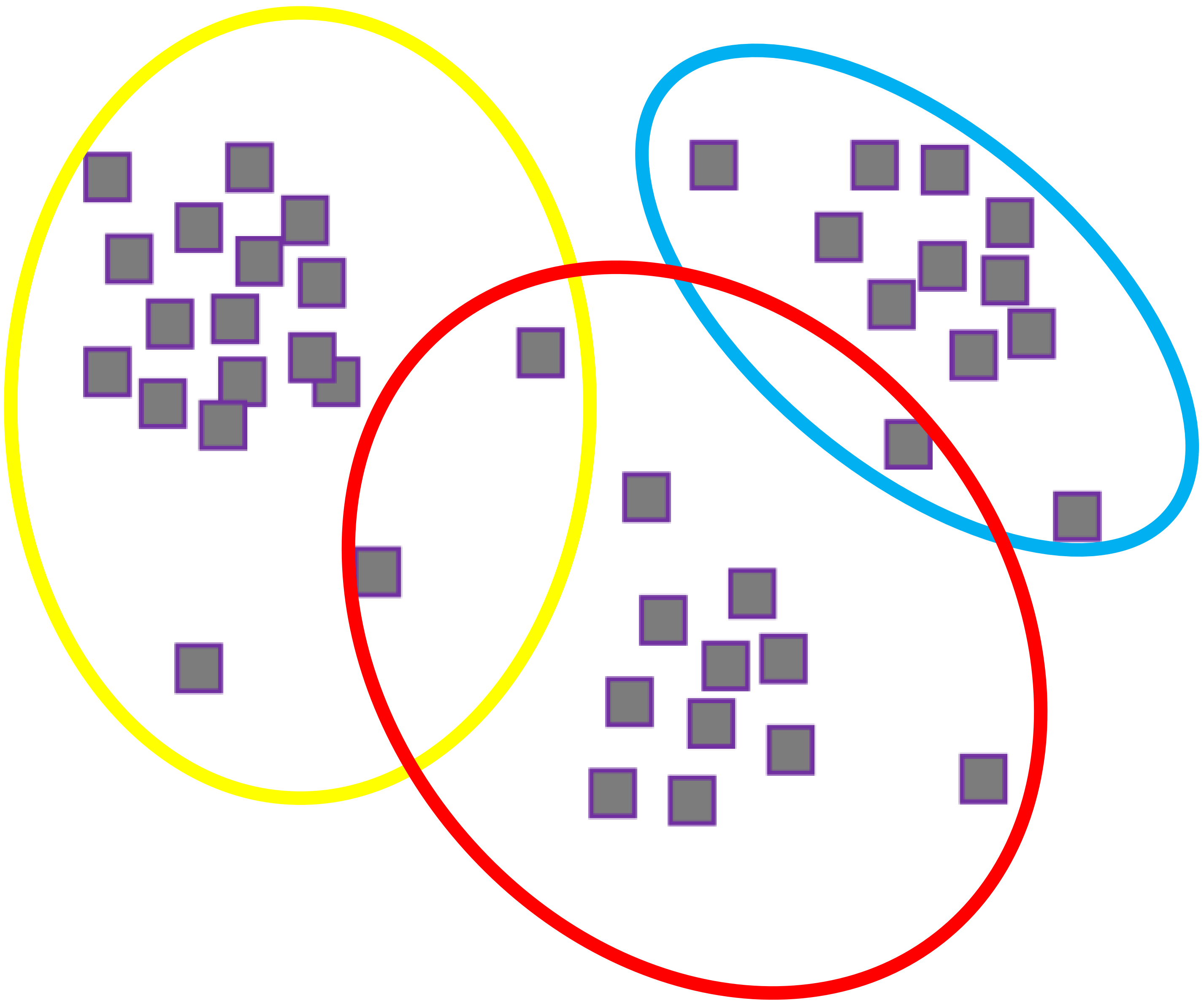
Clustering

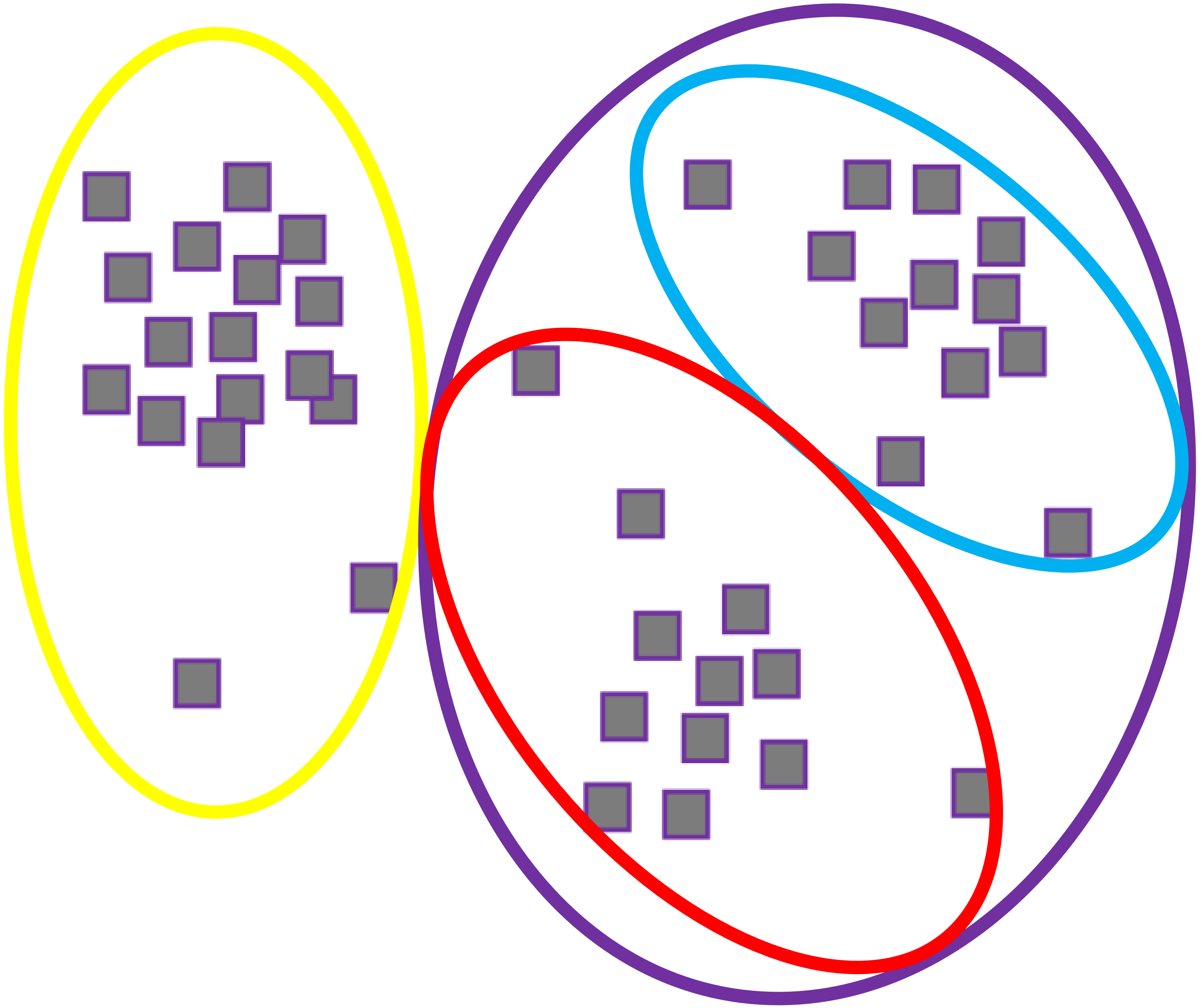
- Partition points into k sets such that
  - Any two points in the same set are similar
  - Any two points in different sets are dissimilar
- k can be an external parameter or a value determined by the algorithm
- Partitions can be categorical (in/out) (“hard clustering”) or probabilistic (each point is in each cluster with some weight, with the weights summing to 1) (“soft clustering”)
- Clusters can be disjoint, overlapping, or hierarchical

Clustering

- Partition points into k sets such that
  - Any two points in the same set are similar
  - Any two points in different sets are dissimilar
- k can be an external parameter or a value determined by the algorithm
- Partitions can be categorical (in/out) (“hard clustering”) or probabilistic (each point is in each cluster with some weight, with the weights summing to 1) (“soft clustering”)
- Clusters can be disjoint, overlapping, or hierarchical
- Typically presumes the existence of a measure D($\overline{x_1}$, $\overline{x_2}$) giving a “distance” between two points D($\overline{x_1}$, $\overline{x_2}$) – such as Euclidean distance







Clustering

- Partition points into k sets such that
  - Any two points in the same set are similar
  - Any two points in different sets are dissimilar
- k can be an external parameter or a value determined by the algorithm
- Partitions can be categorical (in/out) (“hard clustering”) or probabilistic (each point is in each cluster with some weight, with the weights summing to 1) (“soft clustering”)
- Clusters can be disjoint, overlapping, or hierarchical
- Typically presumes the existence of a measure D($\overline{x_1}$, $\overline{x_2}$) giving a “distance” between two points D($\overline{x_1}$, $\overline{x_2}$) – such as Euclidean distance

Clustering

(Somewhat) more formally:

Given:

- a set D of N examples D={$\overline{x_1}$, $\overline{x_2}$, …, $\bar{x}_N$ }

  where each $\overline{x_i}$ is in some $\mathcal{X}$ (often $\mathcal{X}$ = R^n)

Find:

- k sets {S_1, …, S_k} such that each $S_i \subseteq$ D and $\bigcup_1^k S_i$ = D

  (often: and for $1 \leq i \neq j \leq k$, $S_i \cap S_j = \emptyset$)

  where $\sum_{i=1}^{k} \sum_{\overline{x_1}, \overline{x_2} \epsilon S_i} D(\overline{x_1}, \overline{x_2})$ is "small" and

  $\sum_{1 \leq i \neq j \leq k} \sum_{\overline{x_1} \epsilon S_i, \overline{x_2} \epsilon S_j} D(\overline{x_1}, \overline{x_2})$ is "big"

Clustering

(Somewhat) more formally:

Given:

- a set D of N examples D={$\overline{x}_1$, $\overline{x}_2$, …, $\overline{x}_N$ }

  where each $\overline{x}_i$ is in some $\mathcal{X}$ (often $\mathcal{X}$ = R^n)

Find:

- k sets {S_1, …, S_k) such that each $S_i \subseteq D$ and $\bigcup_1^k S_i = D$

  (often: and for $1 \leq i \neq j \leq k$, $S_i \cap S_j = \emptyset$)

