Cluster Analysis

Center-based, k-centers, k-means, k-means++

By Ilai Fallach (ifallach@gmail.com)

(*) Based on https://www.cs.cornell.edu/jeh/book.pdf

Lecture Outline

- Introduction to clustering - definition & motivation
- Mathematical oriented
- Center-based clustering algorithms
- Some benchmarking

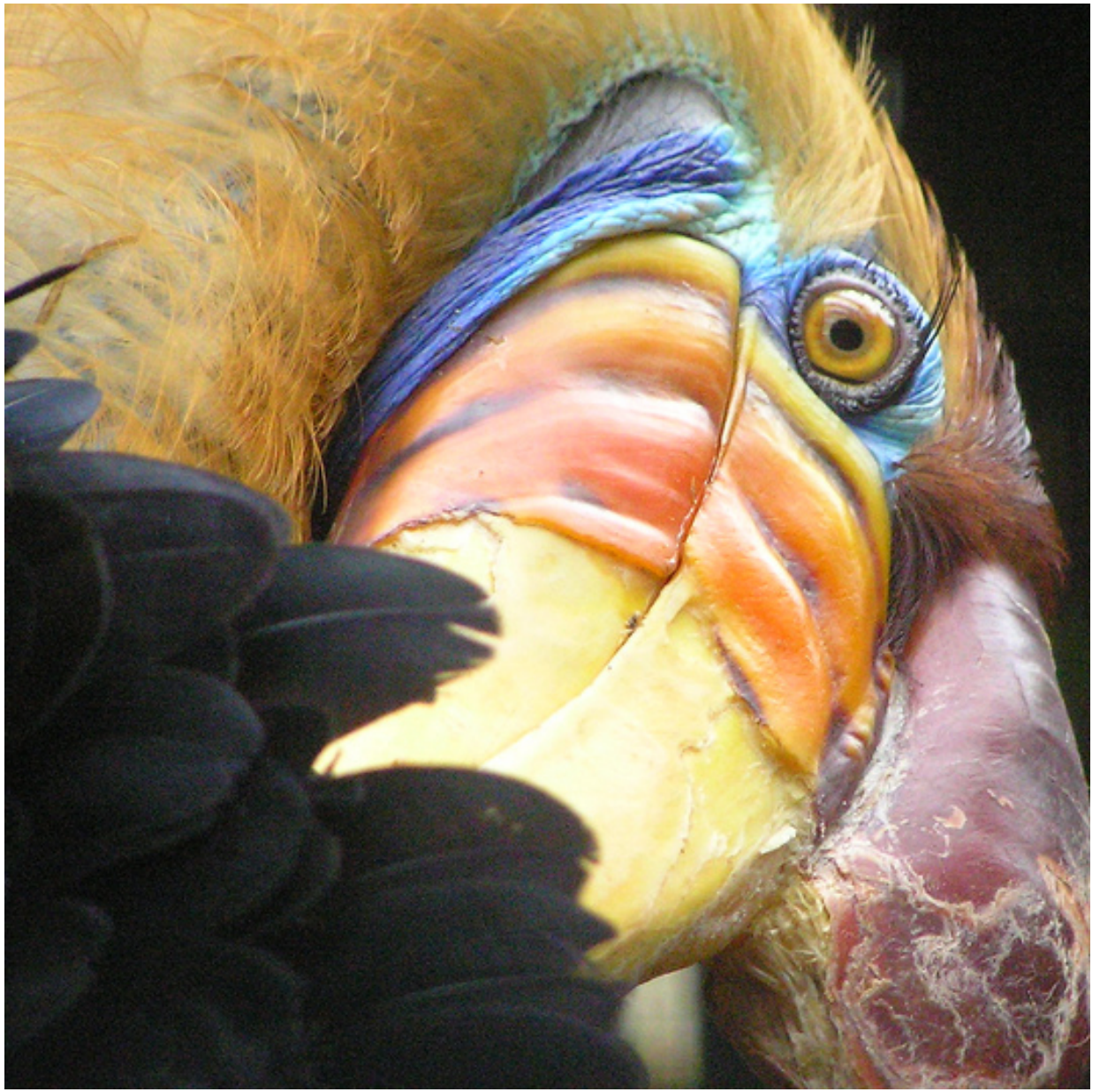
Clustering Problem Definition

Partitioning a set of objects into subsets according to some desired criterion.

- In plain words: form groups of similar things.
- An unsupervised learning problem.

Motivation

Compressing an image by reducing the number of colors it contains.



Images taken from: http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=MachineLearning&doc=exercises%2Fex9%2Fex9.html

Motivation

Image is compressed by choosing 16 most "representative" colors.

Images taken from: http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=MachineLearning&doc=exercises%2Fex9%2Fex9.html

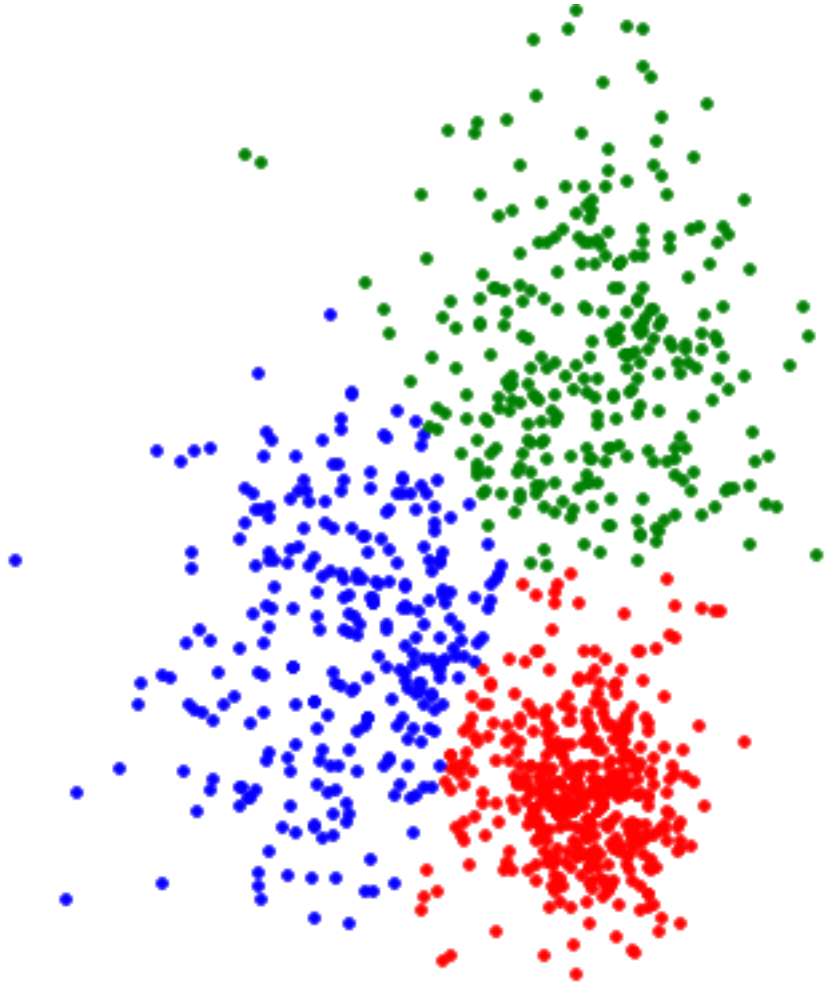
Applications

- Fire station, warehouse, hospital locationing
- Image segmentation
- Market segmentation (group similar customers)
- Group similar images, documents, medical patients
- Compression (Image, audio, video)
- Bioinformatics
- Recommender systems
- And more…

Center-based clusters

Clustering can be defined by k "center points" $c_1, \ldots, c_k$, with each data point assigned to whichever center point is closest to it (by some metric).

We need a objective (cost) function to define what is a good center. Three standard objectives often used are **k-center**, **k-median**, and **k-means** clustering.



k-center

Clustering

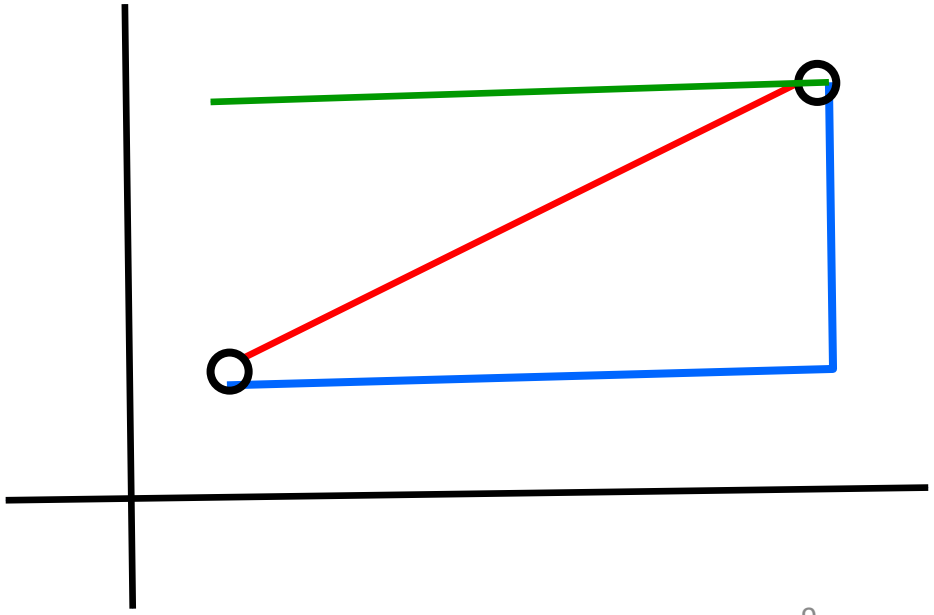
Metric Space

A set X and a function $d: X \times X \to \mathbb{R}$ s.t:

1. $d(x, y) \geq 0$ (Non-negativity)
2. $d(x, y) = 0 \Leftrightarrow x = y$ (Identity)
3. $d(x, y) = d(y, x)$ (Symmetry)
4. $d(x, y) \leq d(x, z) + d(z, y)$ (Trinagle enq)

Examples

1. $L_2: d(x, y) = \sqrt{\left(x_1^{(1)} - x_2^{(1)}\right)^2 + \left(x_1^{(2)} - x_2^{(2)}\right)^2}$
2. $L_1: d(x, y) = \left|x_1^{(1)} - x_2^{(1)}\right| + \left|x_1^{(2)} - x_2^{(2)}\right|$
3. $L_\infty: d(x, y) = \max\left\{\left|x_1^{(1)} - x_2^{(1)}\right|, \left|x_1^{(2)} - x_2^{(2)}\right|\right\}$



k-center clustering

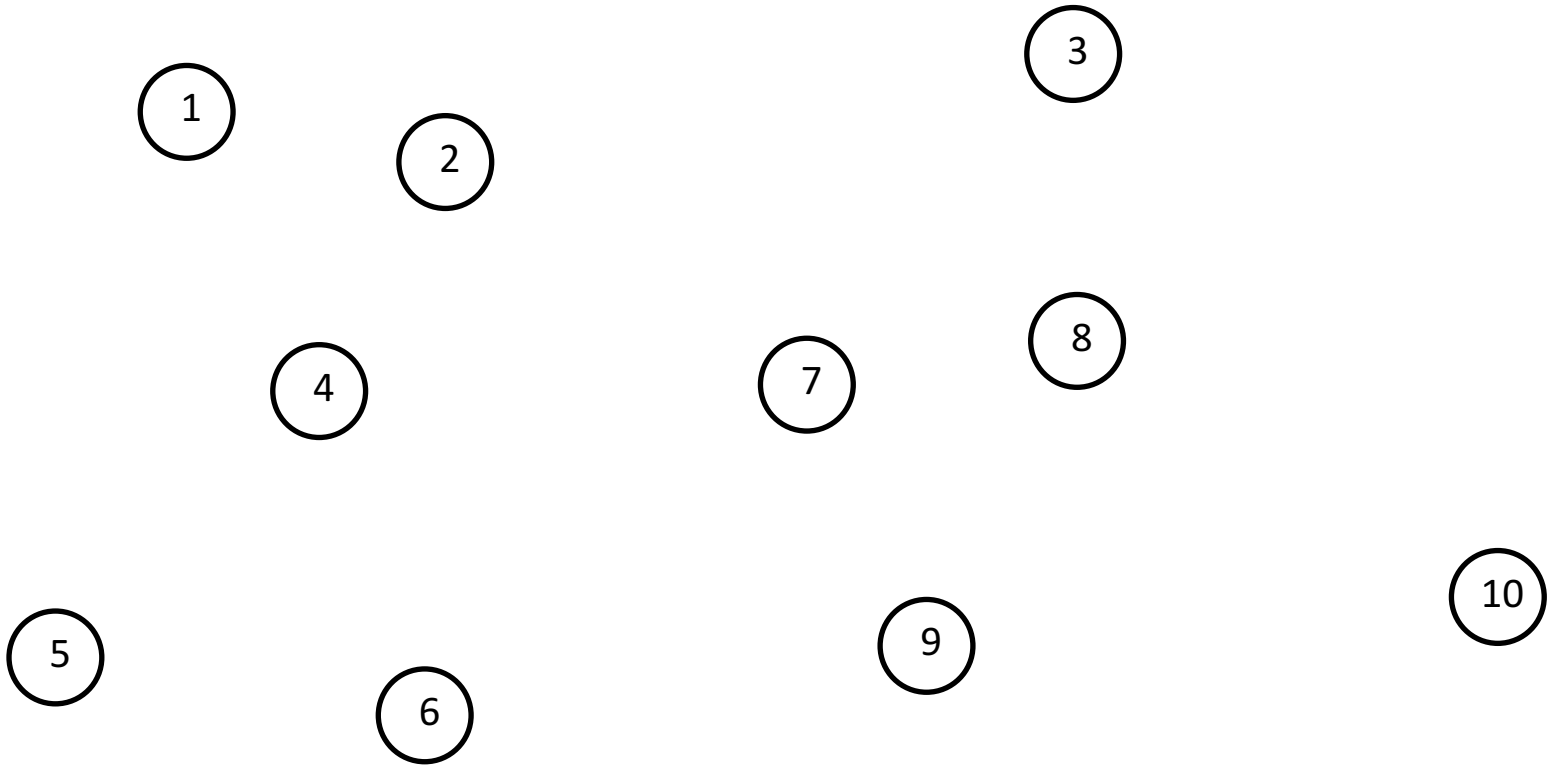
Find a partition $C = \{C_1, \ldots, C_k\}$ of our data into k clusters, with corresponding centers $\{c_1, \ldots, c_k\}$, to minimize the *maximum* distance between any data point and the center of its cluster:

$$\phi_{kcenter}(C) = \max_{j=1}^{k} \max_{a_i \in C_j} |a_i - c_j|$$

Many times called the "fire-station location problem".

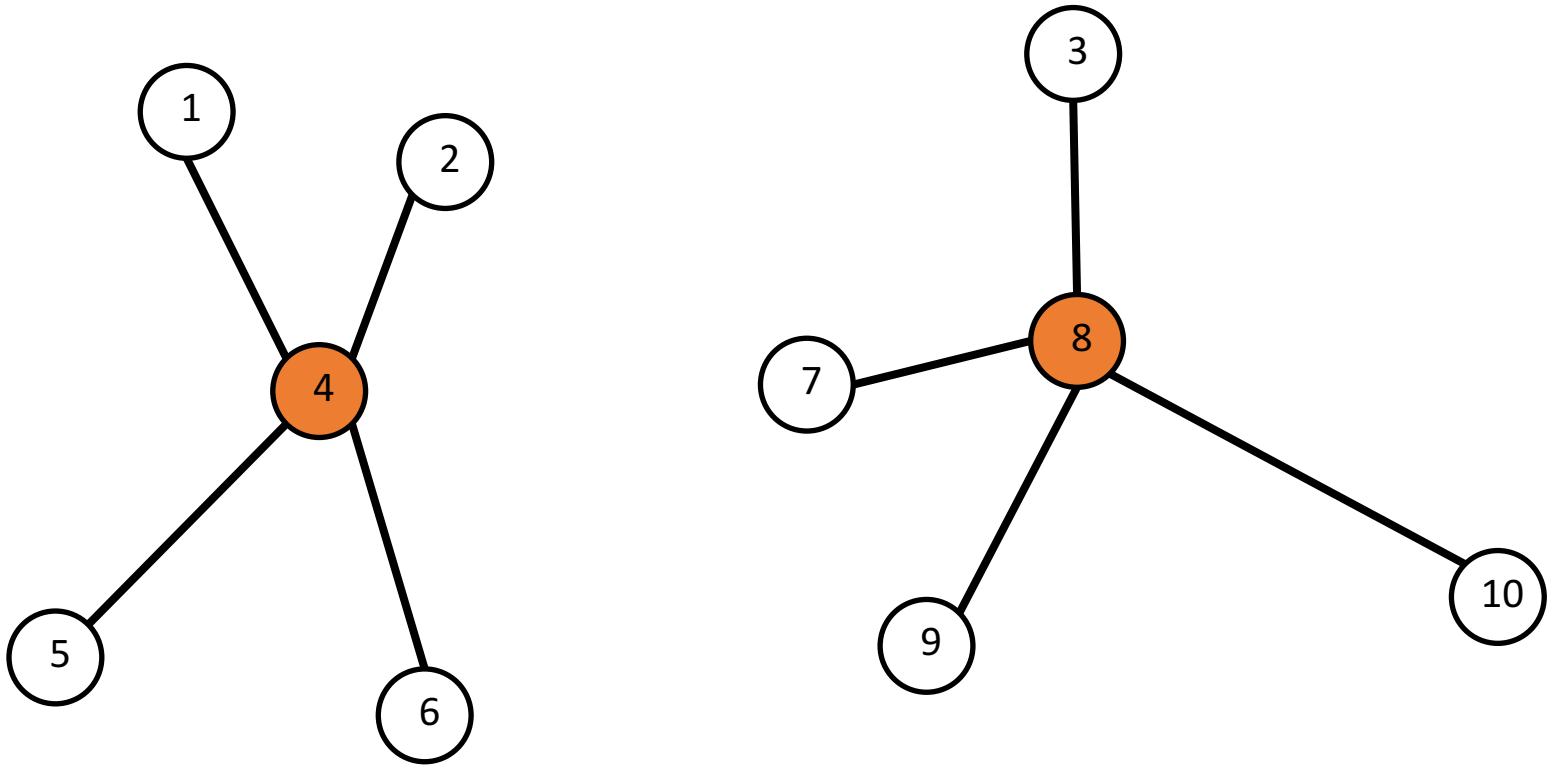
k-center clustering

Suppose $k = 2$ and we could pick only from existing points:



k-center clustering

Suppose $k = 2$ and we could pick only from existing points:



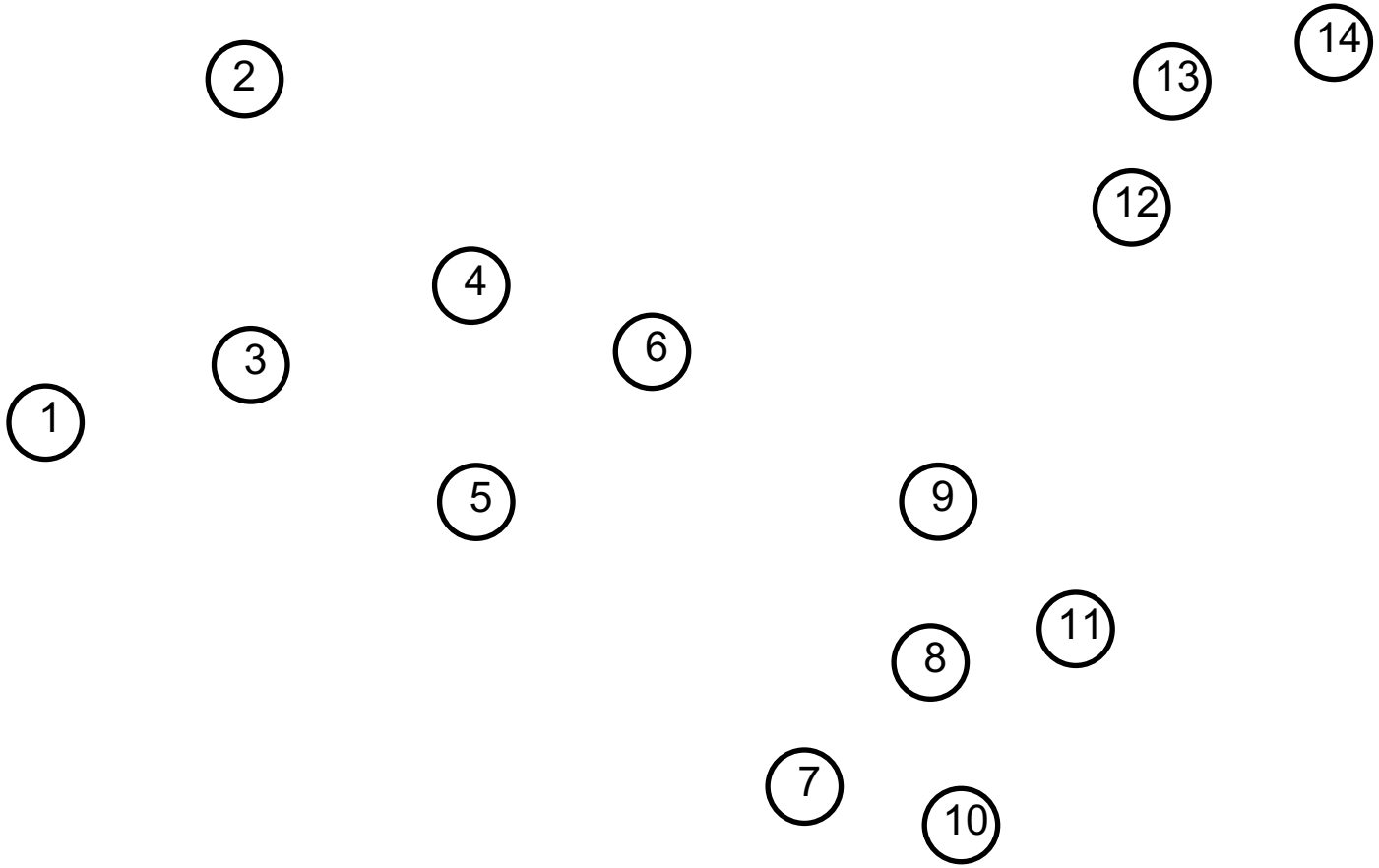
Any ideas for such an algorithm?

