A Nearly Tight Analysis of Greedy k-means++

Christoph Grunau, Ahmet Alper Özüdogru, Václav Rozhoň, Jakub Tětek

Presentation overview

I will:

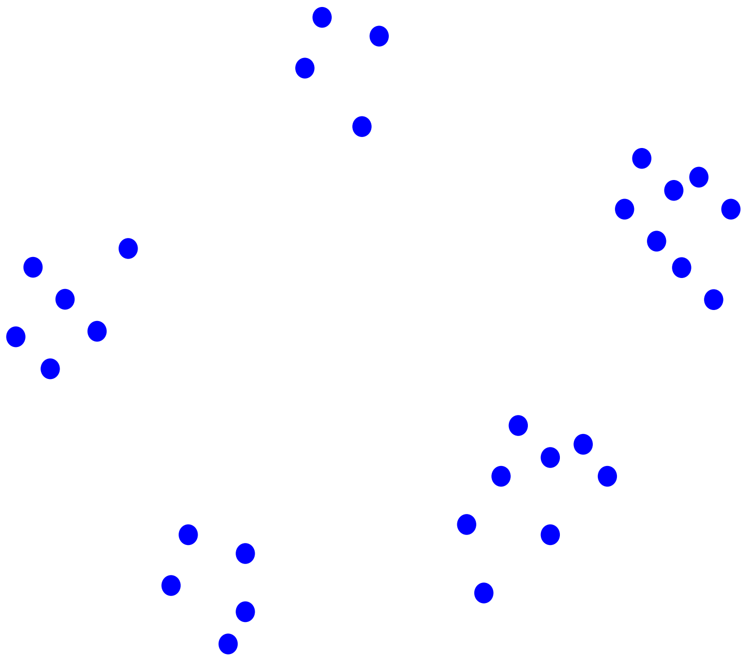
- Introduce the (greedy) k-means++ algorithm
- Explain why is the greedy version harder to understand
- (if time allows) Briefly sketch some ideas of the analysis

(1)
Introducing (greedy) k-means++

The k-means problem

Commonly used formalization of clustering

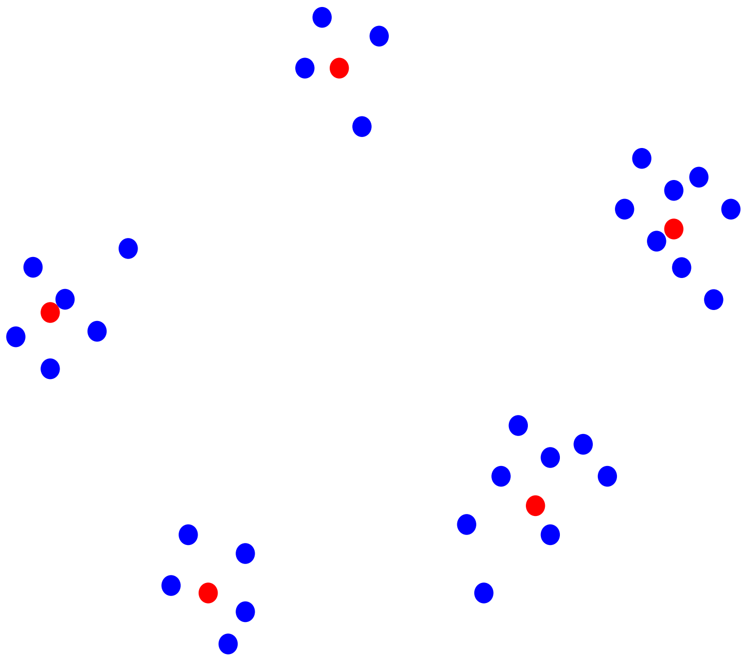
For a set $X \subseteq \mathbb{R}^d$ find a set of k centers C that minimizes $\Sigma_{x \in X} \min_{c \in C} d(x,c)^2$



The k-means problem

Commonly used formalization of clustering

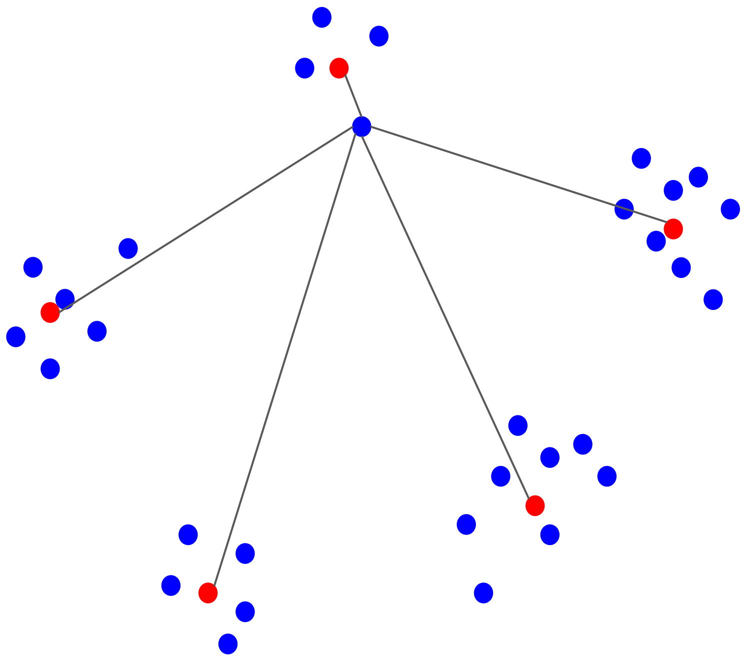
For a set $X \subseteq \mathbb{R}^d$ find a set of k centers C that minimizes $\Sigma_{x \in X} \min_{c \in C} d(x,c)^2$



The k-means problem

Commonly used formalization of clustering

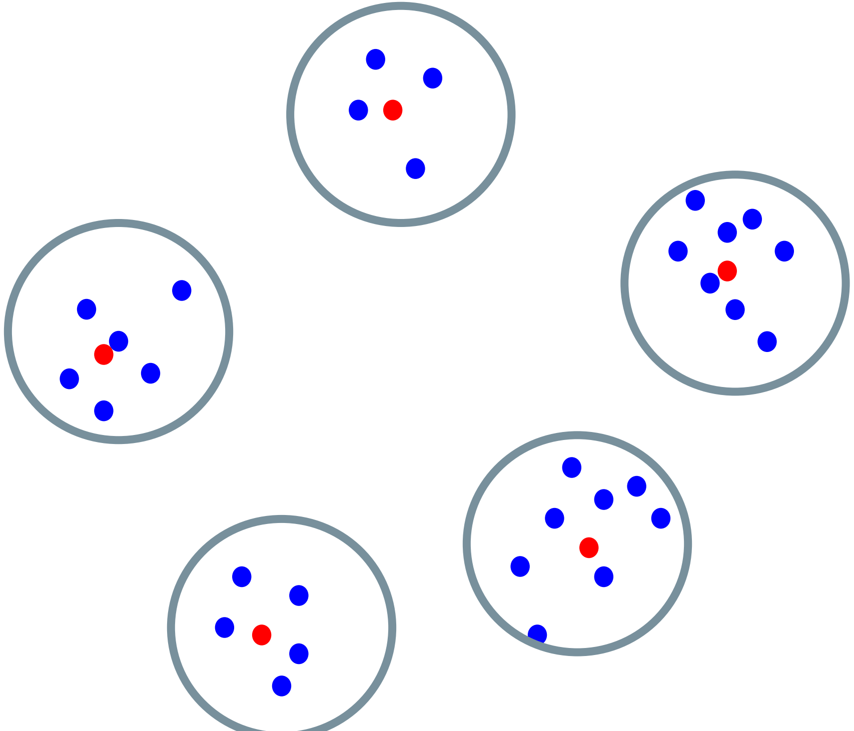
For a set $X \subseteq \mathbb{R}^d$ find a set of k centers C that minimizes $\Sigma_{x \in X} \min_{c \in C} d(x,c)^2$



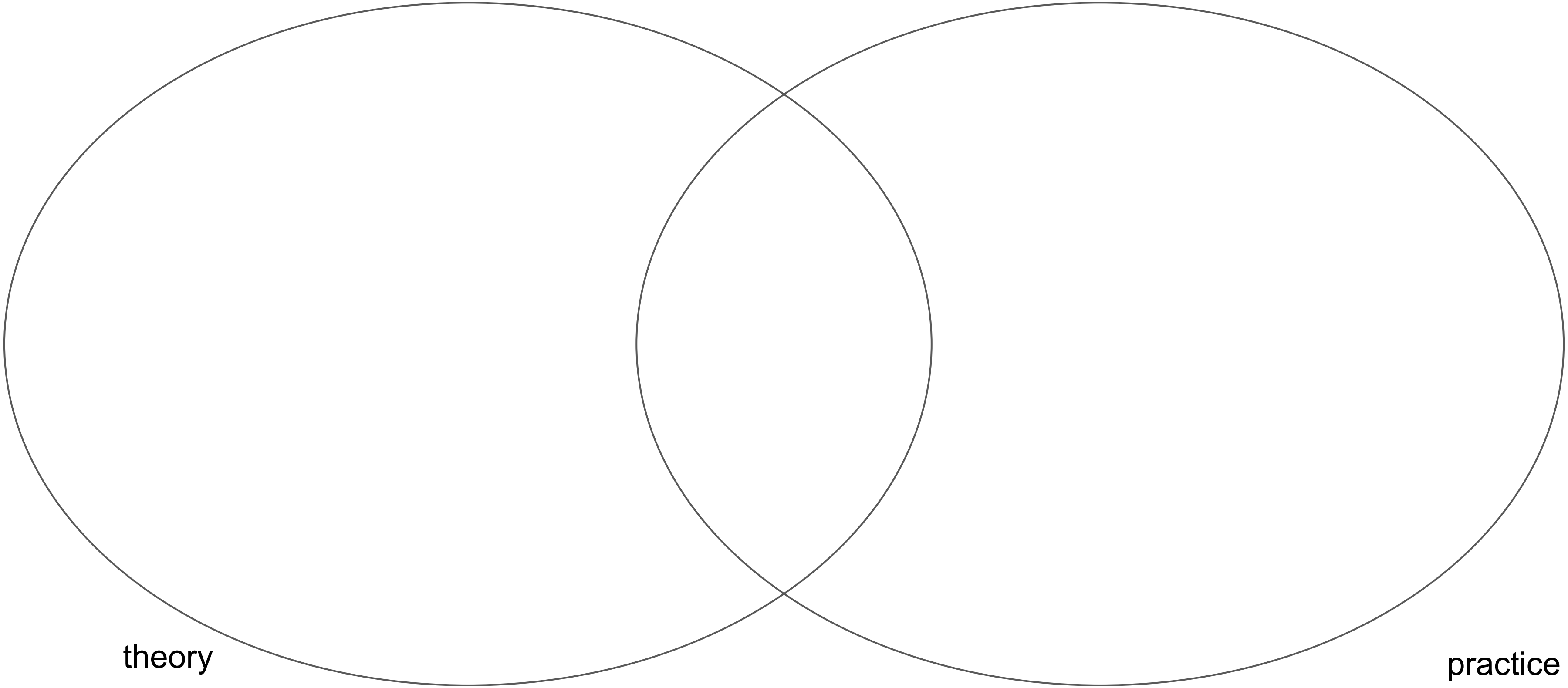
The k-means problem

Commonly used formalization of clustering

For a set $X \subseteq \mathbb{R}^d$ find a set of k centers C that minimizes $\Sigma_{x \in X} \min_{c \in C} d(x,c)^2$



