Unifying Proportional Fairness in Centroid and Non-Centroid Clustering

Ben Cookson

University of Toronto

First presented at NeurIPS 2025

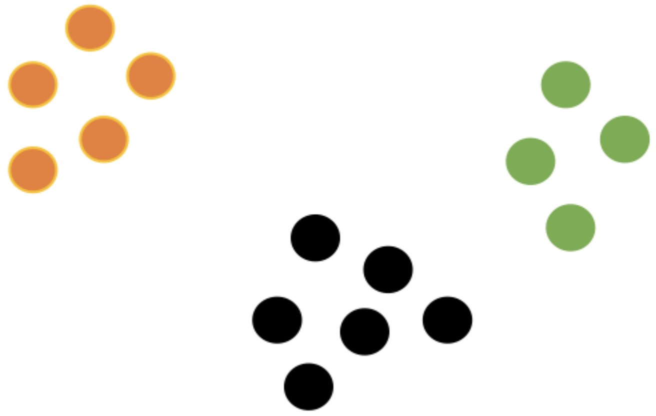
Joint work with **Nisarg Shah** and **Ziqi Yu**

Clustering

Given a set of data points, return a grouping of those data points into related **clusters**, where data points in the same cluster are "similar" to one another.

Clustering

Given a set of data points, return a grouping of those data points into related **clusters**, where data points in the same cluster are "similar" to one another.

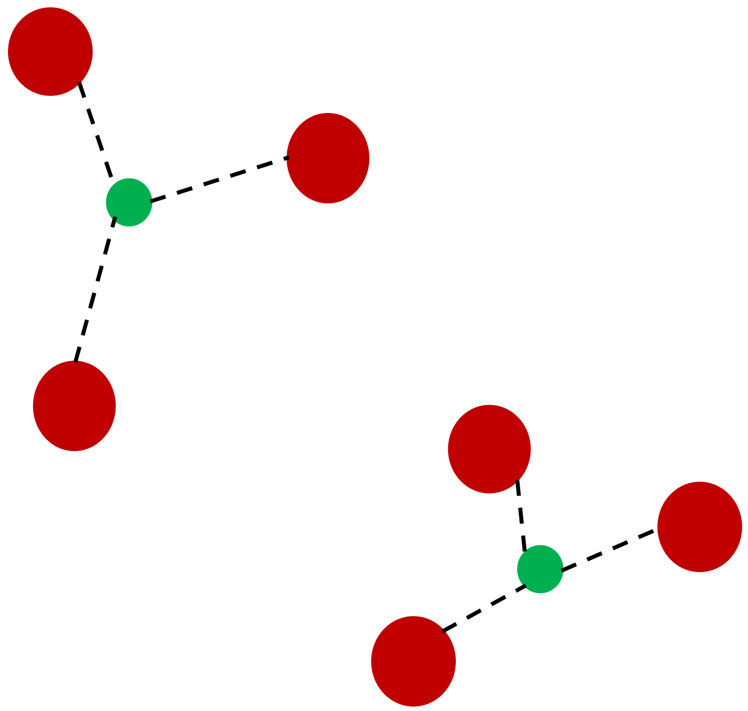


Clustering in Machine Learning

- Common Clustering Methods
  - k-means
  - k-medians

- These are great for applications like segmenting pixels of images into key groups.

- How do we do this fairly using principals from social choice?

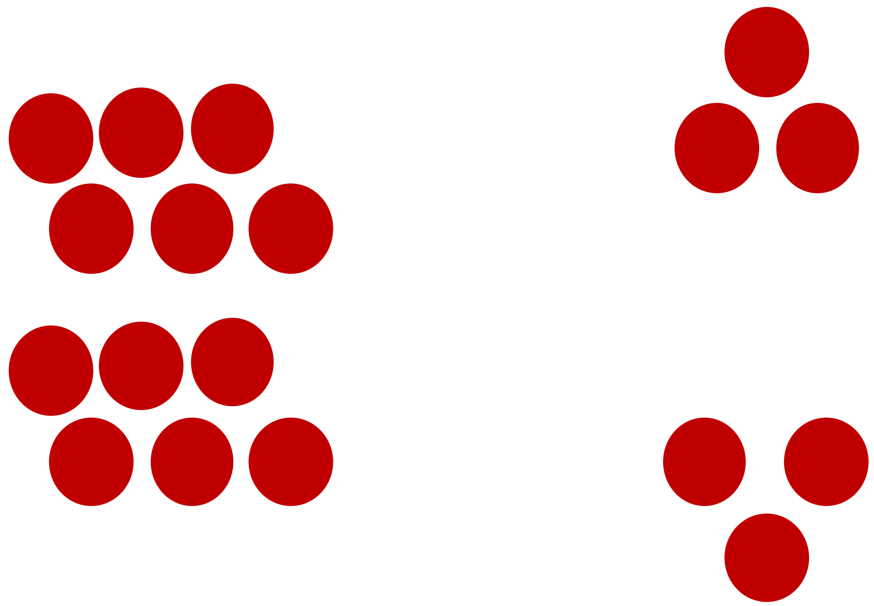


$$\min_{C_1,\ldots,C_k,x_1,\ldots,x_k} \sum_{t=1}^{k} \sum_{i\in C_t} d(i,x_t)^2.$$

k-means objective

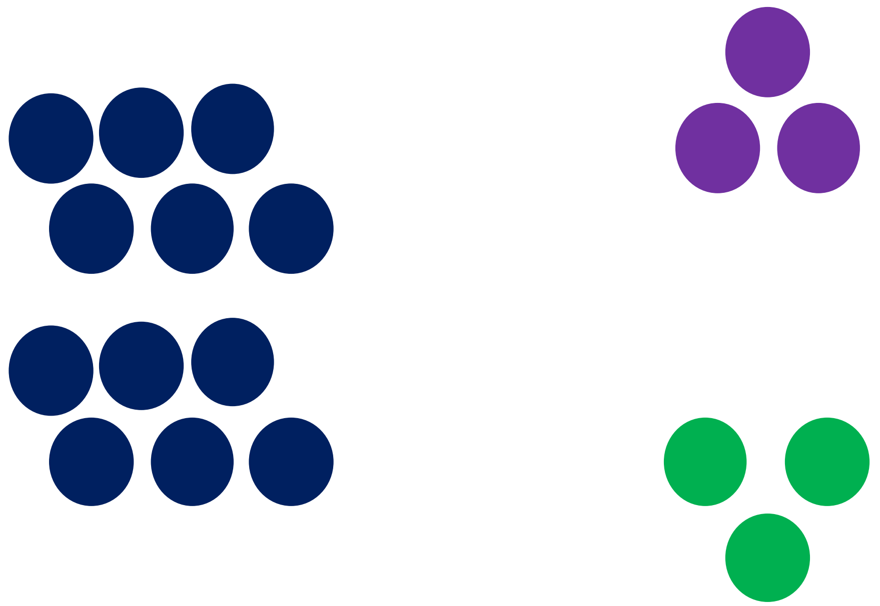
Fair Clustering Algorithms

- Assume the points have preferences over the clusters they are placed in.
  - Want to be placed in clusters with points that are similar to them.



Fair Clustering Algorithms

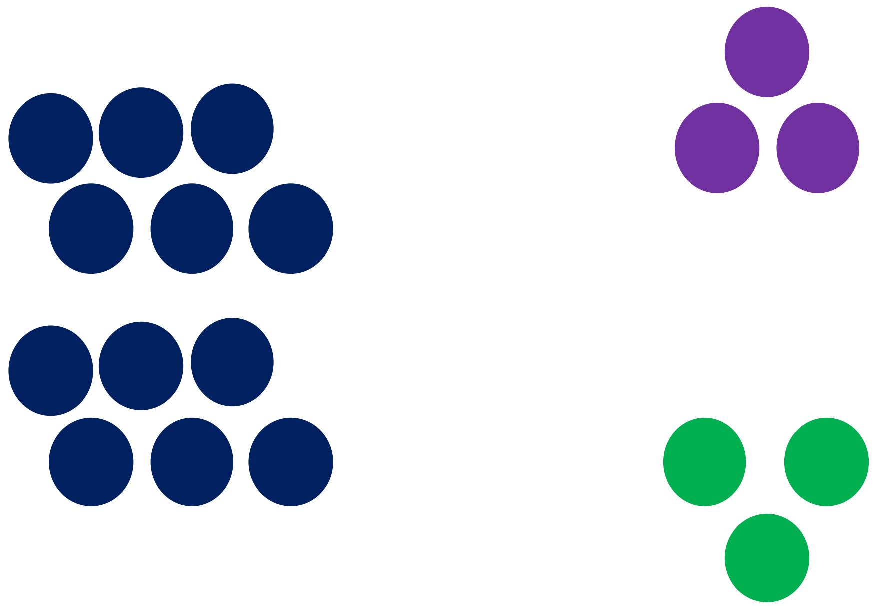
- Assume the points have preferences over the clusters they are placed in.
  - Want to be placed in clusters with points that are similar to them.
- Common clustering algorithms do not take such preferences into account.



Fair Clustering Algorithms

- Assume the points have preferences over the clusters they are placed in.
  - Want to be placed in clusters with points that are similar to them.

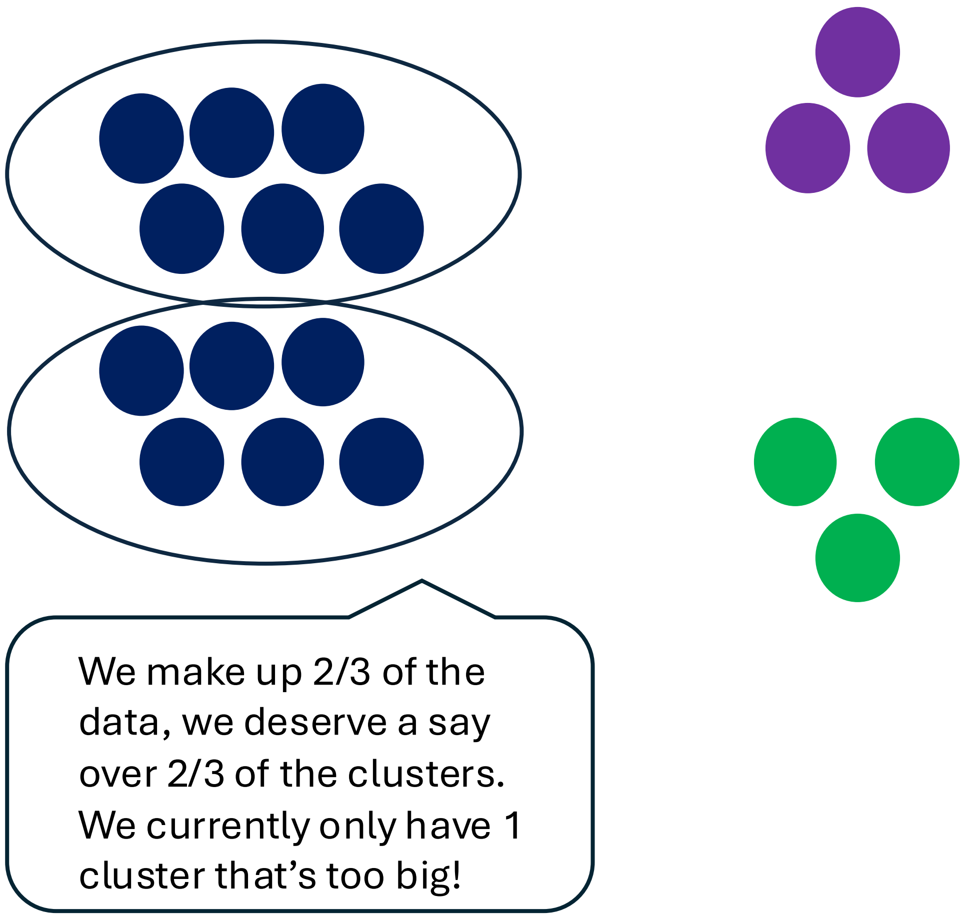
- **Proportional Representation:**
  - A group of x% of the points should get a say over how x% of clusters are decided



Fair Clustering Algorithms

- Assume the points have preferences over the clusters they are placed in.
  - Want to be placed in clusters with points that are similar to them.

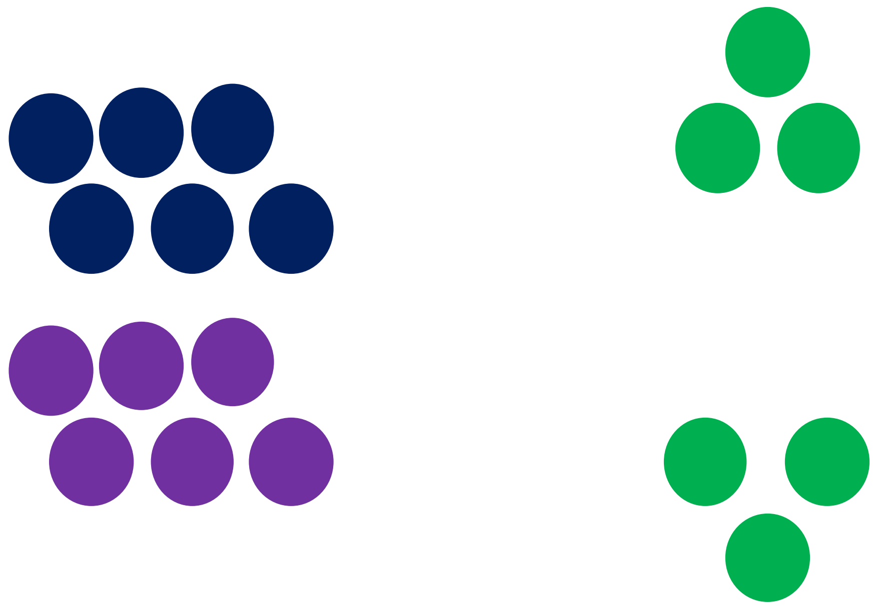
- **Proportional Representation:**
  - A group of x% of the points should get a say over how x% of clusters are decided



Fair Clustering Algorithms

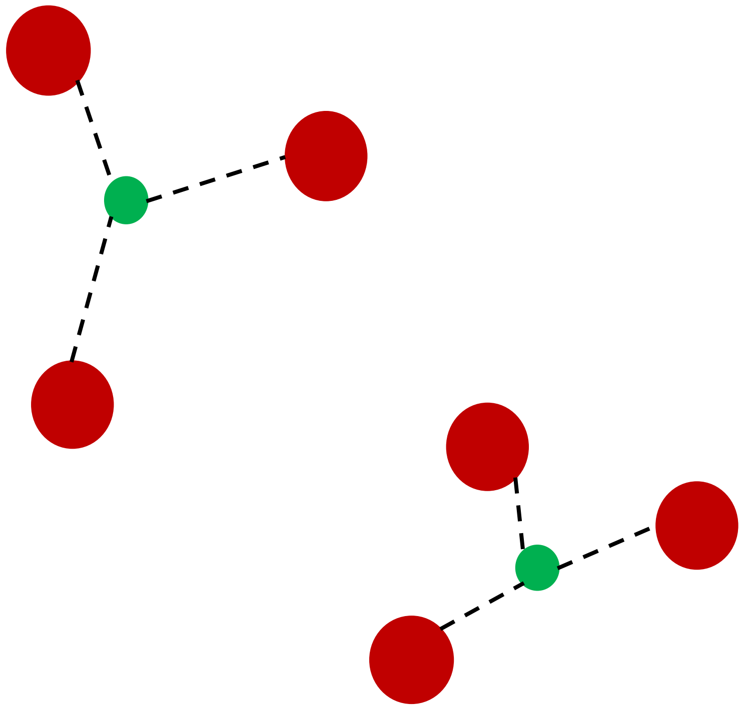
- Assume the points have preferences over the clusters they are placed in.
  - Want to be placed in clusters with points that are similar to them.

- **Proportional Representation:**
  - A group of x% of the points should get a say over how x% of clusters are decided



Formal Clustering Setup

- **Given**
  - Metric space (M, d)
  - Set of data points (agents) $N \subseteq M$
  - Set of possible cluster centers $\mathrm{X} \subseteq \mathrm{M}$
- Goal: Find a **Clustering**
  $C = \{(\mathrm{C}_1, \mathrm{x}_1), (\mathrm{C}_2, \mathrm{x}_2), \ldots, (\mathrm{C}_\mathrm{k}, \mathrm{x}_\mathrm{k})\}$
  - $\mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_\mathrm{k}$ is a disjoint partition of N into k groups
  - Assign each group C_t a center $\mathrm{x}_\mathrm{t} \in \mathrm{X}$



Formal Clustering Setup

- For a possible cluster (C_t, x_t), the loss of each agent i $\in C_t$ is denoted as $\ell_i(C_t, x_t)$.