Farthest-first traversal algorithm

1. Pick a random data point to be the first cluster center.
2. At time t, for $t = 2,3, \ldots, k$, pick the farthest data point from any existing cluster center; make it the t_{th} cluster center.

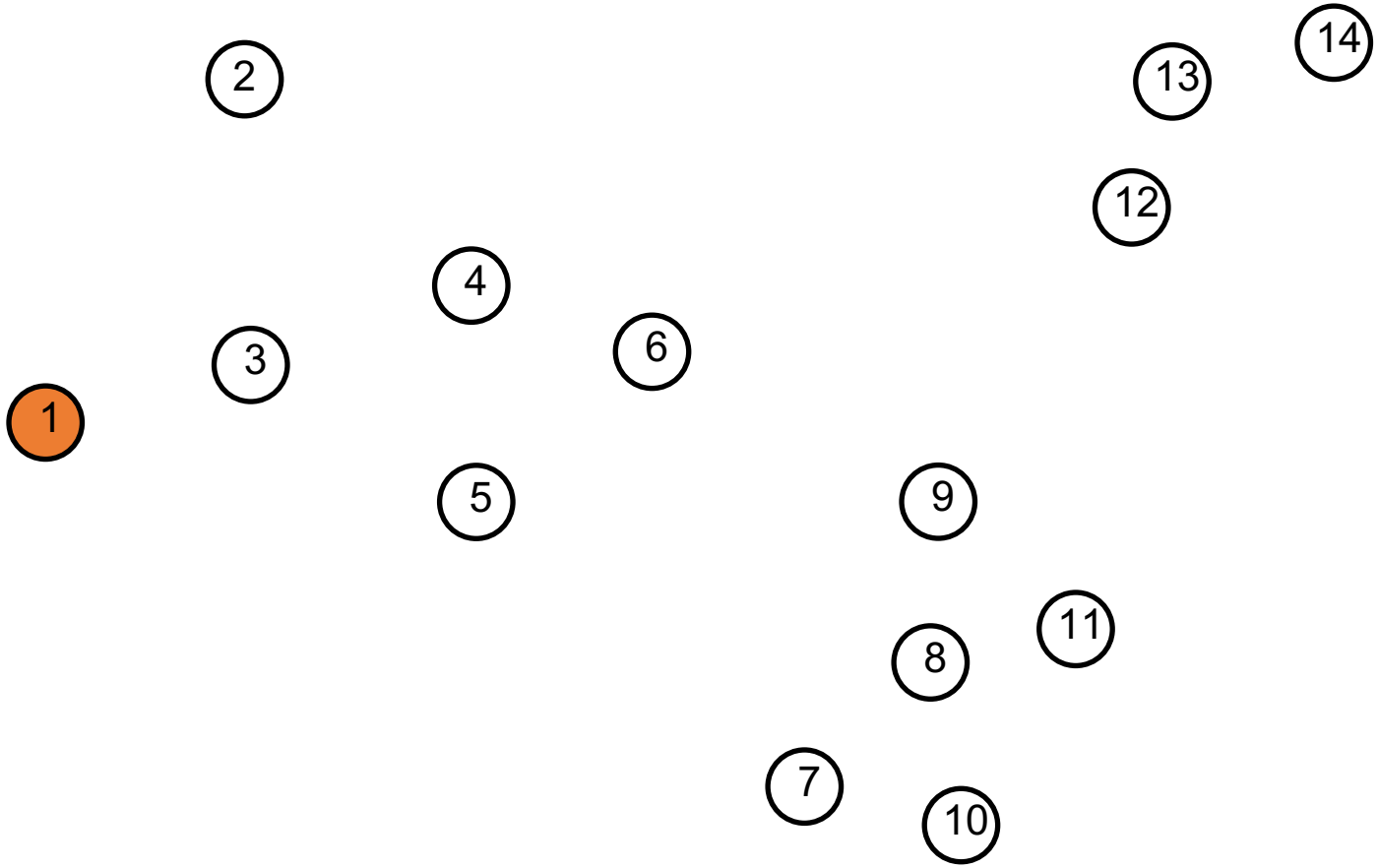
Example

Step t=0



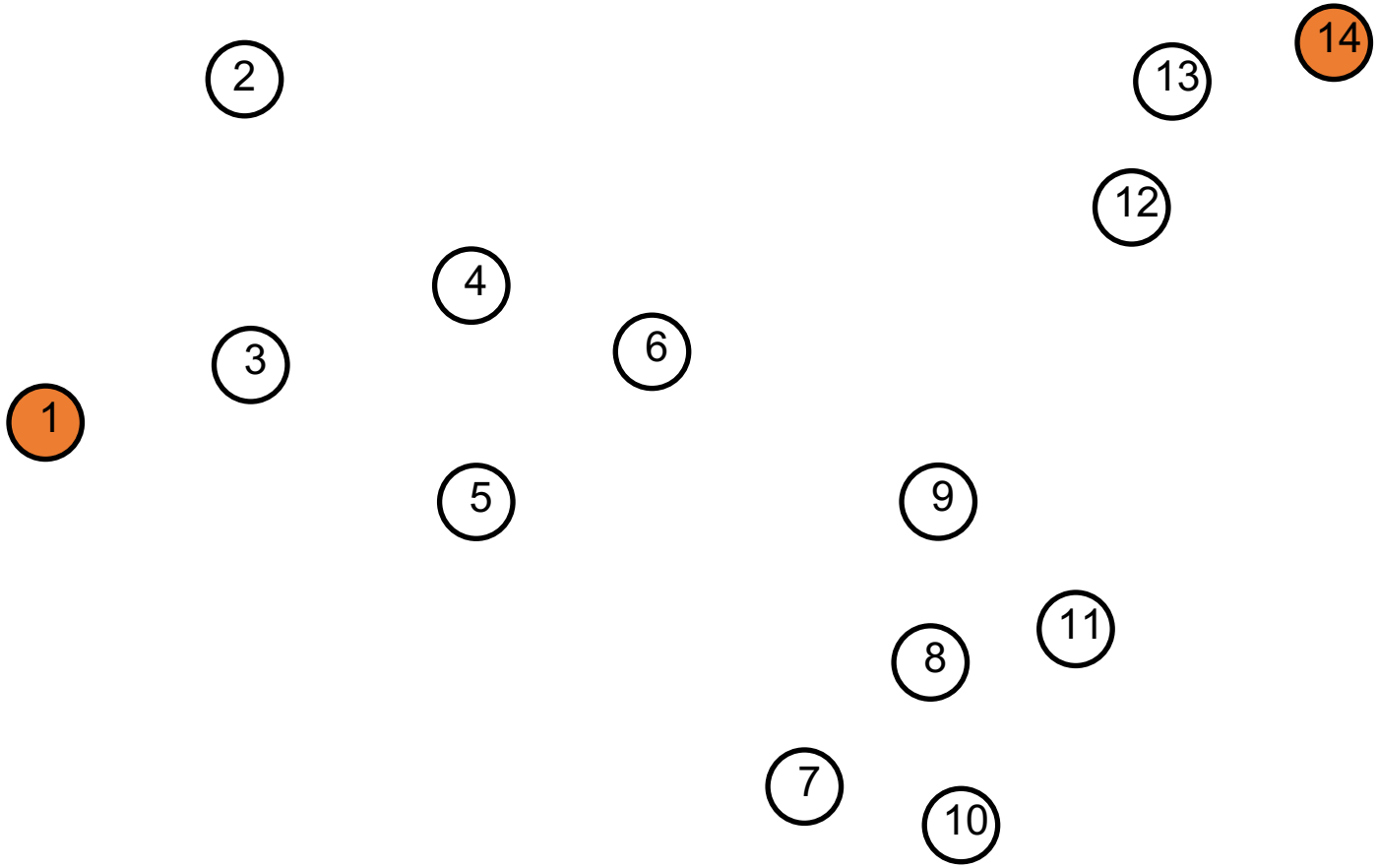
Example

Step t=1



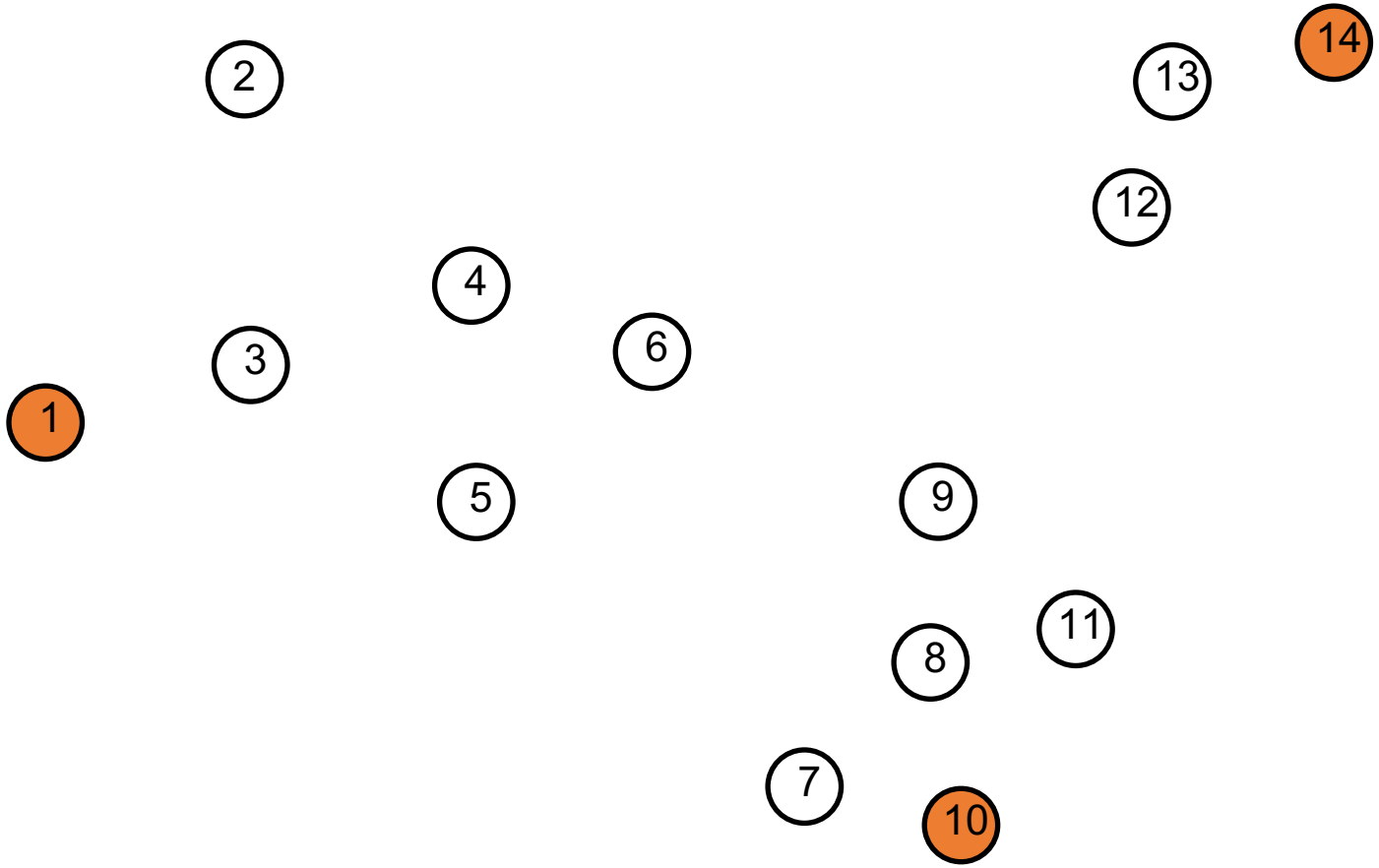
Example

Step t=2



Example

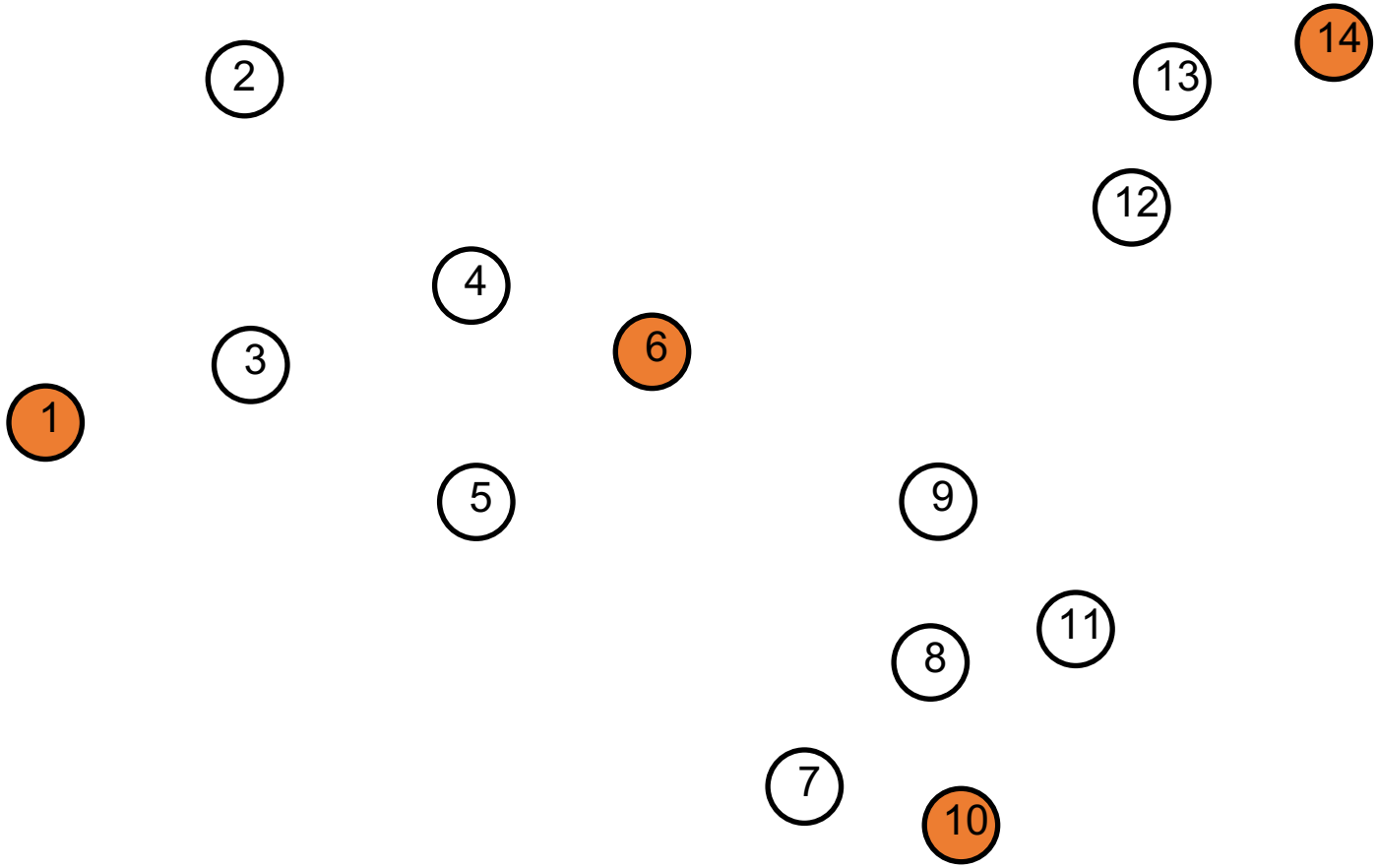
Step t=3



Example

Step t=4

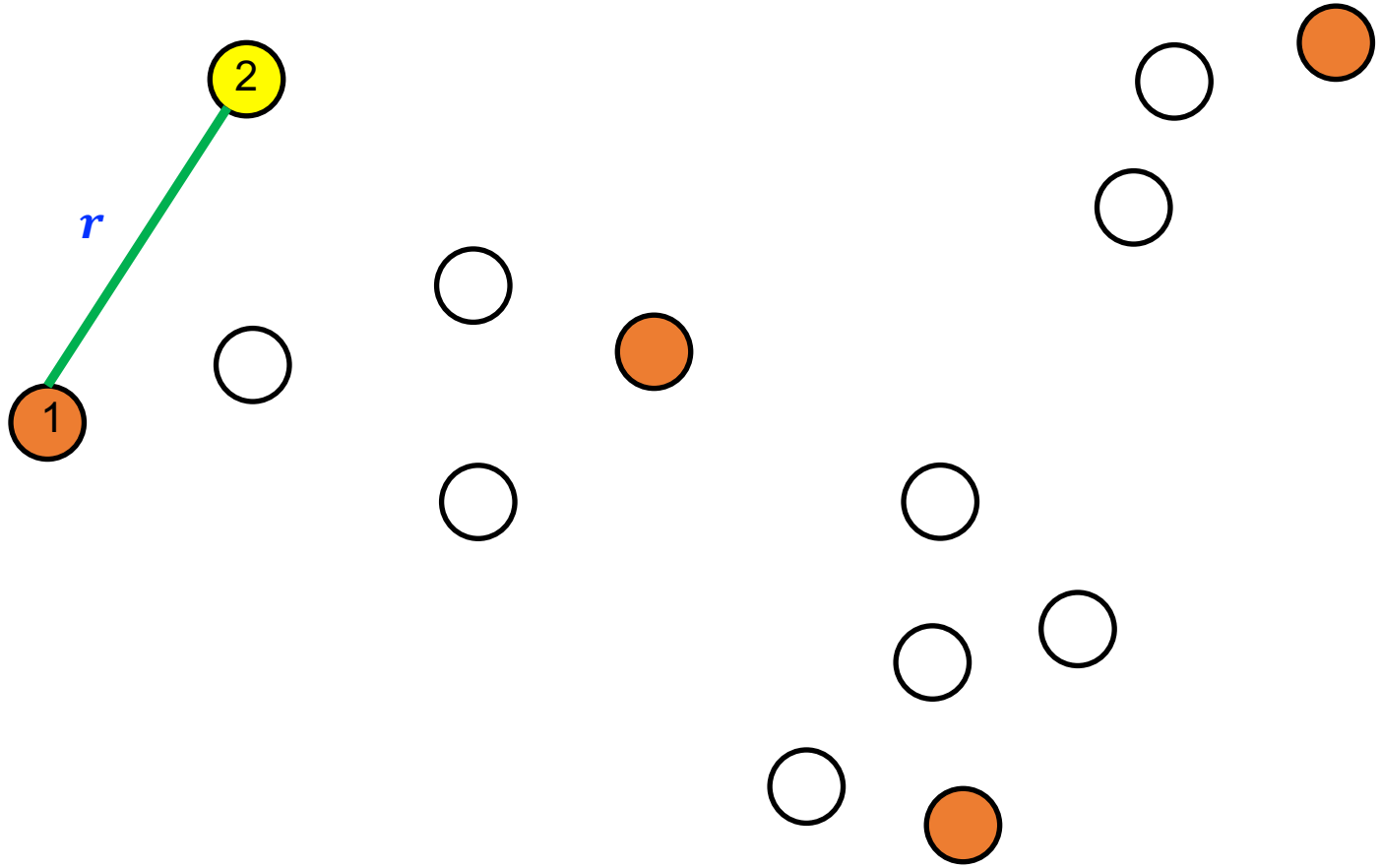
Which point and center have the maximum distance?