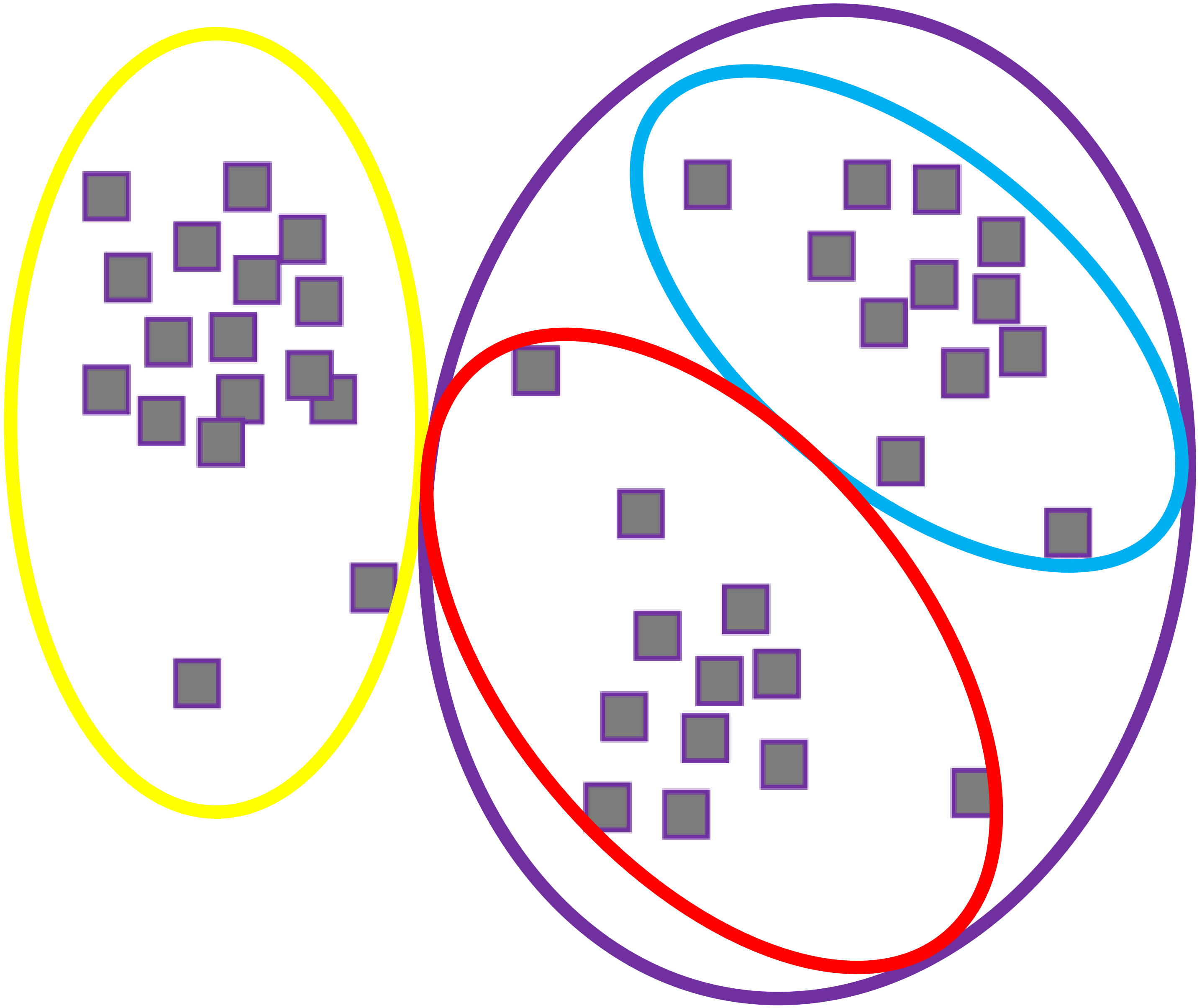
  where $\sum_{i=1}^{k} \sum_{\overline{x_1},\overline{x_2} \epsilon S_i} D(\overline{x_1}, \overline{x_2})$ is “small” and

  $\sum_{1 \leq i \neq j \leq k} \sum_{\overline{x_1} \epsilon S_i,\, \overline{x_2} \epsilon S_j} D(\overline{x_1}, \overline{x_2})$ is “big”

Assign items in D to k (possibly disjoint) subsets of D so that items in the same set are “close” and items in two different sets are “far”