- For a clustering $C = \{(C_1, x_1), (C_2, x_2), \ldots, (C_k, x_k)\}$, $\ell_i(C) = \ell_i(C_t, x_t)$. Where $i \in C_t$

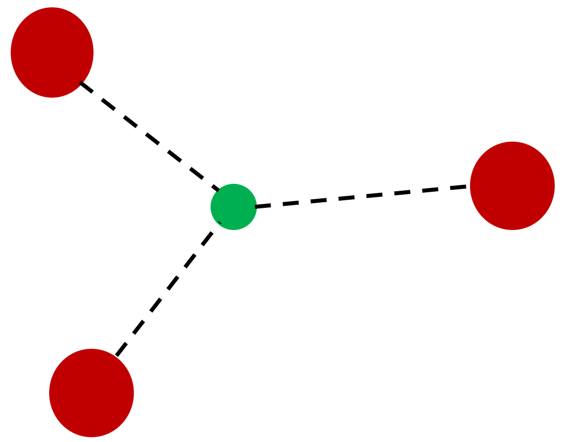
- ***α*-Approximate Core**
  - For any α ≥ 1, a clustering is in *the α-approximate core* if there does not exist a group S of at least n/k agents which can deviate and form a new cluster with a center y ∈ X, and all improve their loss **by at least a factor of *α*** in the process.

$$\alpha \cdot \ell_i(S, y) < \ell_i(C) \; \forall i \in S$$

Two Kinds of Fair Clustering Algorithms

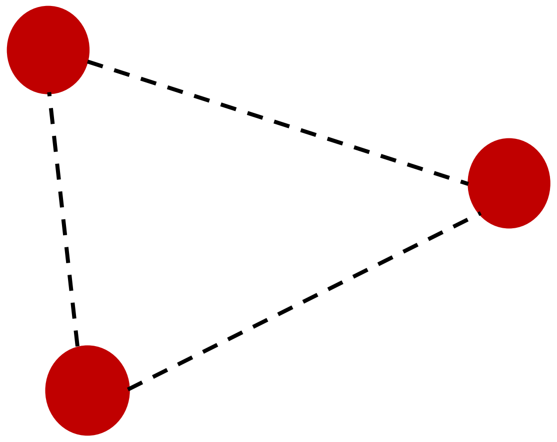
- How do we determine the loss function?
  - **Most Common:** Want to be placed in clusters with points that are "close" to them.

Centroid Preferences [CFLM19]



Each agent wants to be as close as possible to the center of their assigned cluster.
Chen, Fain, Lyu, and Munagala [2019]

Non-Centroid Preferences [CMS24]



Each agent wants to be as close as possible to other agents in their cluster.
Caragiannis, Micha, and Shah [2024]

Centroid Clustering [CFLM19]

- Given
  - Metric space (M, d)
  - Set of data points $N \subseteq M$
  - Set of possible cluster centers $X \subseteq M$

- Each data point's loss is its distance to its assigned center.
  - $\ell_i(C, x) = d(i, x)$

- Common clustering methods like optimizing k-means or k-medians cannot guarantee a constant approximation of the core in this model.

Centroid Clustering [CFLM19]

A simple algorithm called **Greedy Capture** guarantees a constant approximation of the core.

Greedy Capture

$\delta \leftarrow 0$; $C \leftarrow 0$

While $N \neq 0$ *do*

    Smoothly increase δ

    While $\exists\, c \in C$ *such that* $|B(c,\delta) \cap N| \geq 1$ *do*

        $C: N \leftarrow N \setminus (B(c,\delta) \cap N)$

    While $\exists\, c \in M \setminus C$ *such that* $|B(c,\delta) \cap N| \geq n/k$ *do*

        $C \leftarrow C \cup c$

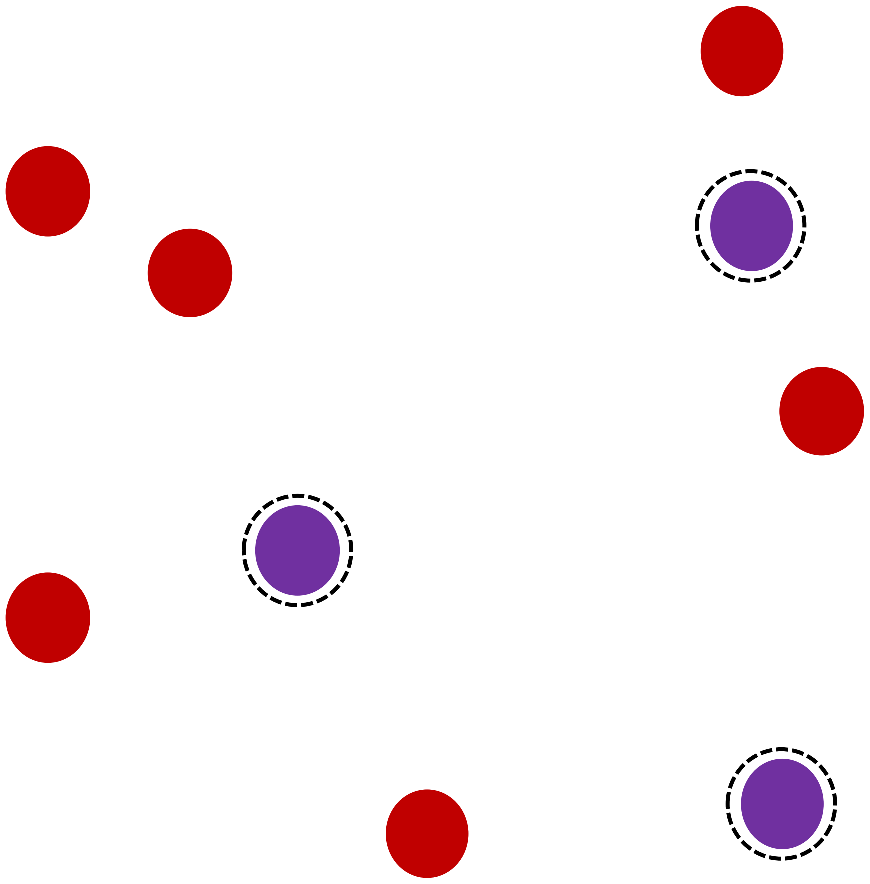
        $N \leftarrow N \setminus (B(c,\delta) \cap N)$

Return C

Centroid Clustering [CFLM19]

Greedy Capture

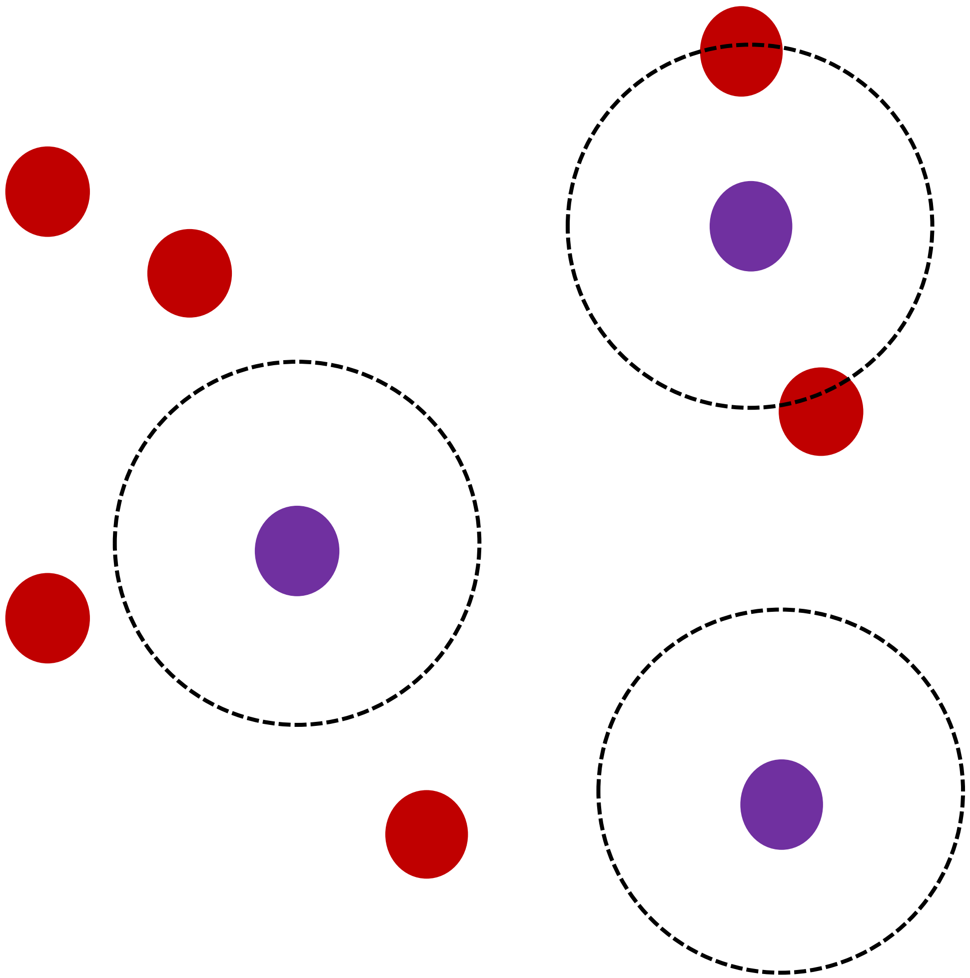
$n = 6, k = 2$



Centroid Clustering [CFLM19]

Greedy Capture

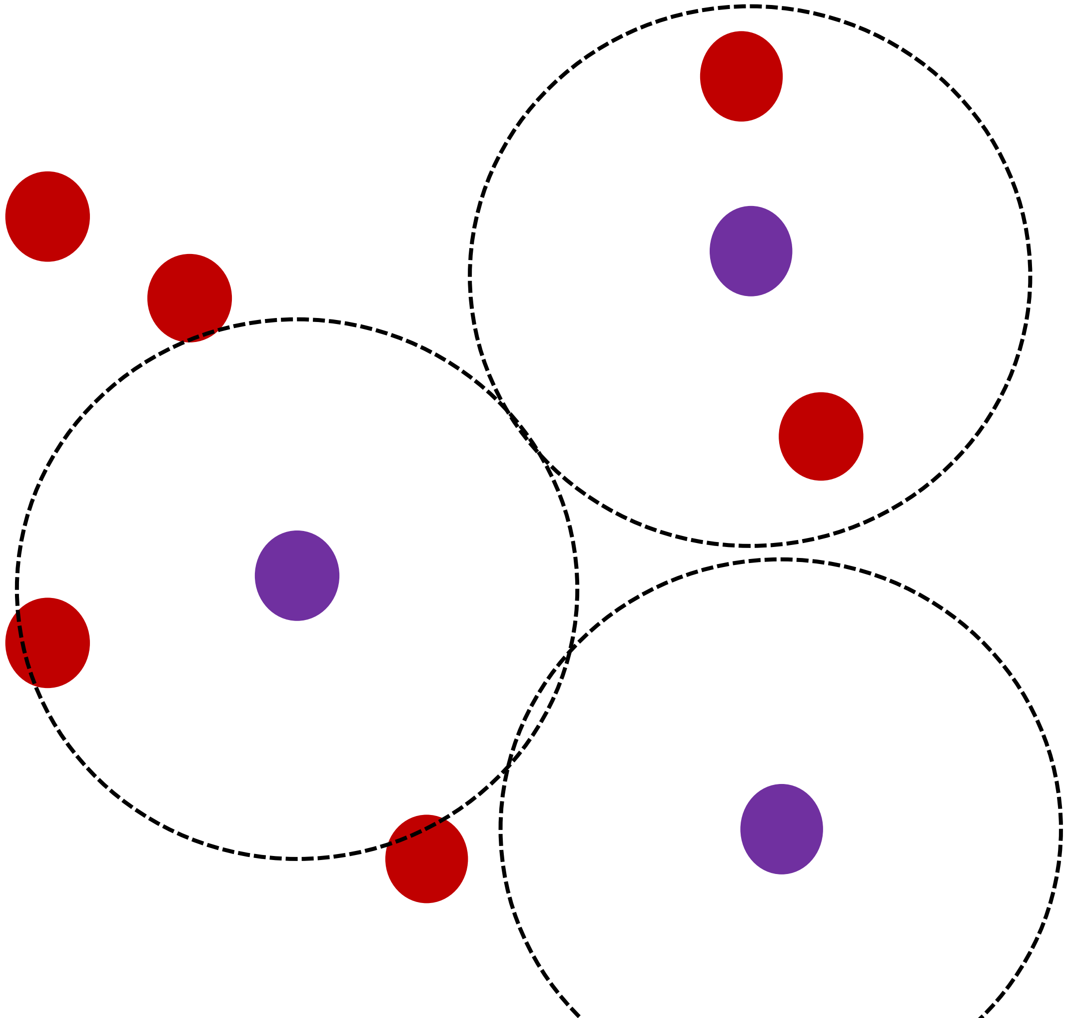
$n = 6, k = 2$



Centroid Clustering [CFLM19]