Example

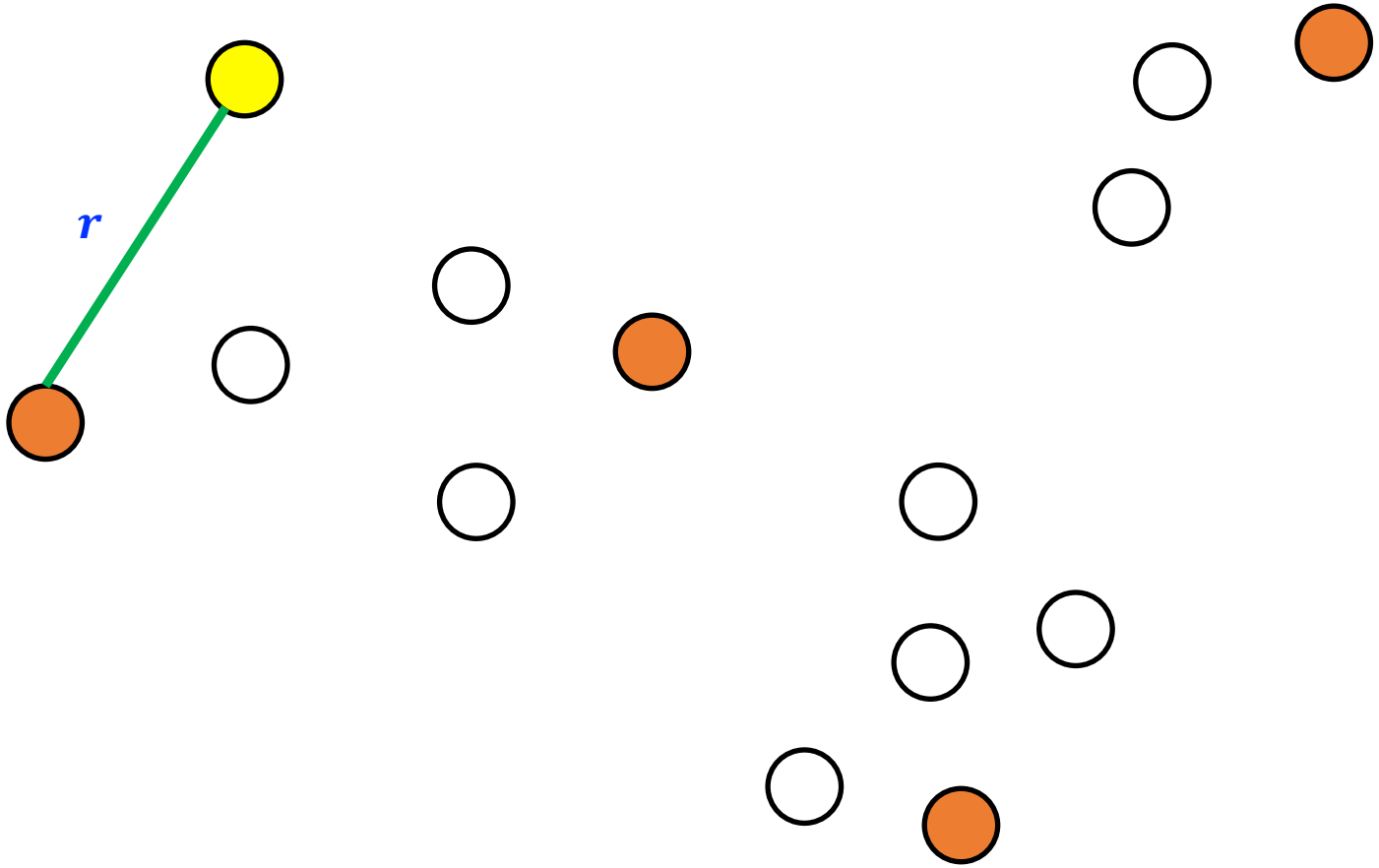
Step t=4

Which point and center have the maximum distance?



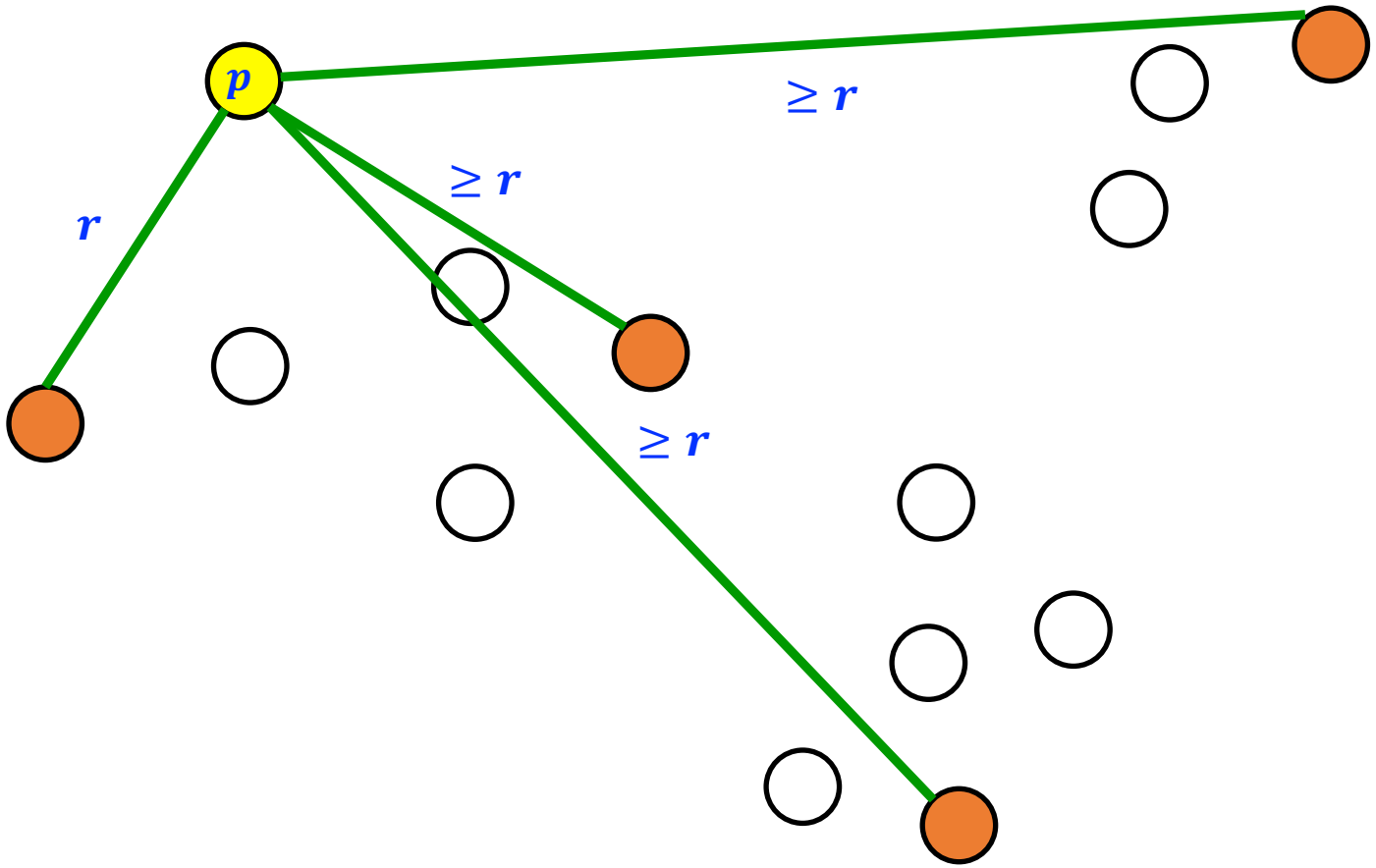
What can we say about this?

Theorem 7.3 *If there is a k-clustering of radius $\frac{r}{2}$, then the above algorithm finds a k-clustering with radius at most r.*



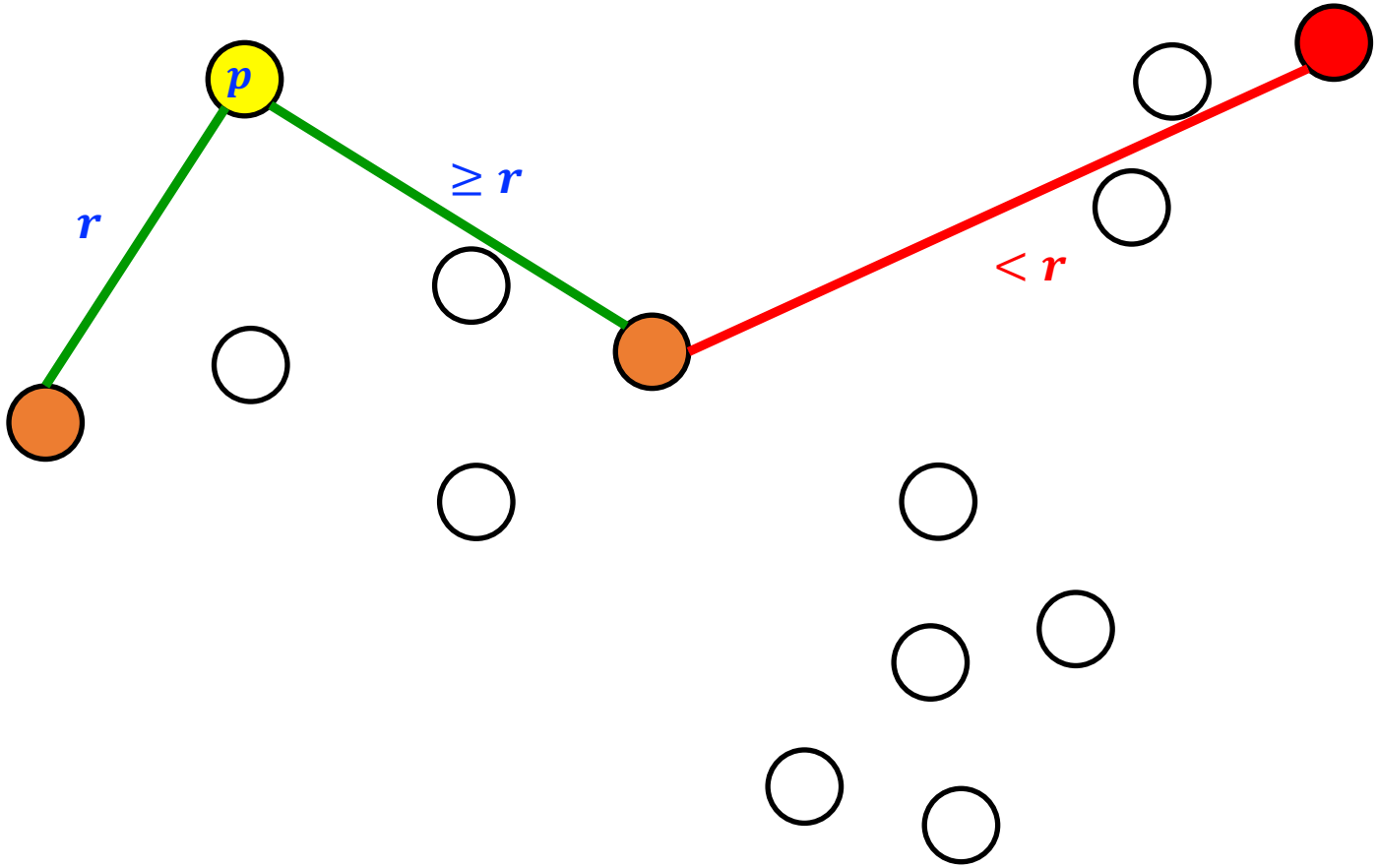
Proof

Distance of p from all other centers $\geq r$



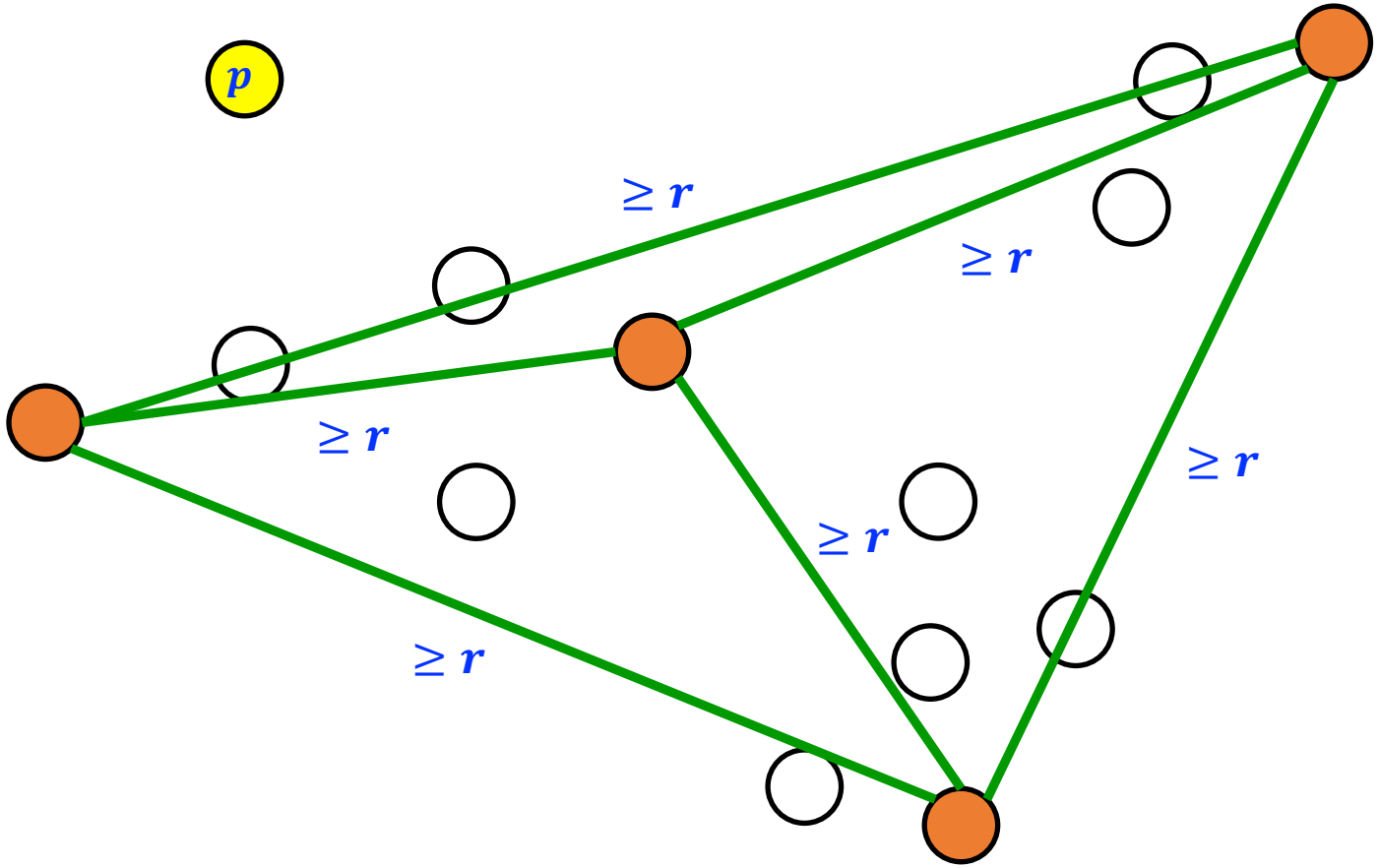
Proof

Distance between centers $\geq r$, since otherwise we could have picked p before one of the existing centers; in this example the red center.



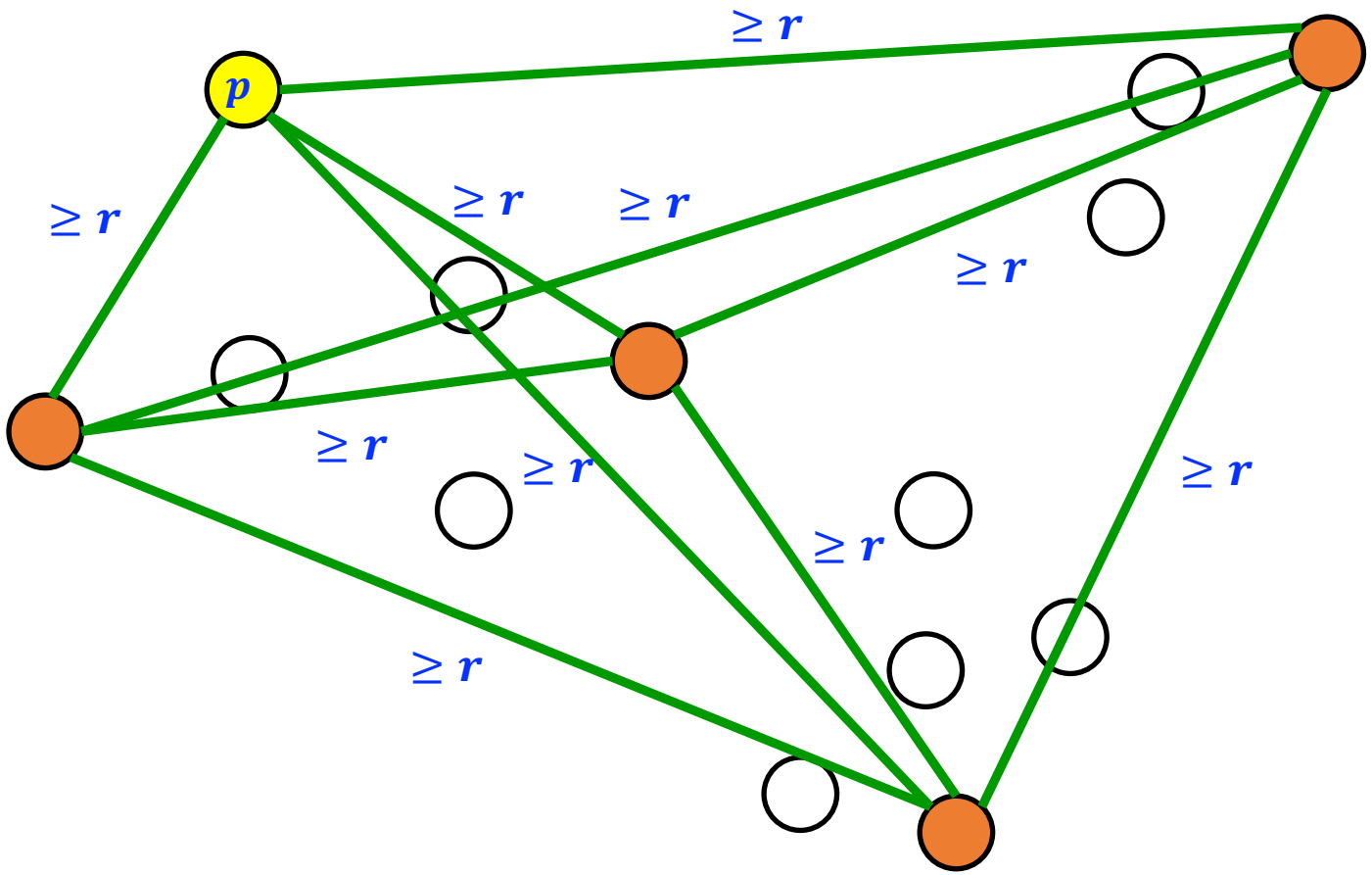
Proof

Distance between centers $\geq r$, since otherwise we could have picked p before one of the existing centers.