k-means: theory versus practice



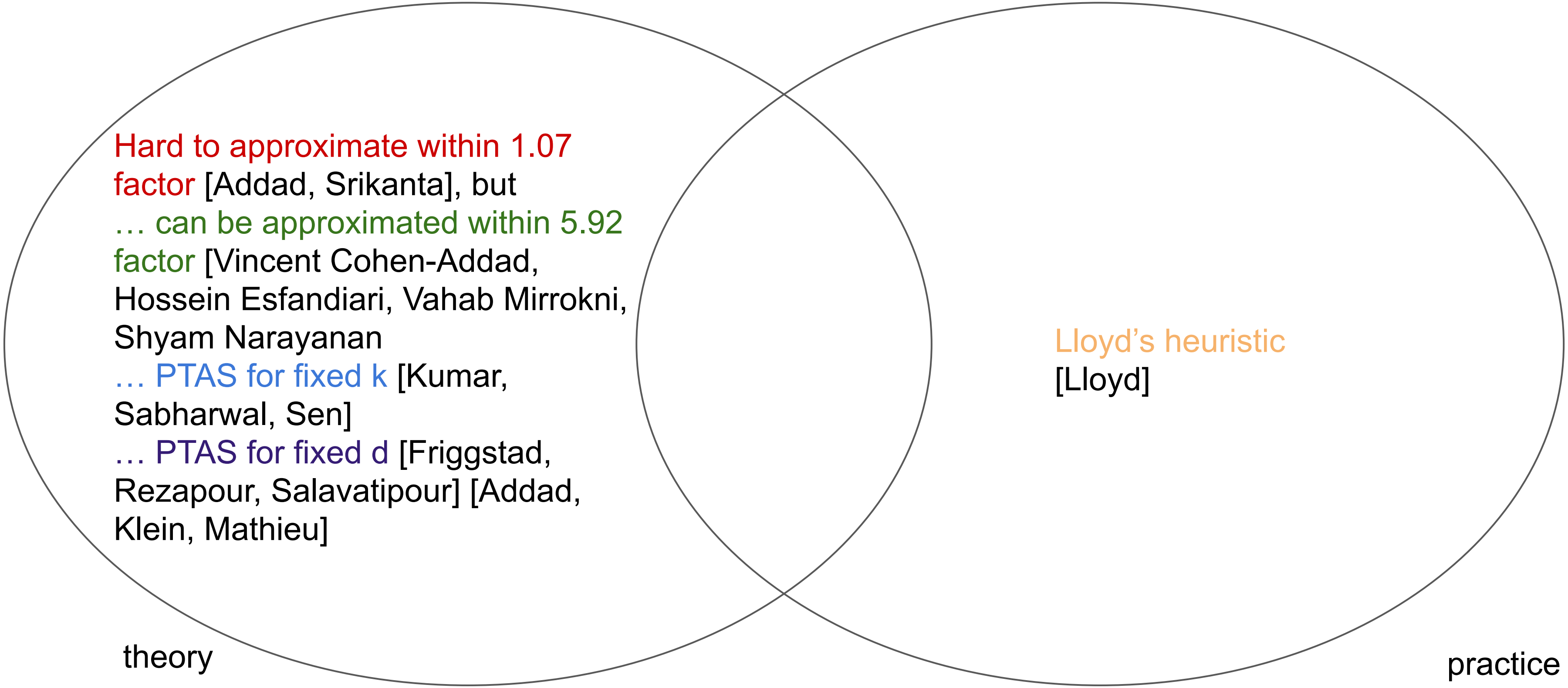
k-means: theory versus practice

Hard to approximate within 1.07 factor [Addad, Srikanta], but
… can be approximated within 5.92 factor [Vincent Cohen-Addad, Hossein Esfandiari, Vahab Mirrokni, Shyam Narayanan
… PTAS for fixed k [Kumar, Sabharwal, Sen]
… PTAS for fixed d [Friggstad, Rezapour, Salavatipour] [Addad, Klein, Mathieu]