Greedy Capture

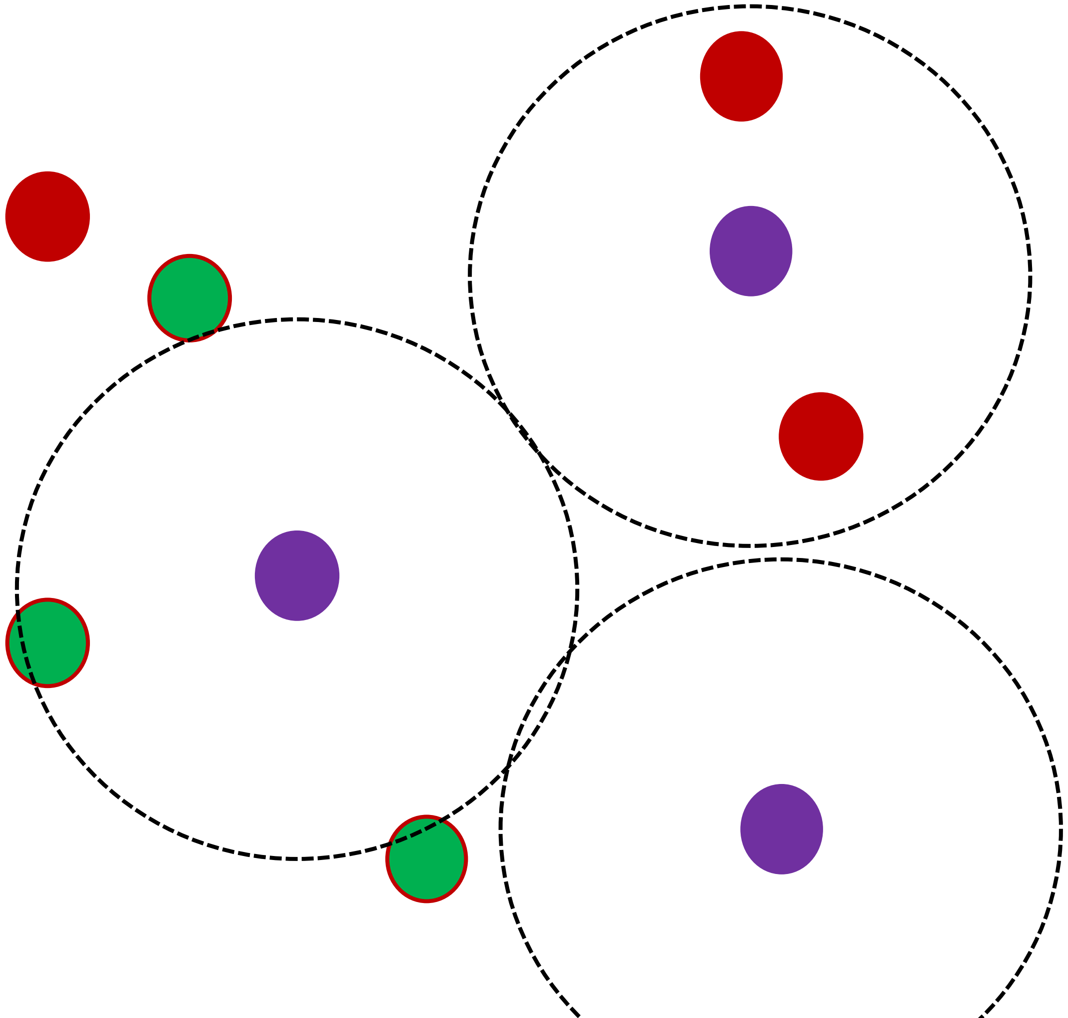
$n = 6, k = 2$



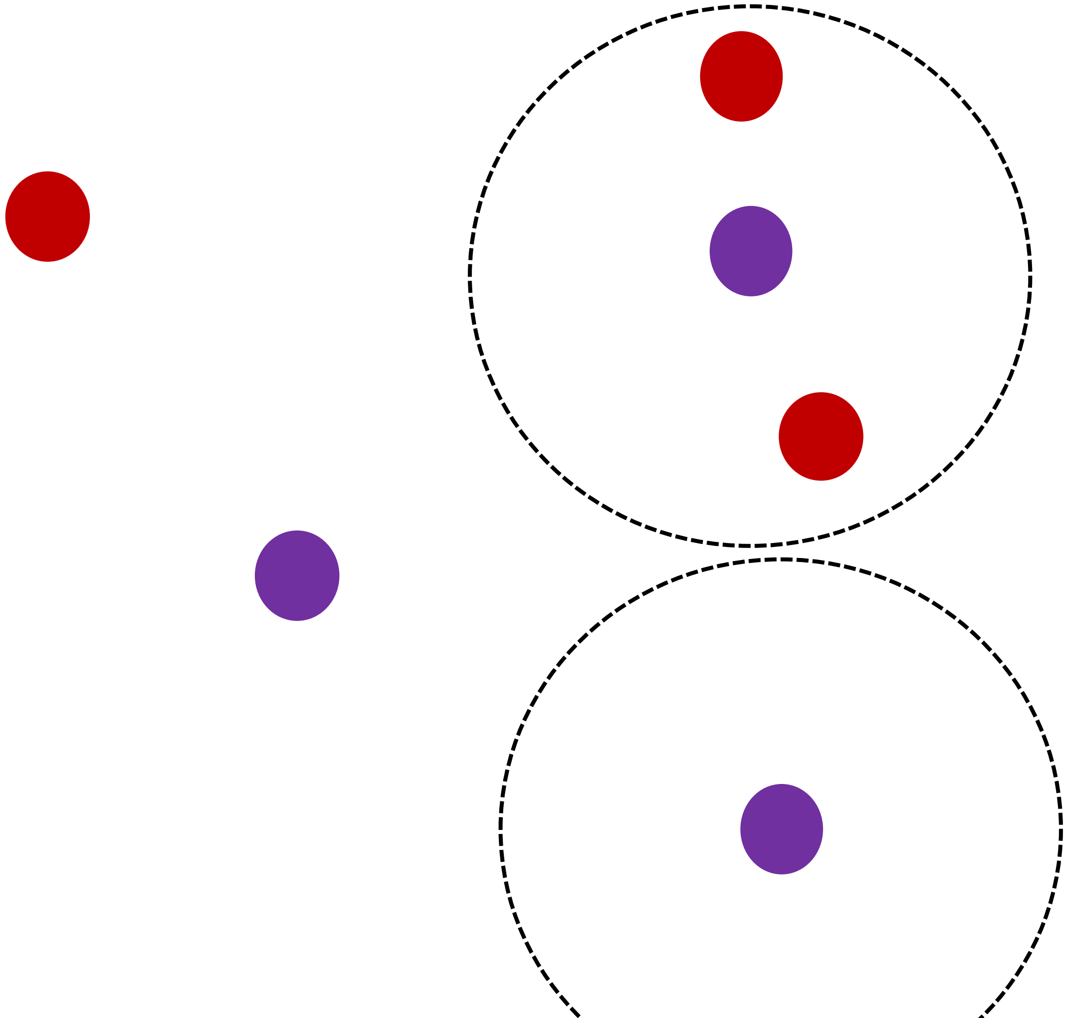
Centroid Clustering [CFLM19]

Greedy Capture

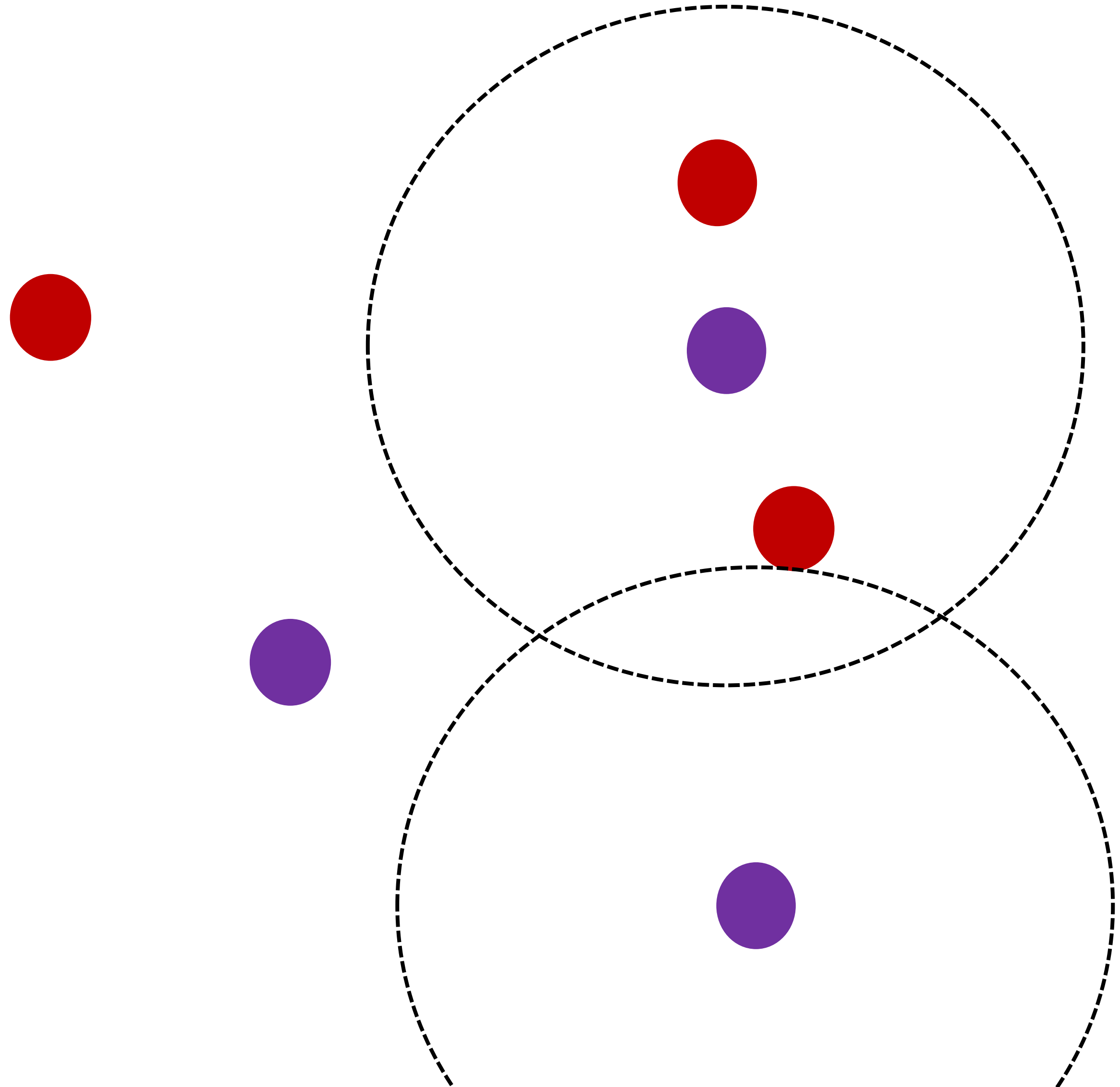
$n = 6, k = 2$



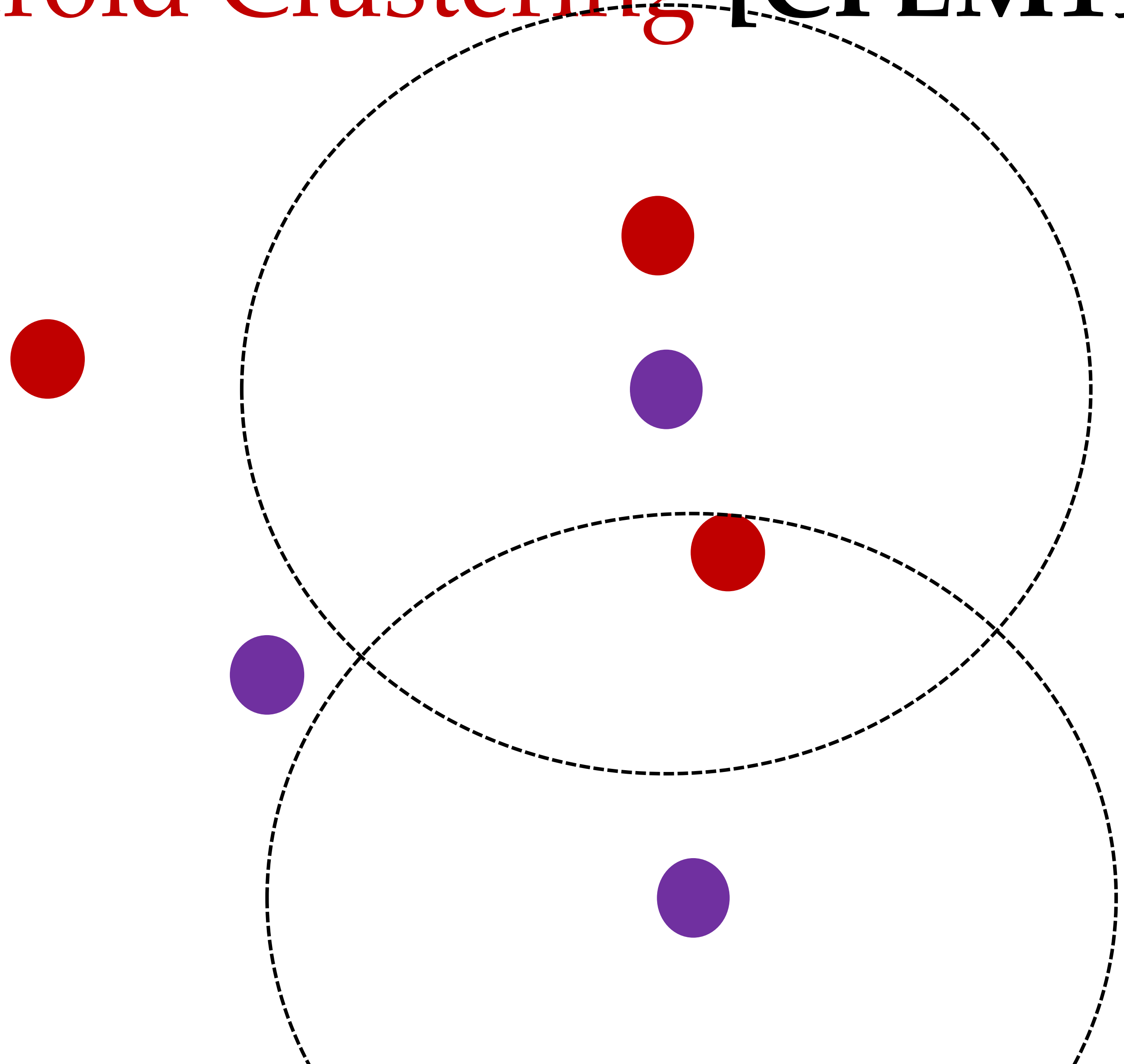
Centroid Clustering **[CFLM19]**



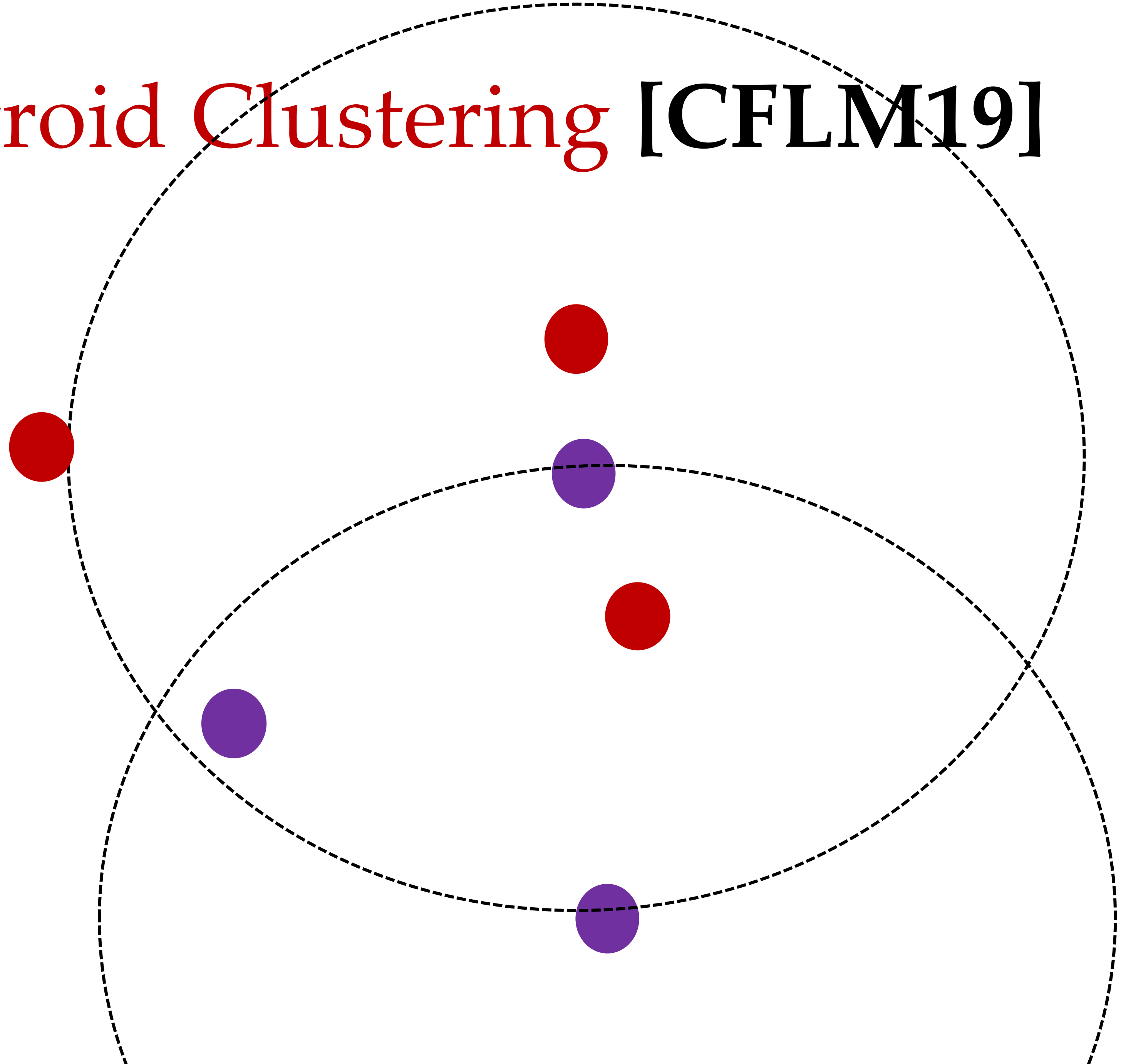
Centroid Clustering **[CFLM19]**



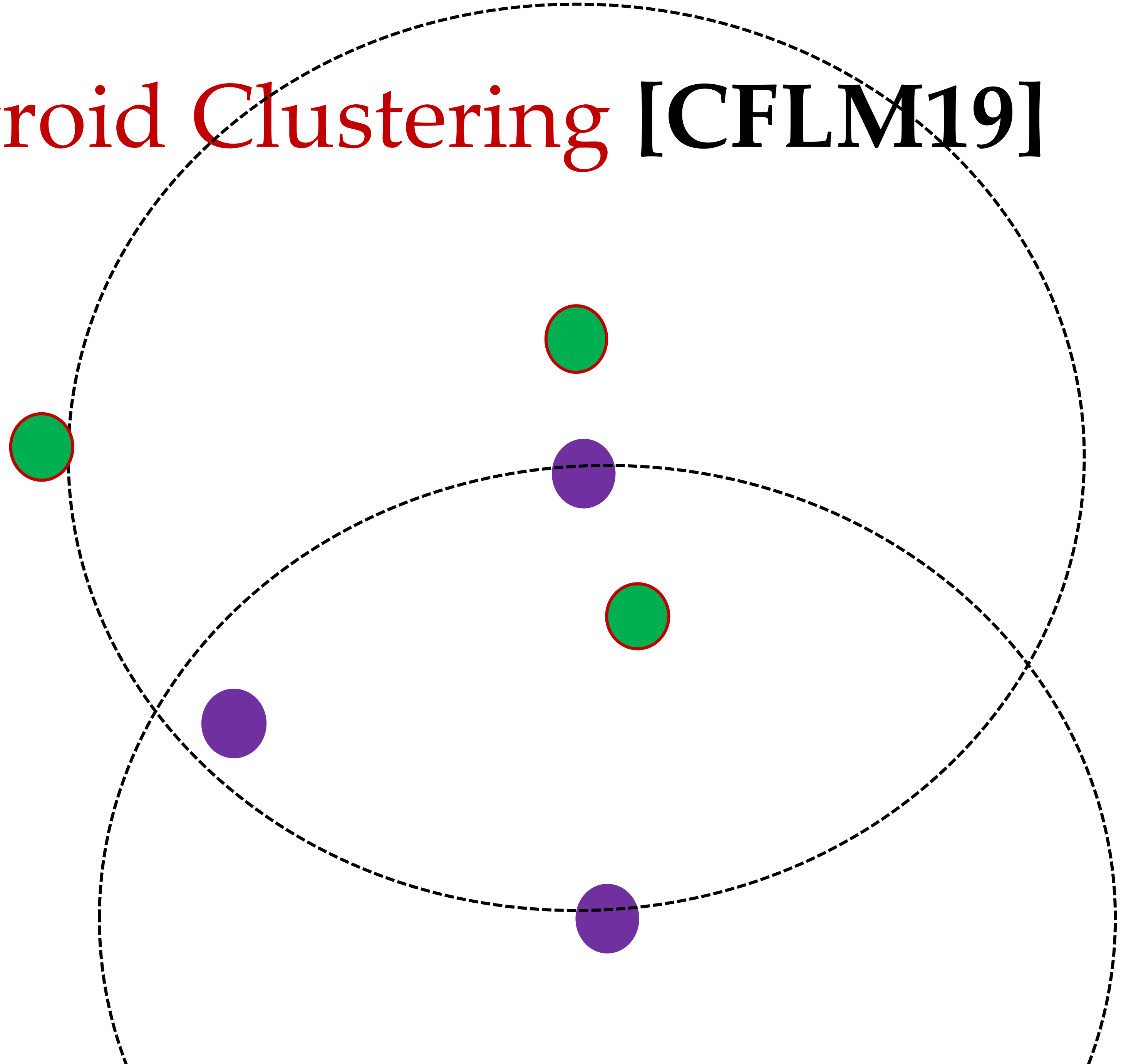
Centroid Clustering [CFLM19]