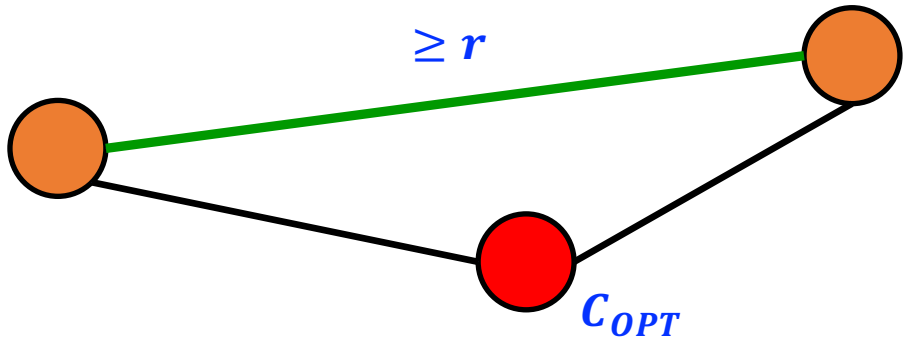
Proof

Distance between centers and p $\geq r$.



Proof

- We have $k+1$ points (k centers + point p), each pair is of distance $\geq r$.
- In OPT solution at least 2 of these points are assigned to the same center.



- This center must be of distance $\geq \frac{r}{2}$ from at least one of them ■

K-means

Clustering

k-means clustering

- A fundamental problem in data analysis & machine learning.
- By far the most popular clustering algorithm [Berkhin ‘02].
- Identified as one of the top 10 algorithms in data mining [Wu et al ‘07].

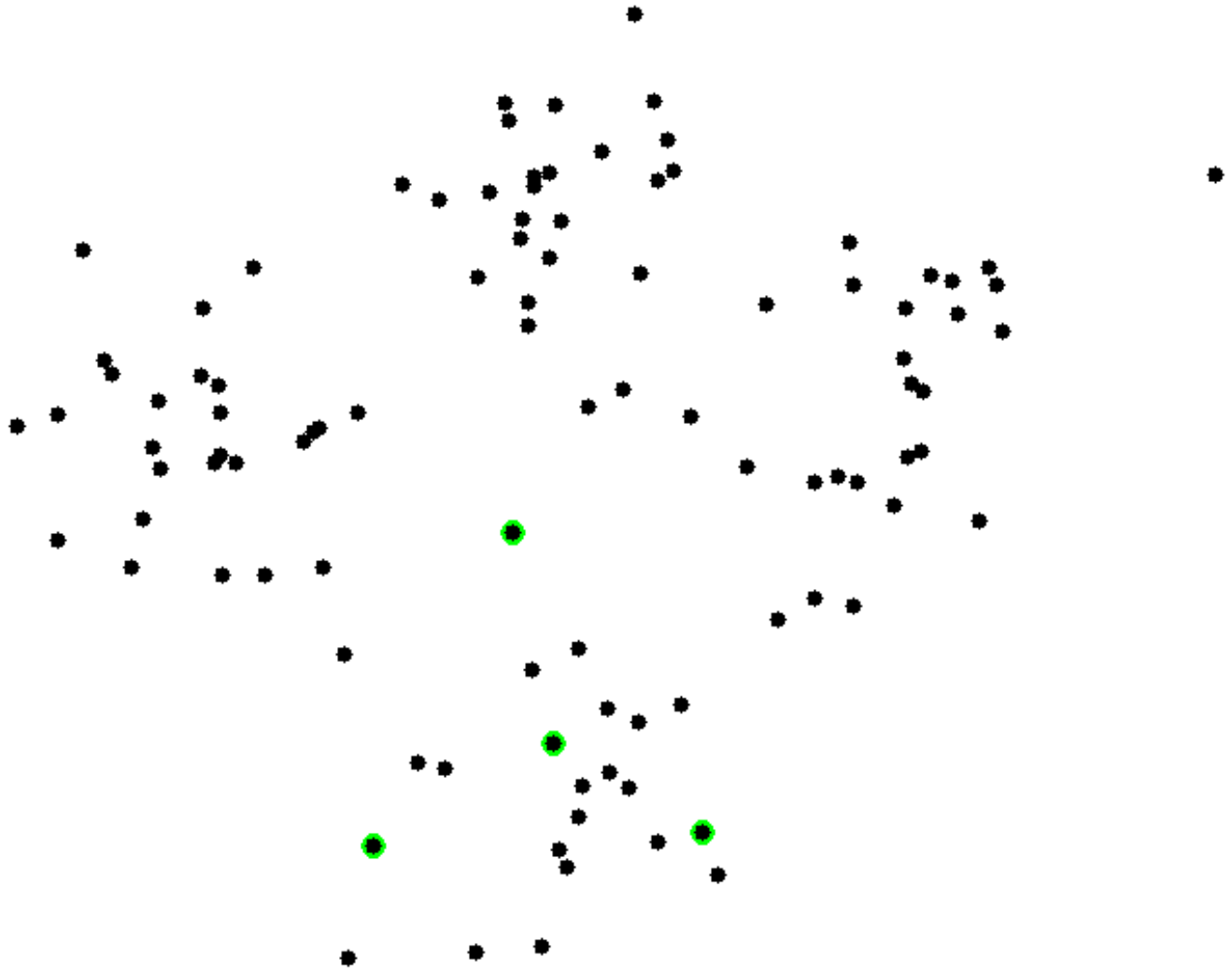
k-means clustering

Given integer k and n data points $X \subset R^d$. We wish to choose k centers C so as to minimize the objective function

$$\phi_{kmeans}(X) = \sum_{x \in X} \min_{c \in C} |x - c|^2$$

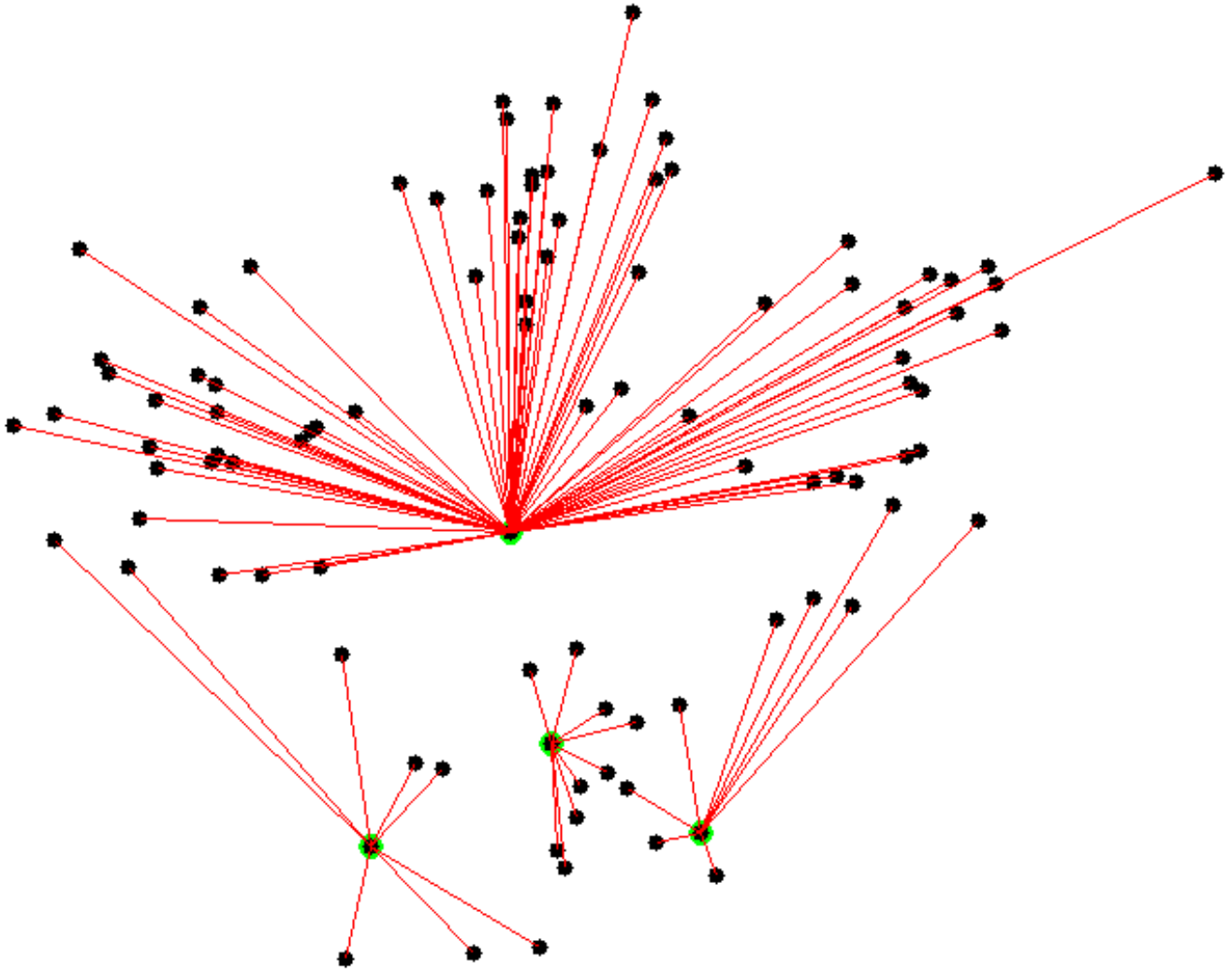
k-means clustering

Initialization



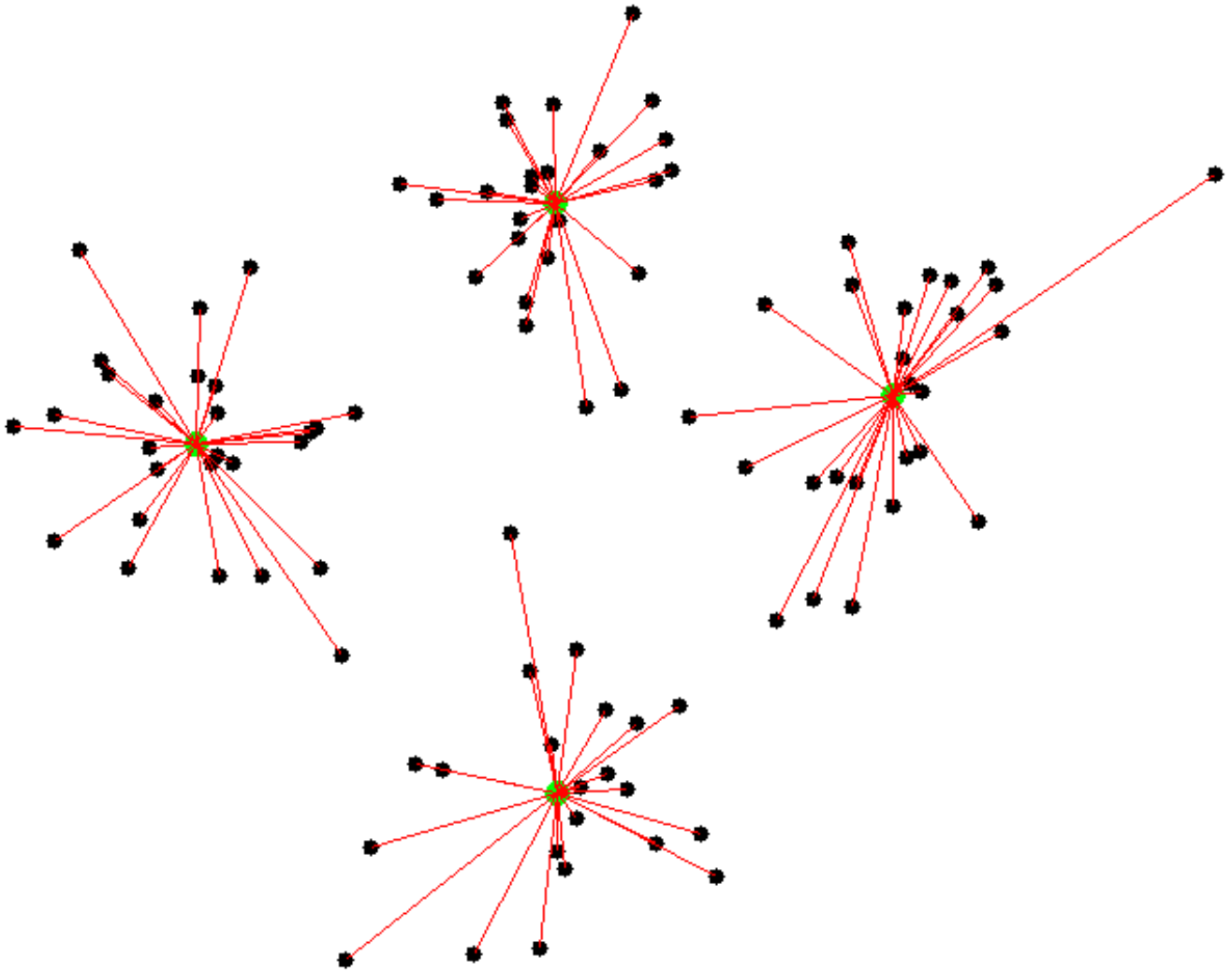
k-means clustering

Closest centroid



k-means clustering

Final solution



Motivation for using k-means (1)

Suppose that the data originates from an equal weight mixture of k spherical well-separated Gaussian densities centered at $\mu_1, \mu_2, \ldots, \mu_k$, each with variance 1 in every direction. The density of the mixture is

$$P(x) = \frac{1}{\sqrt{(2\pi)}^d} \frac{1}{k} \sum_{i=1}^{k} e^{-|x-u_i|^2}$$

Denote by $\mu(x)$ the center nearest to x. Since the exponential function falls off fast we can approximate $\sum_{i=1}^{k} e^{-|x-u_i|^2} \approx e^{-|x-\mu(x)|^2}$. Thus

$$P(x) \approx \frac{1}{\sqrt{(2\pi)}^d k} e^{-|x-\mu(x)|^2}$$

Motivation for using k-means (2)

The likelihood of drawing the sample points $x_1, x_2, \ldots, x_n$ from the mixture, if the centers were $\mu(x_1), \mu(x_2), \ldots, \mu(x_n)$ is approximately

$$P(x_1)P(x_2)\cdots P(x_n) \approx \frac{1}{k^n}\frac{1}{\sqrt{(2\pi)^d}}\prod_{i=1}^{n} e^{-\left|x^{(i)}-\mu\left(x^{(i)}\right)\right|^2} = ce^{-\sum_{i=1}^{n}\left|x^{(i)}-\mu\left(x^{(i)}\right)\right|^2}$$

Minimizing the sum of squared distances to cluster centers finds the maximum likelihood $\mu_1, \mu_2, \ldots, \mu_k$.

The k-means objective (1)

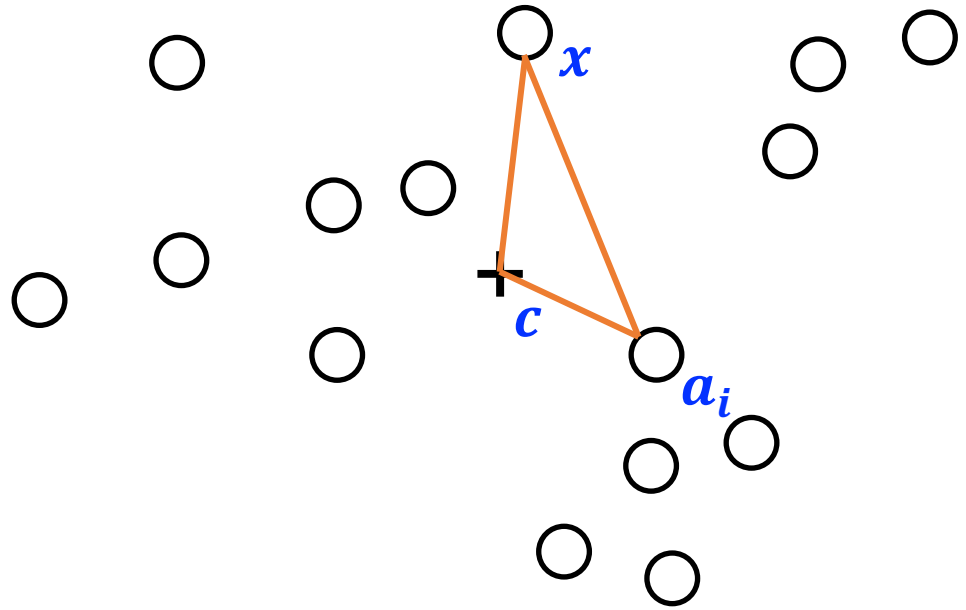
Suppose we have already determined the clustering into $C_1, \ldots, C_k$.

What are the best centers for the clusters?

Lemma 7.1

Let $\{a_1, a_2, \ldots, a_n\}$ be a set of points, if we set $c = \frac{1}{n}\sum_{i=1}^{n} a_i$ (centroid) then,

$$\sum_i |a_i - x|^2 = \underbrace{\sum_i |a_i - c|^2}_{A\ constant} + \underbrace{n|c - x|^2}_{\substack{x=c \to term=0 \\ and\ term \geq 0}}$$



The k-means objective (2)

Proof.

$$\sum_i |a_i - x|^2 = \sum_i |(a_i - c) + (c - x)|^2$$

$$= \sum_i (|a_i - c|^2) + 2(c - x) \cdot \underbrace{\sum_i (a_i - c)}_{=0 \text{ since } c \text{ is a centroid}} + n|c - x|^2$$

$$= \sum_i |a_i - c|^2 + n|c - x|^2$$

Lloyd's Algorithm (/Method)

1. Start with k centers.

2a. Associate each point with the center nearest to it.

2b. Find the centroid of each cluster and replace the set of old centers with the centroids.

Repeat Step 2 until the centers converge (according to some criterion, such as the k-means score no longer improving).

By **Lemma 7.1**, converges to a local minimum of the objective, since we are always minimizing the sum of internal cluster squared distances.