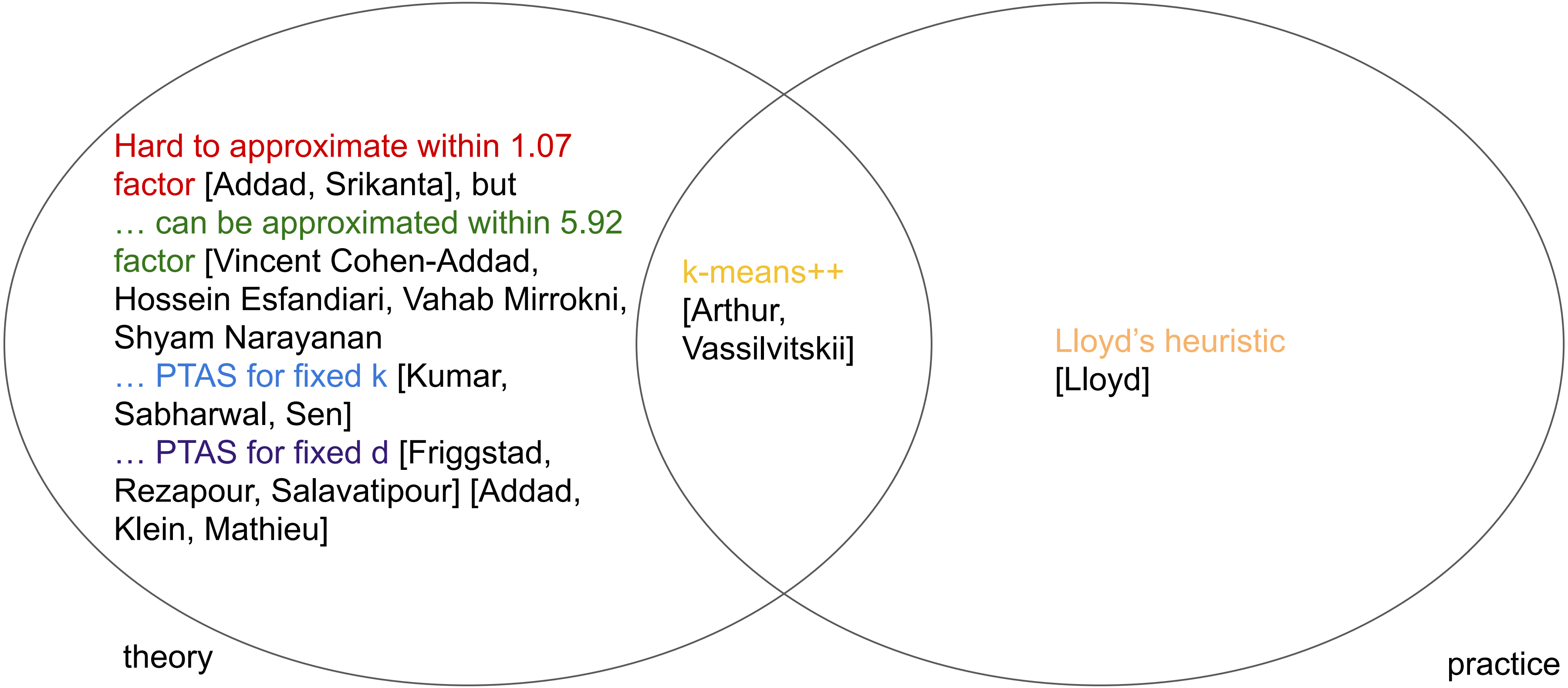
theory

practice

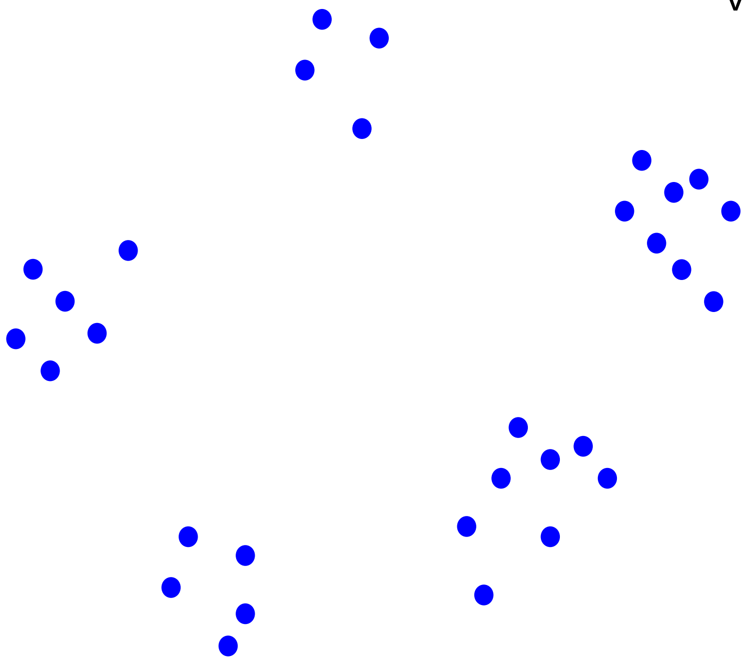
k-means: theory versus practice



k-means: theory versus practice



k-means++

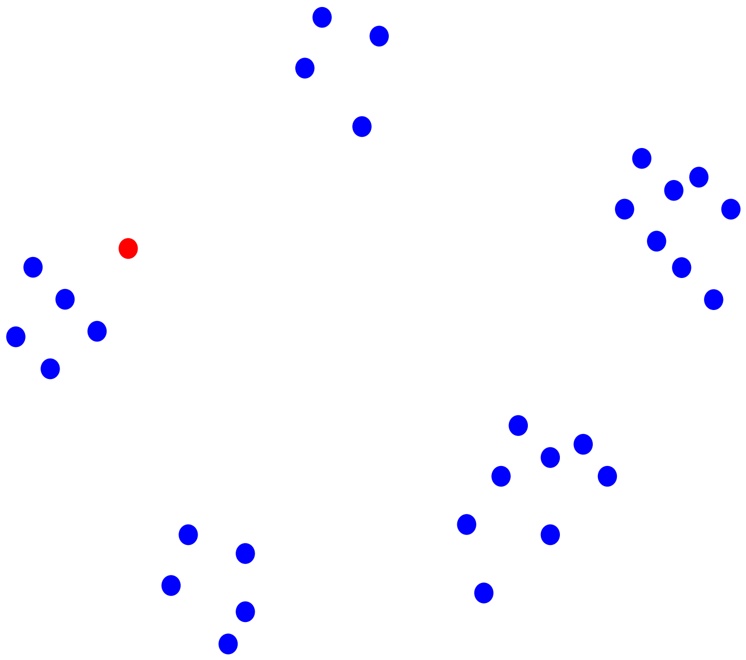


Practice: initial solution for Lloyd’s heuristic

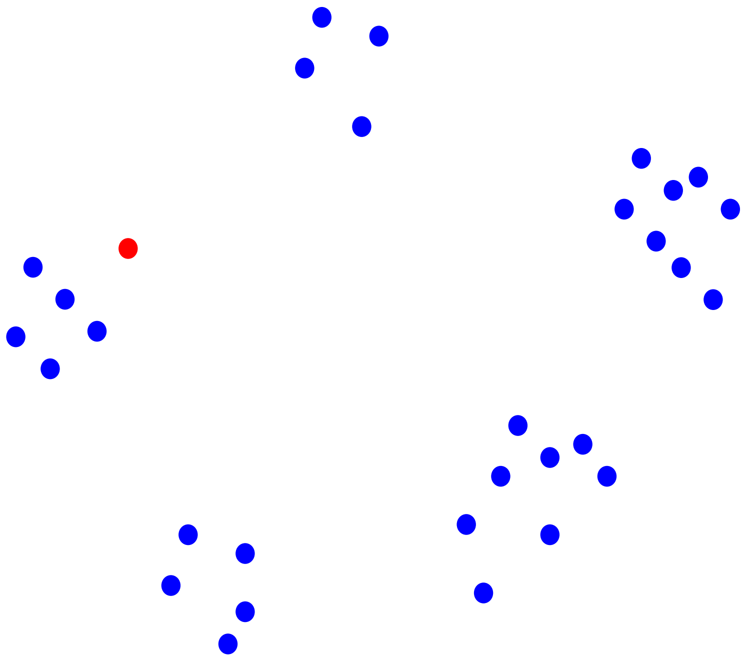
Theory: O(log k) approximation guarantee [Arthur, Vassilvitskii]

k-means++

First center: uniformly at random



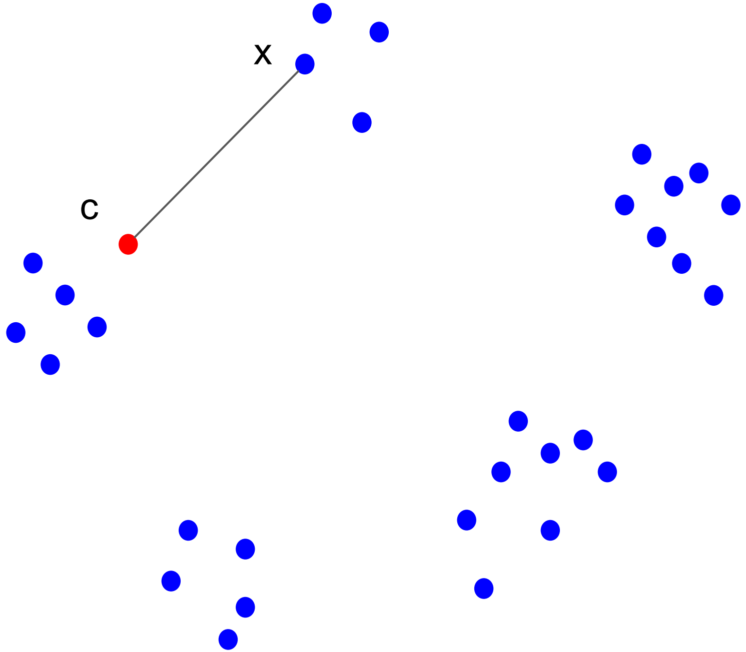
k-means++



First center: uniformly at random

Next k-1 centers: sample a point proportional to its current cost

k-means++

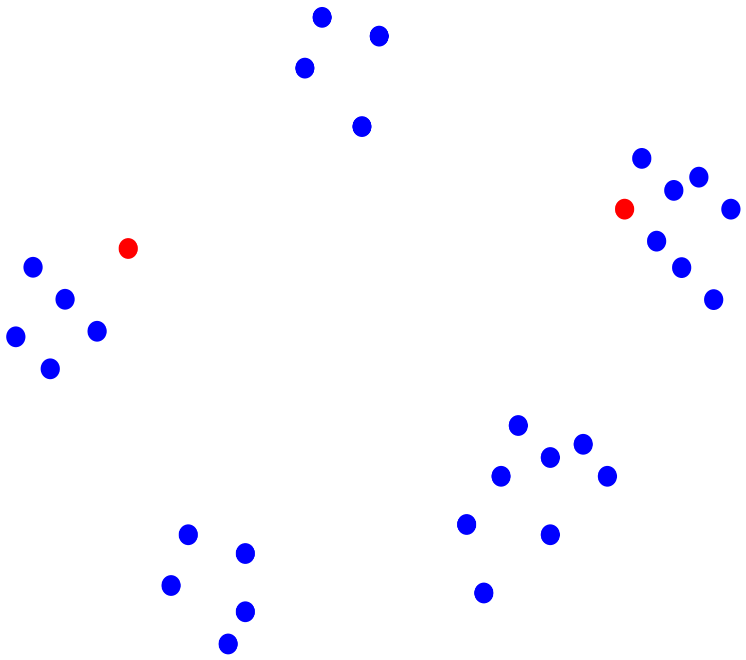


First center: uniformly at random

Next k-1 centers: sample a point proportional to its current cost, i.e.,

$$P(x \text{ sampled}) = d(x, c)^2 / \Sigma_{x' \in X} d(x', c)^2$$

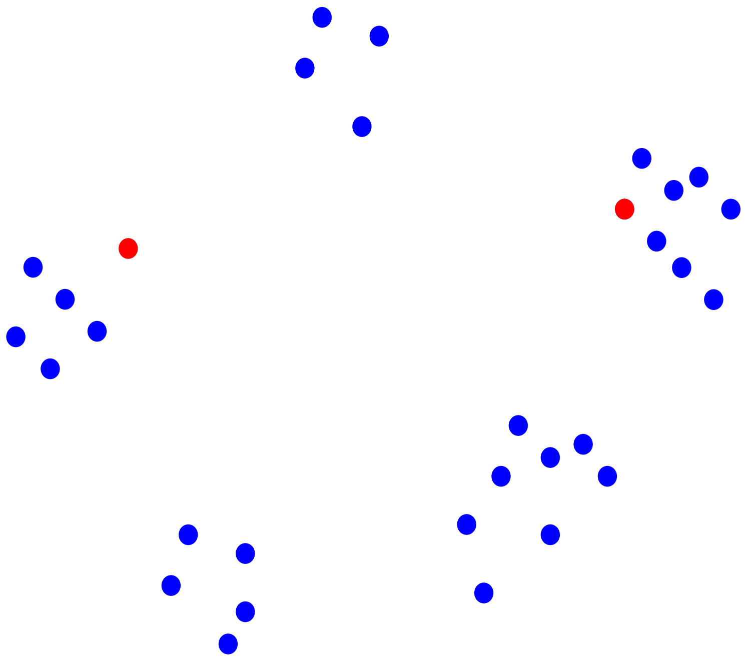
k-means++



First center: uniformly at random

Next k-1 centers: sample a point proportional to its current cost

k-means++

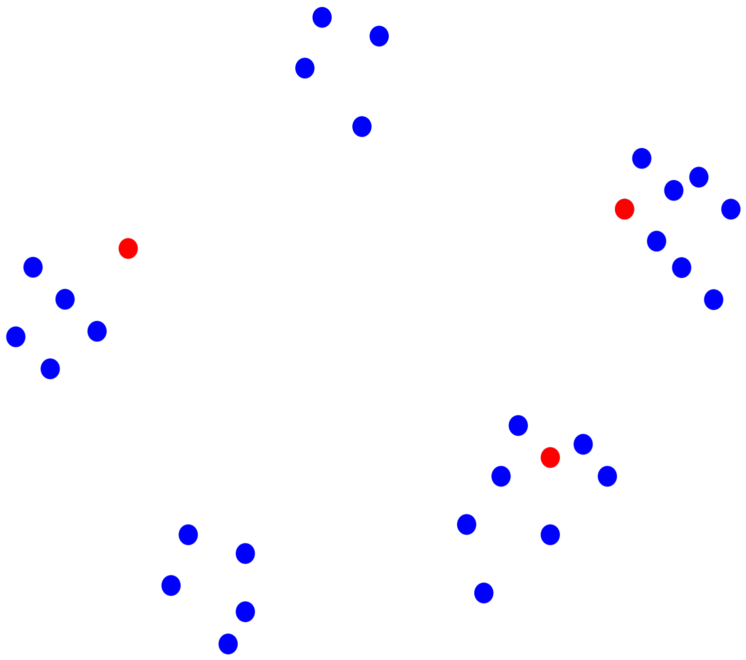


First center: uniformly at random

Next k-1 centers: sample a point proportional to its current cost

$$P(x \text{ sampled}) = \min_{c \in C} d(x, c)^2 \,/\, \Sigma_{x' \in X} \min_{c \in C} d(x', c)^2$$