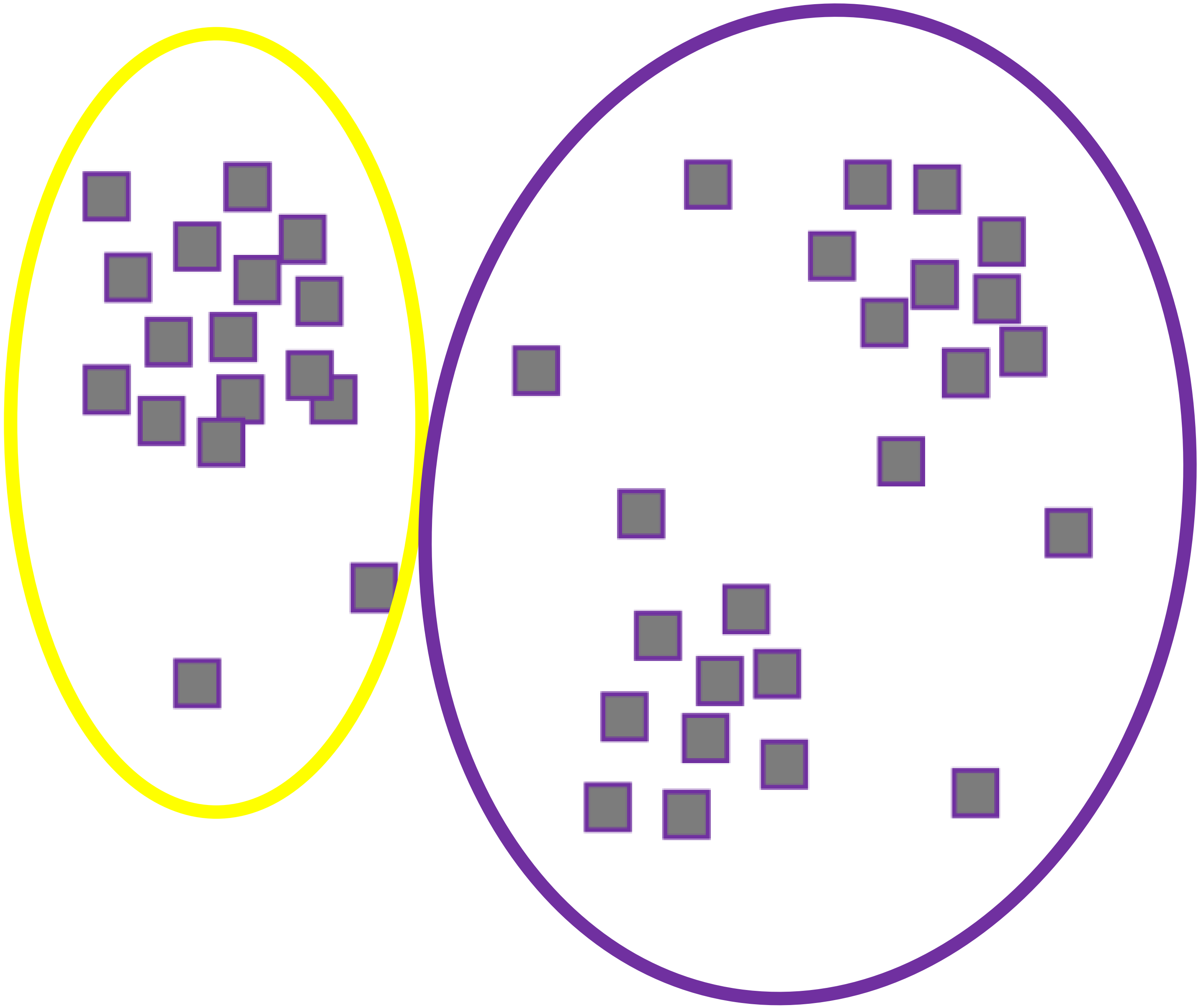
Clustering Approaches

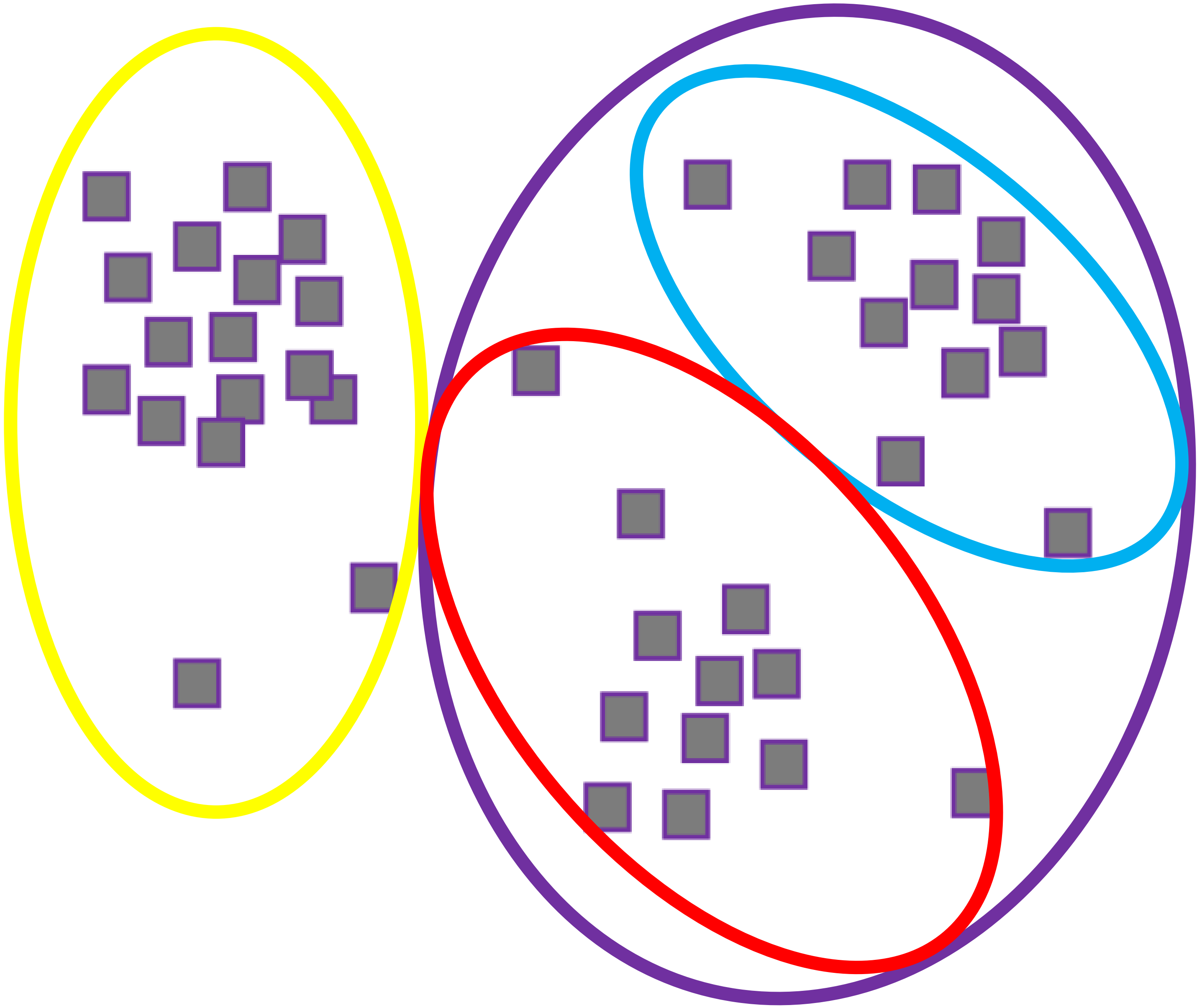
- Hierarchical



Clustering Approaches

- Hierarchical
  - Top down



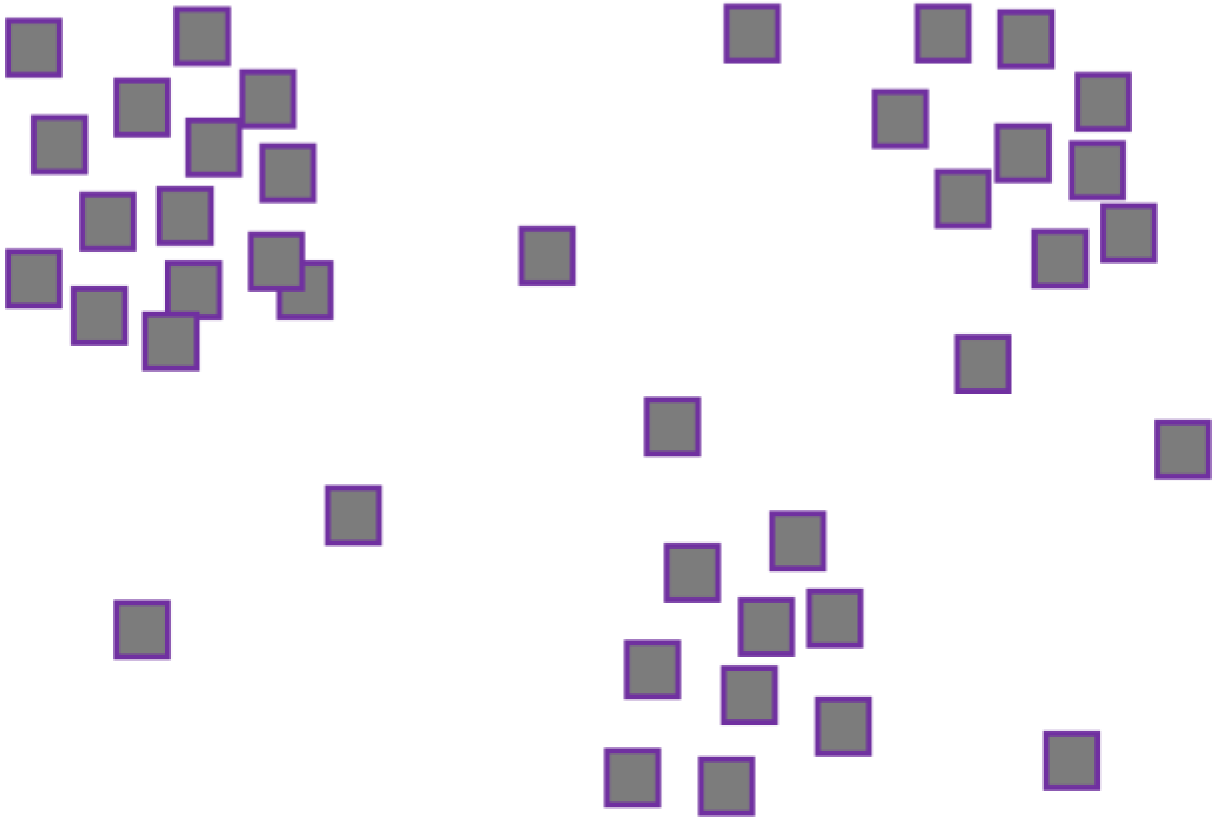


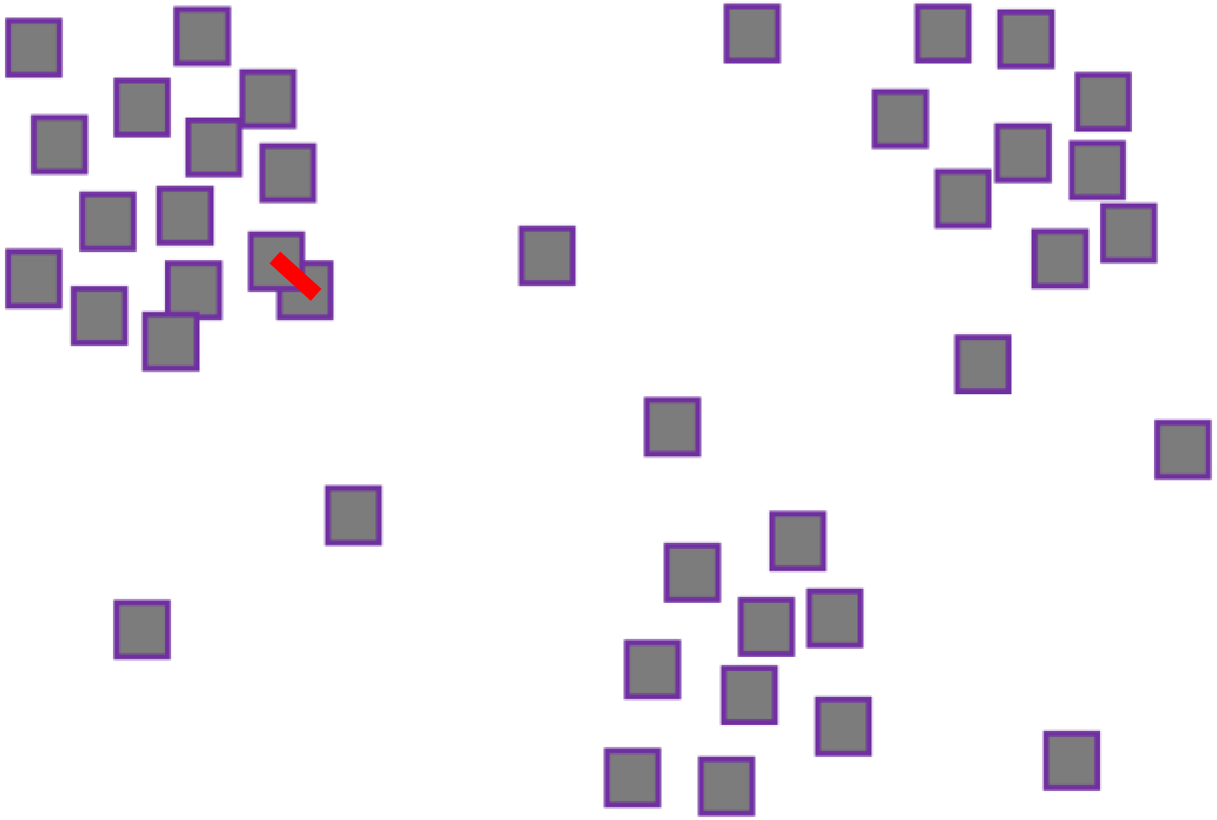