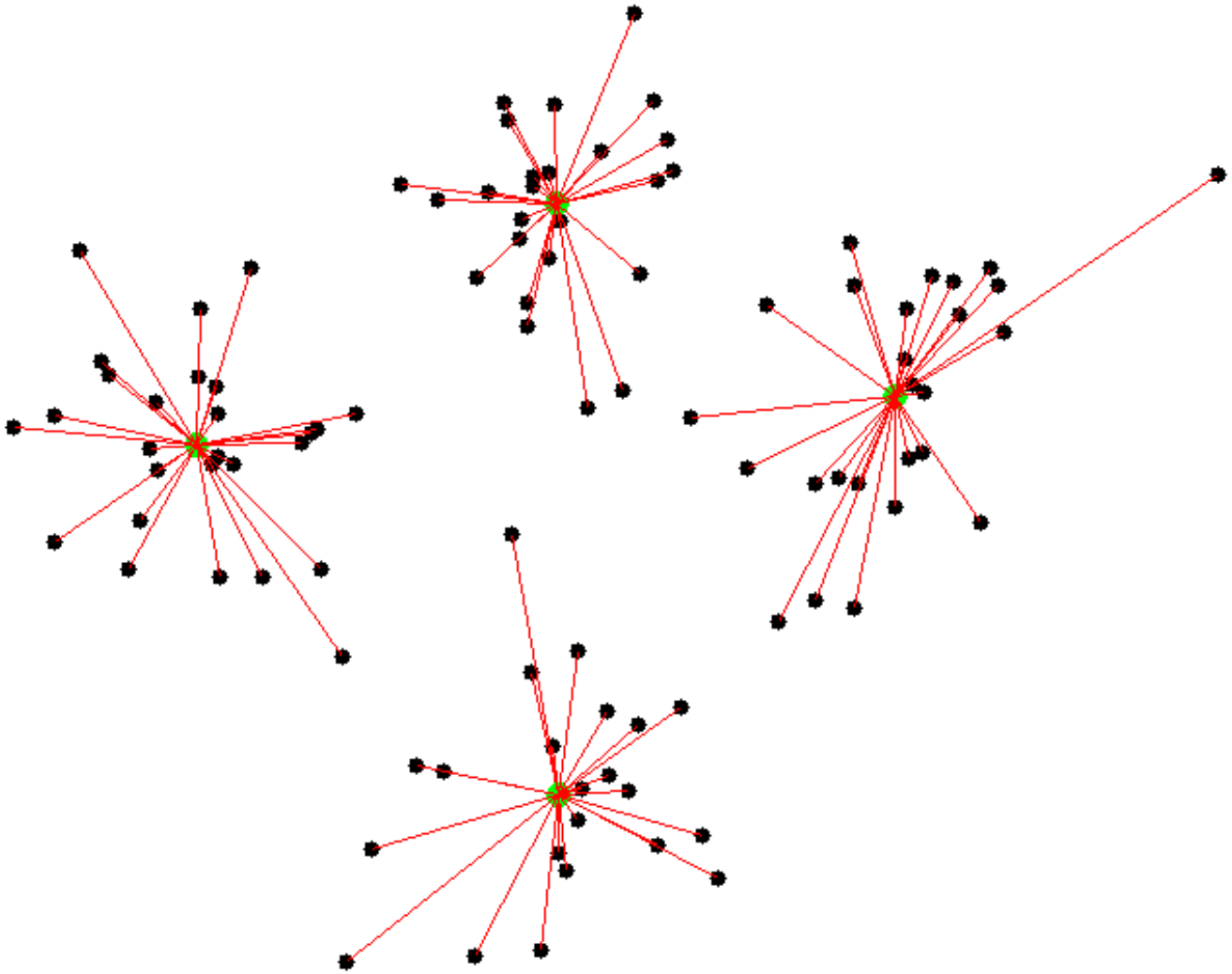
(**How to pick the centers?**)

Lloyd's Algorithm

Click to run simulation

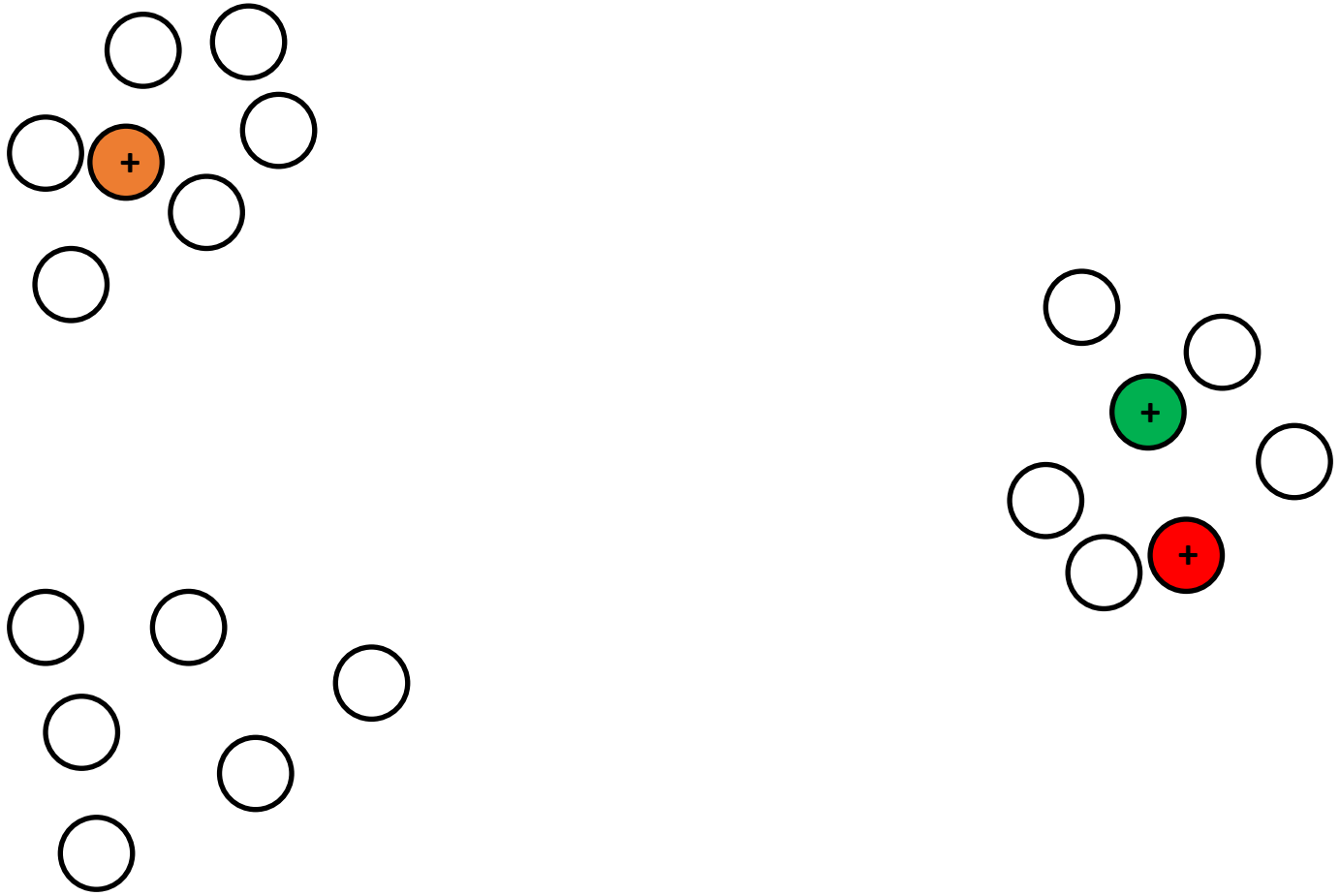
Lloyd's Algorithm

Lloyd's Algorithm

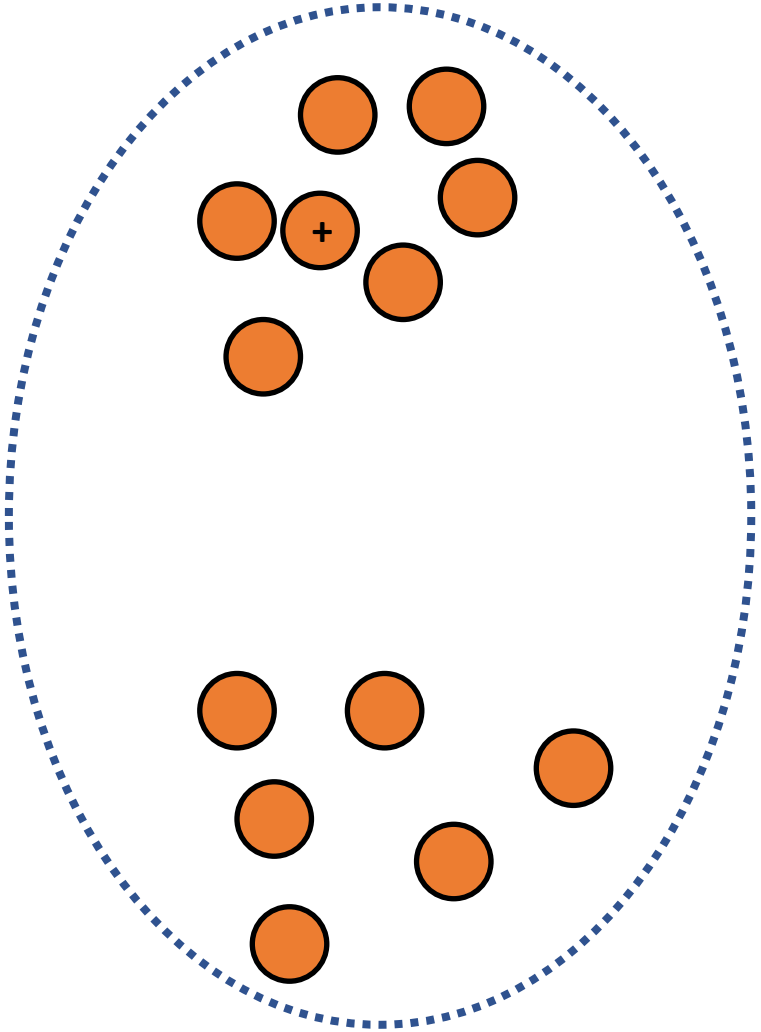


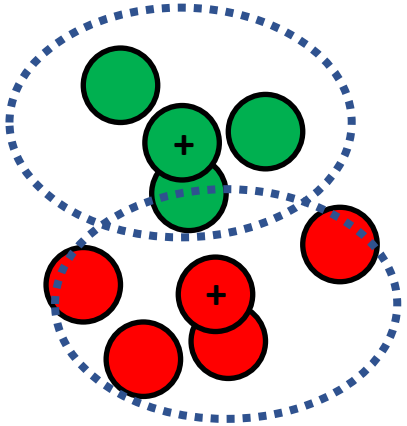
Lloyd's Algorithm – Local and not Global

For the following initialization



Lloyd's Algorithm – Local and not Global

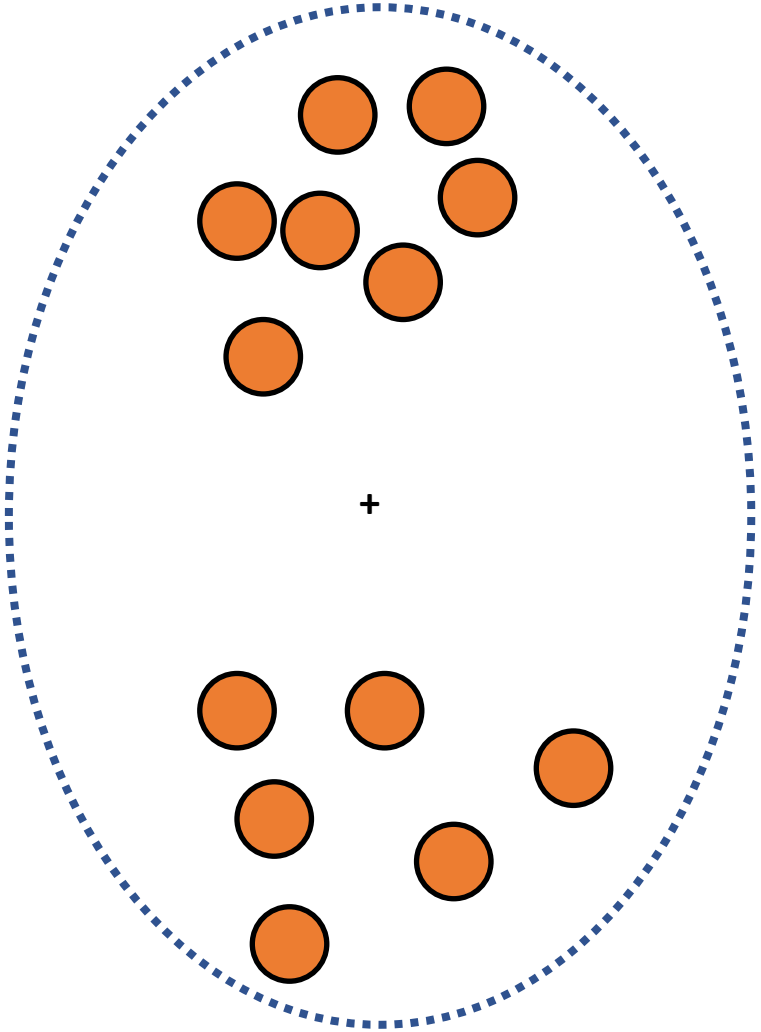


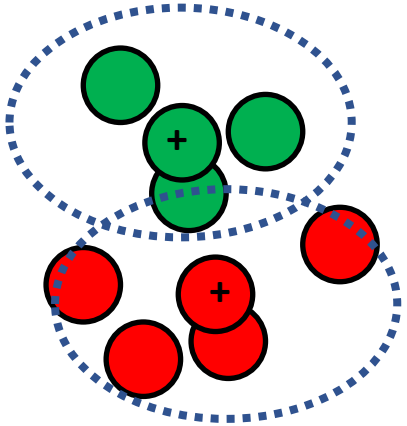


Lloyd's Algorithm – Local and not Global

We receive a local optimum **too far** from the global one.

How can we improve the algorithm?

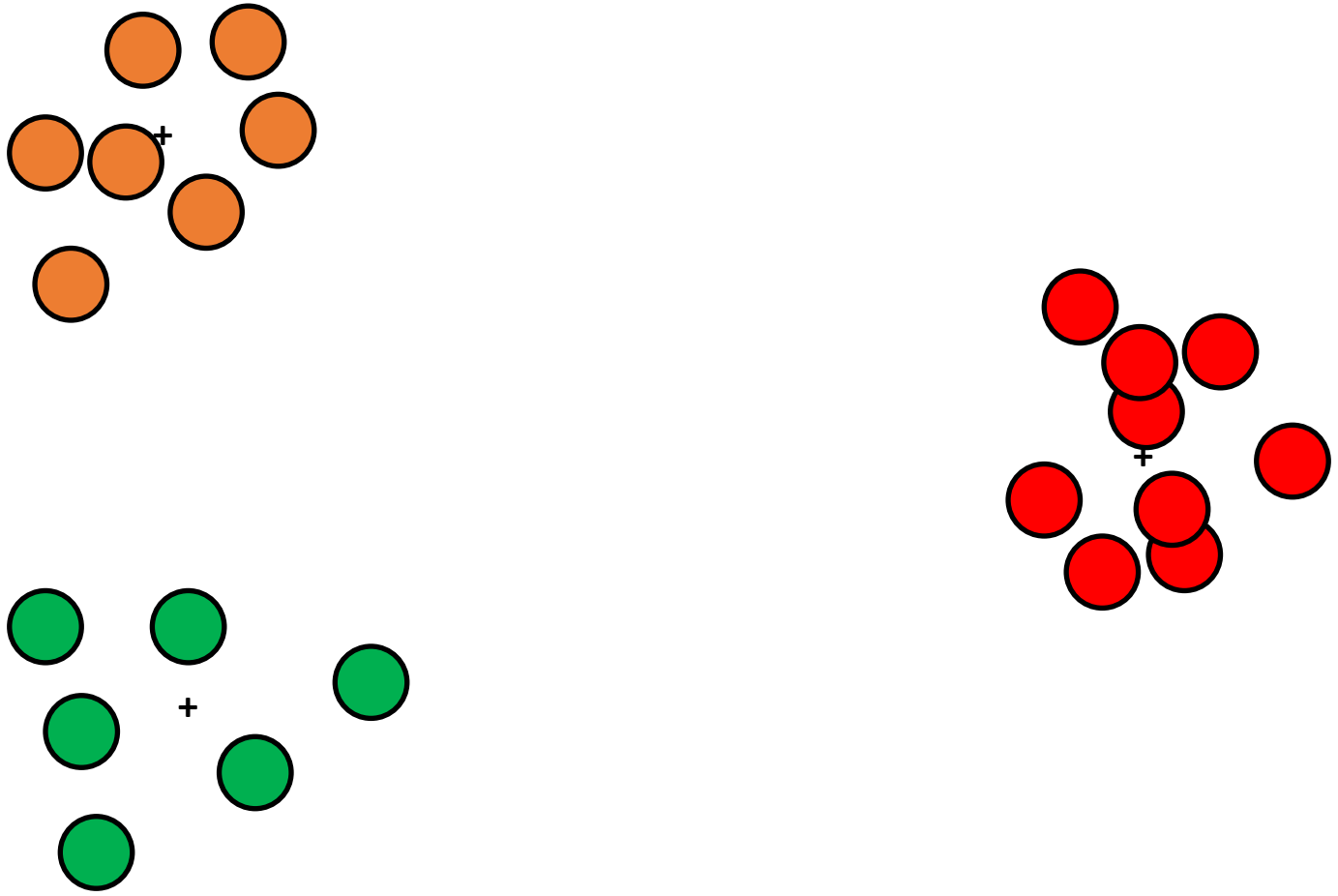




Lloyd's Algorithm – Local and not Global

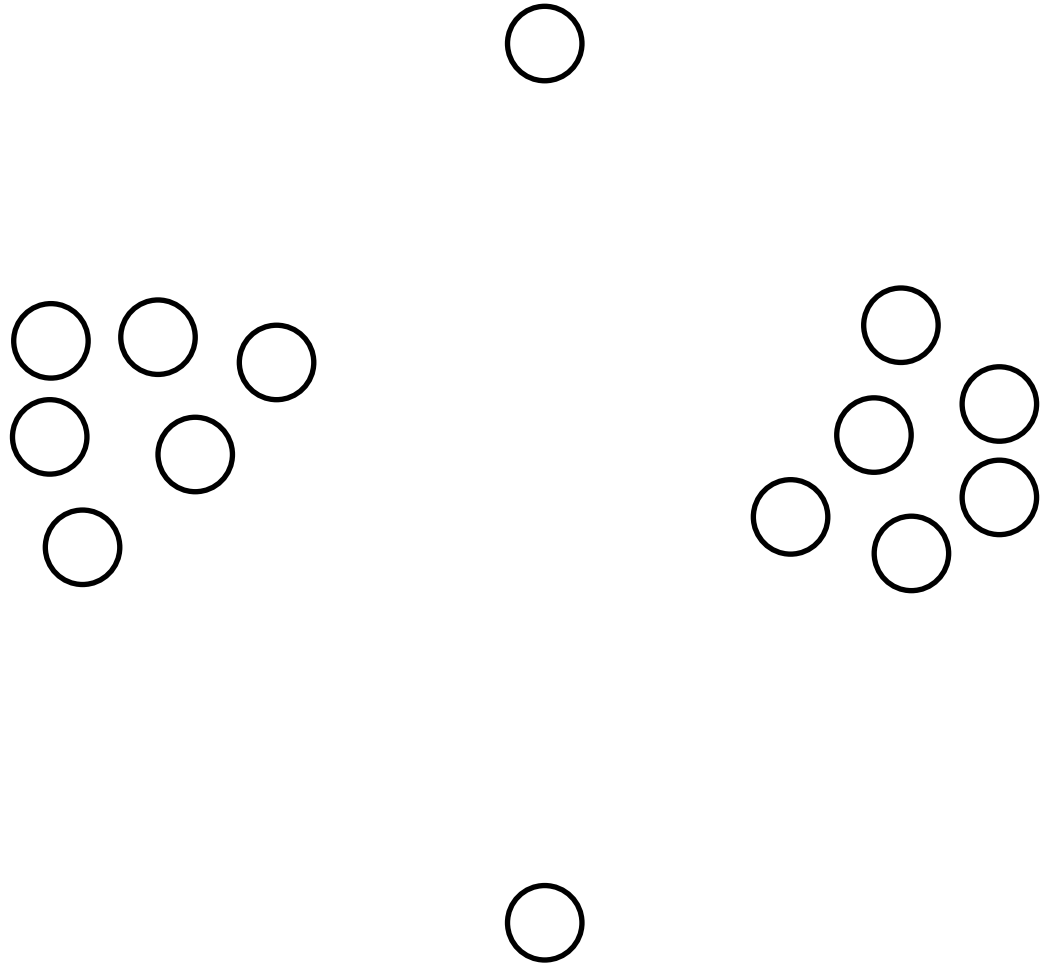
The global solution.