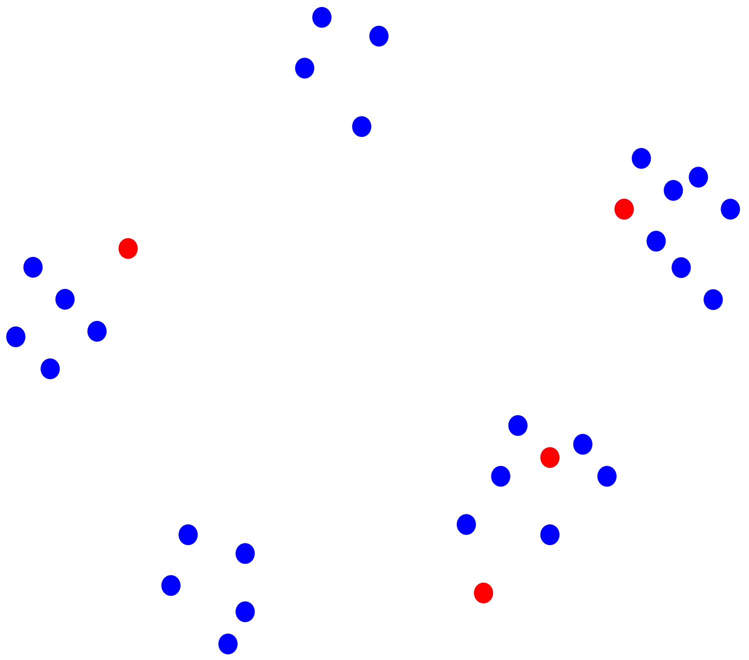
k-means++



First center: uniformly at random

Next k-1 centers: sample a point proportional to its current cost

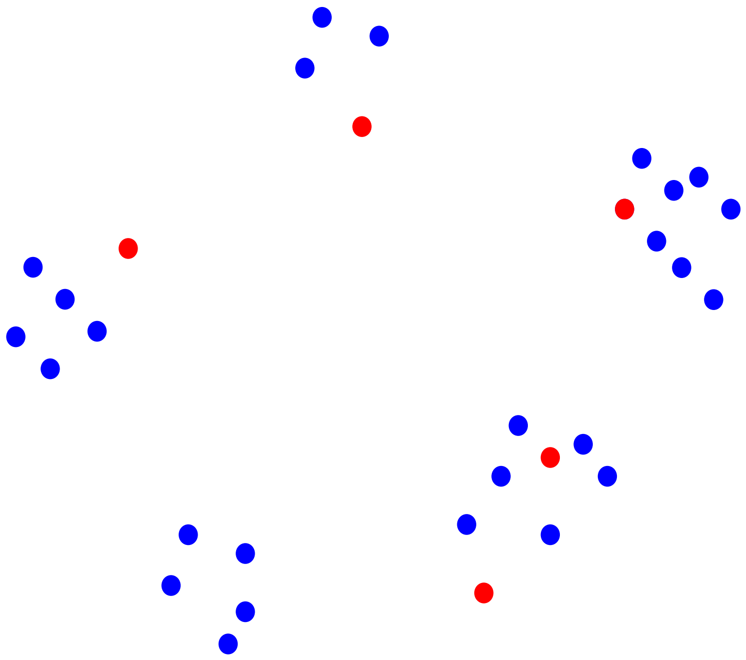
k-means++



First center: uniformly at random

Next k-1 centers: sample a point proportional to its current cost

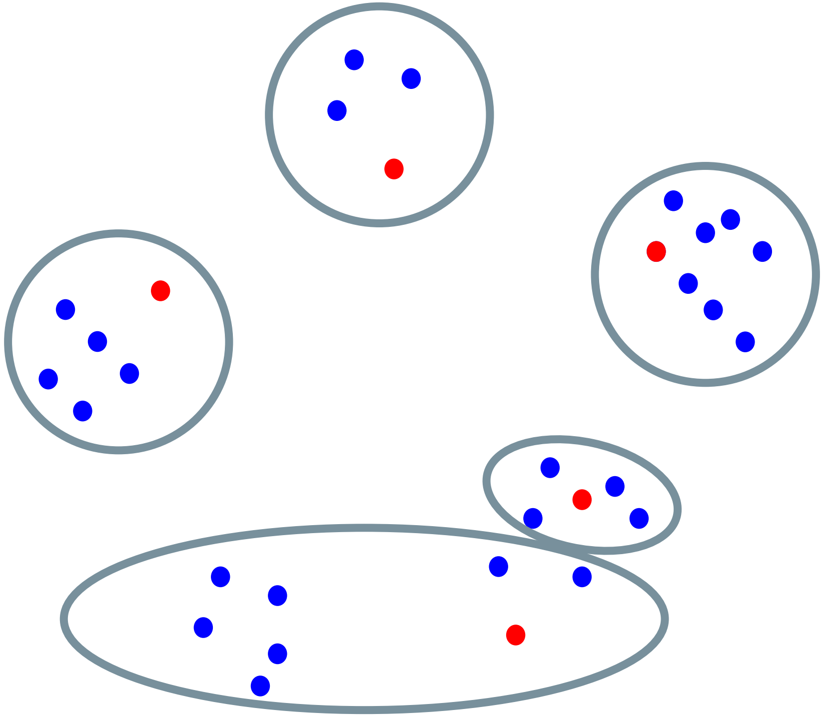
k-means++



First center: uniformly at random

Next k-1 centers: sample a point proportional to its current cost

k-means++



First center: uniformly at random

Next k-1 centers: sample a point proportional to its current cost

k-means++

[Arthur, Vassilvitskii] k-means++ is Θ(log k)-approximate.

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

Algorithm 4 k-means++ seeding

Input: X, k

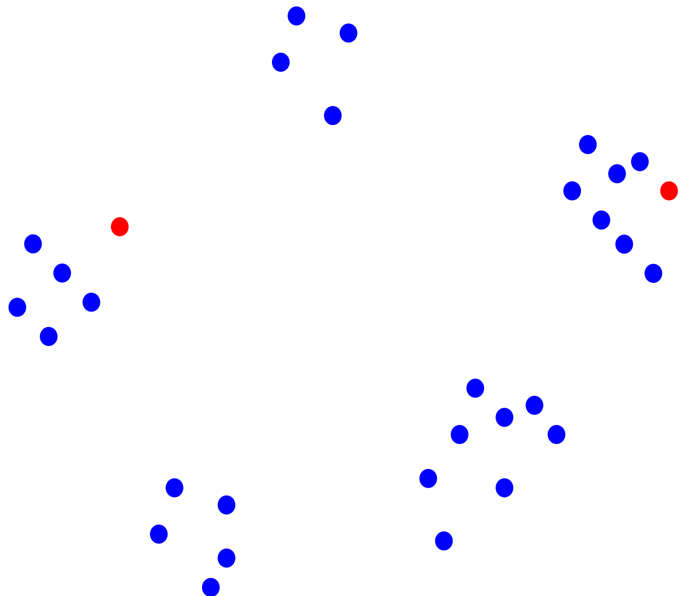
1: Uniformly sample $x \in X$ and set $C_1 = \{x\}$.
2: **for** $i \leftarrow 1, 2, \ldots, k-1$ **do**
3: Sample $x \in X$ with probability $\frac{\min_{c \in C_i} d(x,c)^2}{\sum_{x \in X} \min_{c \in C_i} d(x,c)^2}$ and set $C_{i+1} = C_i \cup \{x\}$.
4: **return** $C := C_k$