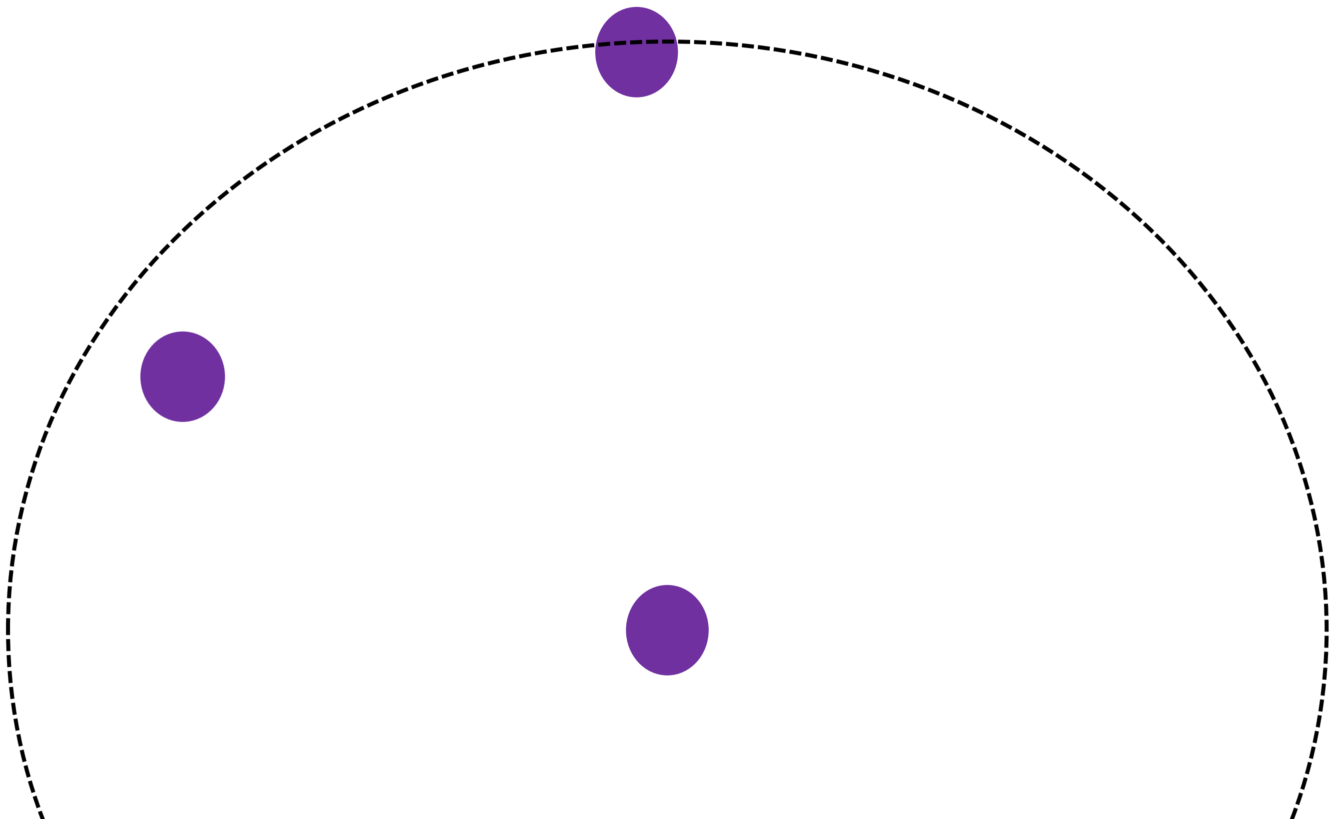
Centroid Clustering [CFLM19]



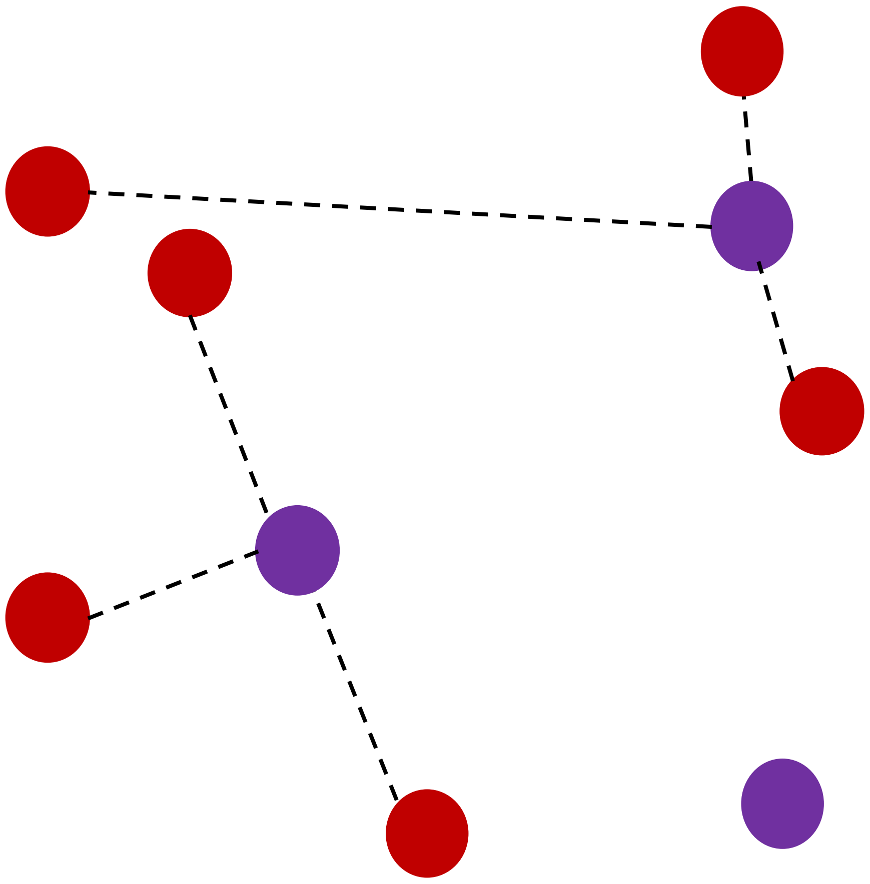
Centroid Clustering [CFLM19]



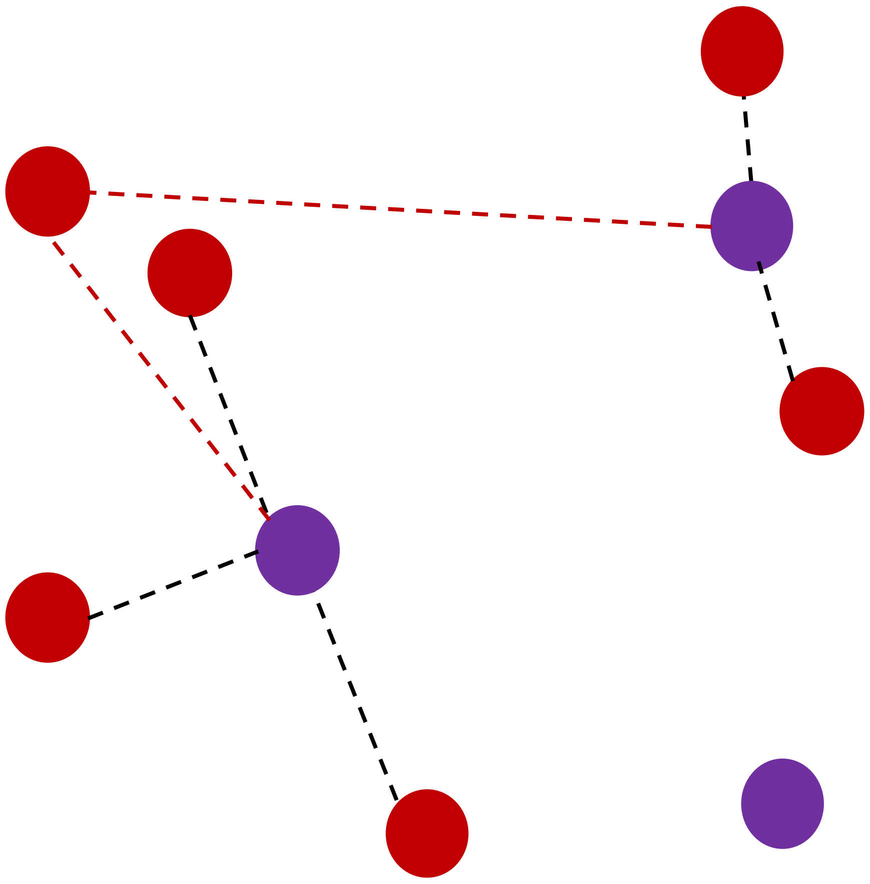
Centroid Clustering [CFLM19]



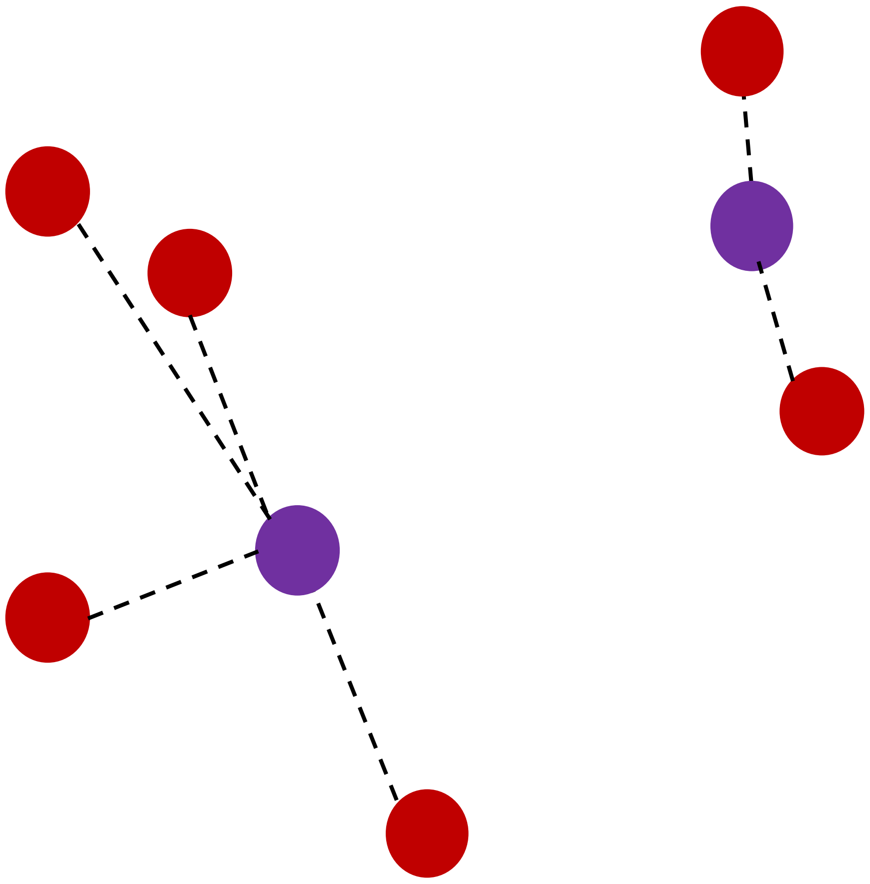
Centroid Clustering [CFLM19]



Centroid Clustering [CFLM19]



Centroid Clustering [CFLM19]



Centroid Clustering [CFLM19]

Greedy Capture gets a $1 + \sqrt{2} \approx 2.414$ approximation of the centroid core

Centroid Clustering [CFLM19]

Greedy Capture gets a $1 + \sqrt{2} \approx 2.414$ approximation of the centroid core

Proof:

Let $X' = \{x_1, x_2, \ldots, x_k\}$ be the cluster centers returned by greedy capture.

For contradiction, there exists a set of agents S of size $\lceil n/k \rceil$ and a center $y \in M \setminus X'$ such that:

$$(1 + \sqrt{2})d(i, y) < d(i, x) \; \forall i \in S, x \in X'$$

Centroid Clustering **[CFLM19]**

Greedy Capture gets a $1 + \sqrt{2} \approx 2.414$ approximation of the centroid core

Proof:

Let $X' = \{x_1, x_2, \dots, x_k\}$ be the cluster centers returned by greedy capture.

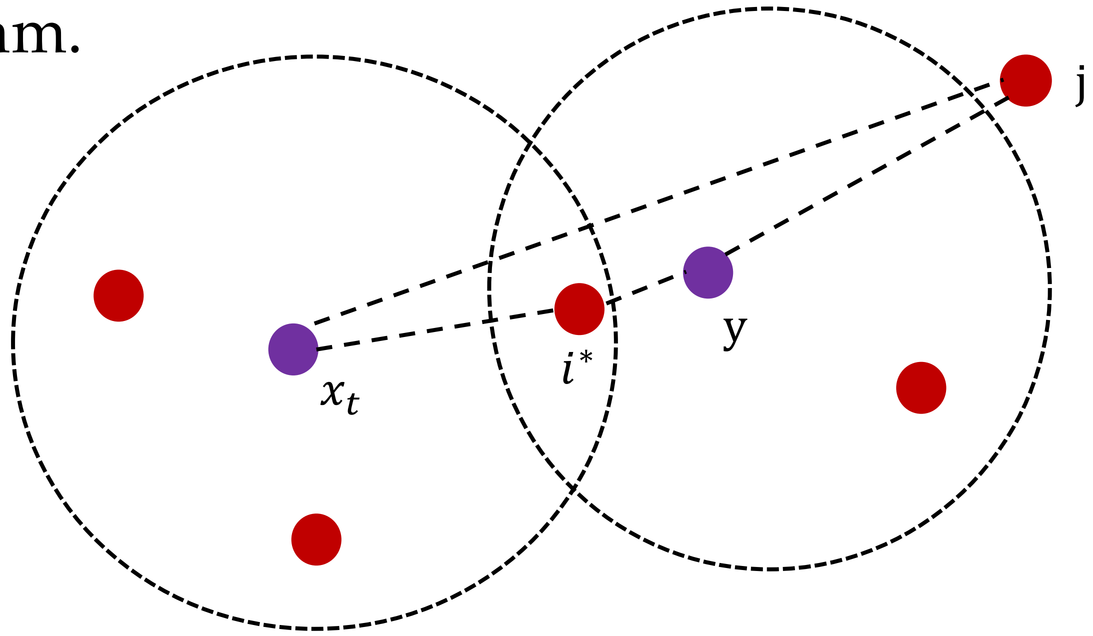
For contradiction, there exists a set of agents S of size $\lceil n/k \rceil$ and a center $y \in M \setminus X'$ such that:

$$(1+\sqrt{2})d(i,y) < d(i,x) \; \forall i \in S, x \in X'$$

i^* is the first agent from S captured by the algorithm.

x_t is the center that captured i^*.

At the timestep where x_t captures, y has not yet captured S. Thus, there must be some $j \in S$ with $d(j,y) \geq d(i^*, x_t)$.