How can we improve the algorithm?



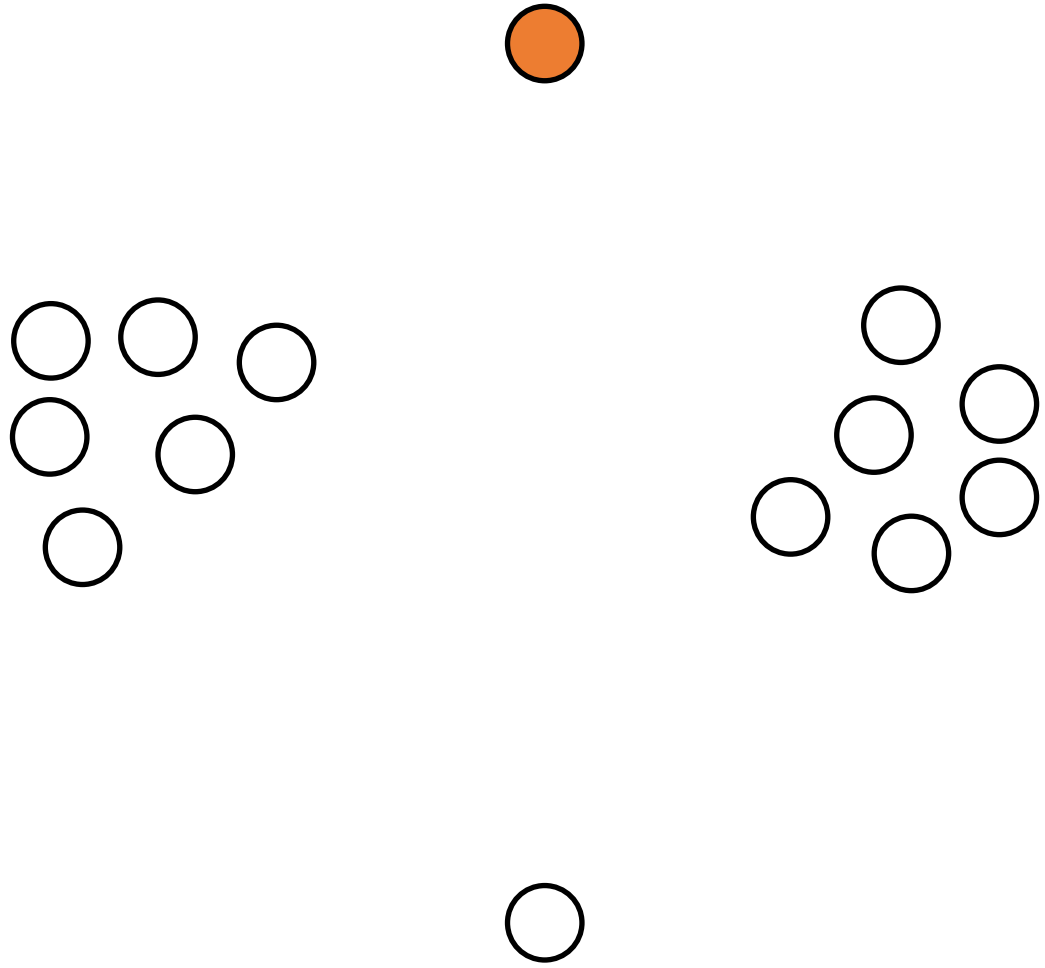
Farthest-first traversal robustness

Can be easily fooled by outliers. For $k = 2$:



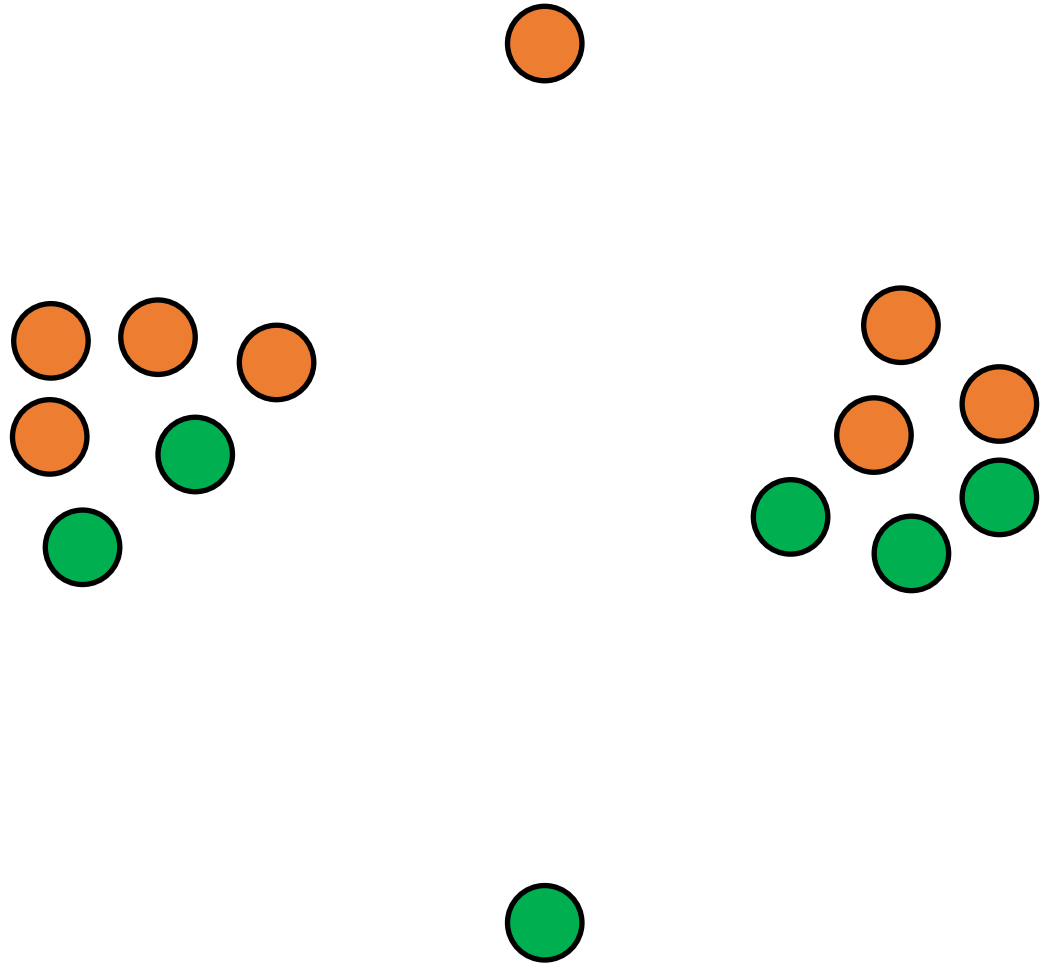
Farthest-first traversal robustness

Can be easily fooled by outliers. For $k = 2$:



Farthest-first traversal robustness

Can be easily fooled by outliers. For $k = 2$:



A common but simple initialization method

Randomly choose k initial centroids for the clusters, then run a chosen algorithm. Repeat the two steps multiple times.

In the end, choose the clustering that yields the best optimization criterion.

PCA Initialization

Find the first k principle components (eigenvectors) and set them as the initial centroids.

K-means++

Initialization Method

(Original paper: http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf)

K-means++ Overview

Choose centers at random from the data points, but weight the data points (the probability of choosing them) according to their squared distance from the closest center already chosen.

Definitions

- Given clustering C with objective ϕ and points X, we let $\phi(A) = \sum_{x \in A} \min_{c \in C} |x - c|^2$ denote the **contribution** of $A \subset X$ to the objective.
- let $D(x) = \min_{c \in C} |x - c|$ denote the distance from a data point x to the **closest** center we have already chosen.

The k-means++ algorithm

1a. Take one center c_1, chosen uniformly at random from X.

1b. Take a new center c, choosing a $\in X$ with probability $\frac{D(a)^2}{\sum_{x \in X} D(x)^2}$.

1c. Repeat Step 1b. until we have taken k centers altogether.

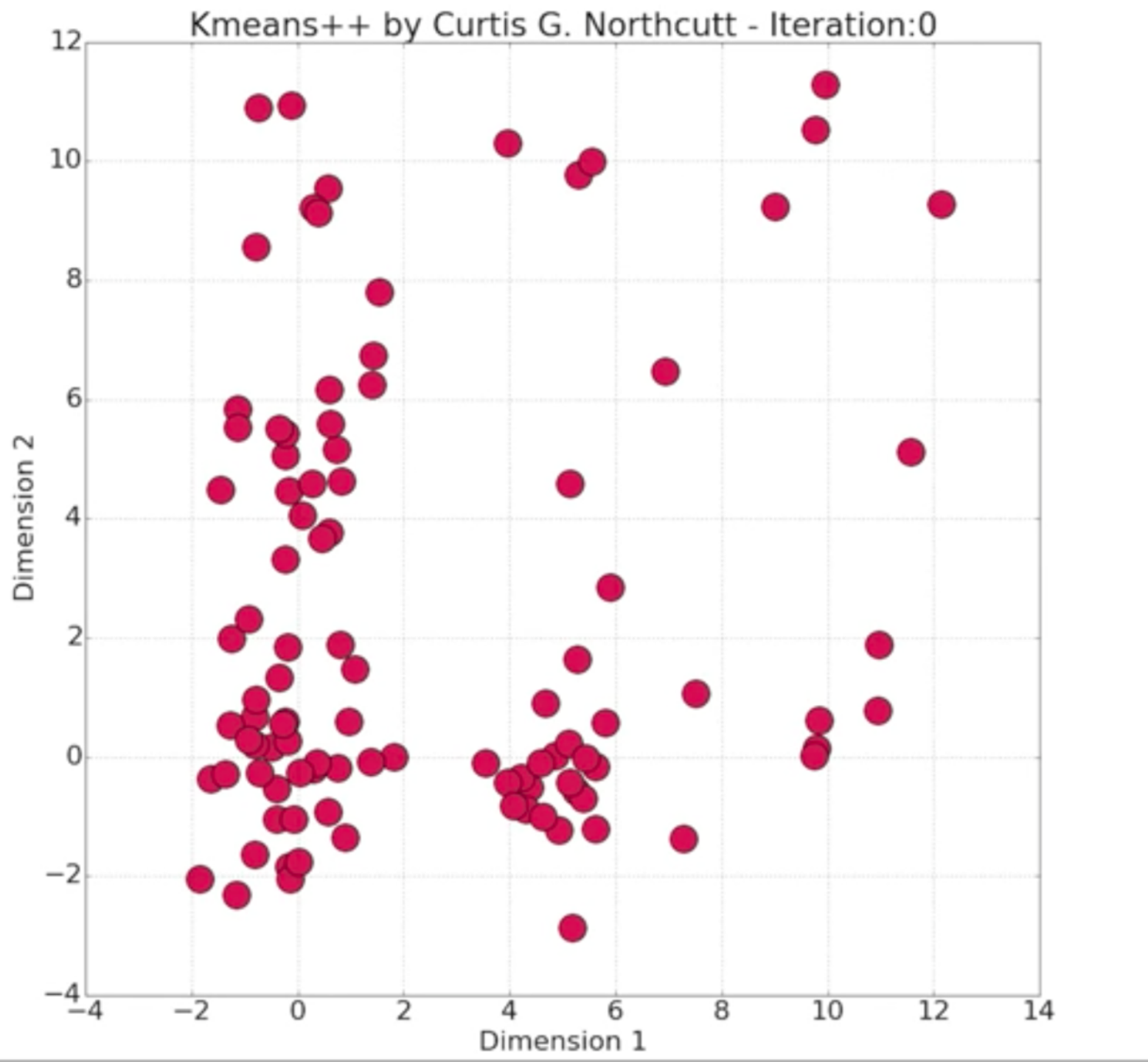
Then: Proceed as with the chosen k-means algorithm (Lloyd).

- We call the weighting used in Step 1b simply "D^2 weighting".
- The algorithm takes $O(n \cdot k^2)$.

Test Run

Step k=0

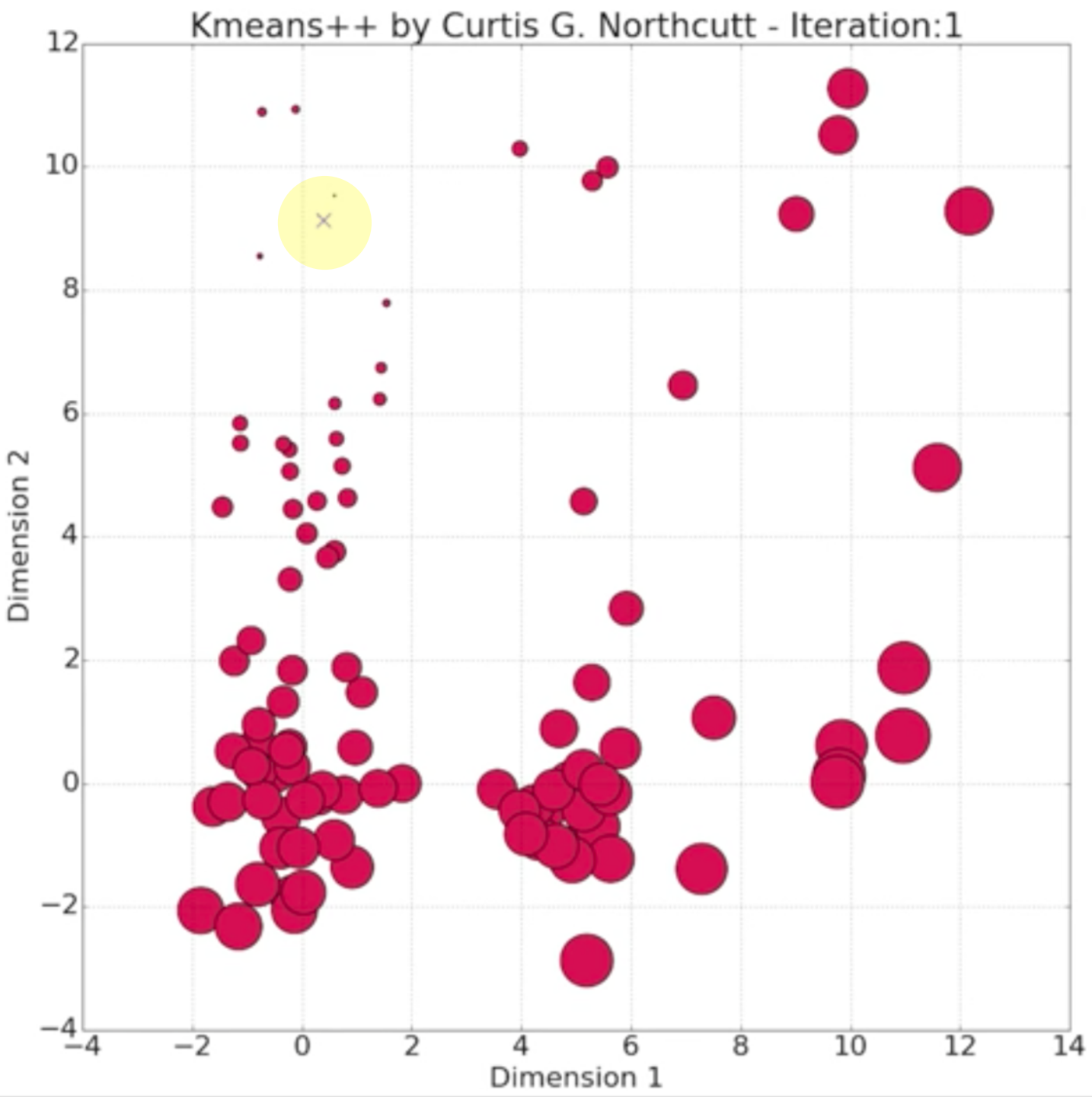
Sample data set for clustering. Size of the points represent the relative chance to choose them. Choosing first center uniformly at random yields same size for all points.



Test Run

Step k=1

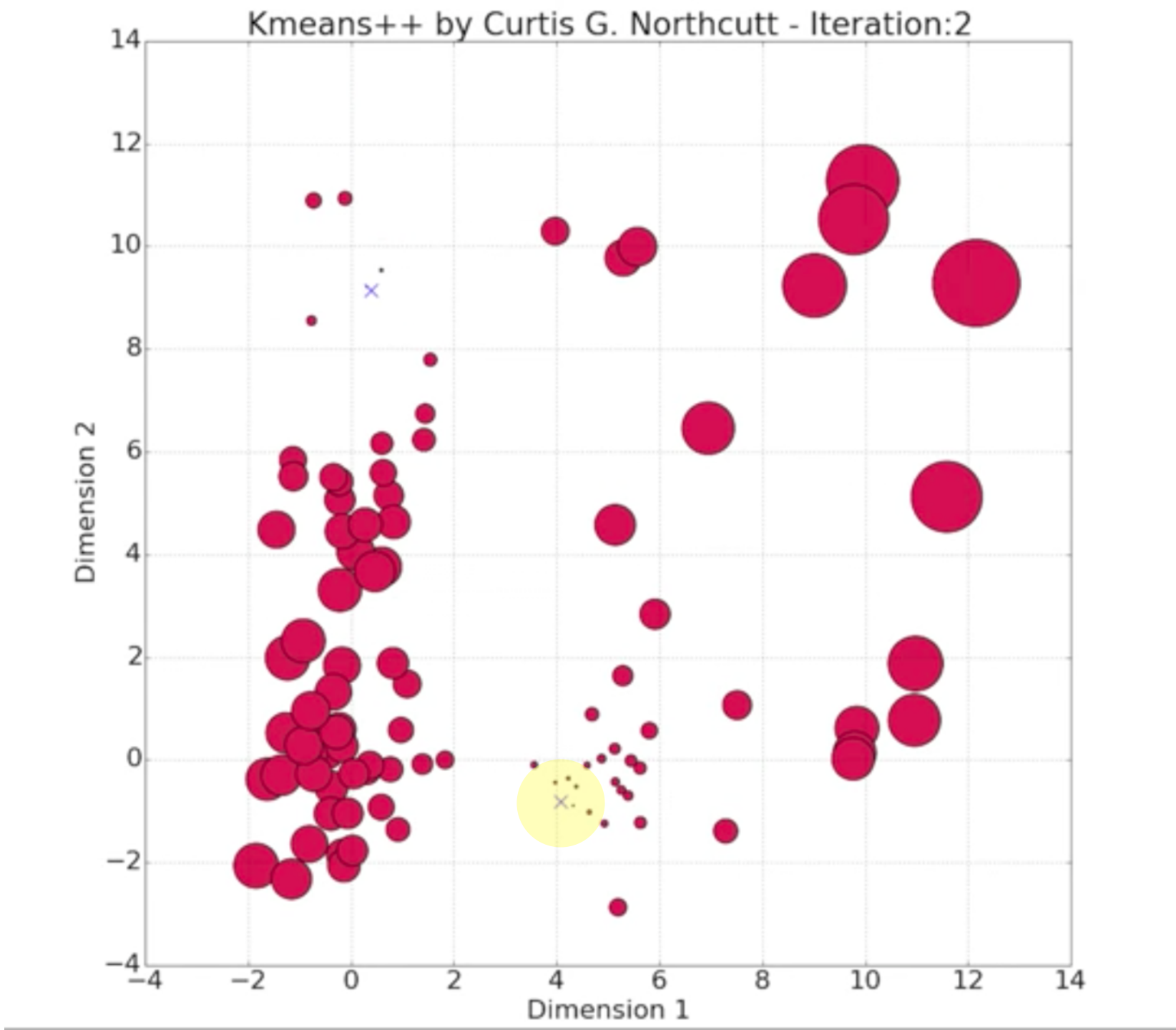
Second center will most likely be chosen far from the first one.