Greedy k-means++

Greedy Vs. Non-greedy

In each step, pick ℓ potential centers

Select the one that causes the biggest cost decrease

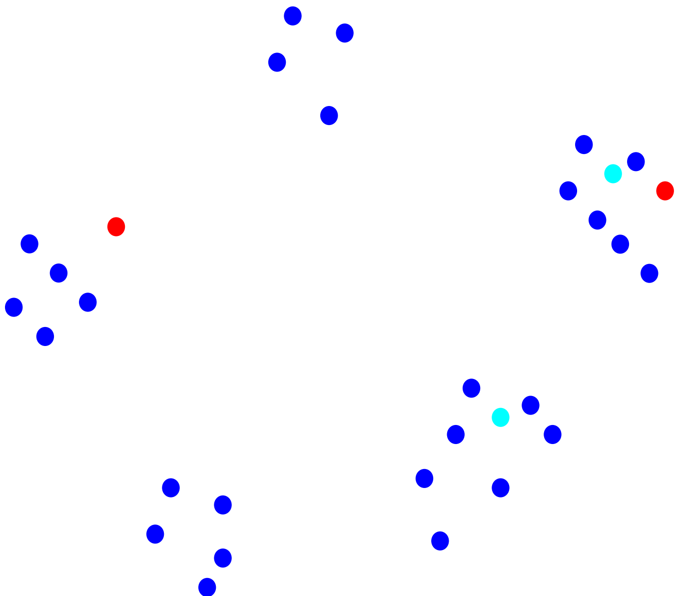


Greedy k-means++

Greedy Vs. Non-greedy

In each step, pick ℓ potential centers

Select the one that causes the biggest cost decrease



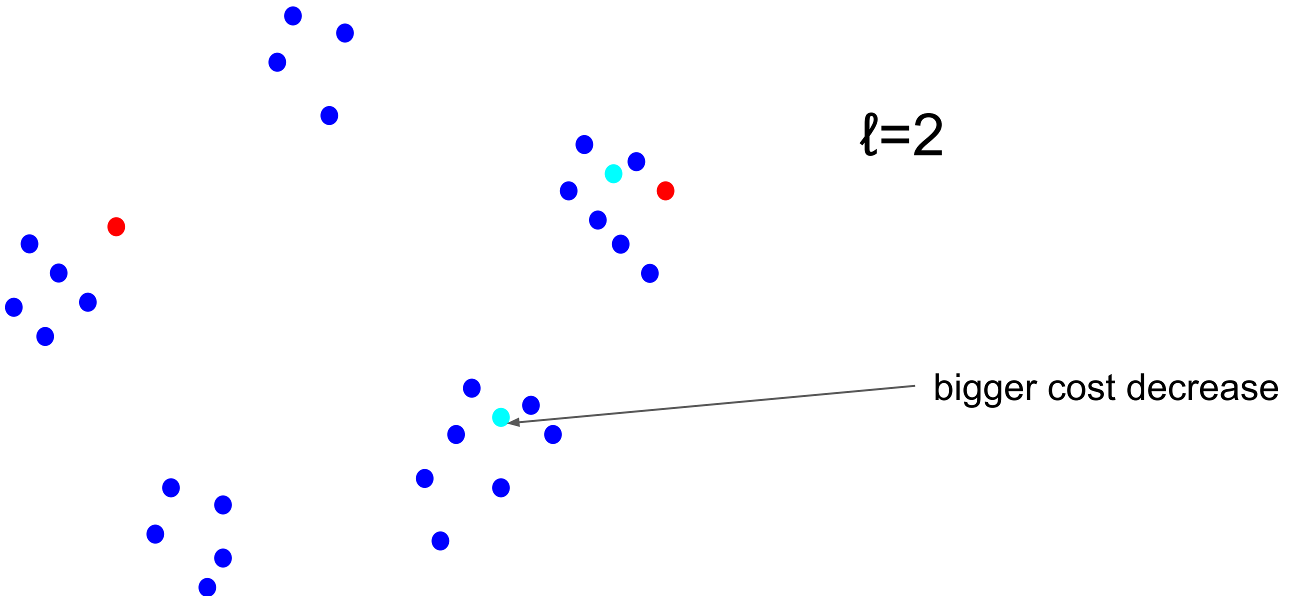
ℓ=2

Greedy k-means++

Greedy Vs. Non-greedy

In each step, pick ℓ potential centers

Select the one that causes the biggest cost decrease



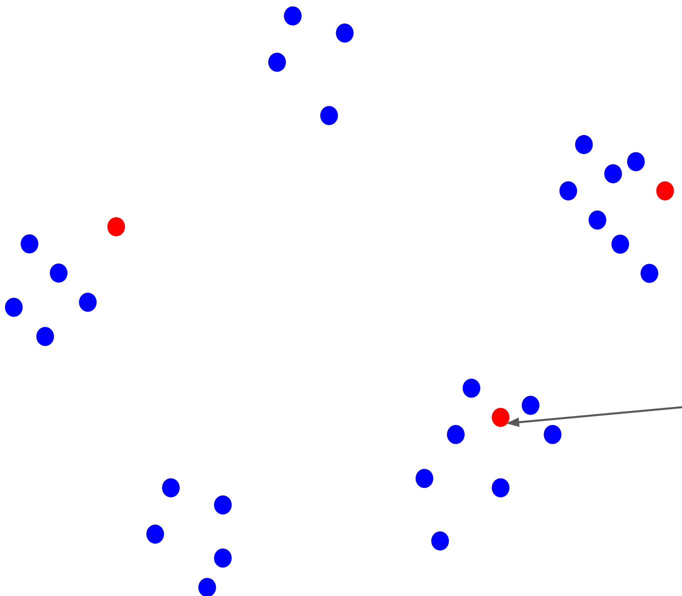
Greedy k-means++

Greedy Vs. Non-greedy

In each step, pick ℓ potential centers

Select the one that causes the biggest cost decrease

ℓ=2



bigger cost decrease

Greedy k-means++

A different variant of k-means++ commonly used in e.g. the scikit-learn library.

[Arthur, Vassilvitskii] asked for its analysis.

Algorithm 5 Greedy k-means++ seeding

Input: X, k, ℓ

Uniformly independently sample $c_1^1, \ldots, c_1^\ell \in X$;

Let $c_1 = \arg\min_{c \in \{c_1^1, \ldots, c_1^\ell\}} \sum_{x \in X} d(x, c)^2$ and set $C_1 = \{c_1\}$.

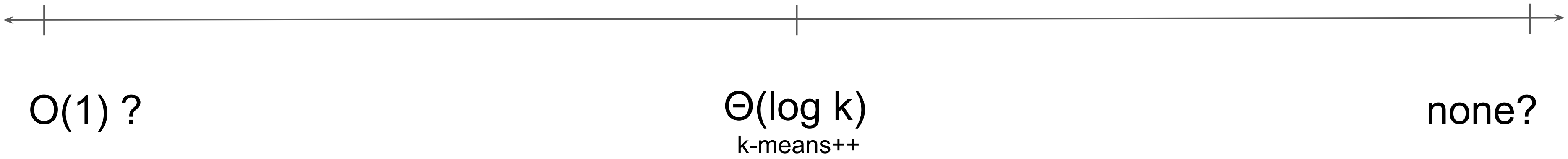
for $i \leftarrow 1, 2, \ldots, k-1$ **do**

  Sample $c_{i+1}^1, \ldots, c_{i+1}^\ell \in X$ independently, sampling x with probability $\frac{\min_{c \in C_i} d(x,c)^2}{\sum_{x \in X} \min_{c \in C_i} d(x,c)^2}$.

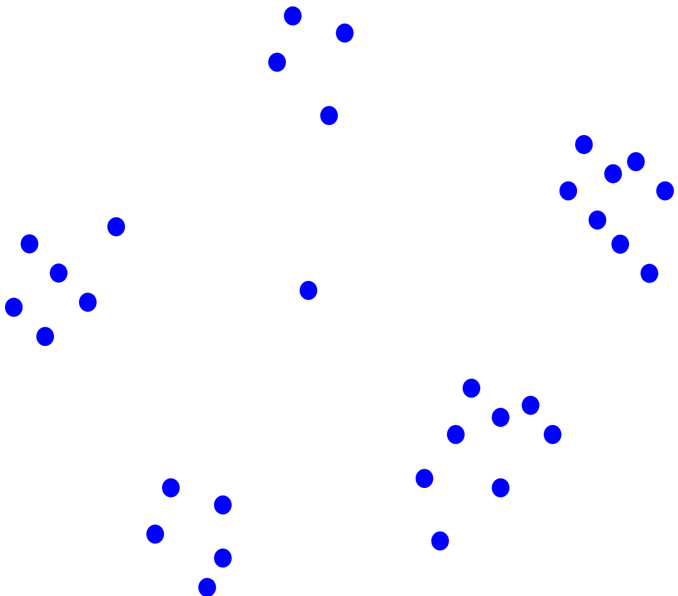
  Let $c_{i+1} = \arg\min_{c \in \{c_i^1, \ldots, c_i^\ell\}} \sum_{x \in X} \min_{c' \in C_i \cup \{c\}} d(x, c')^2$ and set $C_{i+1} = C_i \cup \{c_{i+1}\}$.

return $C := C_k$

Guarantees for this algorithm? (say ℓ=2)



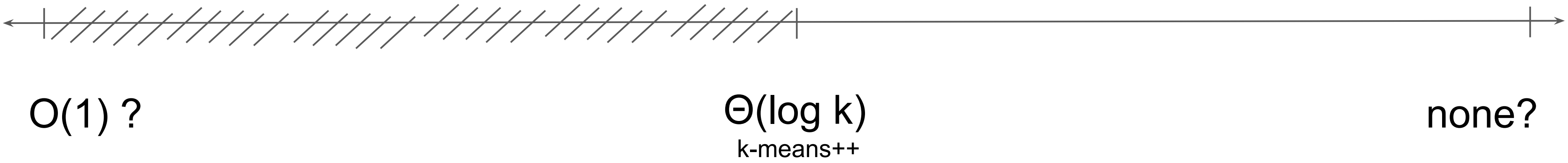
Greedy k-means++



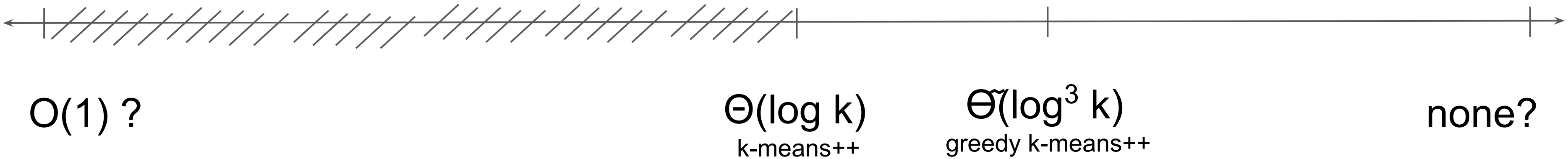
In the worst case, greedy is not better!

[Bhattacharya, Eube, Röglin, Schmidt]: $\Omega(\ell * \log k)$-approximate

Guarantees for this algorithm? (say ℓ=2)



Guarantees for this algorithm? (say ℓ=2)



Our results:

$O(\ell^3 \times \log^3 k)$ upper bound,

$\Omega(\ell^3 \times \log^3 k \,/\, \log^2(\ell \times \log k))$ lower bound

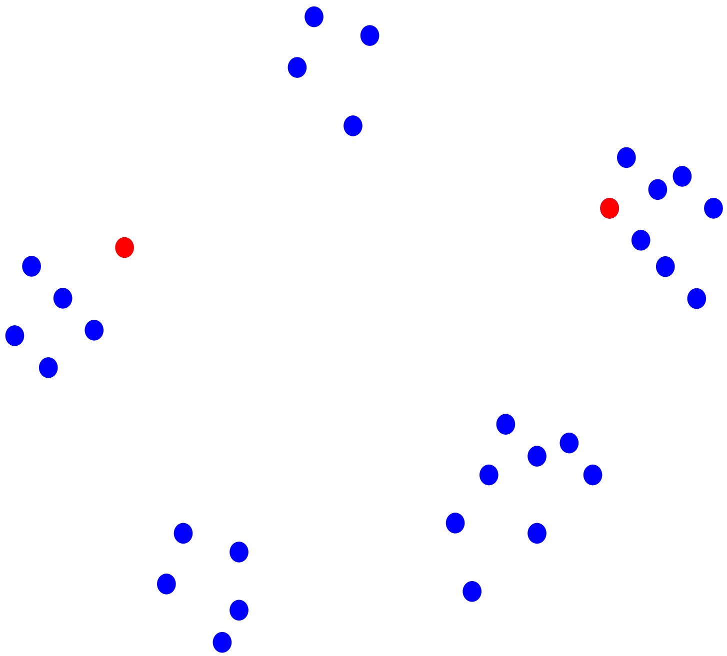
In scikit-learn $\ell = \Theta(\log k)$, hence the algorithm is $\tilde{\Theta}(\log^6 k)$ approximate!

(2)
Why can’t we just recycle k-means++ analysis?

The main k-means++ Lemma
from [Arthur, Vassilvitski]

The main k-means++ Lemma

from [Arthur, Vassilvitski]



Lemma: Condition on sampling c from some optimal cluster K. Then,

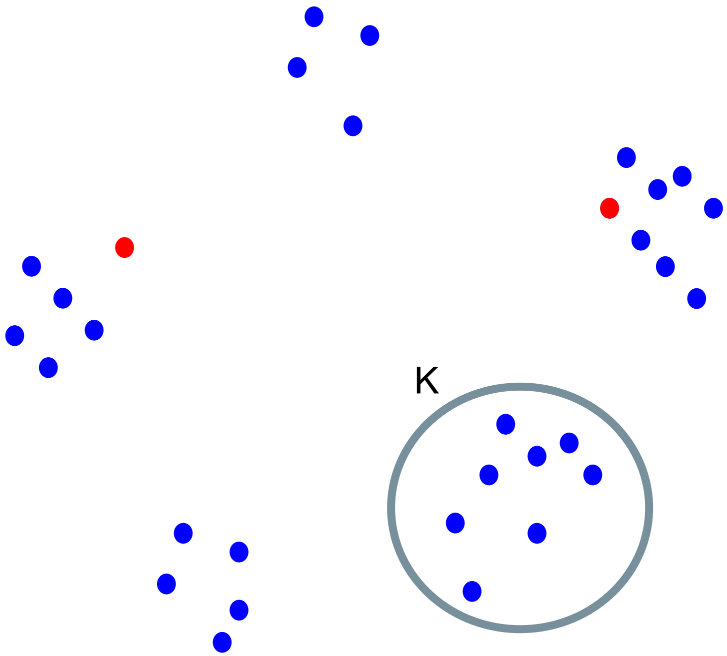
$$E\left[\Sigma_{x \in K} \min_{c' \in (C \cup c)} d(x,c')^2 \right] \leq 8 \times \Sigma_{x \in K} d(x,\mu(K))^2$$