Centroid Clustering [CFLM19]

Greedy Capture gets a $1 + \sqrt{2} \approx 2.414$ approximation of the centroid core

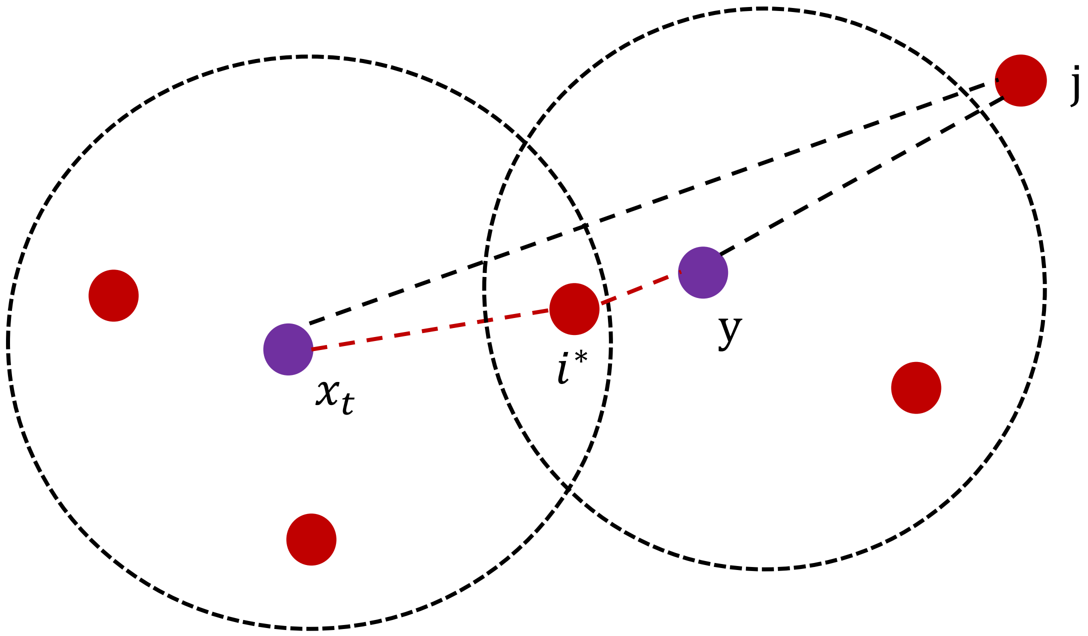
Proof:

Let $X' = \{x_1, x_2, \ldots, x_k\}$ be the cluster centers returned by greedy capture.

For contradiction, there exists a set of agents S of size $\lceil n/k \rceil$ and a center $y \in M \setminus X'$ such that:

$$(1+\sqrt{2})d(i,y) < d(i,x) \; \forall i \in S, x \in X'$$

$$\min\left(\frac{d(i^*,x_t)}{d(i^*,y)}, \frac{d(j,x_t)}{d(j,y)}\right)$$



Centroid Clustering [CFLM19]

Greedy Capture gets a $1+\sqrt{2} \approx 2.414$ approximation of the centroid core

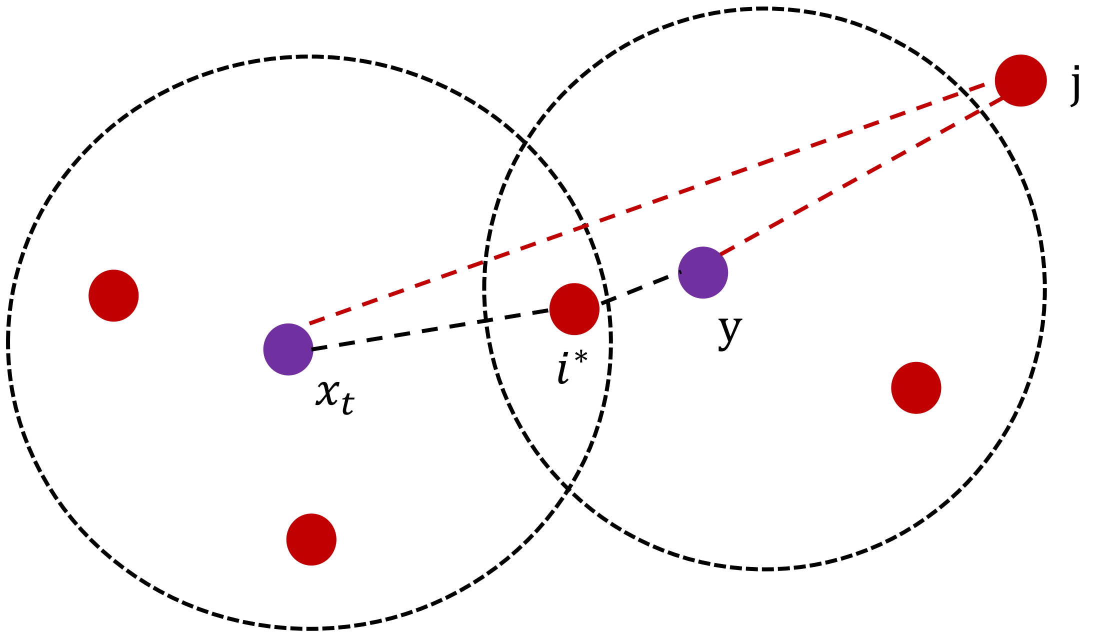
Proof:

Let $X' = \{x_1, x_2, \dots, x_k\}$ be the cluster centers returned by greedy capture.

For contradiction, there exists a set of agents S of size $\lceil n/k \rceil$ and a center $y \in M \setminus X'$ such that:

$$(1+\sqrt{2})d(i,y) < d(i,x) \;\forall i \in S, x \in X'$$

$$\min\left(\frac{d(i^*,x_t)}{d(i^*,y)}, \frac{d(j,x_t)}{d(j,y)}\right)$$



Centroid Clustering [CFLM19]

Greedy Capture gets a $1 + \sqrt{2} \approx 2.414$ approximation of the centroid core

Proof:

Let $X' = \{x_1, x_2, \ldots, x_k\}$ be the cluster centers returned by greedy capture.

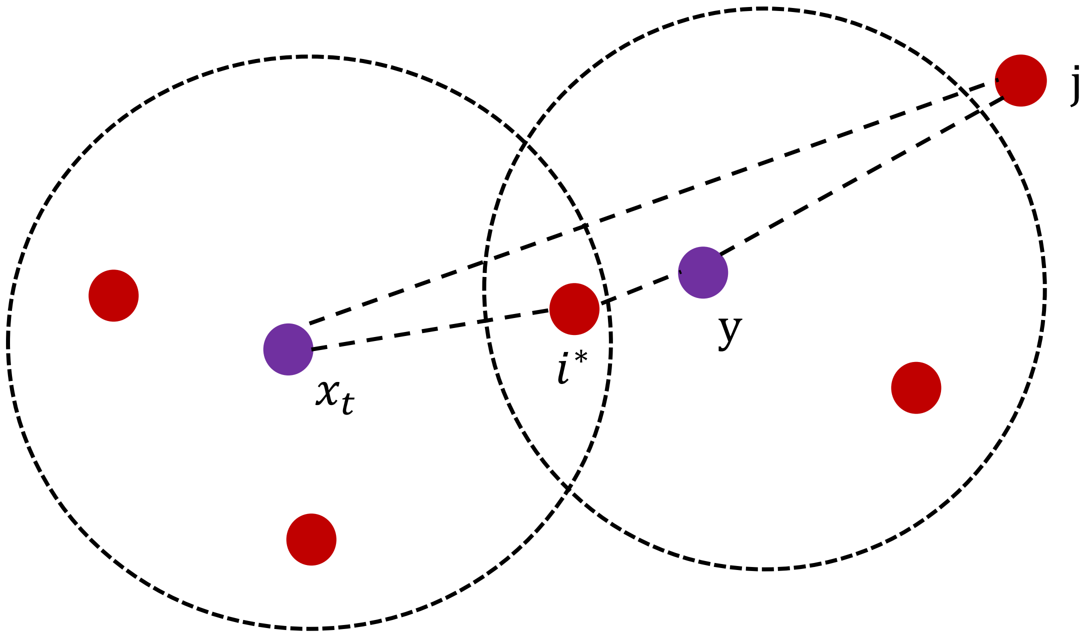
For contradiction, there exists a set of agents S of size $\lceil n/k \rceil$ and a center $y \in M \setminus X'$ such that:

$$(1+\sqrt{2})d(i,y) < d(i,x) \; \forall i \in S, x \in X'$$

$$\min\left(\frac{d(i^*,x_t)}{d(i^*,y)}, \frac{d(j,x_t)}{d(j,y)}\right) \le$$

$$\min\left(\frac{d(i^*,x_t)}{d(i^*,y)}, \frac{d(j,y)+d(y,i^*)+d(i^*,x_t)}{d(j,y)}\right) \le$$

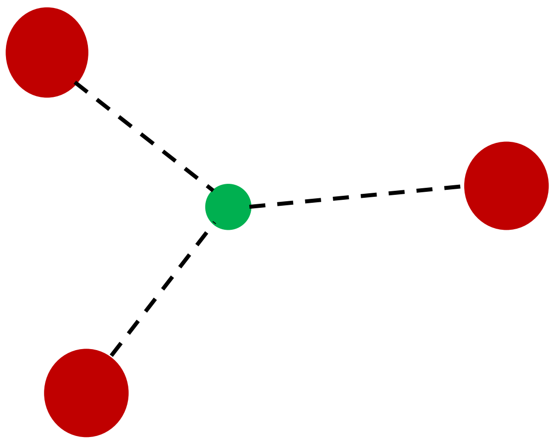
$$\min\left(\frac{d(j,y)}{d(i^*,y)}, 2 + \frac{d(i^*,y)}{d(j,y)}\right) \le 1+\sqrt{2}$$



Two Kinds of Fair Clustering Algorithms

- Assume the points have preferences over the clusters they are placed in.
  - **Most Common:** Want to be placed in clusters with points that are close to them.

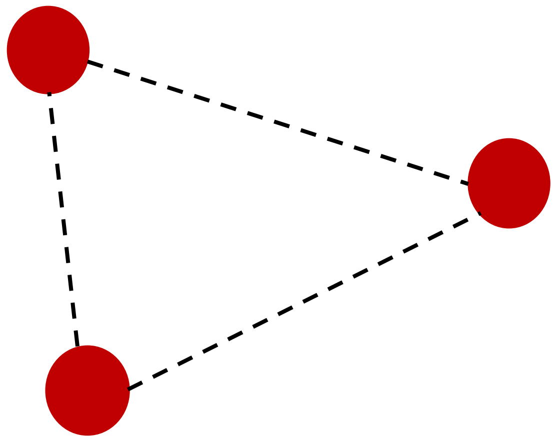
Centroid Preferences [CFLM19]



Each agent wants to be as close as possible to the center of their assigned cluster.

Chen, Fain, Lyu, and Munagala [2019]

Non-Centroid Preferences [CMS24]



Each agent wants to be as close as possible to other agents in their cluster.

Caragiannis, Micha, and Shah [2024]

Non-Centroid Clustering **[CMS24]**

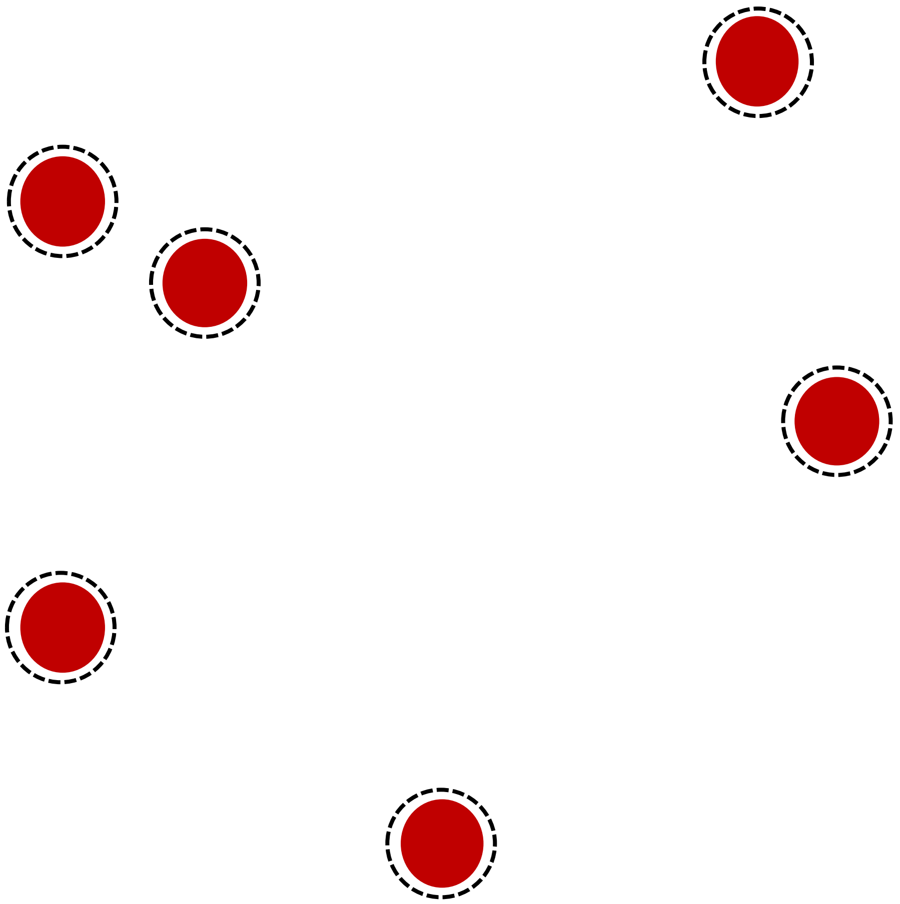
- **Given**
  - Metric space (M, d)
  - Set of data points $N \subseteq M$
  - ~~Set of possible cluster centers $X \subseteq M$~~

- Each data point's loss is its distance to the farthest agent in its cluster.
  - $\ell_i(C, x) = \max_{j \in C} d(i, j)$

Non-Centroid Clustering [CMS24]

Greedy Capture

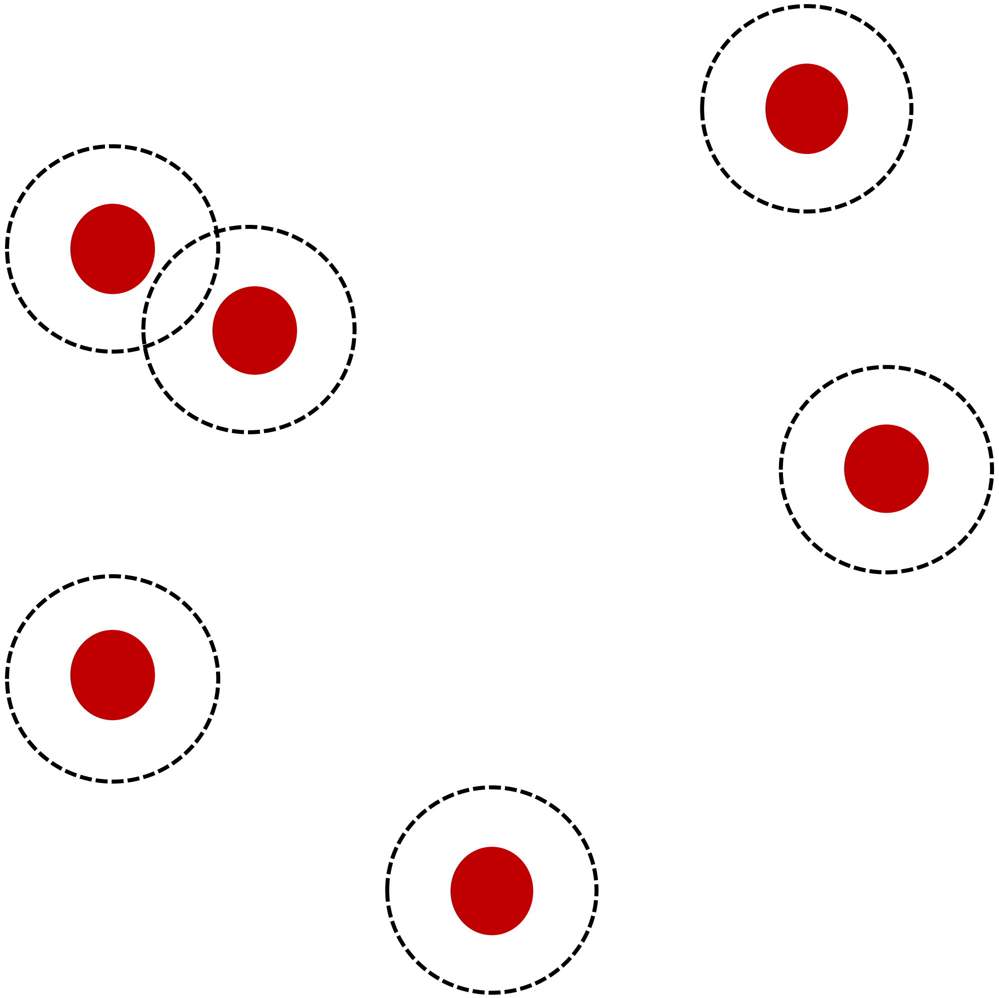
$n = 6, k = 2$



Non-Centroid Clustering [CMS24]