Test Run

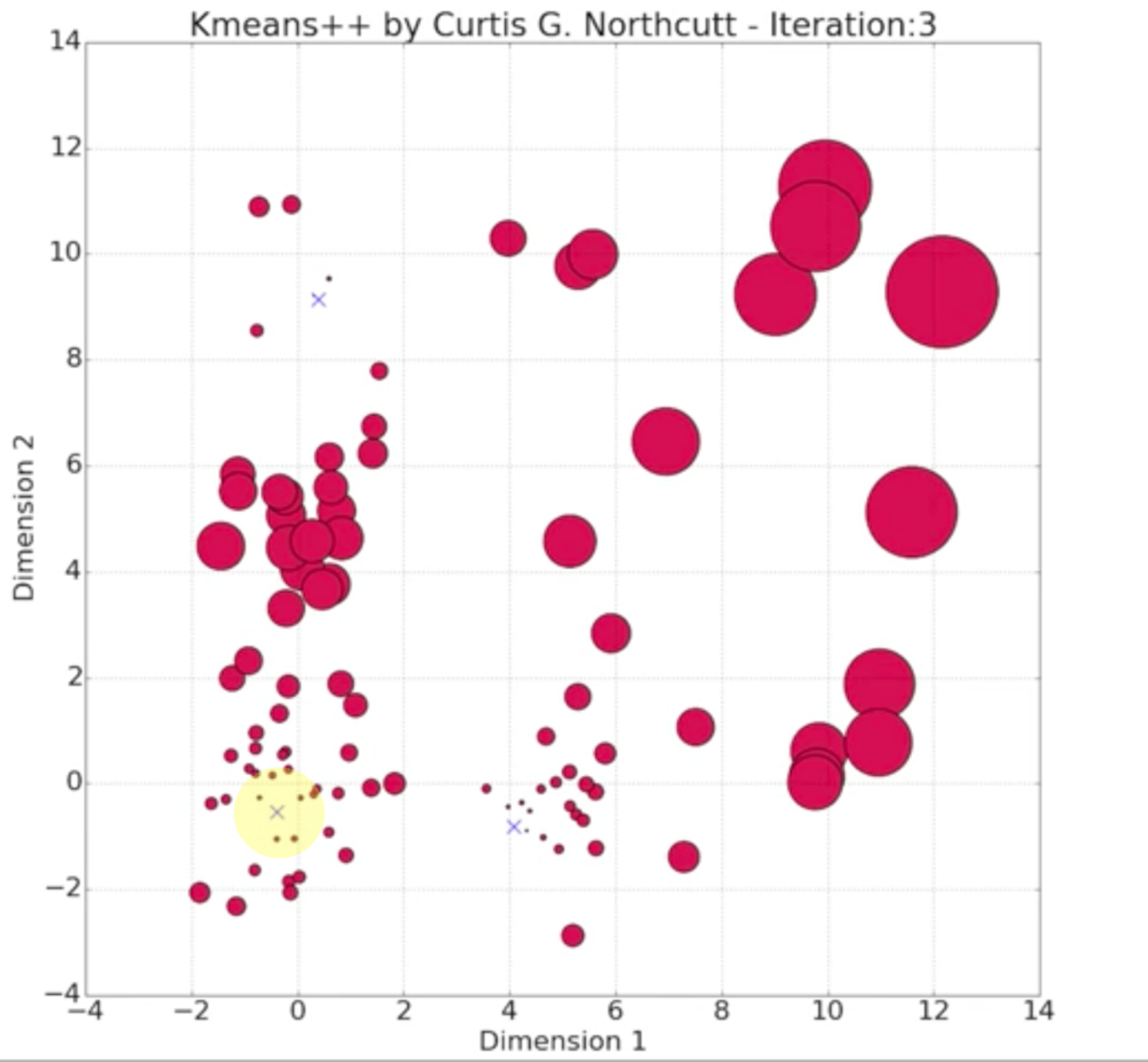
Step k=2

The next one will most likely be chosen far away from the 2 already chosen centers.



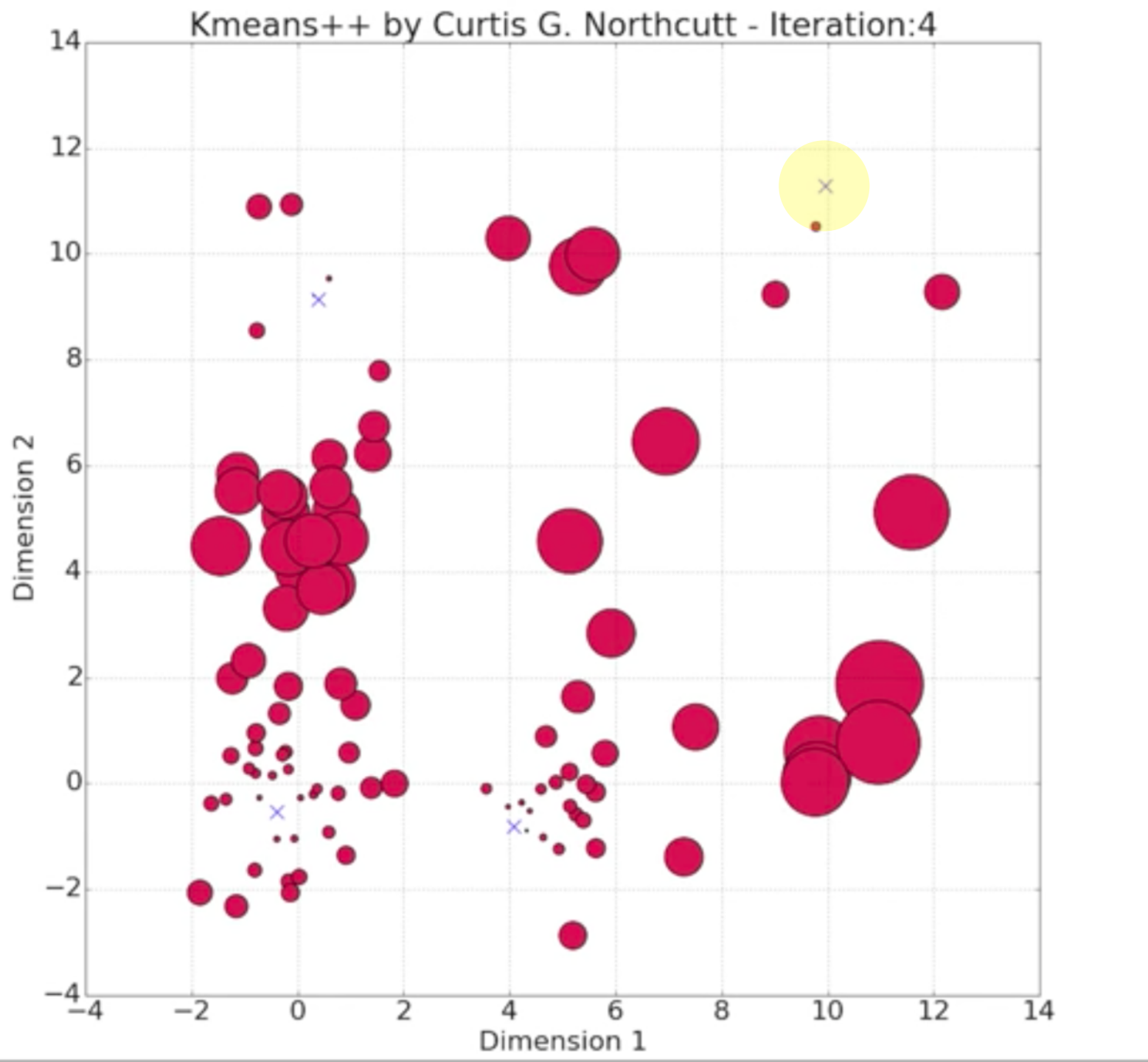
Test Run

Step k=3



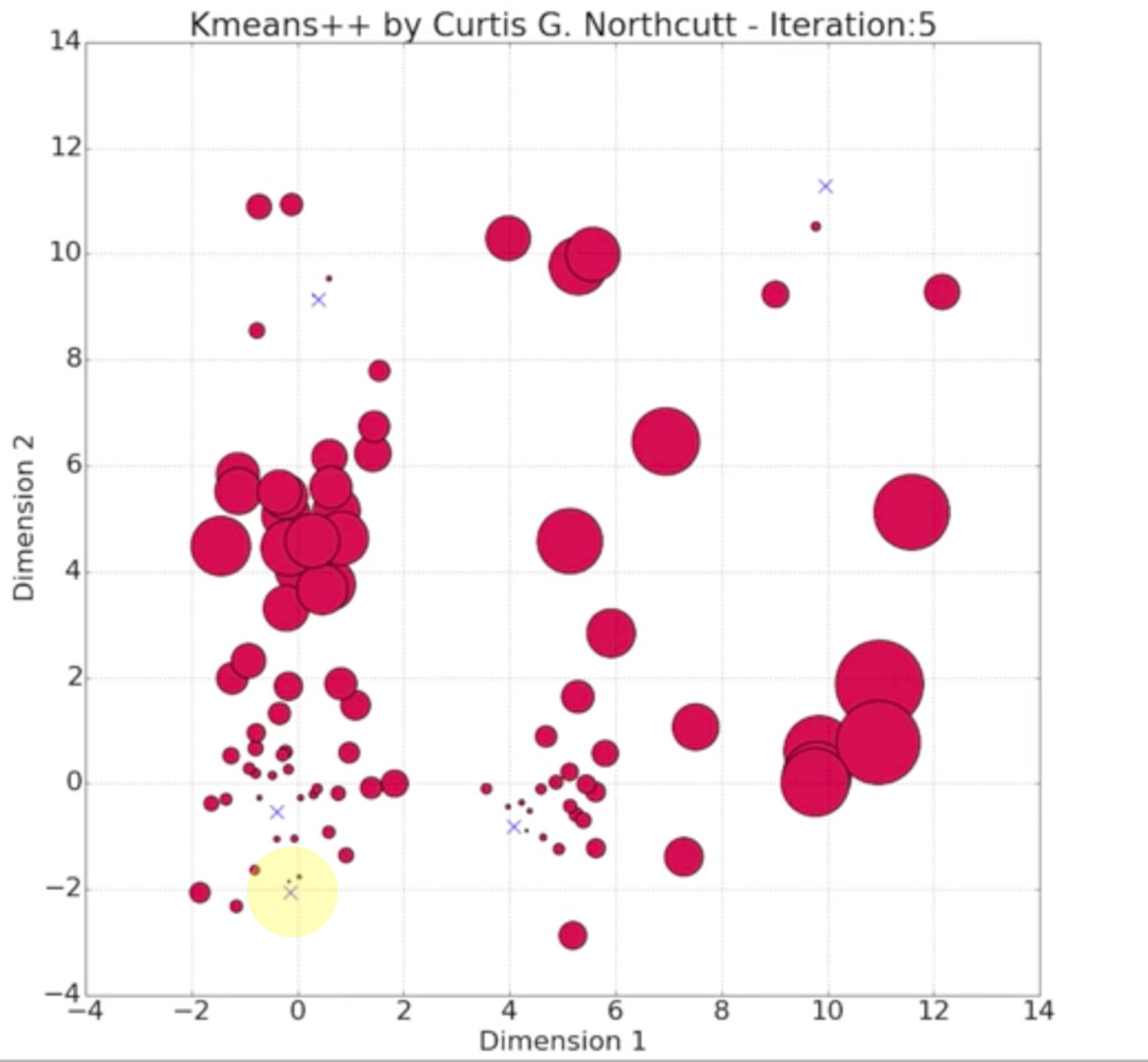
Test Run

Step k=4



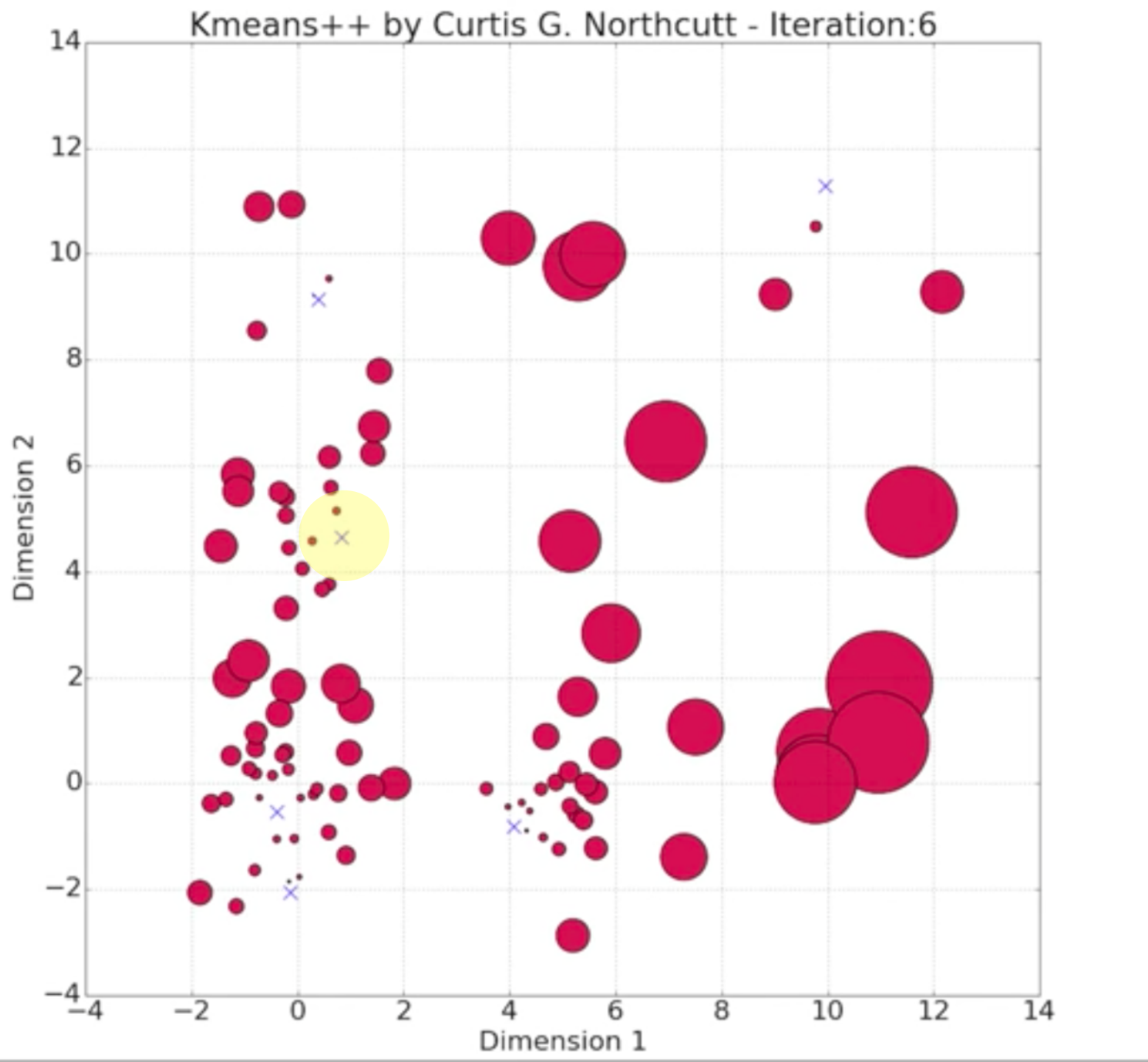
Test Run

Step k=5



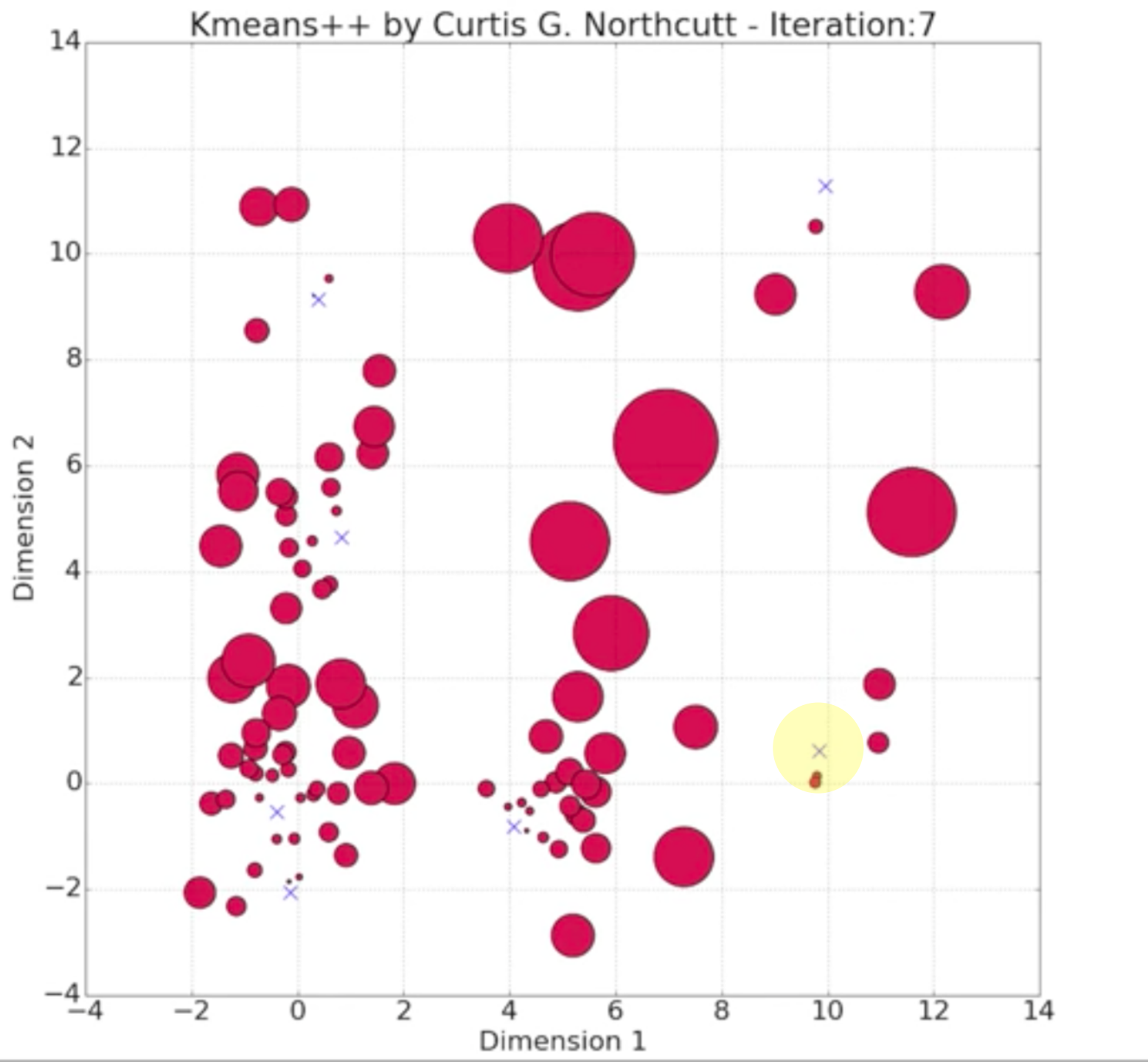
Test Run

Step k=6



Test Run

Step k=7



K-means++ is $O(log\ k)$-Competitive

Theorem 3.1 *If C is constructed with k-means++, then the corresponding objective ϕ satisfies, $E[\phi] \leq 8(ln\, k + 2)\phi_{OPT}$.*

In fact, this holds after only Step 1 of the algorithm above. As noted above, the rest of the algorithm (Lloyd) can only decrease ϕ.

Part 1 – First center chosen

Reminder (Lloyd's) - Lemma 2.1 *Let* $\{a_1, a_2, \dots, a_n\}$ *be a set of points, if we set* $c = \frac{1}{n}\sum_{i=1}^{n} a_i$ *(centroid) then,* $\sum_i |a_i - x|^2 = \sum_i |a_i - c|^2 + n|c - x|^2$.

Lemma 3.2 *Let* A *be an arbitrary cluster in* C_{OPT} *(optimal clustering), and let* C *be the clustering with just one center, which is chosen uniformly at random from* A*. Then,* $E[\phi(A)] = 2 \cdot \phi_{OPT}(A)$.