Clustering Approaches

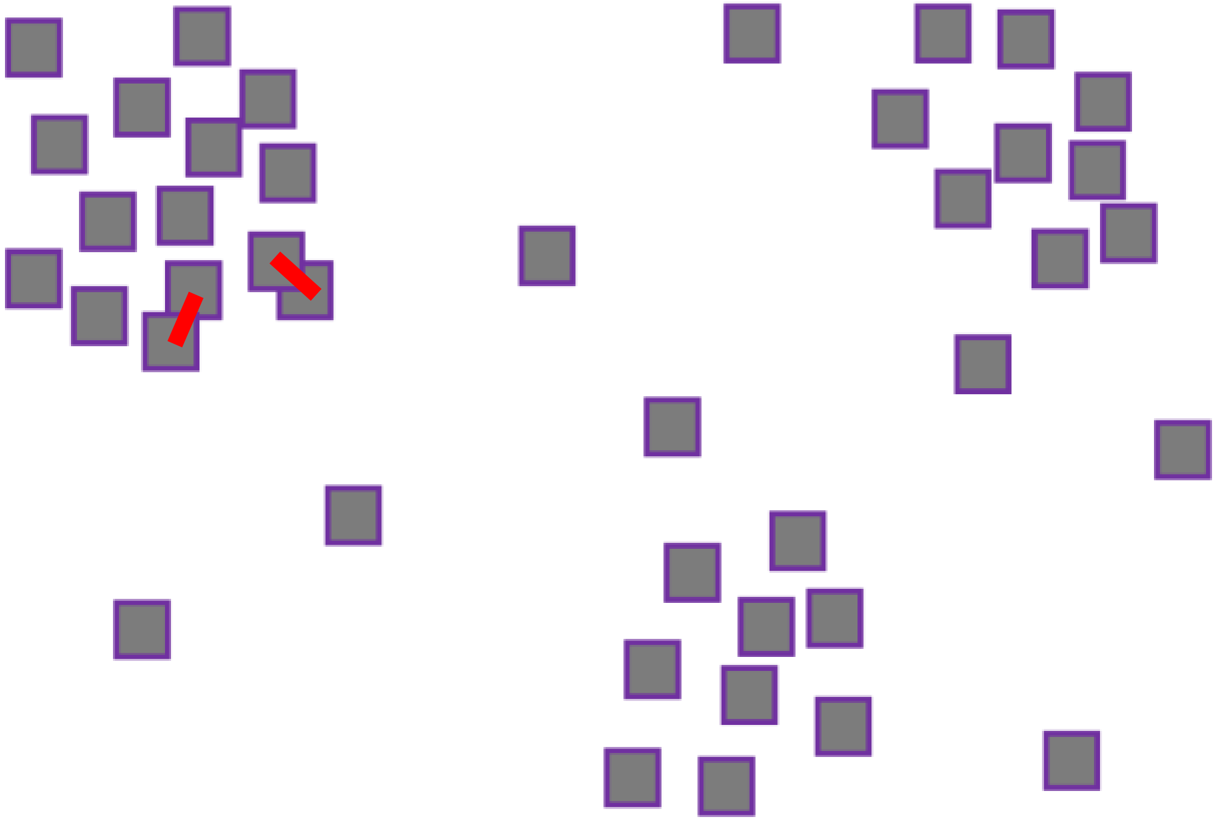
- Hierarchical
  - Top down

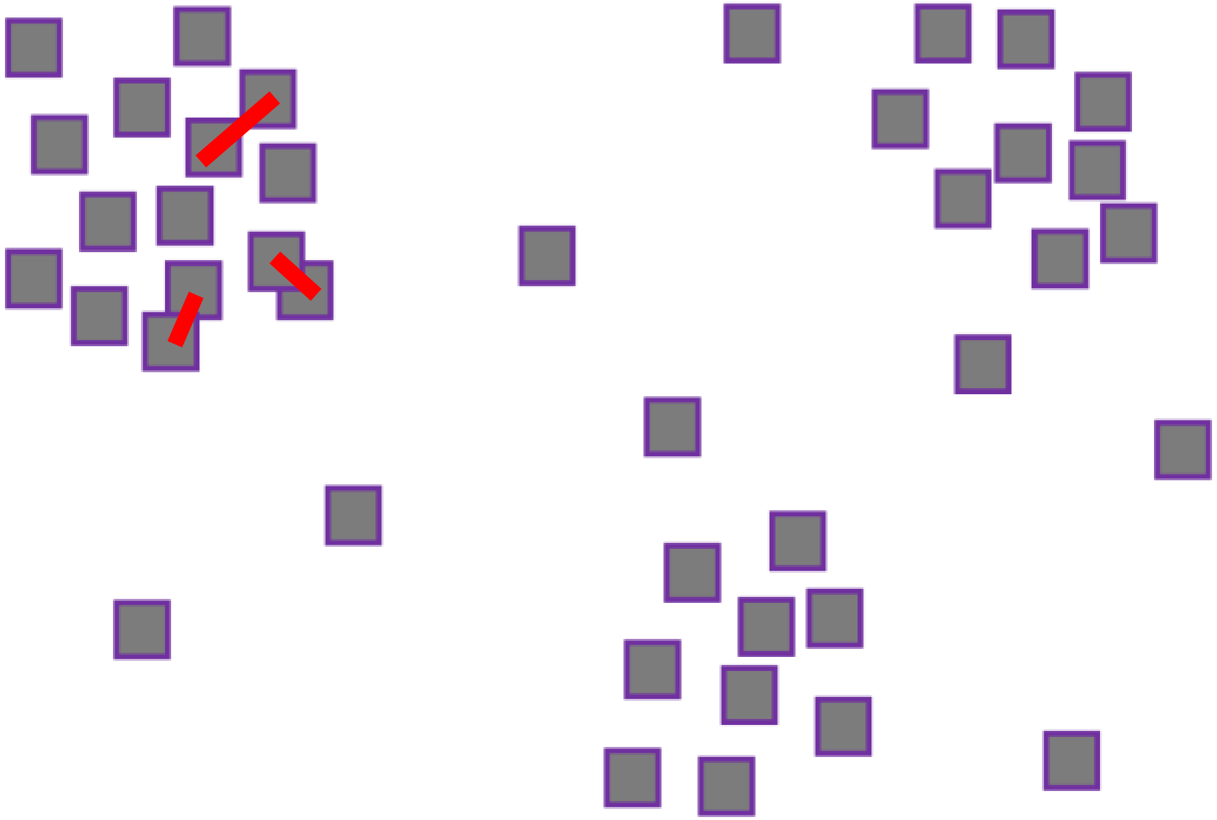
Clustering Approaches

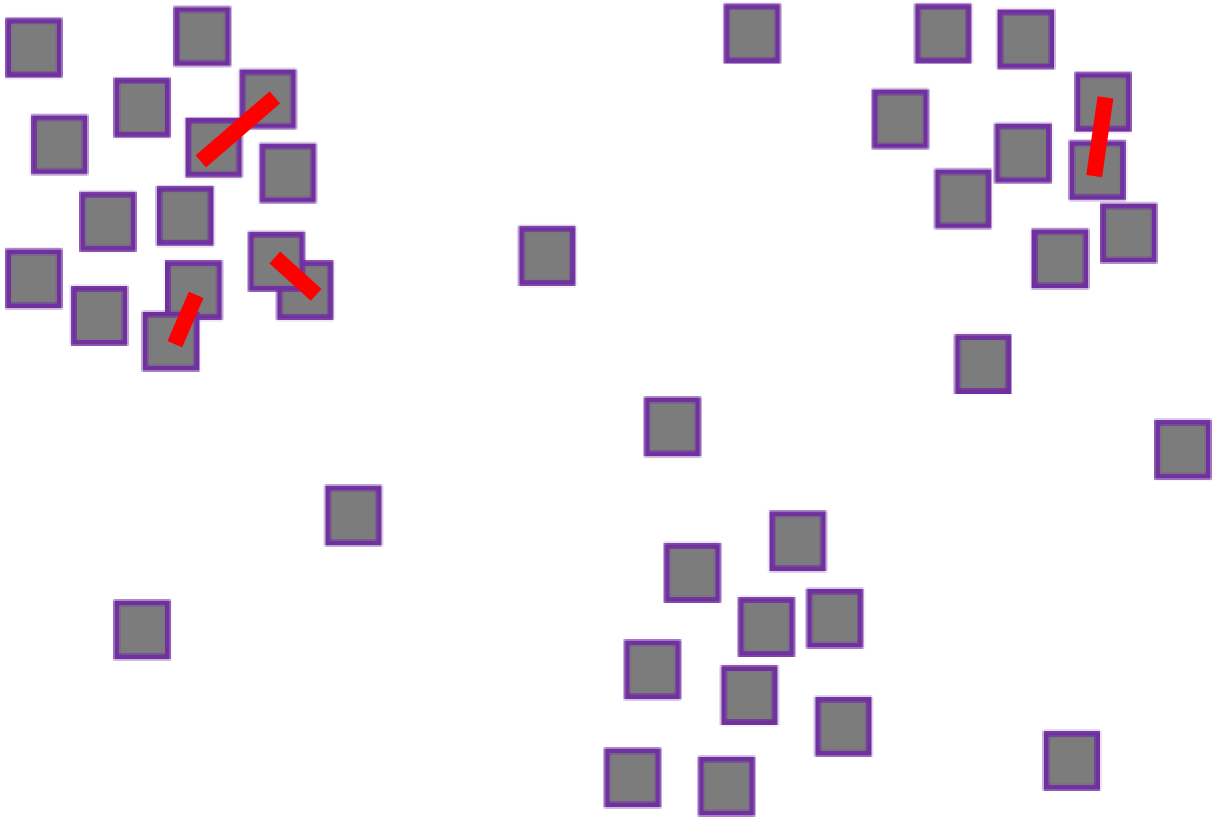
- Hierarchical
  - Top down
  - Bottom up / “agglomerative”