The main k-means++ Lemma

from [Arthur, Vassilvitski]

Lemma: Condition on sampling c from some optimal cluster K. Then,

$$E\left[\ \Sigma_{x\in K}\min_{c'\in(C\cup c)} d(x,c')^2\ \right] \le 8 \times \Sigma_{x\in K} d(x,\mu(K))^2$$

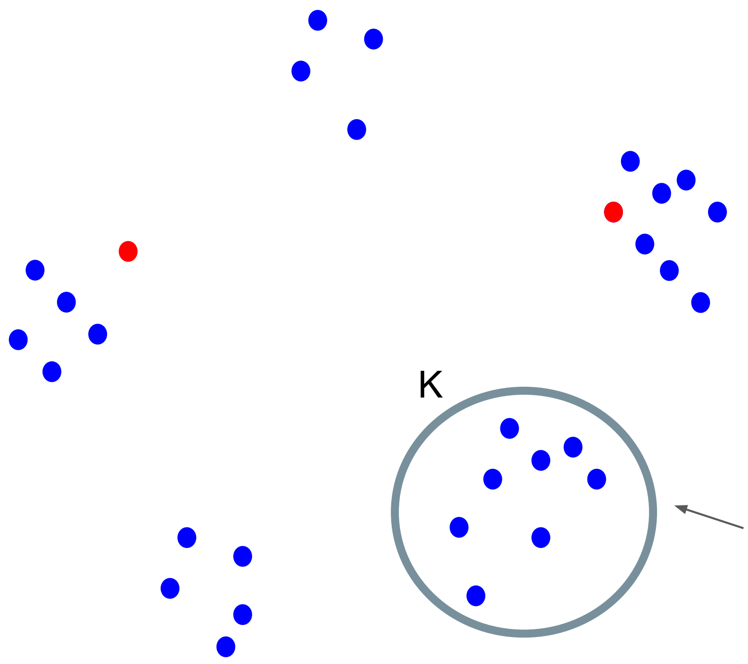


The main k-means++ Lemma

from [Arthur, Vassilvitski]

Lemma: Condition on sampling c from some optimal cluster K. Then,

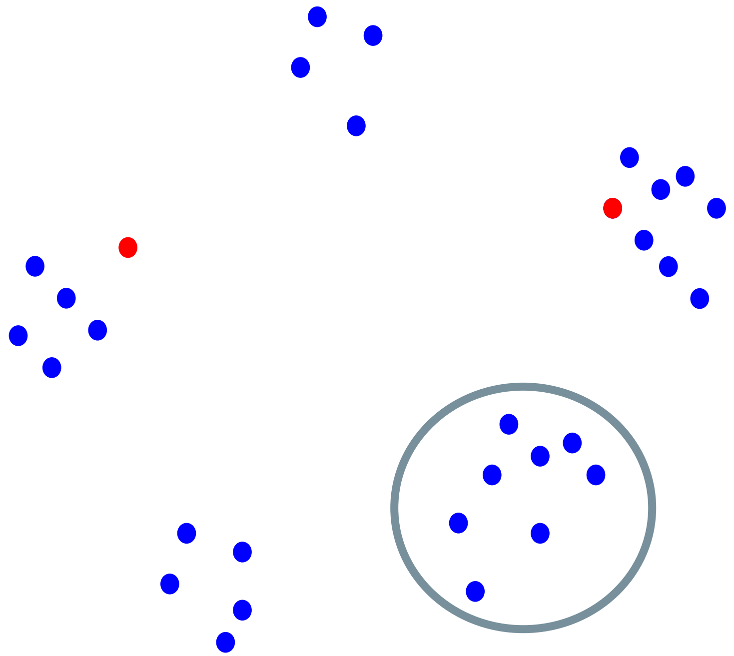
$$E\left[\Sigma_{x \in K} \min_{c' \in (C \cup c)} d(x,c')^2 \right] \leq 8 \times \Sigma_{x \in K} d(x,\mu(K))^2$$



With the new center, its cost gets at most 8 times worse than the optimal cost, in expectation.

The main k-means++ Lemma

from [Arthur, Vassilvitski]



Lemma: Condition on sampling c from some optimal cluster K. Then,

$$E\left[\Sigma_{x \in K} \min_{c' \in (C \cup c)} d(x, c')^2 \right] \leq 8 \times \Sigma_{x \in K} d(x, \mu(K))^2$$

Proof sketch:

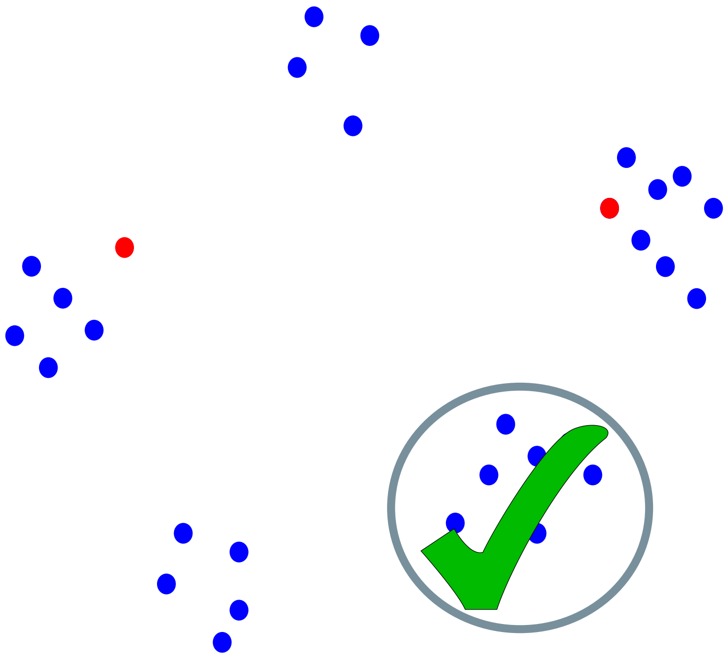
1. Prove it for uniform distribution.
2. In general,
    a. or current centers are far from K (reduces to 1)
    b. at least one center is close to K (done)

The main k-means++ Lemma

from [Arthur, Vassilvitski]

Lemma: Condition on sampling c from some optimal cluster K. Then,

$$\cancel{\mathbb{E}[}\ \Sigma_{x \in K} \min_{c' \in (C \cup c)} d(x, c')^{\cancel{2}}\ ]\ \cancel{\le 8} \times \Sigma_{x \in K} d(x, \mu(K))^2$$

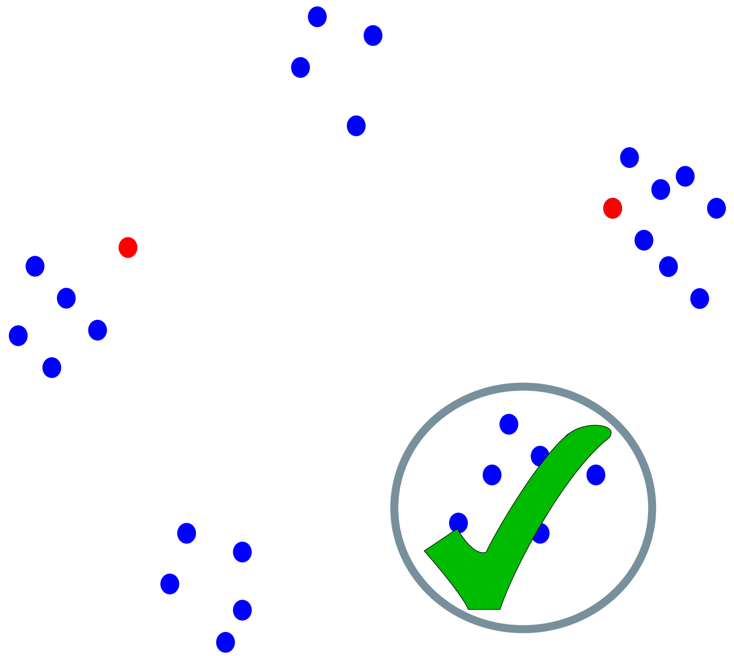


The main k-means++ Lemma

from [Arthur, Vassilvitski]

Lemma: Condition on sampling c from some optimal cluster K. Then,

$$\cancel{\mathbb{E}[}\ \Sigma_{x\in K}\min_{c'\in(C\cup c)} d(x,c')^{\cancel{2}}\ ] \le \cancel{8} \times \Sigma_{x\in K} d(x,\mu(K))^2$$



The rest of the analysis is about computing the probability of sampling from an already “covered” cluster.

The problem with sampling ℓ >1 points

Algorithm 5 Greedy k-means++ seeding

Input: X, k, ℓ

1: Uniformly independently sample $c_1^1, \ldots, c_1^\ell \in X$;

2: Let $c_1 = \cancel{\arg\min_{c \in \{c_1^1, \ldots, c_1^\ell\}} \sum_{x \in X} d(x, c)^2}$ and set $C_1 = \{c_1\}$.

adversarially

3: **for** $i \leftarrow 1, 2, \ldots, k-1$ **do**

4: Sample $c_{i+1}^1, \ldots, c_{i+1}^\ell \in X$ independently, sampling x with probability $\frac{\min_{c \in C_i} d(x,c)^2}{\sum_{x \in X} \min_{c \in C_i} d(x,c)^2}$.