Greedy Capture

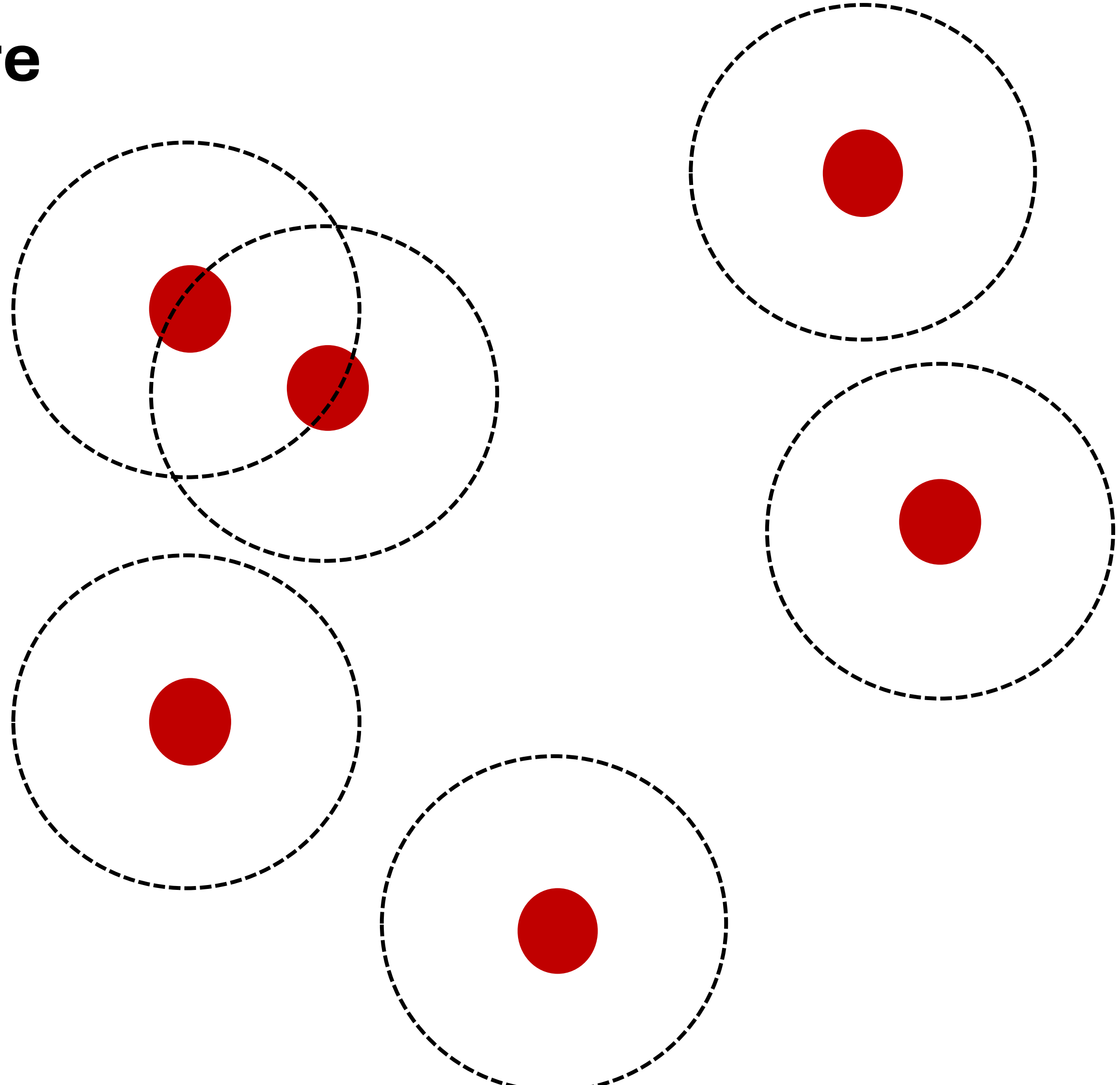
$n = 6, k = 2$



Non-Centroid Clustering [CMS24]

Greedy Capture

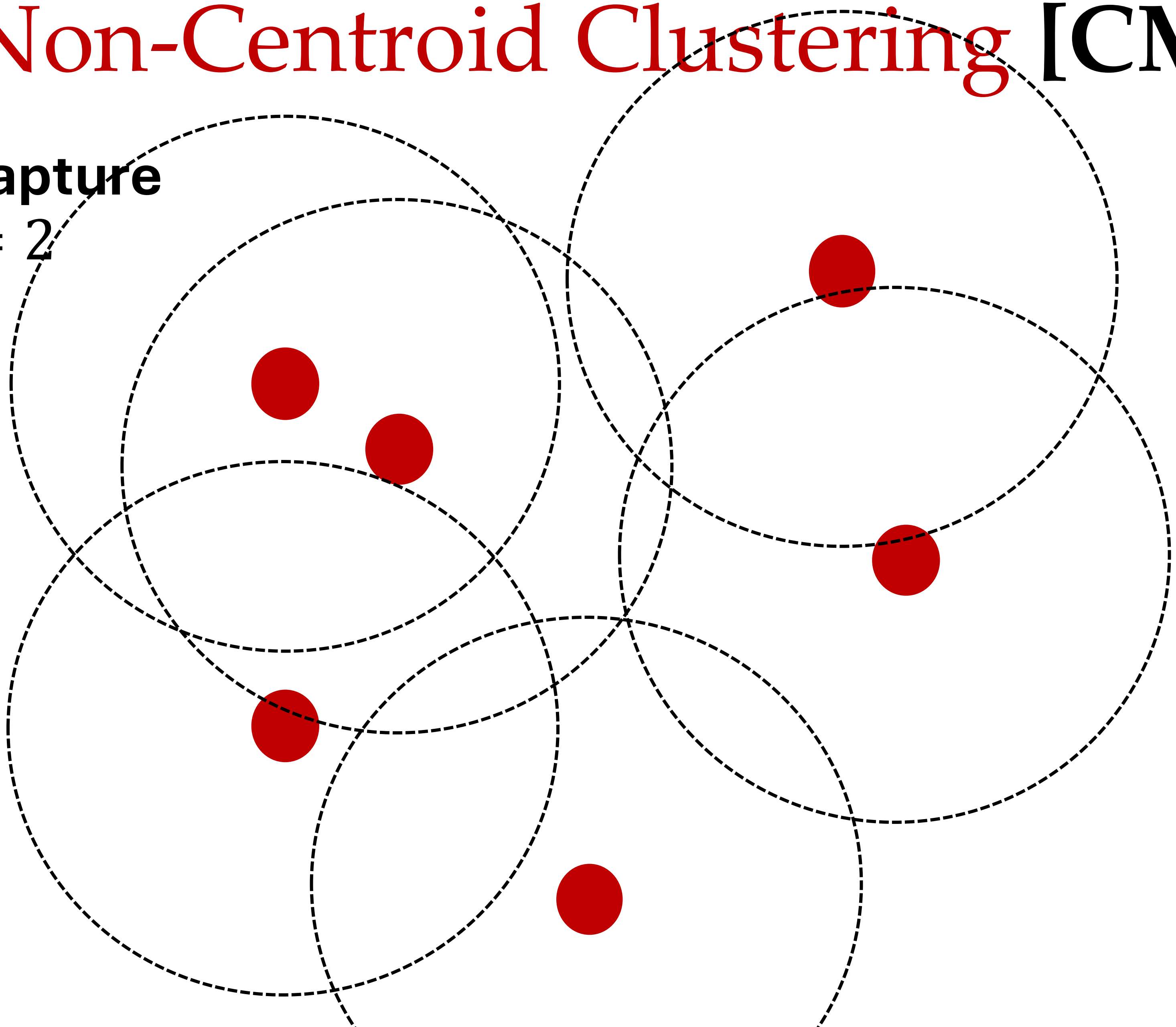
$n = 6, k = 2$



Non-Centroid Clustering **[CMS24]**

Greedy Capture

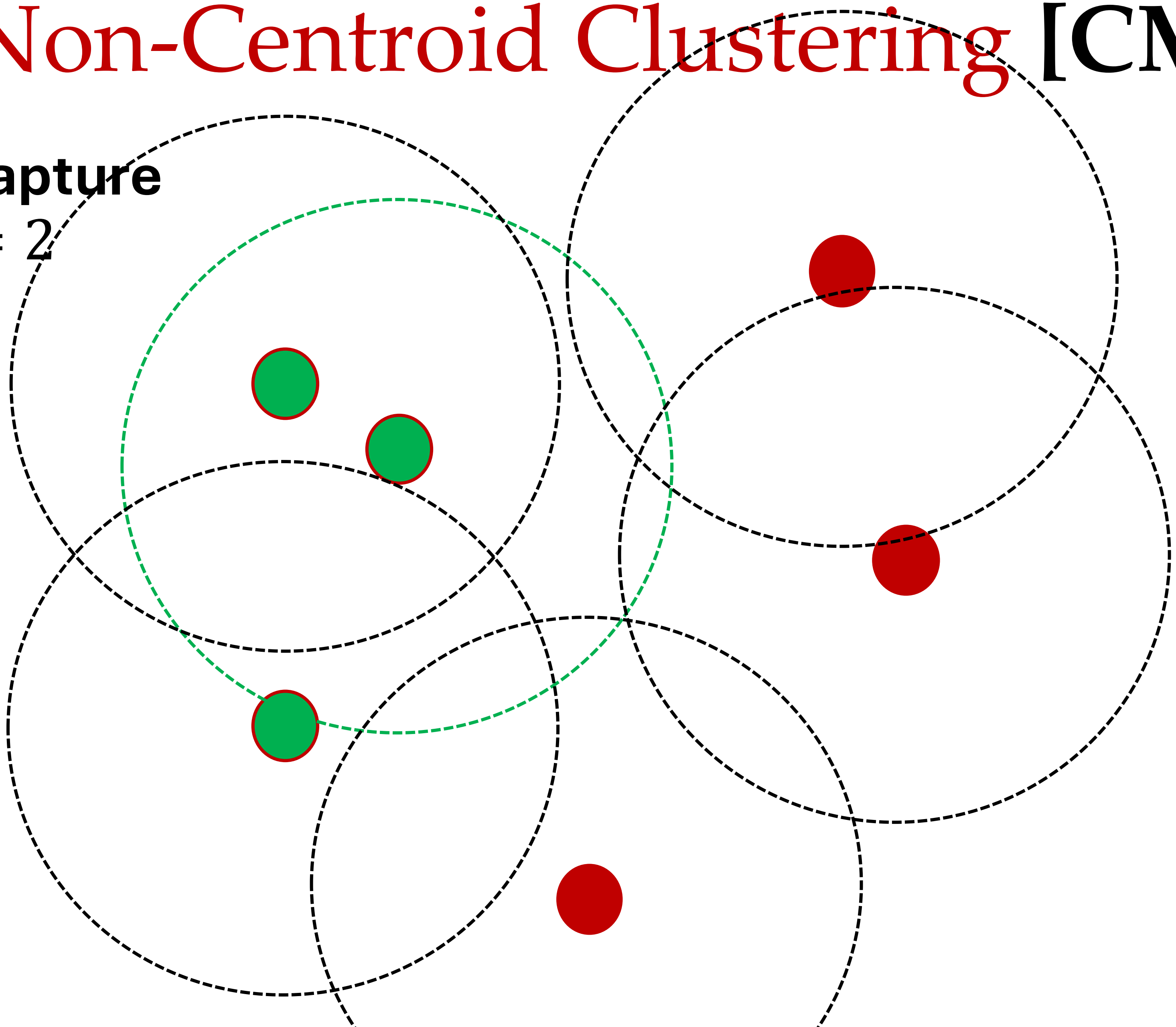
$n = 6, k = 2$



Non-Centroid Clustering [CMS24]

Greedy Capture

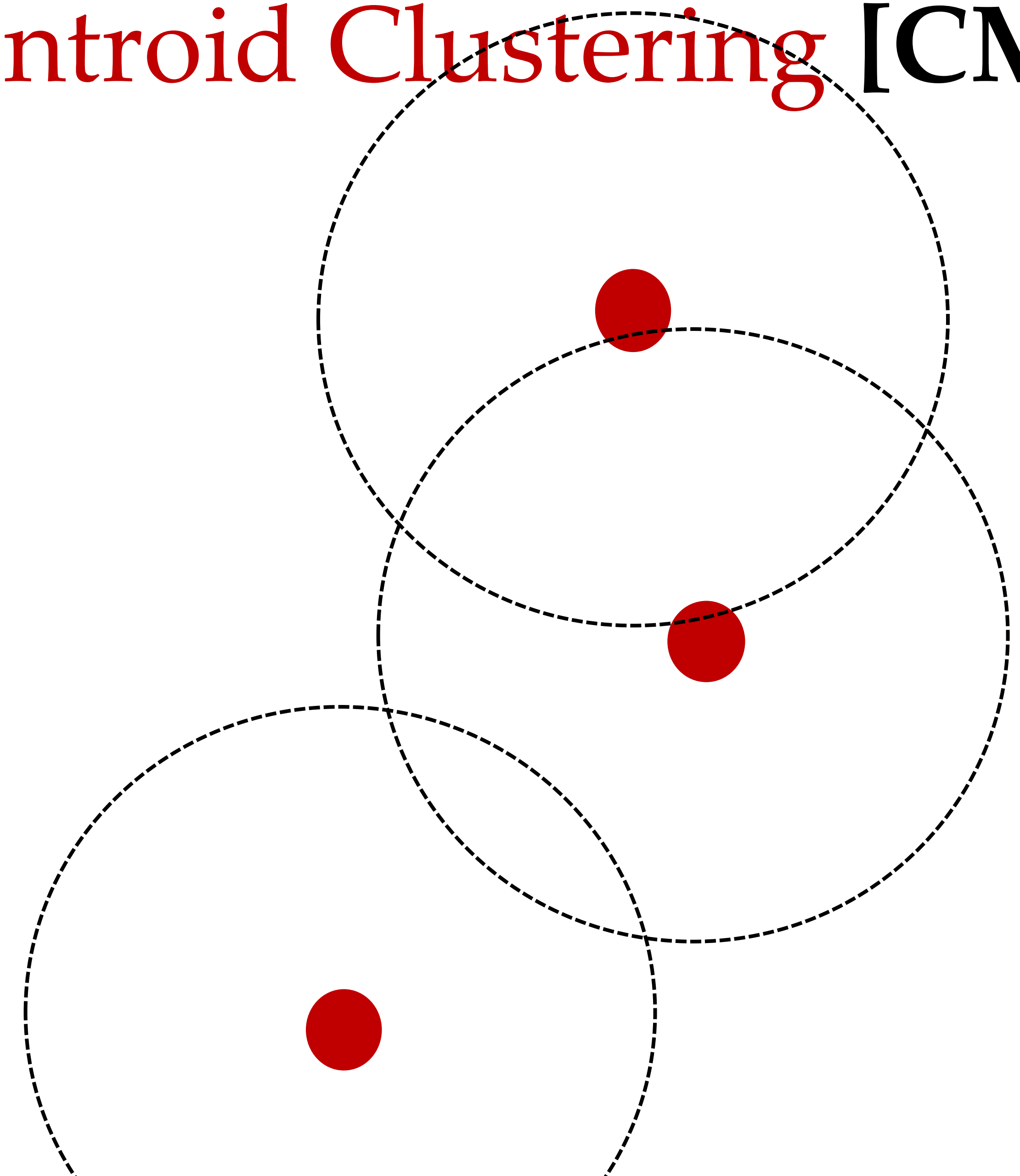
$n = 6, k = 2$



Non-Centroid Clustering [CMS24]