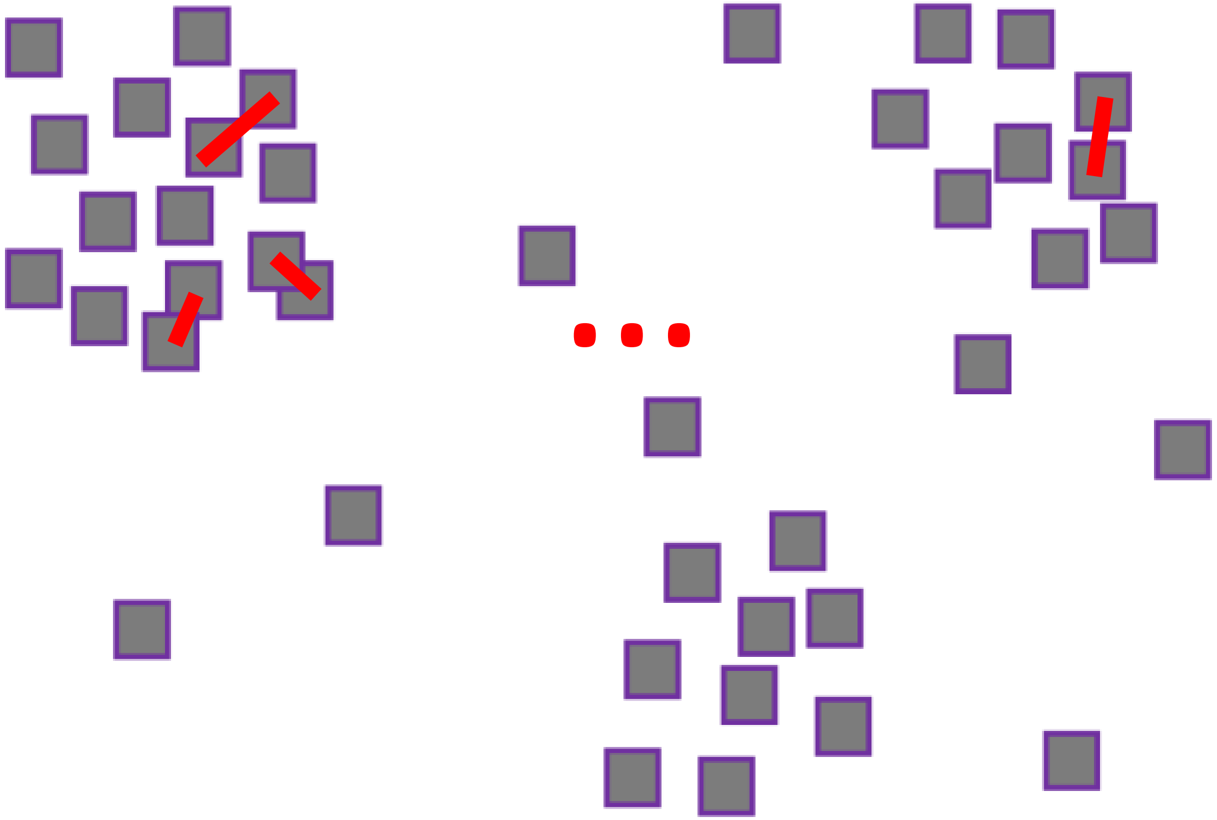


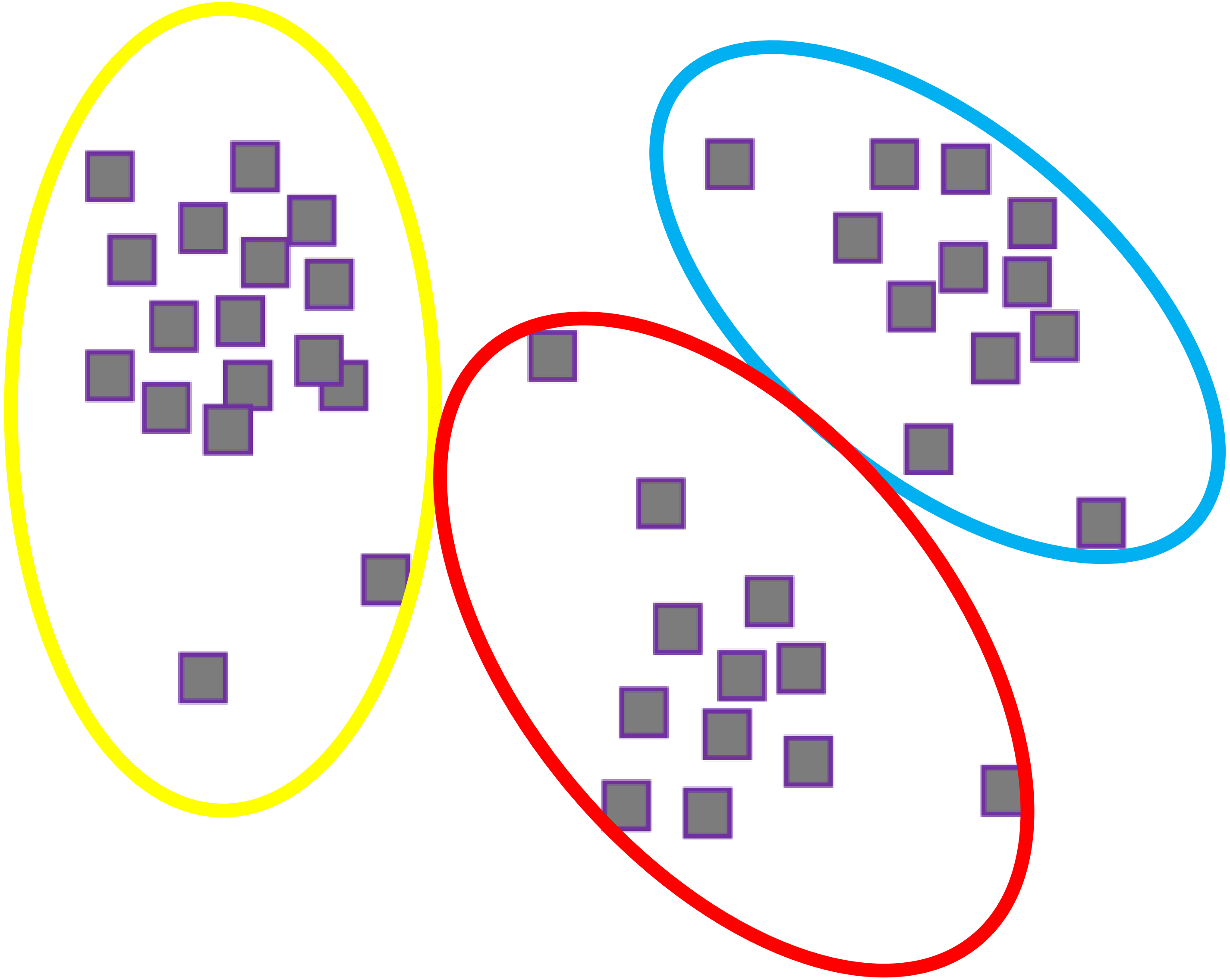


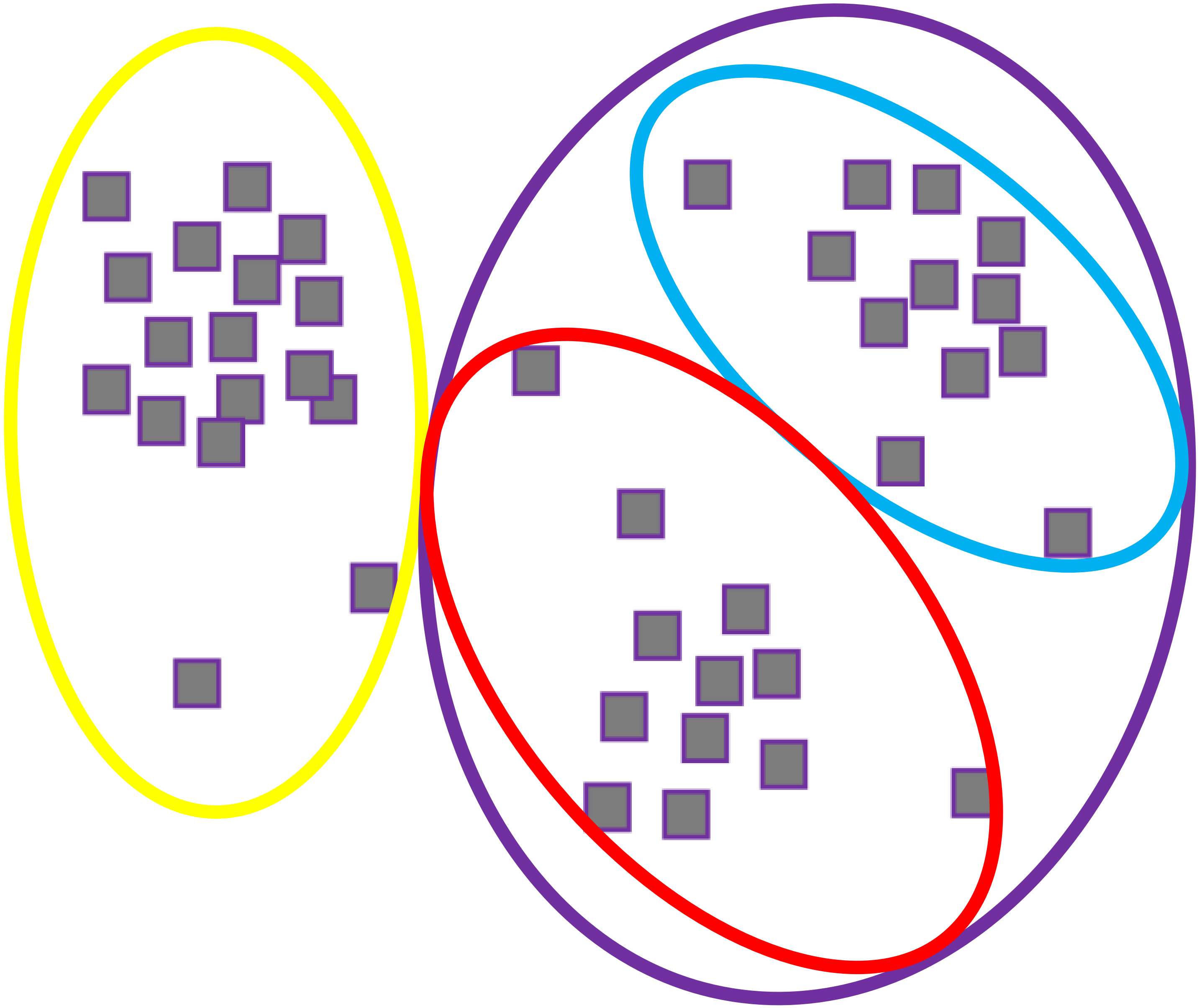












Clustering Approaches

- Hierarchical
  - Top down
  - Bottom up

Clustering Approaches

- Hierarchical
  - Top down
  - Bottom up
- Distributional/centroid-based
  - k-means clustering

k-Mean Clustering

General idea
(k is given)

1. Pick k points at random from D - call them $\overline{c_1}$, $\bar{c}_2$, …, $\bar{c}_k$ (“c” for “centroid”)

k-Mean Clustering

General idea
(k is given)

1. Pick k points at random from D - call them $\overline{c_1}$, $\bar{c}_2$, …, $\bar{c}_k$ (“c” for “centroid”)
2. Define S_i = { $\bar{x} \in$ D | argmin_i D($\bar{x}$, $\overline{c_i}$) } for each $1 \leq \mathrm{i} \leq$ k

   - this sets each S_i to all points in D that are closer to $\overline{c_i}$ than any other centroid

k-Mean Clustering

General idea
(k is given)

1. Pick k points at random from D - call them $\overline{c_1}$, $\bar{c}_2$, …, $\bar{c}_k$ ("c" for "centroid")
2. Define S_i = { $\bar{x}$ ϵ D | argmin_i D($\bar{x}$, $\overline{c_i}$) } for each $1 \leq i \leq k$
   - this sets each S_i to all points in D that are closer to $\overline{c_i}$ than any other centroid
3. Define $\overline{c_i}$ = "average" of all s ϵ S_i for each $1 \leq i \neq j \leq k$
   - this resets each $\overline{c_i}$ to the center ("centroid") of all of the points in S_i

k-Mean Clustering

General idea
(k is given)

1. Pick k points at random from D - call them $\overline{c_1}$, $\bar{c}_2$, …, $\bar{c}_k$ (“c” for “centroid”)
2. Define S_i = { $\bar{x}$ ϵ D | argmin_i D($\bar{x}$, $\overline{c_i}$) } for each $1 \leq i \leq k$ }
   - this sets each S_i to all points in D that are closer to $\overline{c_i}$ than any other centroid
3. Define $\overline{c_i}$ = “average” of all s ϵ S_i for each $1 \leq i \neq j \leq k$
   - this resets each $\overline{c_i}$ to the center (“centroid”) of all of the points in S_i
4. If the c_i’s have changed from the previous iteration, go to 2, otherwise end

k-Mean Clustering

General idea
(k is given)

1. Pick k points at random from D - call them $\bar{c}_1, \bar{c}_2, \ldots, \bar{c}_k$ ("c" for "centroid")
2. Define S_i = { $\bar{x} \in D \mid \text{argmin}_i\, D(\bar{x}, \bar{c}_i)$ } for each $1 \leq i \leq k$ }

   - this sets each S_i to all points in D that are closer to $\bar{c}_i$ than any other centroid
3. Define $\bar{c}_i$ = "average" of all $s \in S_i$ for each $1 \leq i \neq j \leq k$

   - this resets each $\bar{c}_i$ to the center ("centroid") of all of the points in S_i
4. If the c_i's have changed from the previous iteration, go to 2, otherwise end

Example of "Expectation Maximization" (EM Algorithm)

k-means Clustering Algorithm

k-means(D,k):,)

For i = 1 to k

$\overline{c_i}$ = random $\bar{x} \in$D not already selected;

Repeat

For i = 1 to k

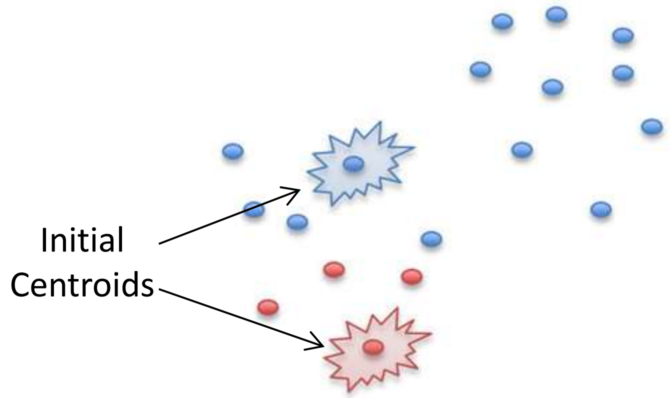
$S_i = \{\bar{x} \in D \mid \underset{i}{\operatorname{argmin}}\ \text{distance}(\overline{c_i}, \bar{x})\ \}$

For i = 1 to k

$$c_i = \frac{\sum_{\mathrm{x} \in S_\mathrm{i}} \bar{x}}{|S_\mathrm{i}|}$$

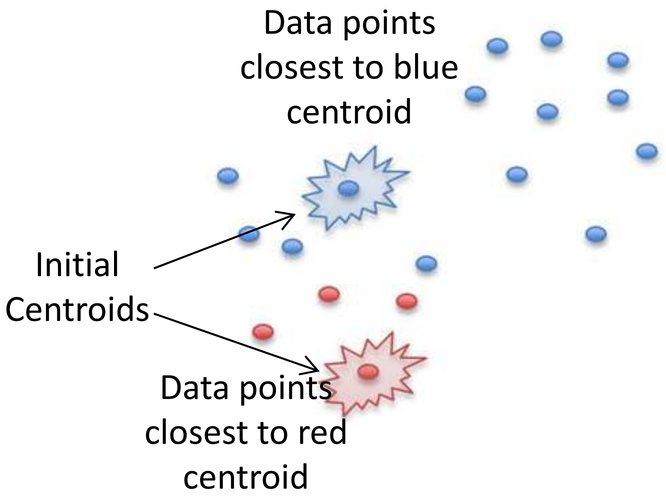
Until <stopping condition> [example: centroids don’t change]

Initial Seeding



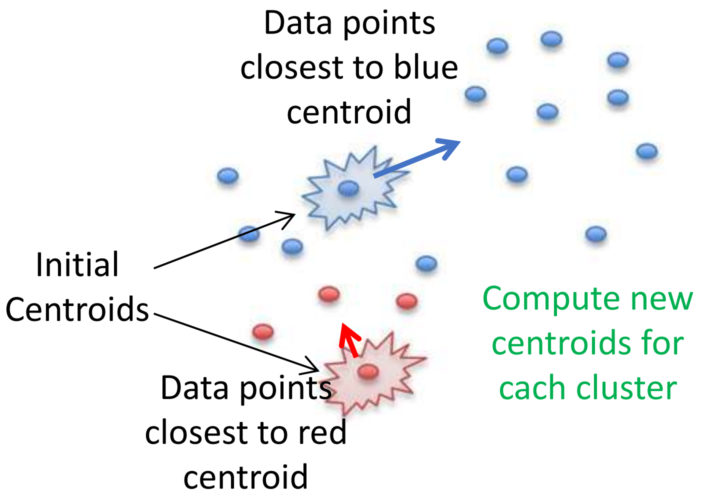
https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805