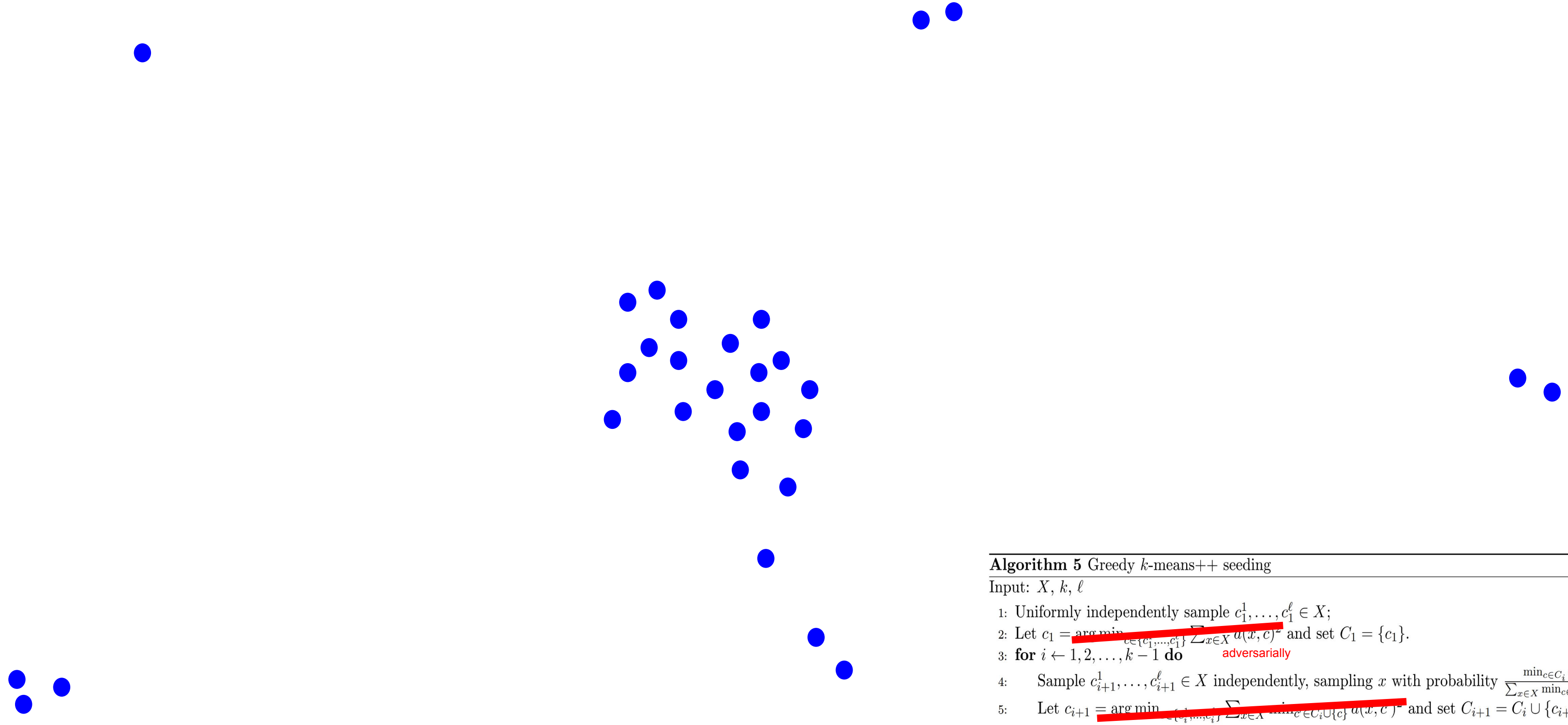
5: Let $c_{i+1} = \cancel{\arg\min_{c \in \{c_i^1, \ldots, c_i^\ell\}} \sum_{x \in X} \min_{c' \in C_i \cup \{c\}} d(x, c')^2}$ and set $C_{i+1} = C_i \cup \{c_{i+1}\}$.

adversarially

6: **return** $C := C_k$

The adversarial version is only $\Omega(k^{1-1/\ell})$ approximate!

$\Omega(k^{1-1/\ell})$ approximation for adversarial algorithm



Algorithm 5 Greedy k-means++ seeding

Input: X, k, ℓ

1: Uniformly independently sample $c_1^1, \dots, c_1^\ell \in X$;

2: Let $c_1 = \cancel{\arg\min_{c \in \{c_1^1, \dots, c_1^\ell\}} \sum_{x \in X} d(x,c)^2}$ and set $C_1 = \{c_1\}$.

adversarially

3: **for** $i \leftarrow 1, 2, \dots, k-1$ **do**

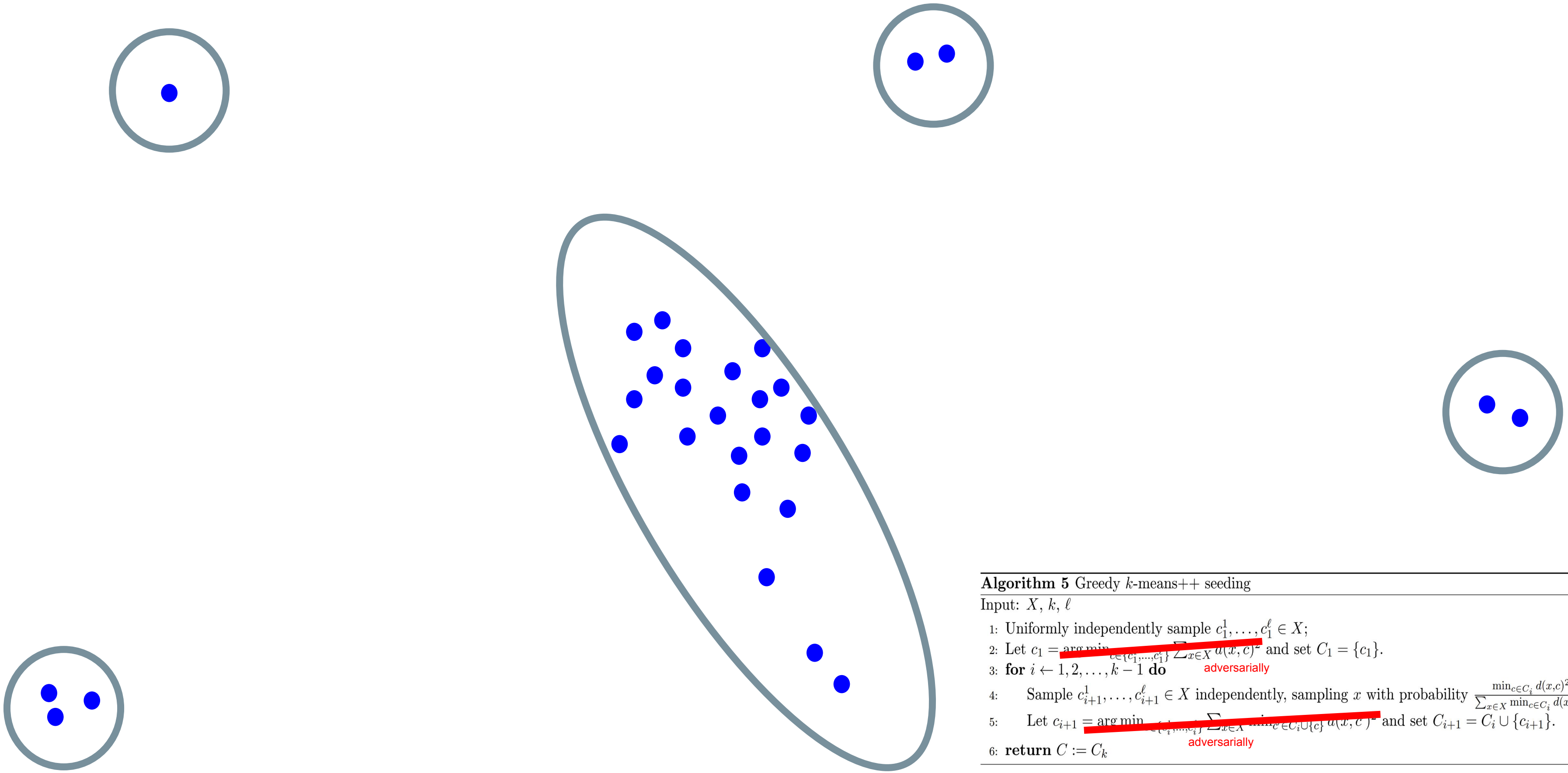
4: Sample $c_{i+1}^1, \dots, c_{i+1}^\ell \in X$ independently, sampling x with probability $\frac{\min_{c \in C_i} d(x,c)^2}{\sum_{x \in X} \min_{c \in C_i} d(x,c)^2}$.

5: Let $c_{i+1} = \cancel{\arg\min_{c \in \{c_{i+1}^1, \dots, c_{i+1}^\ell\}} \sum_{x \in X} \min_{c' \in C_i \cup \{c\}} d(x,c')^2}$ and set $C_{i+1} = C_i \cup \{c_{i+1}\}$.

adversarially

6: **return** $C := C_k$

$\Omega(k^{1-1/\ell})$ approximation for adversarial algorithm



Algorithm 5 Greedy k-means++ seeding

Input: X, k, ℓ

Uniformly independently sample $c_1^1, \ldots, c_1^\ell \in X$;

Let $c_1 = \cancel{\arg\min_{c \in \{c_1^1, \ldots, c_1^\ell\}} \sum_{x \in X} d(x,c)^2}$ adversarially and set $C_1 = \{c_1\}$.

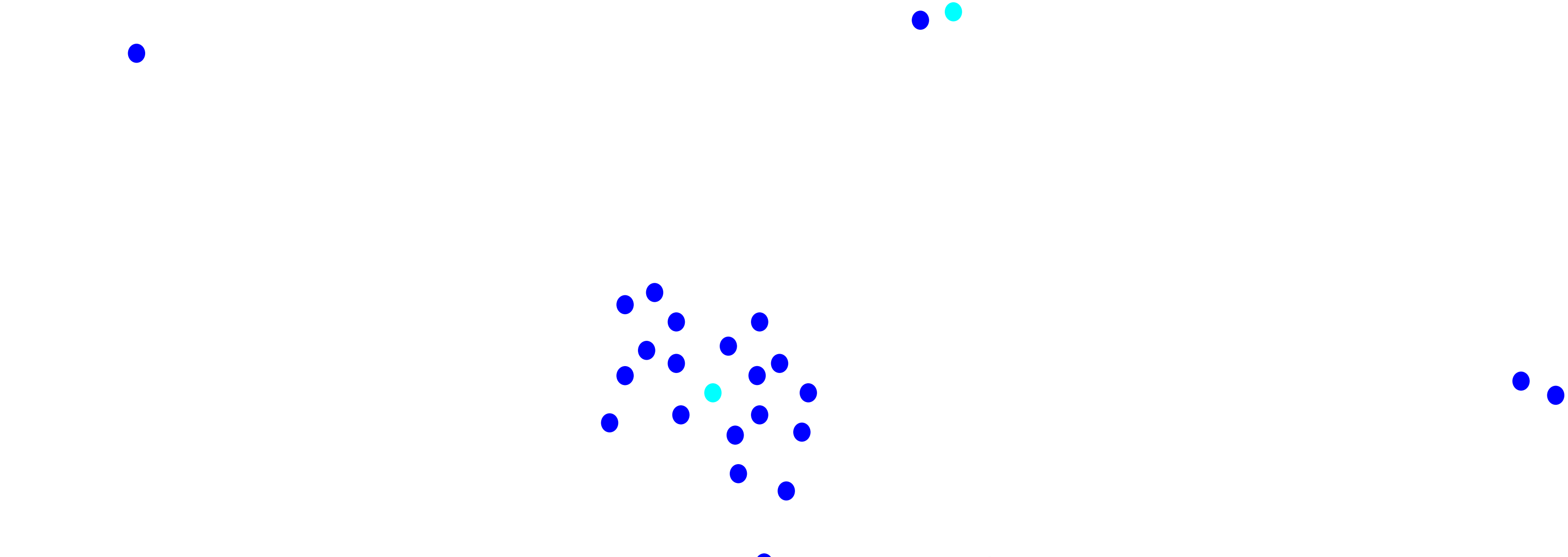
for $i \leftarrow 1, 2, \ldots, k-1$ **do**

  Sample $c_{i+1}^1, \ldots, c_{i+1}^\ell \in X$ independently, sampling x with probability $\frac{\min_{c \in C_i} d(x,c)^2}{\sum_{x \in X} \min_{c \in C_i} d(x,c)^2}$.