Greedy Capture

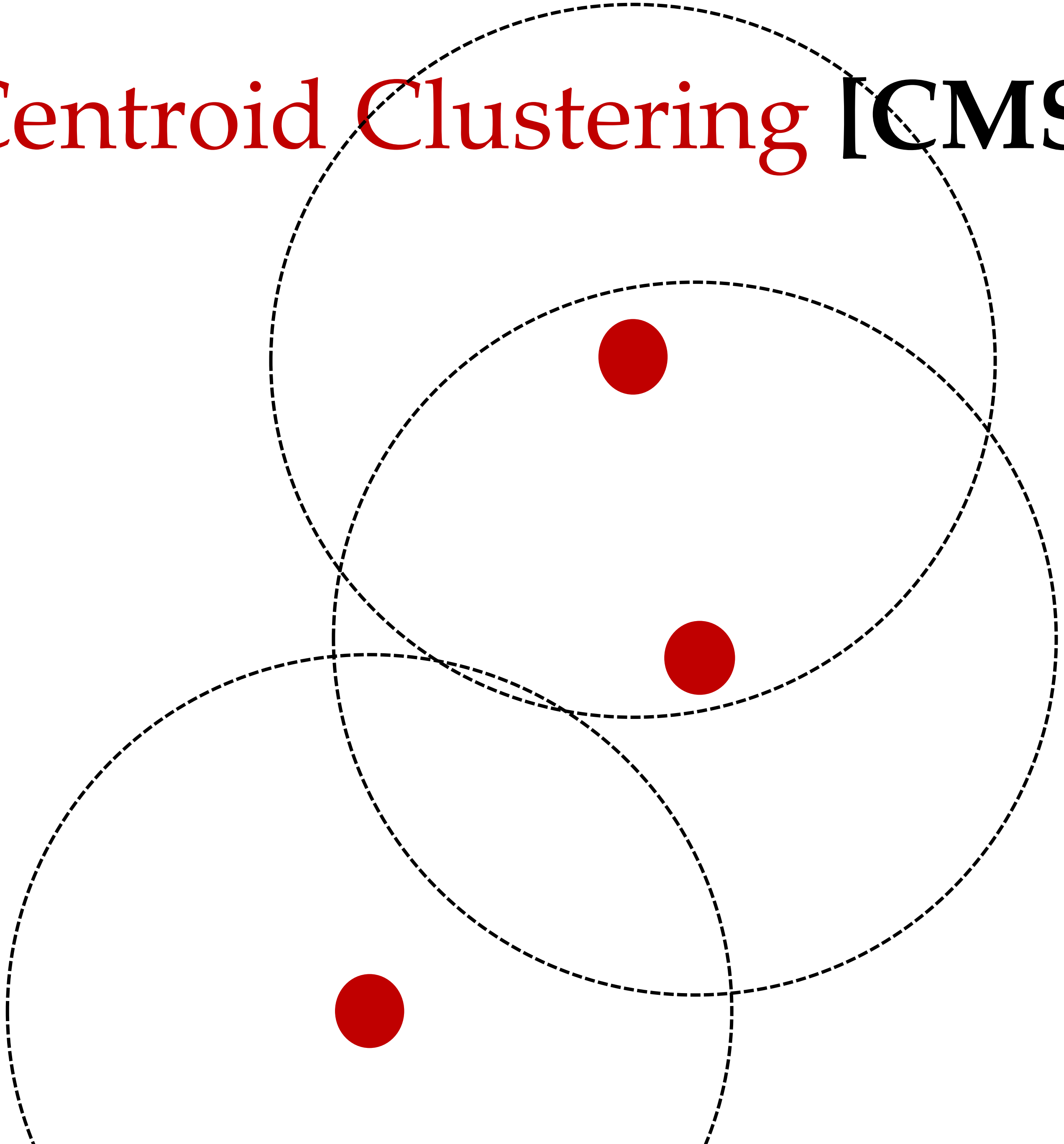
$n = 6, k = 2$



Non-Centroid Clustering [CMS24]

Greedy Capture

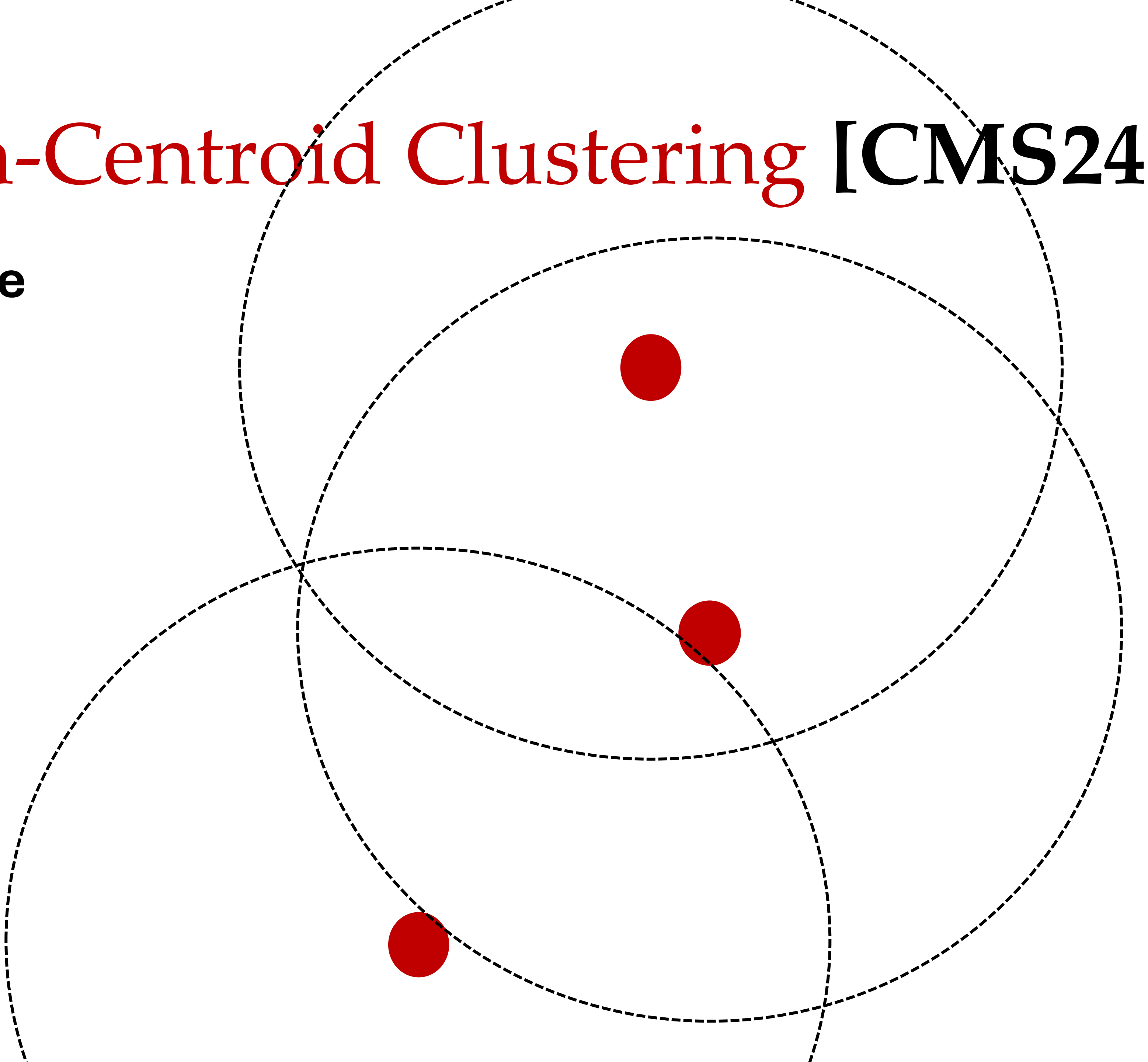
$n = 6, k = 2$



Non-Centroid Clustering [CMS24]

Greedy Capture

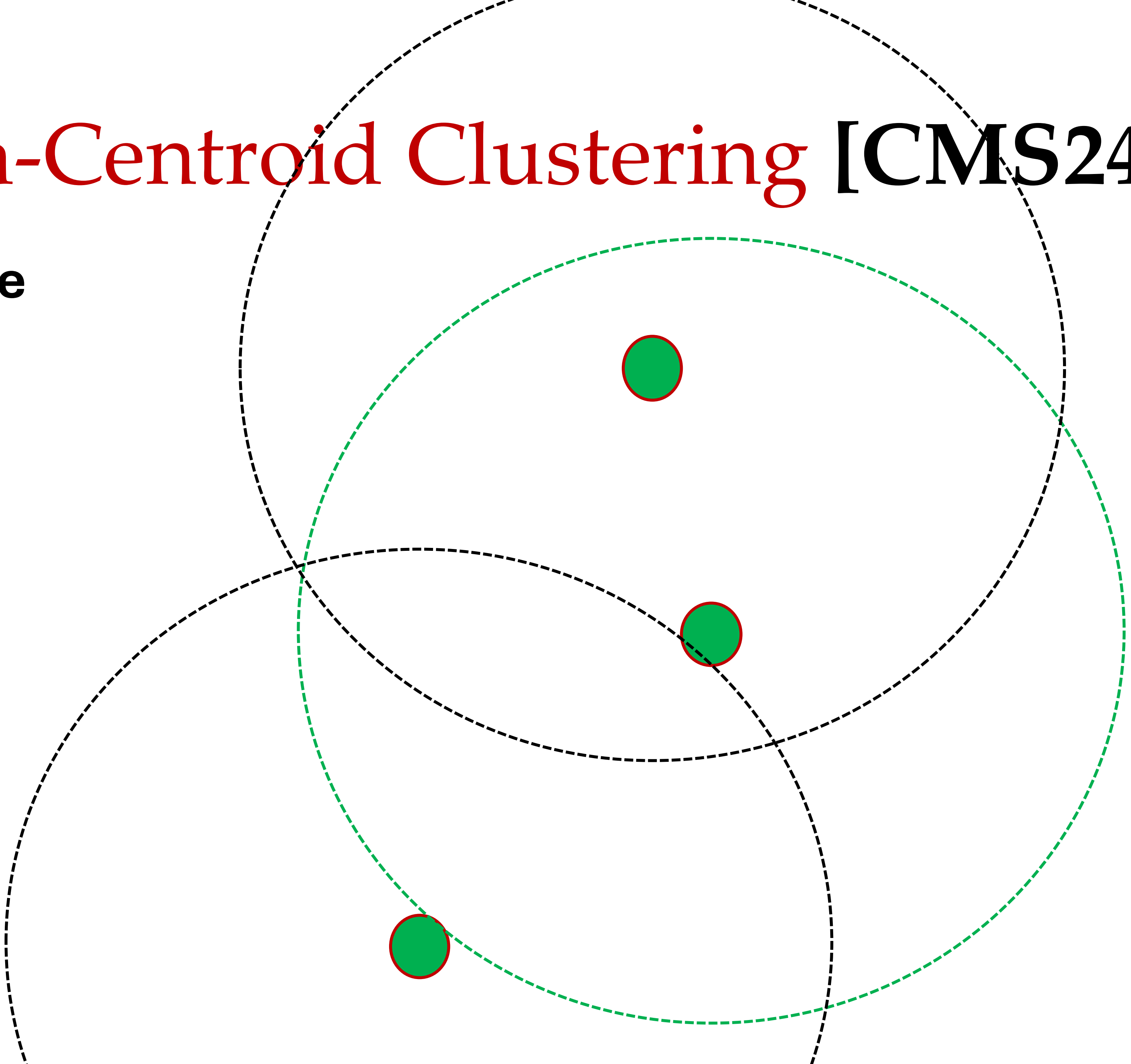
$n = 6, k = 2$



Non-Centroid Clustering [CMS24]

Greedy Capture

$n = 6, k = 2$



Non-Centroid Clustering [CMS24]

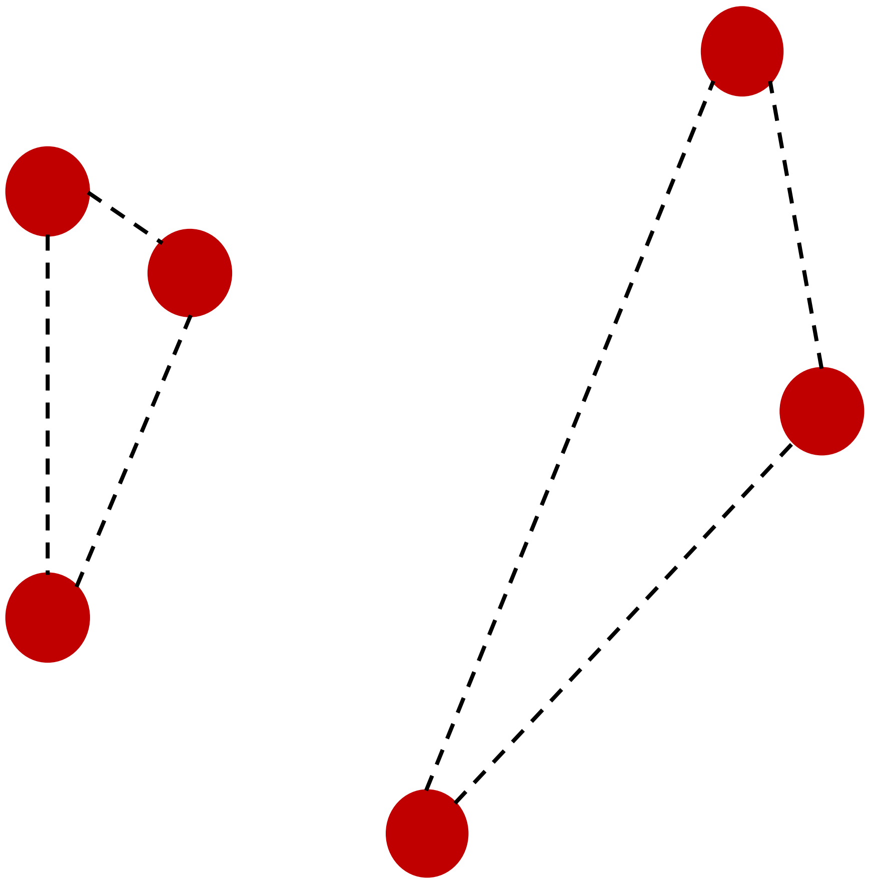
Greedy Capture

$n = 6, k = 2$

Non-Centroid Clustering [CMS24]

Greedy Capture

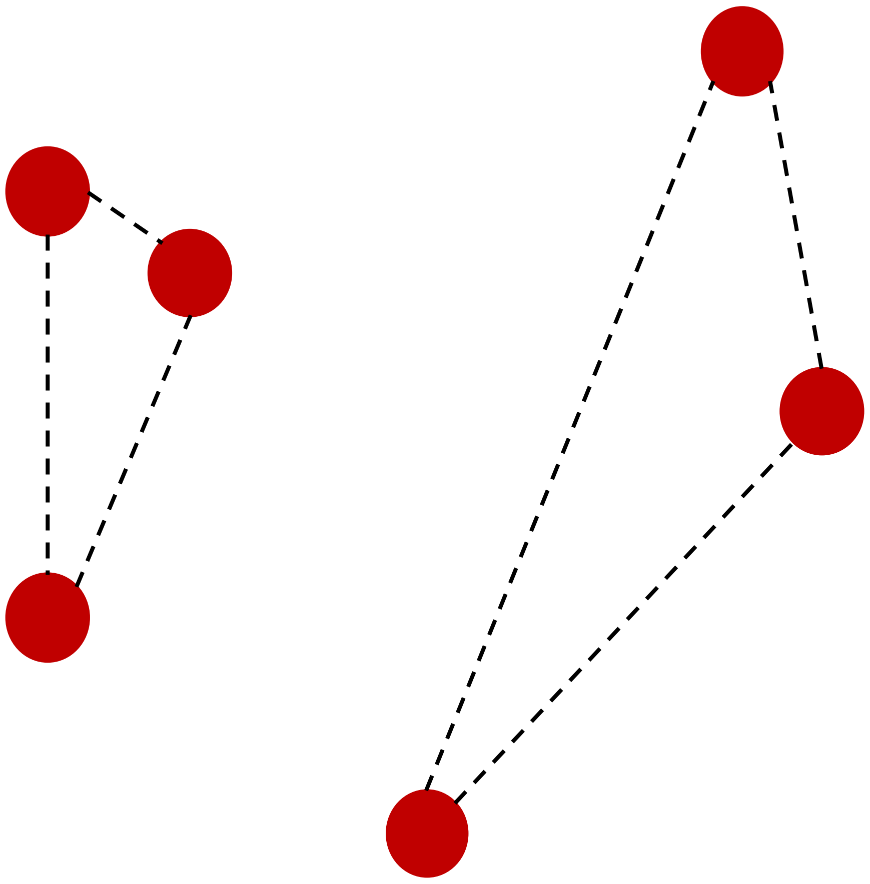
$n = 6, k = 2$



Non-Centroid Clustering **[CMS24]**

Greedy Capture

$n = 6, k = 2$

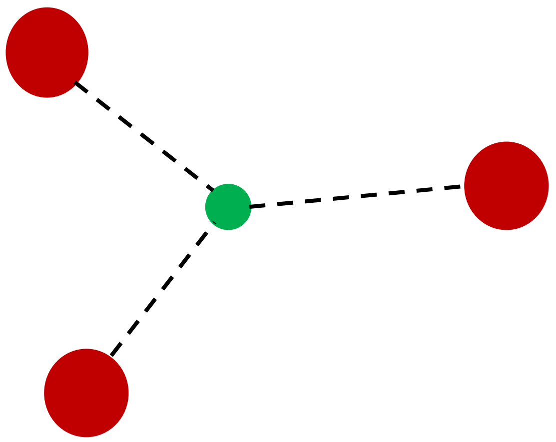


Greedy Capture gets a 2 approximation of the non-centroid core

Two Kinds of Fair Clustering Algorithms

In both settings, Greedy Capture is used to achieve a constant approximation of the core.