Initial Seeding

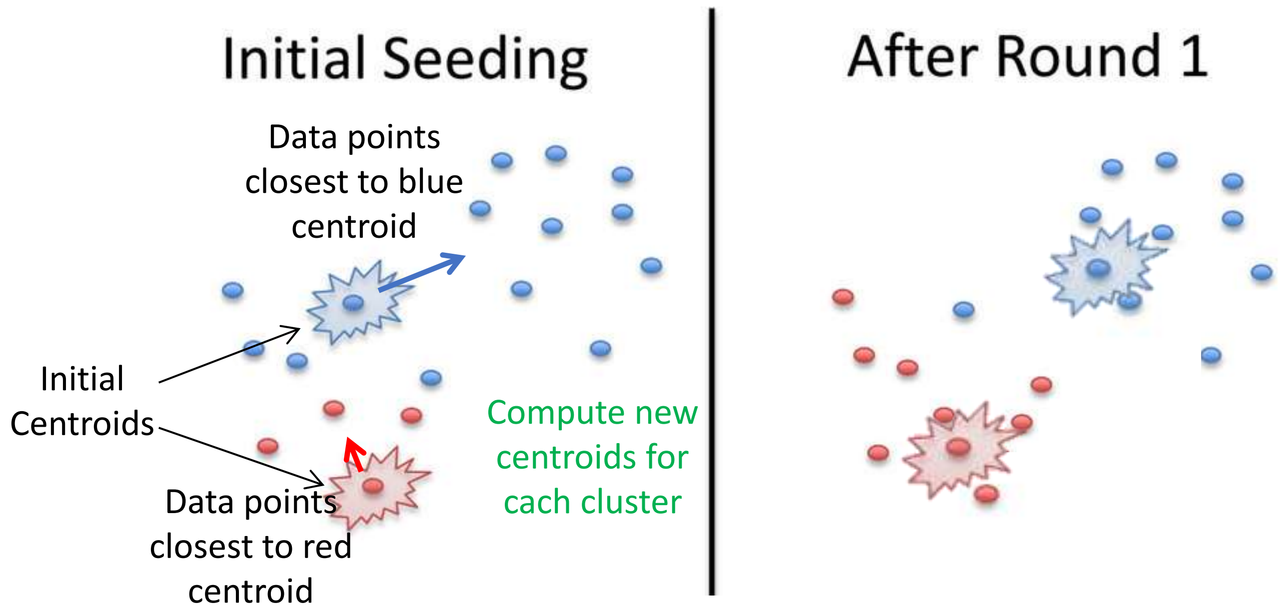


https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805

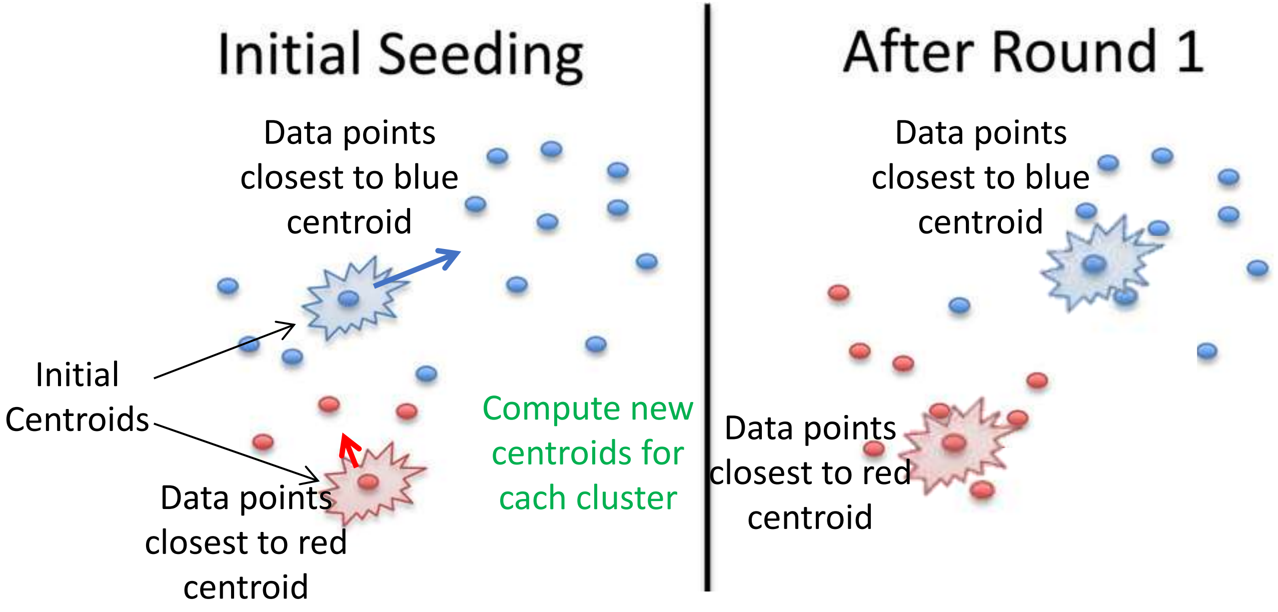
Initial Seeding



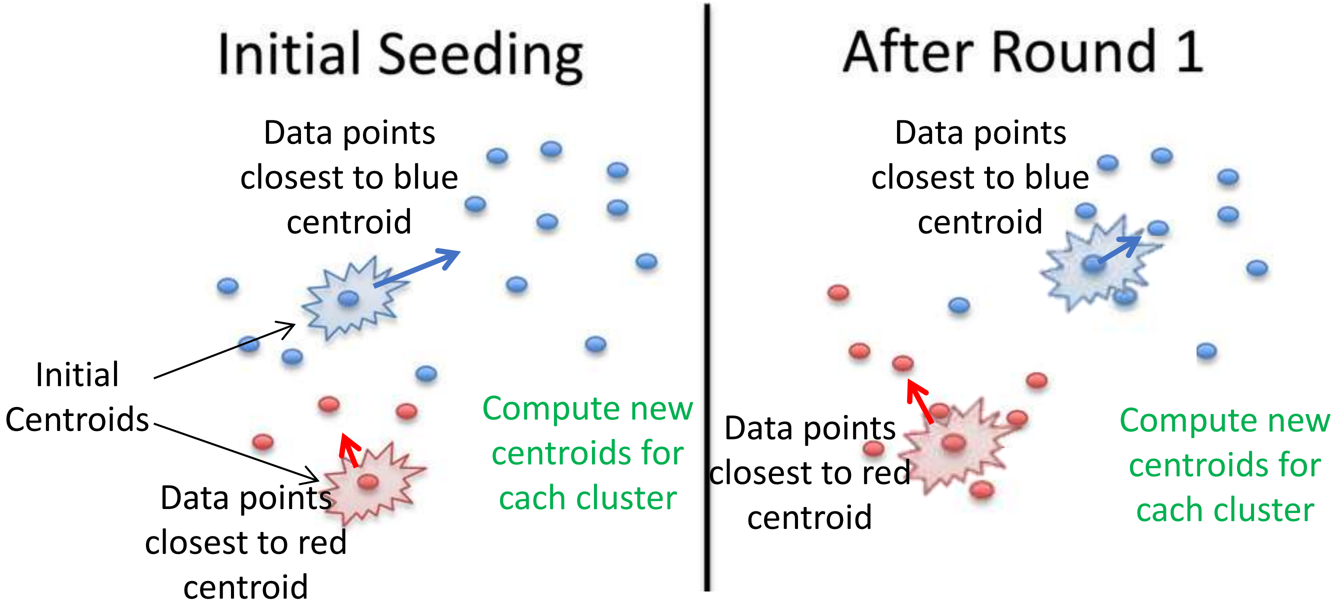
https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805



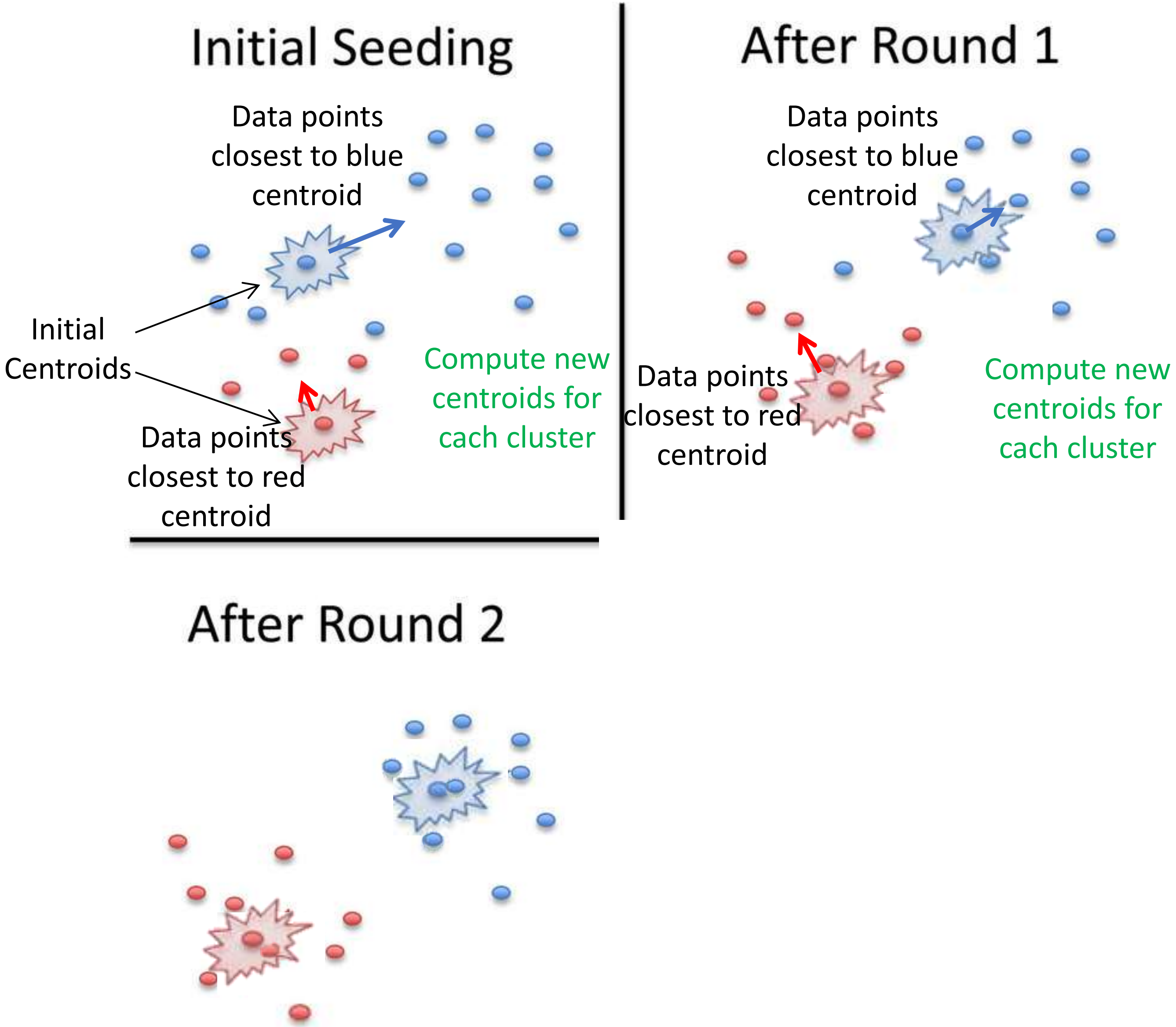
https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805



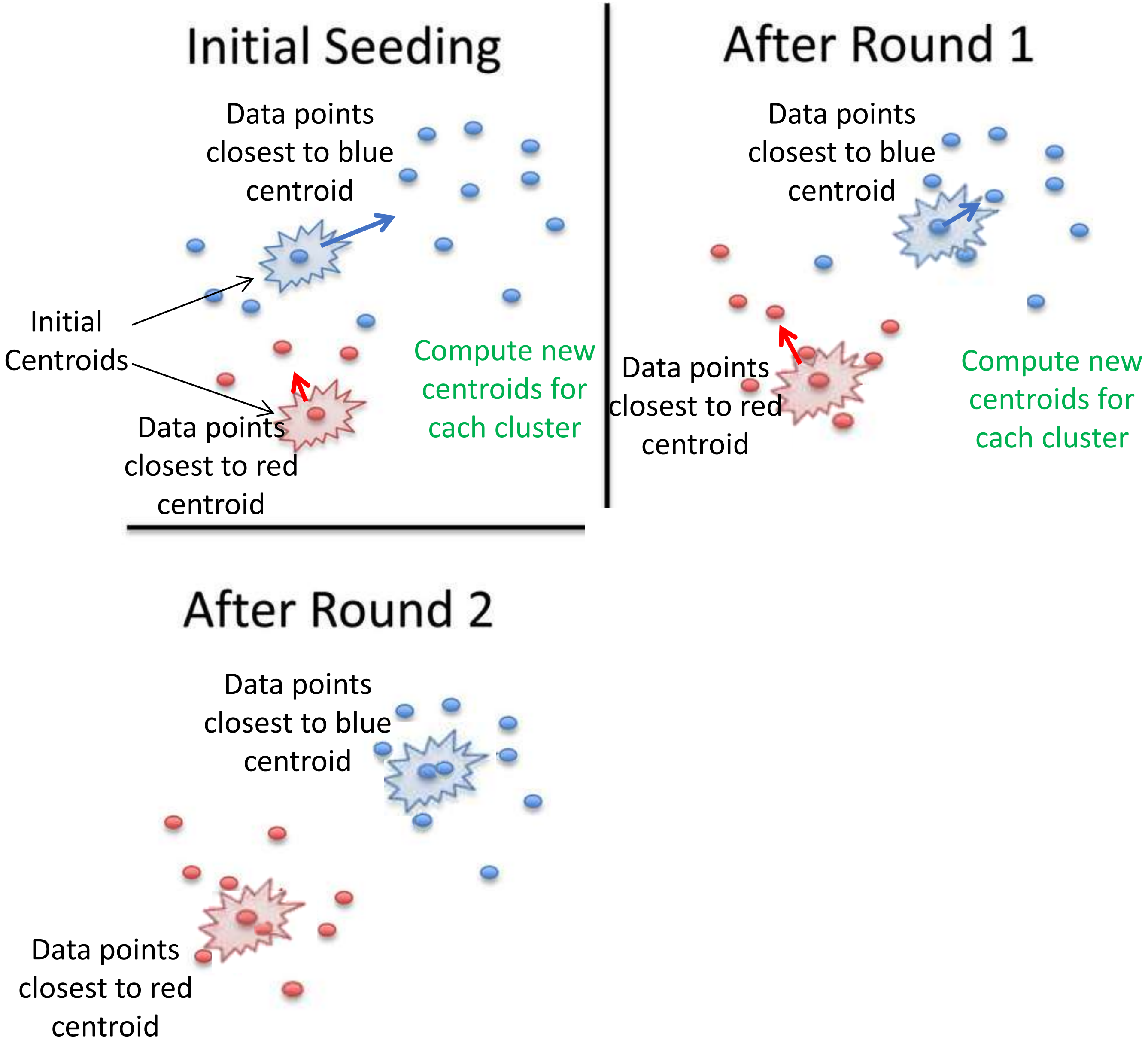
https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805