Part 1 – First center chosen

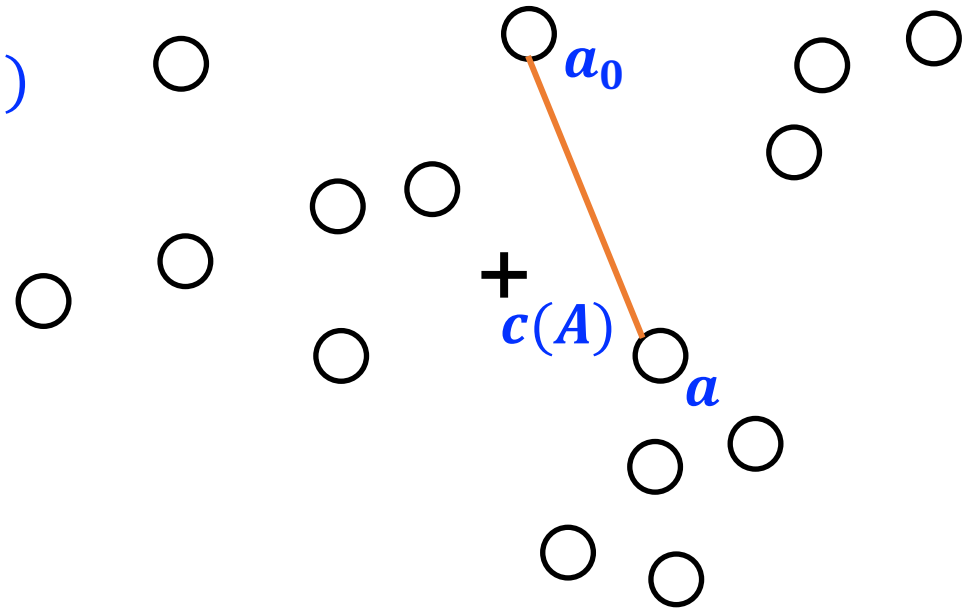
$(E[\phi(A)] = 2\phi_{OPT}(A))$

Proof. Let $c(A)$ denote the centroid of A. By **Lemma 2.1**, we know that since C_{OPT} is optimal, $c(A)$ must be the center in cluster A.

$$E[\phi(A)] \underbrace{=}_{\substack{\text{Choosing each point } a_0 \\ \text{as center uniformly at random}}} \frac{1}{|A|}\sum_{a_0 \in A}\sum_{a \in A}|a - a_0|^2$$

$$\underbrace{=}_{\text{Lemma 2.1}} \frac{1}{|A|}\sum_{a_0 \in A}\left(\sum_{a \in A}|a - c(A)|^2 + |A| \cdot |a_0 - c(A)|^2\right)$$

$$= 2\sum_{a \in A}|a - c(A)|^2 = 2 \cdot \phi_{OPT}(A) \blacksquare$$



Part 2 – Remaining centers

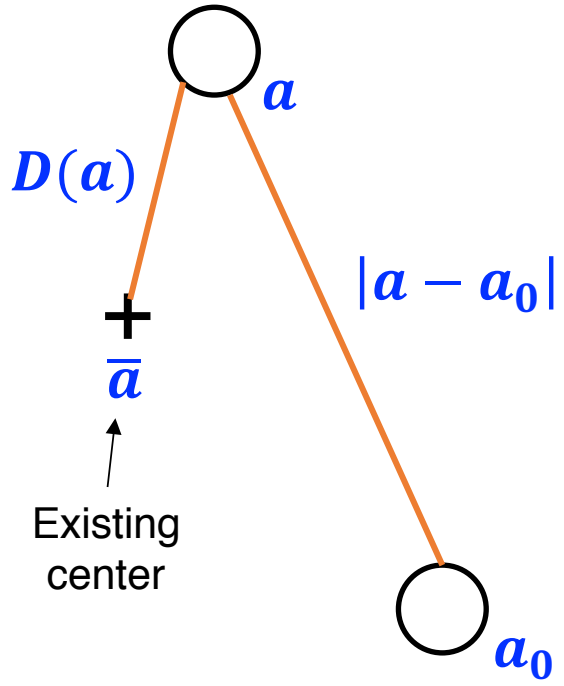
Lemma 3.3 *Let A be an arbitrary cluster in C_{OPT}, and let C be an arbitrary clustering. If we add a center to C from A, chosen with "D^2 weighting", then $E[\phi(A)] \leq 8 \cdot \phi_{OPT}(A)$.*

Part 2 – Remaining centers

Proof. The probability we choose some fixed a_0 as our center, given that we are choosing something **from** $\boldsymbol{A}$, is $\frac{D(a_0)^2}{\sum_{a \in A} D(a)^2}$.

After choosing the center a_0, every point a will contribute to the potential precisely $\min(D(a), |a - a_0|)^2$

Therefore, $\mathrm{E}[\phi(A)] = \sum_{a_0 \in A} \frac{D(a_0)^2}{\sum_{a \in A} D(a)^2} \cdot \sum_{a \in A} \min(D(a), |a - a_0|)^2$.



Part 2 – Remaining centers

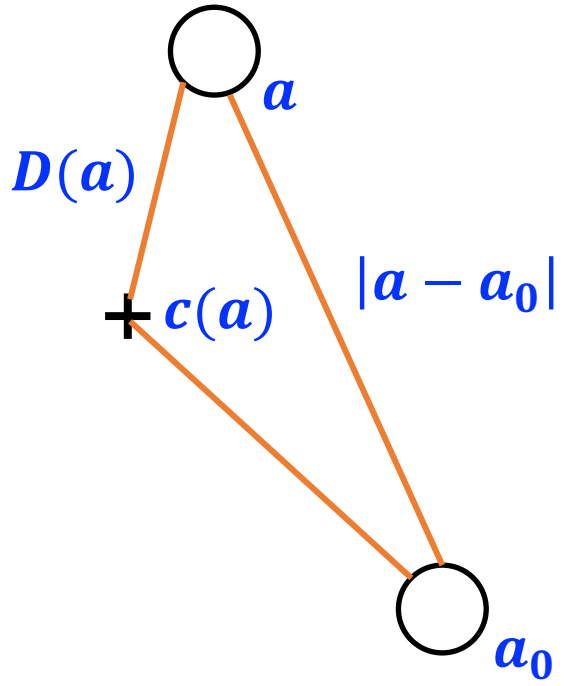
$E[\phi(A)] \leq 8\phi_{OPT}(A)$

Power-mean inequality (PMI): $\sum a_i^2 \geq \frac{1}{m}(\sum a_i)^2$.

From the triangle inequality: $D(a_0) \leq D(a) + |a - a_0|$ for all a, a_0.

Together: $D(a_0)^2 \underbrace{\leq}_{\Delta} (D(a) + |a - a_0|)^2 \underbrace{\leq}_{PMI} 2 \cdot (D(a)^2 + |a - a_0|^2)$.

Summing over a: $D(a_0)^2 \leq \frac{2}{|A|}(\sum_{a \in A} D(a)^2 + \sum_{a \in A} |a - a_0|^2)$.



Part 2 – Remaining centers

$E[\phi(A)] \leq 8\phi_{OPT}(A)$

From $\mathrm{E}[\phi(A)] = \sum_{a_0 \in A}\left(\frac{\boldsymbol{D(a_0)^2}}{\sum_{a\in A} D(a)^2} \sum_{a\in A} \min(D(a), |a-a_0|)^2\right)$

and $\boldsymbol{D(a_0)^2} \leq \frac{2}{|A|}\left(\sum_{a\in A} D(a)^2 + \sum_{a\in A}|a-a_0|^2\right)$ we get

$$\mathrm{E}[\phi(A)] \leq \frac{2}{|A|} \cdot \left(\sum_{a_0\in A} \frac{\sum_{a\in A} D(a)^2}{\sum_{a\in A} D(a)^2} \cdot \sum_{a\in A} \underbrace{\min(D(a), |a-a_0|)^2}_{\leq |a-a_0|^2} + \sum_{a_0\in A} \frac{\sum_{a\in A}|a-a_0|^2}{\sum_{a\in A} D(a)^2}\right.$$

$$\left. \cdot \sum_{a\in A} \underbrace{\min(D(a), |a-a_0|)^2}_{\leq D(a)^2}\right) \leq \frac{4}{|A|} \sum_{a_0\in A} \sum_{a\in A} |a-a_0|^2$$

$$= \quad \boldsymbol{8 \cdot \phi_{OPT}(A)} \blacksquare$$

$$\frac{1}{|A|} \sum_{a_0\in A} \sum_{a\in A} |a-a_0|^2 = 2 \cdot \phi_{OPT}(A)$$

Recap

We have now shown that our seeding technique is competitive ($E[\phi(A)] \leq 8\phi_{OPT}(A)$) as long as it chooses centers from each cluster of C_{OPT}, which completes the first half of our argument.

We can use induction to show the total error in general is at most $O(\log k)$. However, we will only give some intuition for the proof.

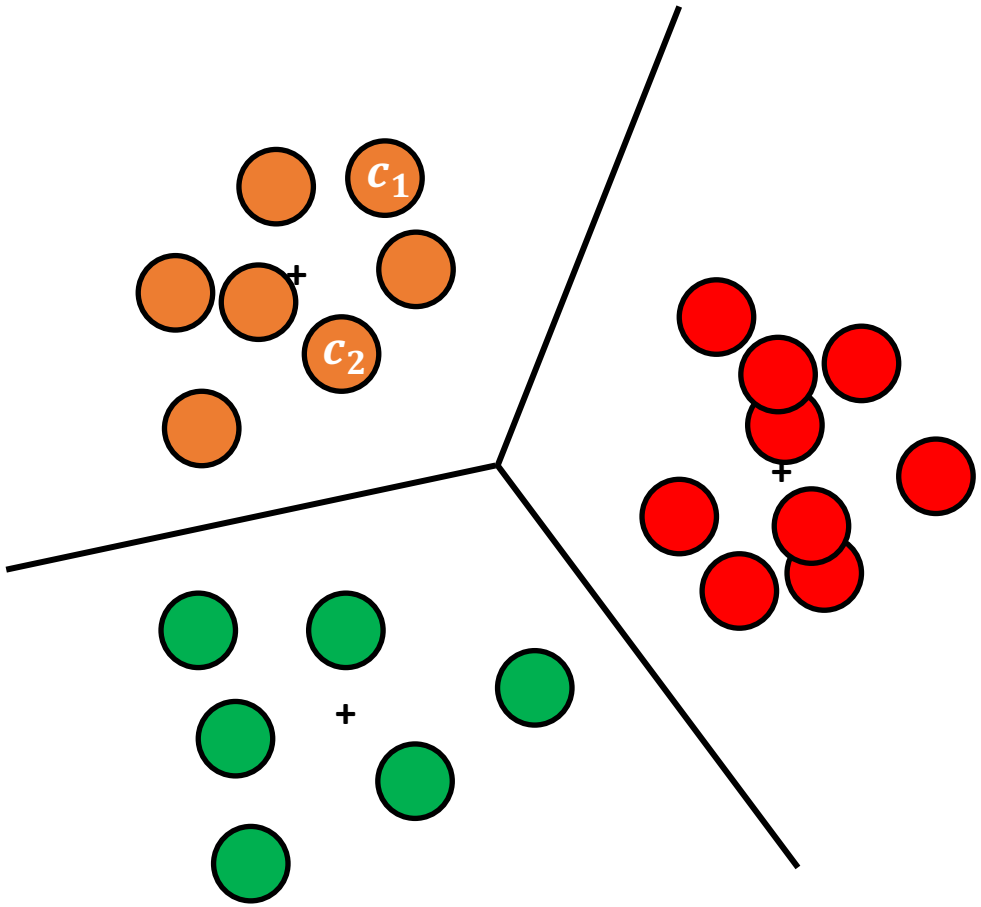
Part 2 – Remaining centers

Intuition: Given the following C_{OPT}, our algorithm could choose 2 centers from the same "optimal" cluster, costing us in the objective's distance from the optimal objective.

Finally, we get

$$\mathbb{E}[\phi] = 8(\ln k + 2) \cdot \phi_{OPT}$$

(i.e. the analysis is tight – proof omitted)



Benchmarking

A simple benchmark on UCI ML's handwritten digits dataset.

digits (k): **10** Samples (n): **1797** Features (d): **64**

Method	# Runs	Intertia (ϕ)	Time (s)
K-means++	1	69657	0.05
Random	10	69676	0.41
PCA	1	70769	0.05

Reference Notebook

Source: http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_digits.html

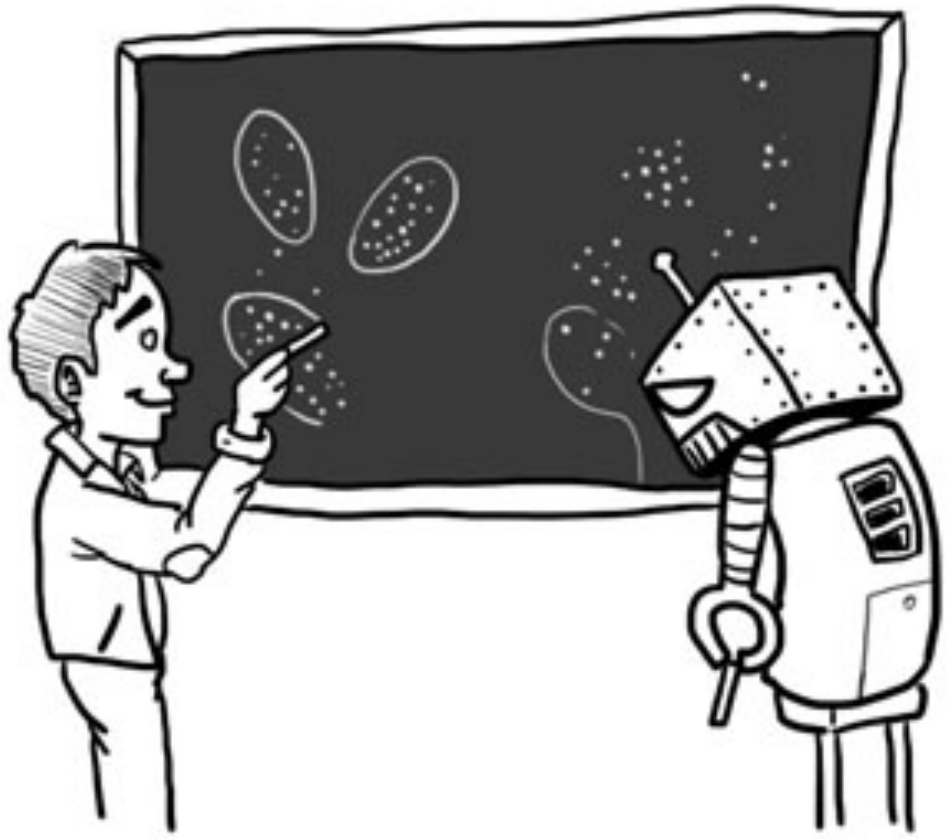
Takeaway


I DON'T ALWAYS
RUN K-MEANS
BUT WHEN I DO,
I USE K-MEANS++
imgflip.com

Summary

- Clustering definition & motivation
- Center-based clustering definition
- K-centers (Fire station problem, Farthest-first traversal)
- K-means & relation to the likelihood of Gaussian mixture
- K-means++ initialization method
- Benchmarking – K-means++ is good!

Happy Clustering!



Part 2 – Remaining centers

Lemma 3.4

- Let C be an arbitrary clustering.
- Choose $u > 0$ "uncovered" clusters from C_{OPT}.
- Let X_u denote the set of points in these clusters.
- Let $X_c = X - X_u$.
- Now suppose we add $t \leq u$ random centers to C, chosen with D^2 weighting.
- Let C' denote the the resulting clustering, and let ϕ' denote the corresponding potential.

Then, $E[\phi'] \leq \left(\phi(X_c) + 8\phi_{OPT}(X_u)\right) \cdot (1 + H_t) + \frac{u-t}{u} \cdot \phi(X_u)$.

- H_t denotes the harmonic sum, $\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{t}$.

Final proof

$$E[\phi'] \le \left(\phi(X_c) + 8\phi_{OPT}(X_u)\right) \cdot (1 + H_t) + \frac{u-t}{u} \cdot \phi(X_u)$$

Proof of **Theorem 3.1** ($E[\phi] \le 8(\ln k + 2)\phi_{OPT}$):

Consider the clustering C after we have completed Step 1.

Let A denote the C_{OPT} cluster in which we chose the first center.

Applying **Lemma 3.4** with $t = u = k - 1$, and with A being the only covered cluster, we have

$$E[\phi] \le \left(\phi(A) + 8\phi_{OPT}(X) - 8\phi_{OPT}(A)\right) \cdot (1 + H_{k-1})$$

$$\le \left(\underbrace{2\phi_{OPT}(A)}_{\textit{Lemma 3.2}} + 8\phi_{OPT}(X_{k-1}) - 8\phi_{OPT}(A)\right) \cdot \left(\underbrace{2 + \ln k}_{H_t \le 1 + \ln k}\right) \le 8(\ln k + 2)\ \phi_{OPT}$$ ■

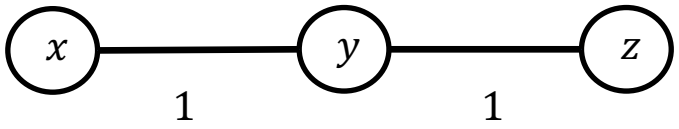
k-means

Given a set of n points A of some metric space X, find a set C of k points in X, such that we minimize $\sum_{x \in A} d^2(x, C)$

Can we use the previous algorithm ?

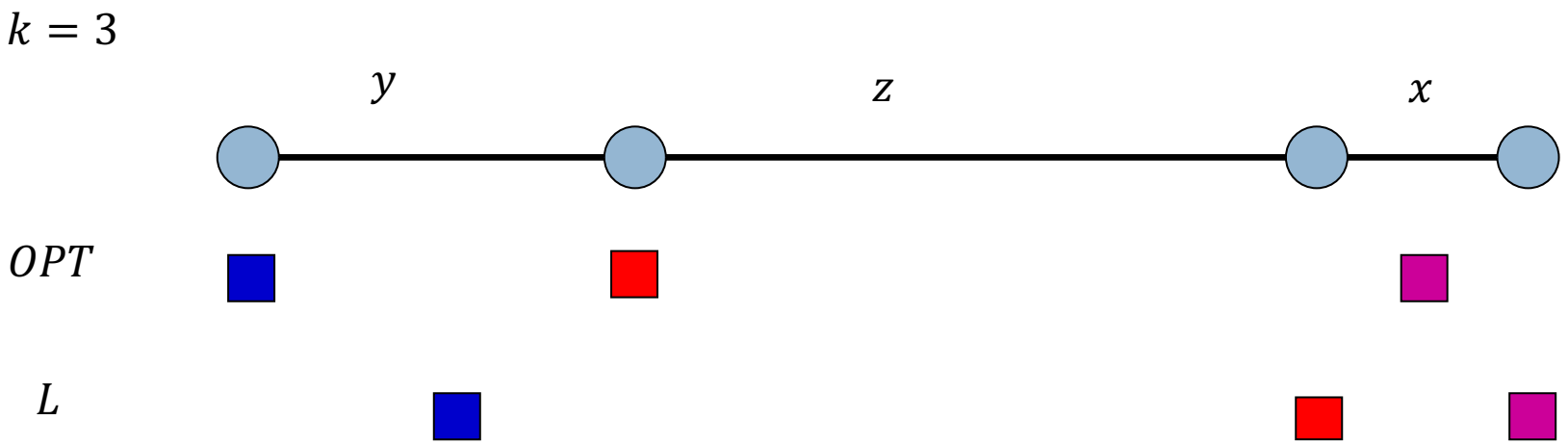
We can but the analysis breaks

$d^2(x, y)$ is not a metric



$$d^2(x, z) \geq d^2(x, y) + d^2(y, z)$$

Quality of the local opt ?



$$\frac{2\frac{y^2}{4}}{2\frac{x^2}{4}} = \frac{y^2}{x^2}$$

Can be made as large as we want