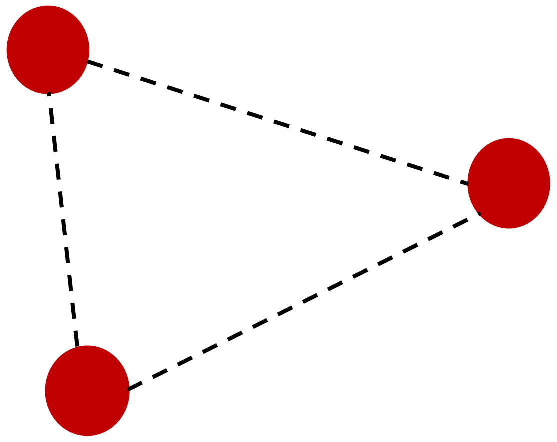
Centroid Preferences [CFLM19]



Each agent wants to be as close as possible to the center of their assigned cluster.

Chen, Fain, Lyu, and Munagala [2019]

Non-Centroid Preferences [CMS24]



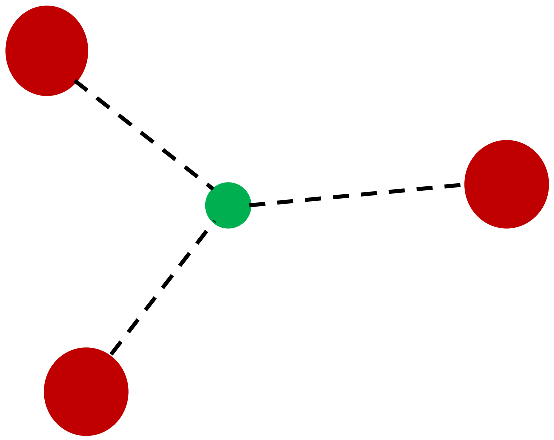
Each agent wants to be as close as possible to other agents in their cluster.

Caragiannis, Micha, and Shah [2024]

Two Kinds of Fair Clustering Algorithms

Question: Does Greedy Capture achieve a constant approximation of the core for both centroid and non-centroid loss in this model?

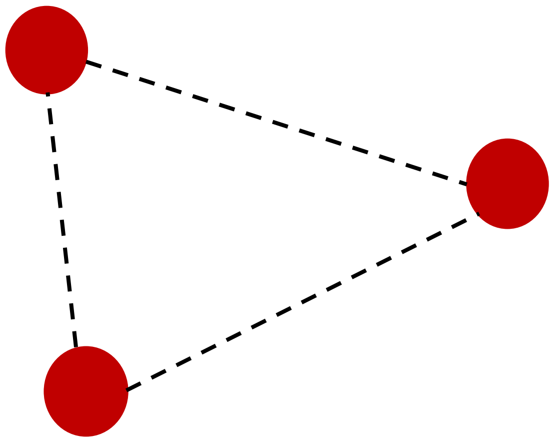
Centroid Preferences [CFLM19]



Each agent wants to be as close as possible to the center of their assigned cluster.

Chen, Fain, Lyu, and Munagala [2019]

Non-Centroid Preferences [CMS24]



Each agent wants to be as close as possible to other agents in their cluster.

Caragiannis, Micha, and Shah [2024]

Unified Clustering

Problem: There are some differences between Greedy Capture in the centroid and non-centroid settings.

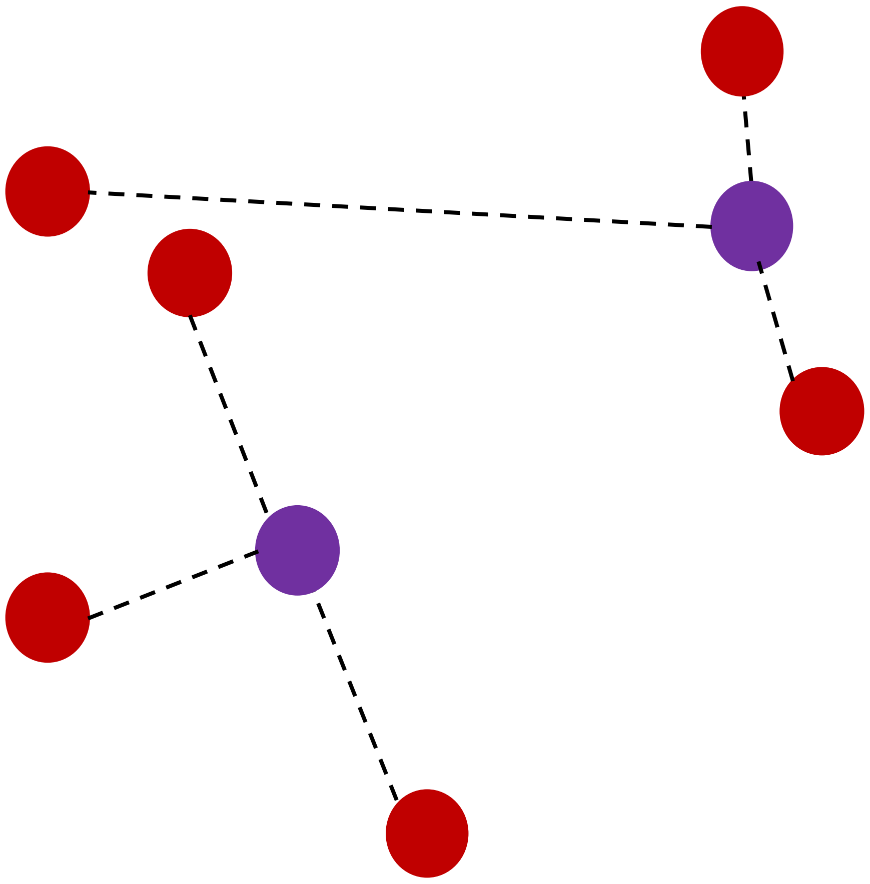
- **Difference 1:** In centroid clustering, the balls grow around the centers. In non-centroid, the balls grow around the agents.
    - This difference isn't a big deal

Unified Clustering

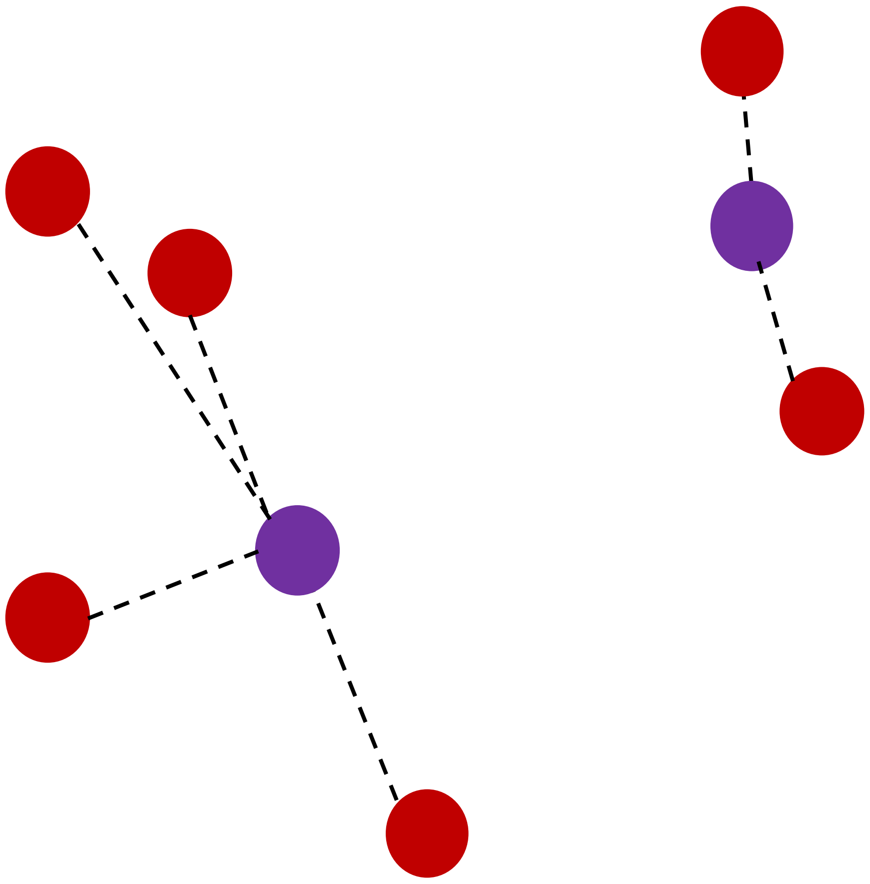
Problem: There are some differences between Greedy Capture in the centroid and non-centroid settings.

- **Difference 1:** In centroid clustering, the balls grow around the centers. In non-centroid, the balls grow around the agents.
  - This difference isn't a big deal
- **Difference 2:** In centroid clustering, agents aren't necessarily mapped to the center that captured them.
  - This is bad.

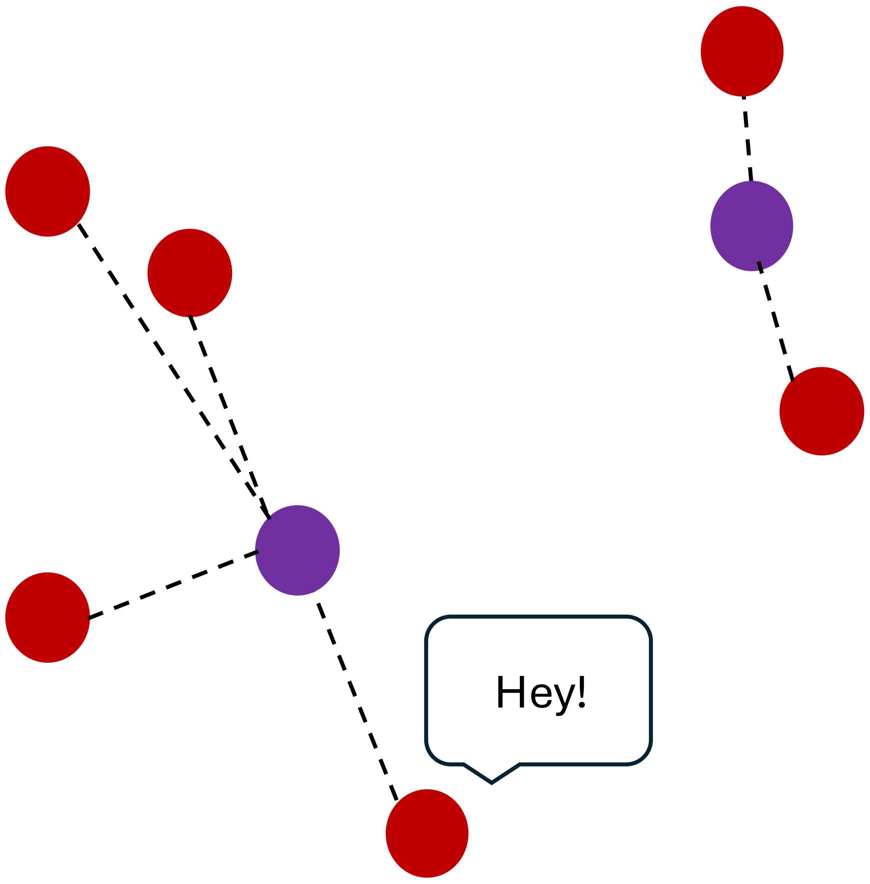
Unified Clustering



Unified Clustering



Unified Clustering



Centroid Clustering [CFLM19]

Greedy Capture gets a $1+\sqrt{2} \approx 2.414$ approximation of the centroid core

Proof:

Let $X' = \{x_1, x_2, \ldots, x_k\}$ be the cluster centers returned by greedy capture.

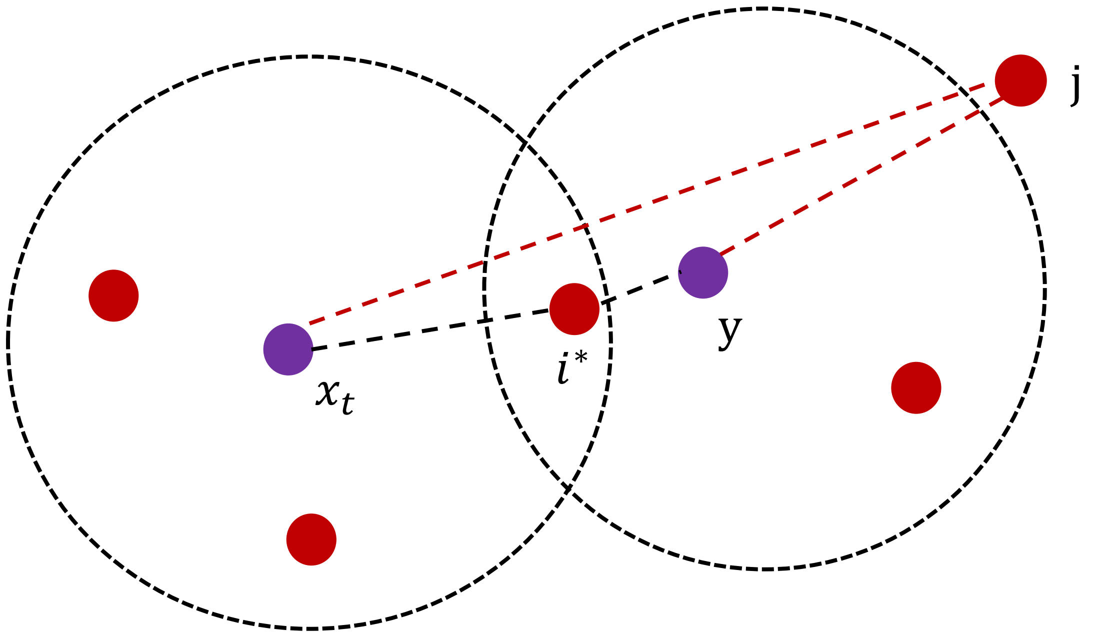
For contradiction, there exists a set of agents S of size $\lceil n/k \rceil$ and a center $y \in M \setminus X'$ such that:

$$(1+\sqrt{2})d(i,y) < d(i,x) \ \forall i \in S, x \in X'$$

$$\min\left(\frac{d(i^*,x_t)}{d(i^*,y)}, \frac{d(j,x_t)}{d(j,y)}\right) \leq$$

$$\min\left(\frac{d(i^*,x_t)}{d(i^*,y)}, \frac{d(j,y)+d(y,i^*)+d(i^*,x_t)}{d(j,y)}\right) \leq$$

$$\min\left(\frac{d(j,y)}{d(i^*,y)}, 2+\frac{d(i^*,y)}{d(j,y)}\right) \leq 1+\sqrt{2}$$



Unified Clustering

Question: Does Greedy Capture achieve a constant approximation of the core for both centroid and non-centroid loss in this model?

Unified Clustering

Question: Does Greedy Capture achieve a constant approximation of the core for both centroid and non-centroid loss in this model?

Answer: NO. In fact, stronger impossibility!

It is impossible to always find a clustering that gets a constant approximation of both types of core.

Unified Clustering

Updated Unified Clustering Model:

- **Given**
  - **Two Metric spaces** $(M, d^c), (M, d^m)$ over the same set of points M.
  - Set of data points $N \subseteq M$
  - Set of possible cluster centers X $\subseteq$ M

Unified Clustering

Updated Unified Clustering Model:

- **Given**
  - **Two Metric spaces** $(M, d^c), (M, d^m)$ over the same set of points M.
  - Set of data points $N \subseteq M$
  - Set of possible cluster centers $\mathrm{X} \subseteq \mathrm{M}$

- Find a clustering $C = \{(\mathrm{C}_1, \mathrm{x}_1), (\mathrm{C}_2, \mathrm{x}_2), \dots, (\mathrm{C}_\mathrm{k}, \mathrm{x}_\mathrm{k})\}$,

- Each kind of loss comes from a different metric space
  - Centroid Loss: $\ell_i^c(C, x) = d^c(i, x)$.
  - Non-Centroid Loss: $\ell_i^m(C, x) = \max_{\mathrm{j} \in \mathrm{C}} d^m(i, j)$.
  - **Unified Loss:** $\ell_\mathrm{i}(\mathrm{C}, \mathrm{x}) = \ell_i^m(C, x) + \ell_i^c(C, x)$

Unified Clustering

In this Unified Model:

There exists an algorithm that gets a **3-approximation** of the core for **any** two metric spaces.

Conclusion

- **For Social Choice**
  - Many interesting theoretical questions remain.
  - Better bounds on the core for both centroid and non-centroid loss.

- **For Machine Learning**
  - Generally, consider the core objective when clustering people.
    - Which common clustering algorithm get the best core approximation in practice?
    - How does the core interact with traditional clustering objectives?