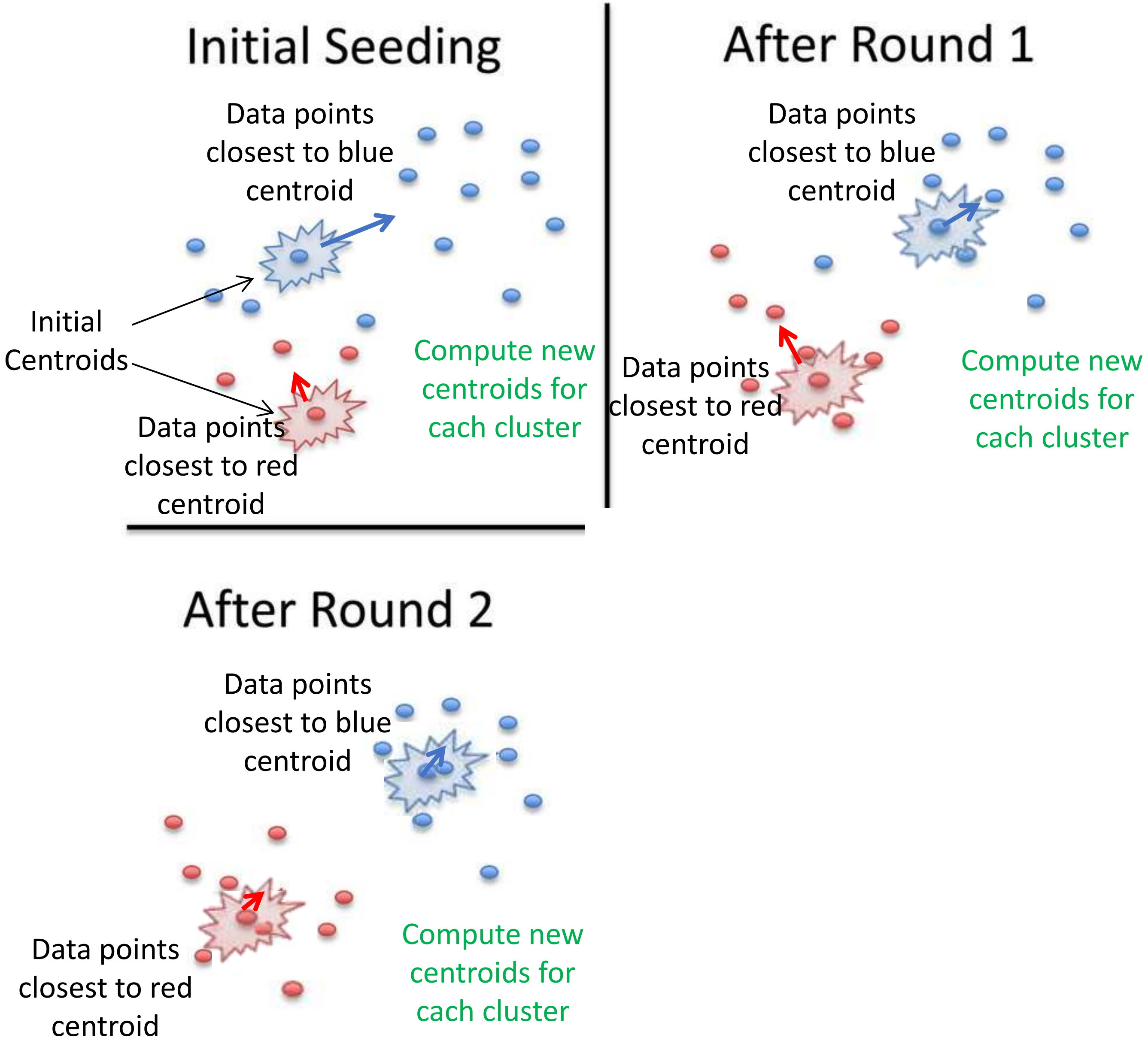
https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805



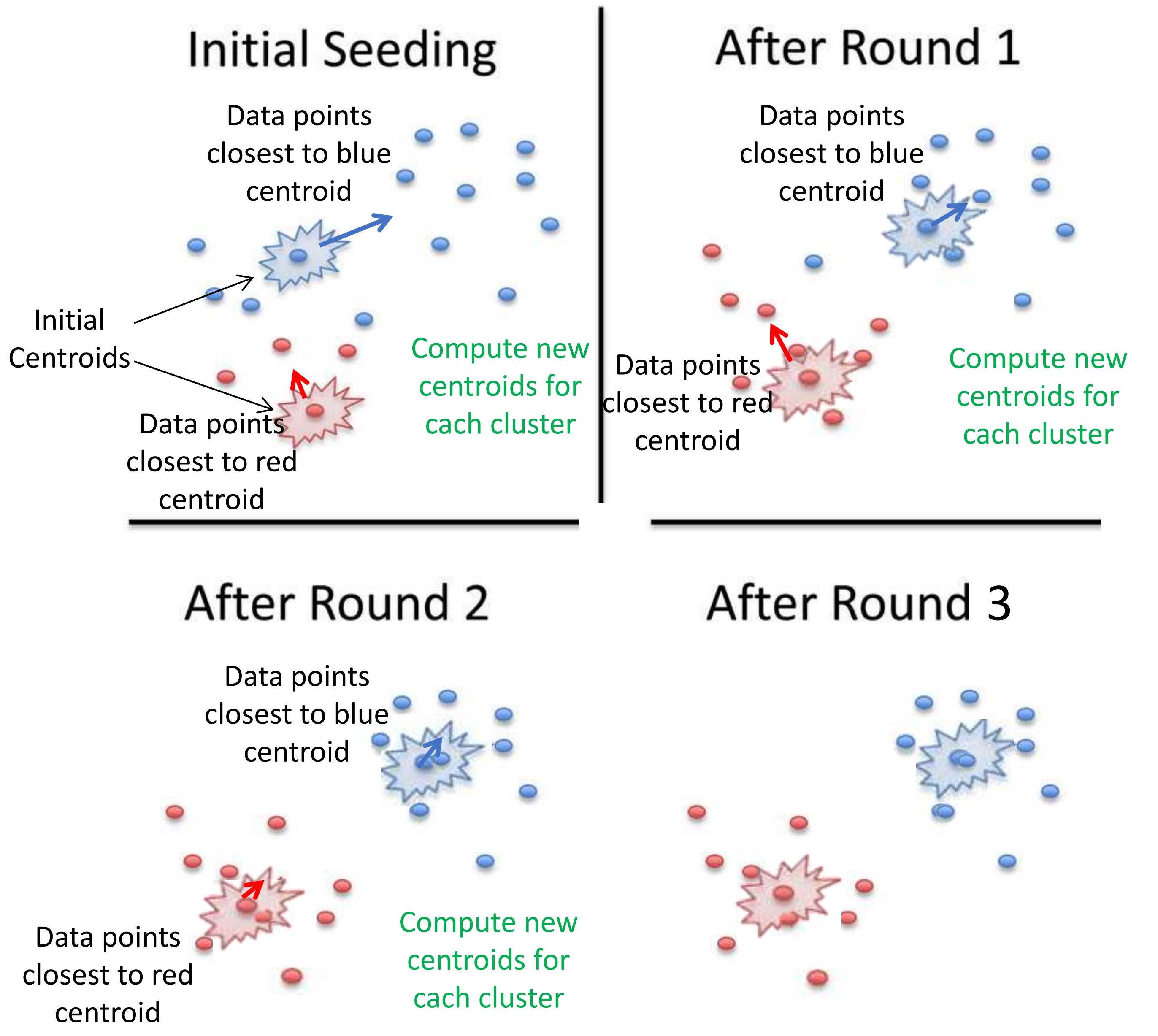
https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805



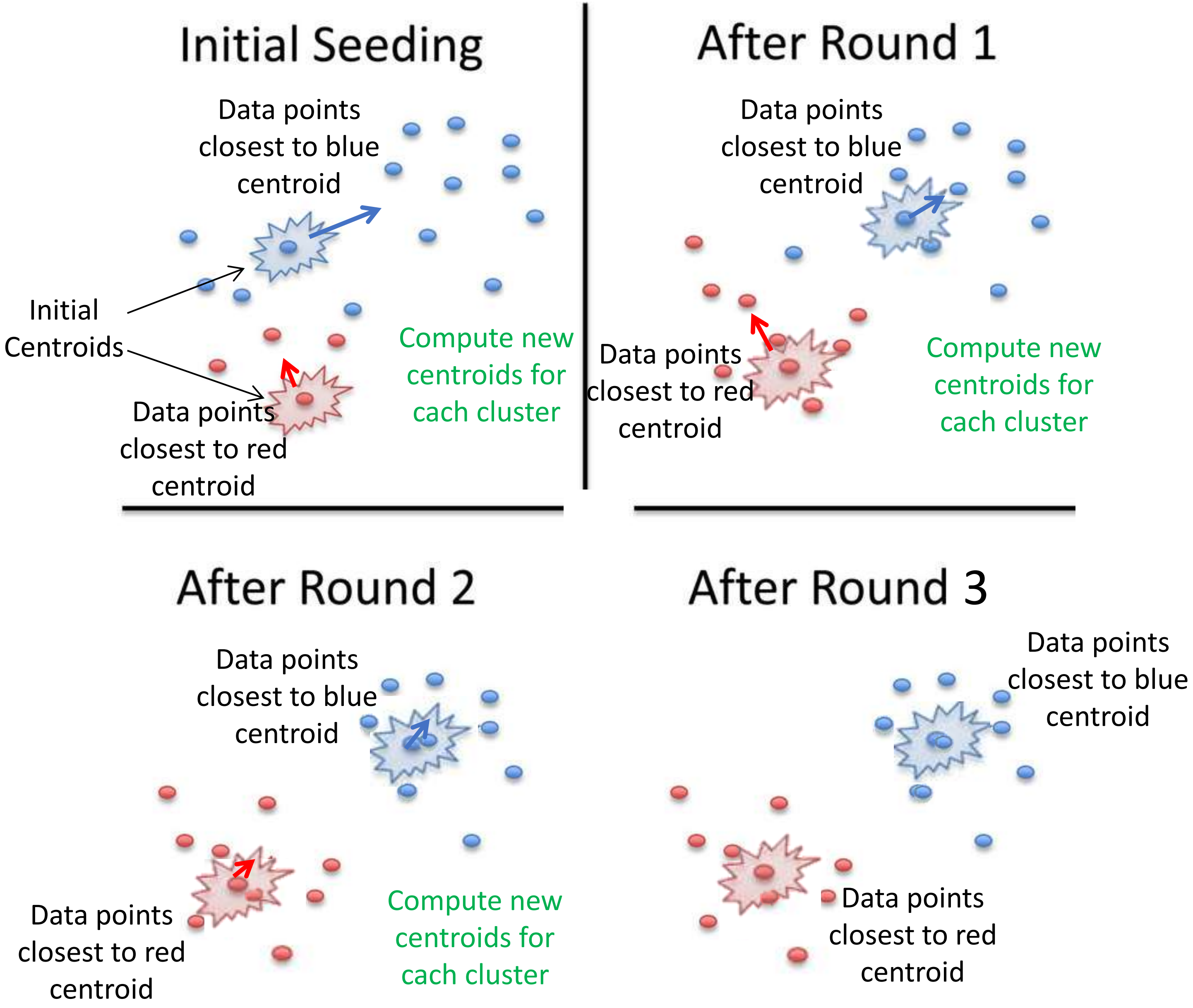
https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805



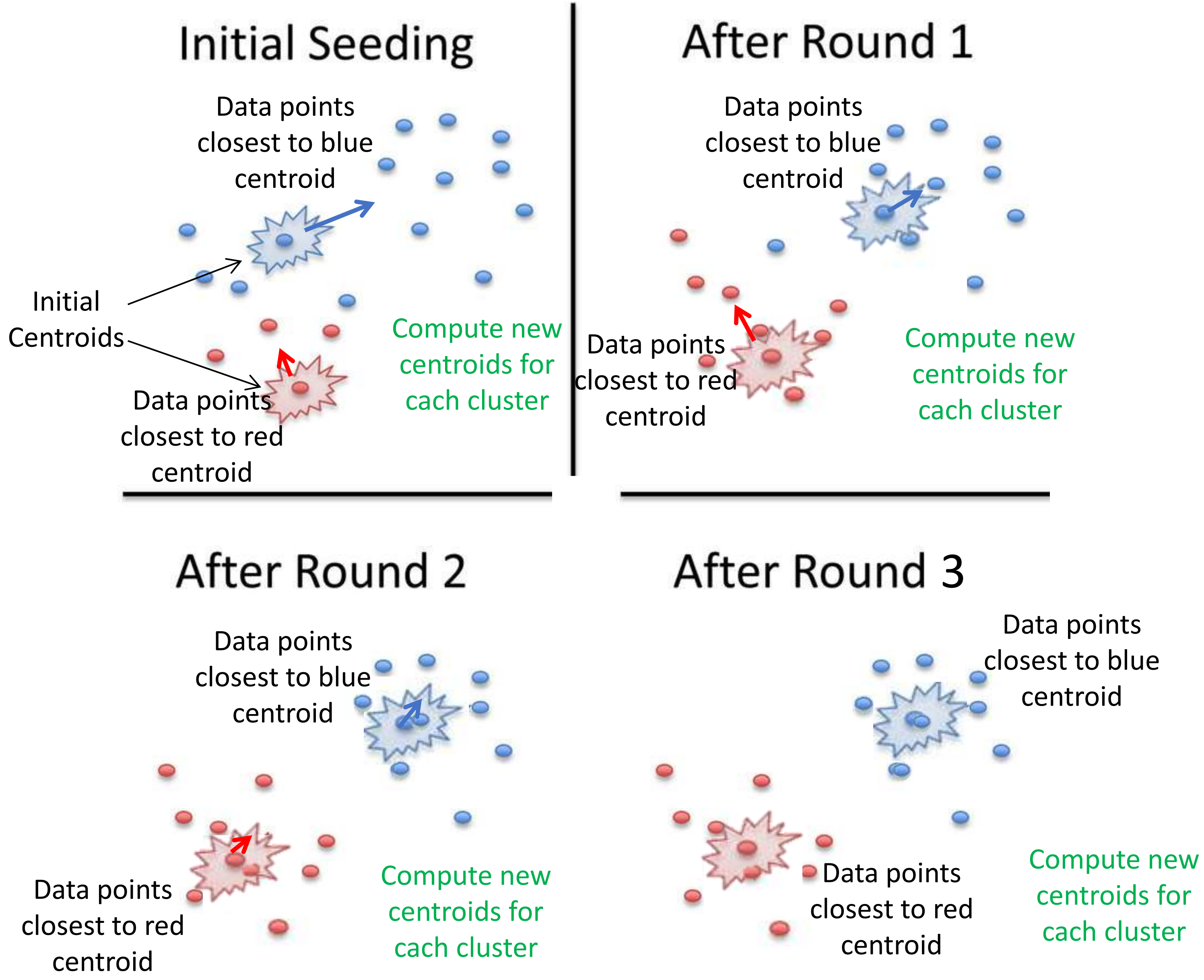
https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805



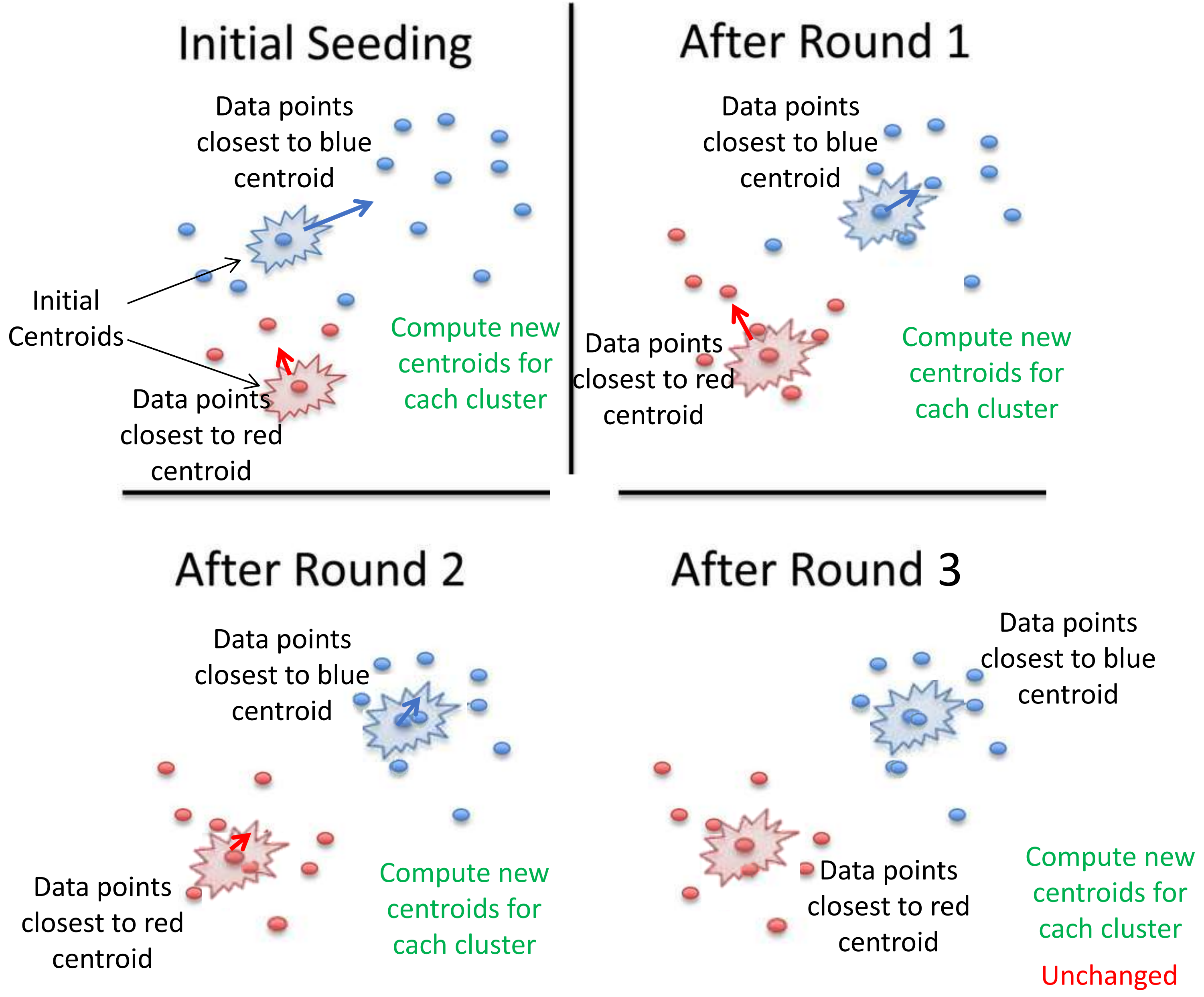
https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805



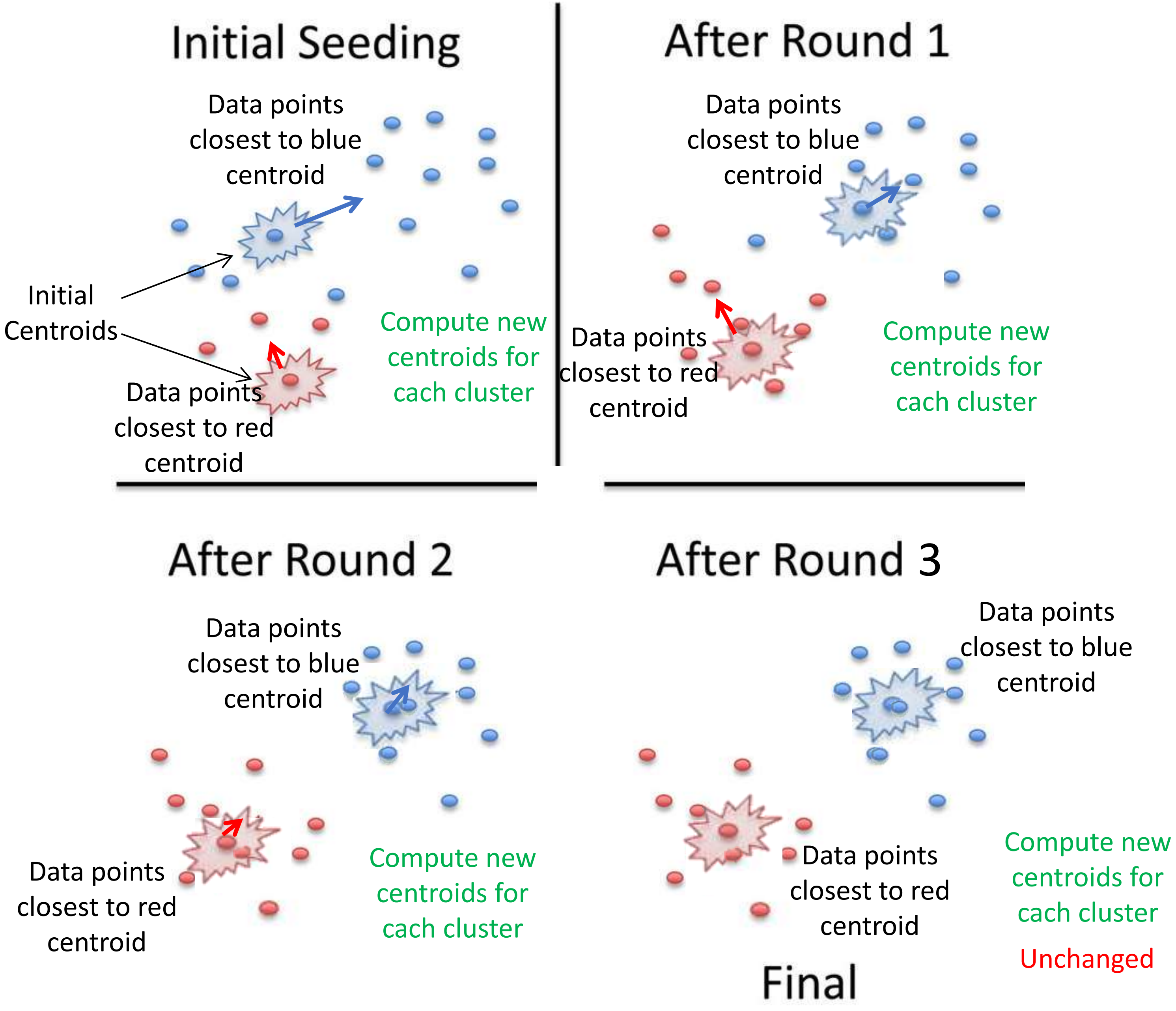
https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805



https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805



https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805



https://www.researchgate.net/figure/K-means-clustering-algorithm-An-example-2-cluster-run-is-shown-with-the-clusters_fig3_268880805

k-Mean Clustering

Many refinements:

- Picking starting points
- Convergence (random restarts)
- Probabilistic membership
    - Example: Data selected from k Gaussian distributions
      Figure out mean and S.D. of each distribution