  Let $c_{i+1} = \cancel{\arg\min_{c \in \{c_i^1, \ldots, c_i^\ell\}} \sum_{x \in X} \min_{c' \in C_i \cup \{c\}} d(x,c')^2}$ adversarially and set $C_{i+1} = C_i \cup \{c_{i+1}\}$.

return $C := C_k$

$\Omega(k^{1-1/\ell})$ approximation for adversarial algorithm



Algorithm 5 Greedy k-means++ seeding

Input: X, k, ℓ

Uniformly independently sample $c_1^1, \ldots, c_1^\ell \in X$;

Let $c_1 = \cancel{\arg\min_{c \in \{c_1^1, \ldots, c_1^\ell\}} \sum_{x \in X} d(x, c)^2}$ and set $C_1 = \{c_1\}$.

adversarially

for $i \leftarrow 1, 2, \ldots, k-1$ **do**

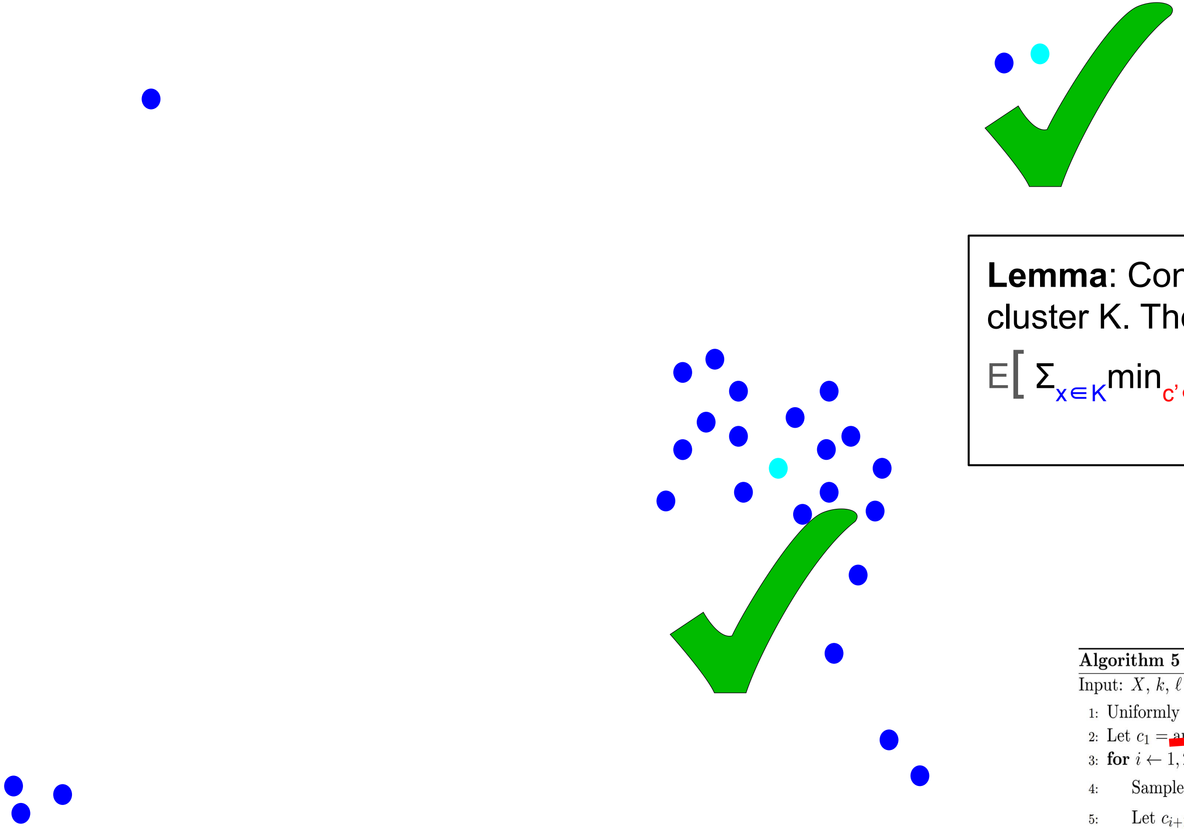
  Sample $c_{i+1}^1, \ldots, c_{i+1}^\ell \in X$ independently, sampling x with probability $\frac{\min_{c \in C_i} d(x,c)^2}{\sum_{x \in X} \min_{c \in C_i} d(x,c)^2}$.

  Let $c_{i+1} = \cancel{\arg\min_{c \in \{c_i^1, \ldots, c_i^\ell\}} \sum_{x \in X} \min_{c' \in C_i \cup \{c\}} d(x, c')^2}$ and set $C_{i+1} = C_i \cup \{c_{i+1}\}$.

adversarially

return $C := C_k$

$\Omega(k^{1-1/\ell})$ approximation for adversarial algorithm



Lemma: Condition on sampling c from some optimal cluster K. Then,

$$E\left[\sum_{x\in K}\min_{c'\in(C\cup c)} d(x,c')^2\right] \leq 8 \times \sum_{x\in K} d(x,\mu(K))^2$$

Algorithm 5 Greedy k-means++ seeding

Input: X, k, ℓ

Uniformly independently sample $c_1^1,\dots,c_1^\ell \in X$;

Let $c_1 = \cancel{\arg\min_{c\in\{c_1^1,\dots,c_1^\ell\}} \sum_{x\in X} d(x,c)^2}$ and set $C_1 = \{c_1\}$.

adversarially

for $i \leftarrow 1,2,\dots,k-1$ **do**

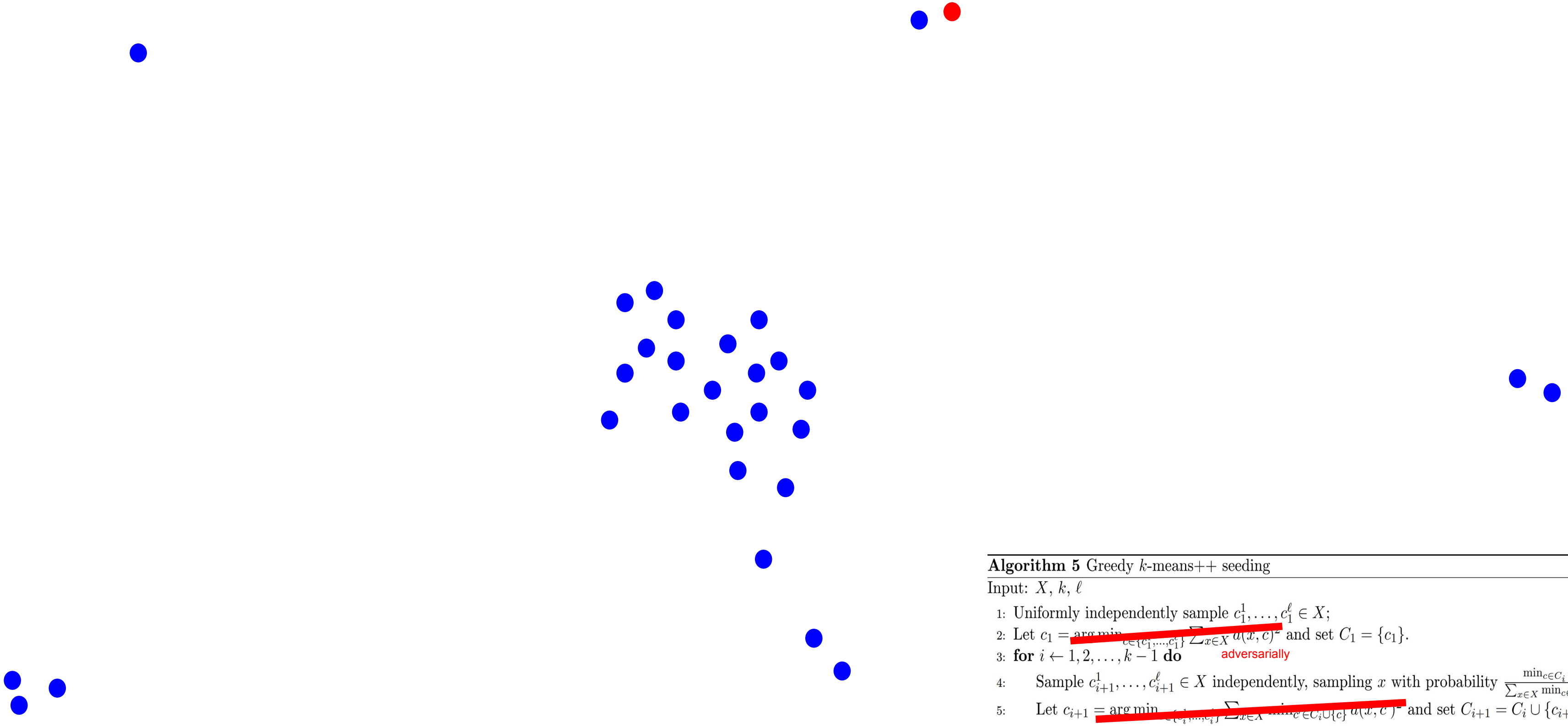
  Sample $c_{i+1}^1,\dots,c_{i+1}^\ell \in X$ independently, sampling x with probability $\frac{\min_{c\in C_i} d(x,c)^2}{\sum_{x\in X}\min_{c\in C_i} d(x,c)^2}$.

  Let $c_{i+1} = \cancel{\arg\min_{c\in\{c_{i+1}^1,\dots,c_{i+1}^\ell\}} \sum_{x\in X}\min_{c'\in C_i\cup\{c\}} d(x,c')^2}$ and set $C_{i+1} = C_i \cup \{c_{i+1}\}$.

adversarially

return $C := C_k$

$\Omega(k^{1-1/\ell})$ approximation for adversarial algorithm



Algorithm 5 Greedy k-means++ seeding

Input: X, k, ℓ

1: Uniformly independently sample $c_1^1, \dots, c_1^\ell \in X$;

2: Let $c_1 = \cancel{\arg\min_{c \in \{c_1^1, \dots, c_1^\ell\}} \sum_{x \in X} d(x,c)^2}$ adversarially and set $C_1 = \{c_1\}$.

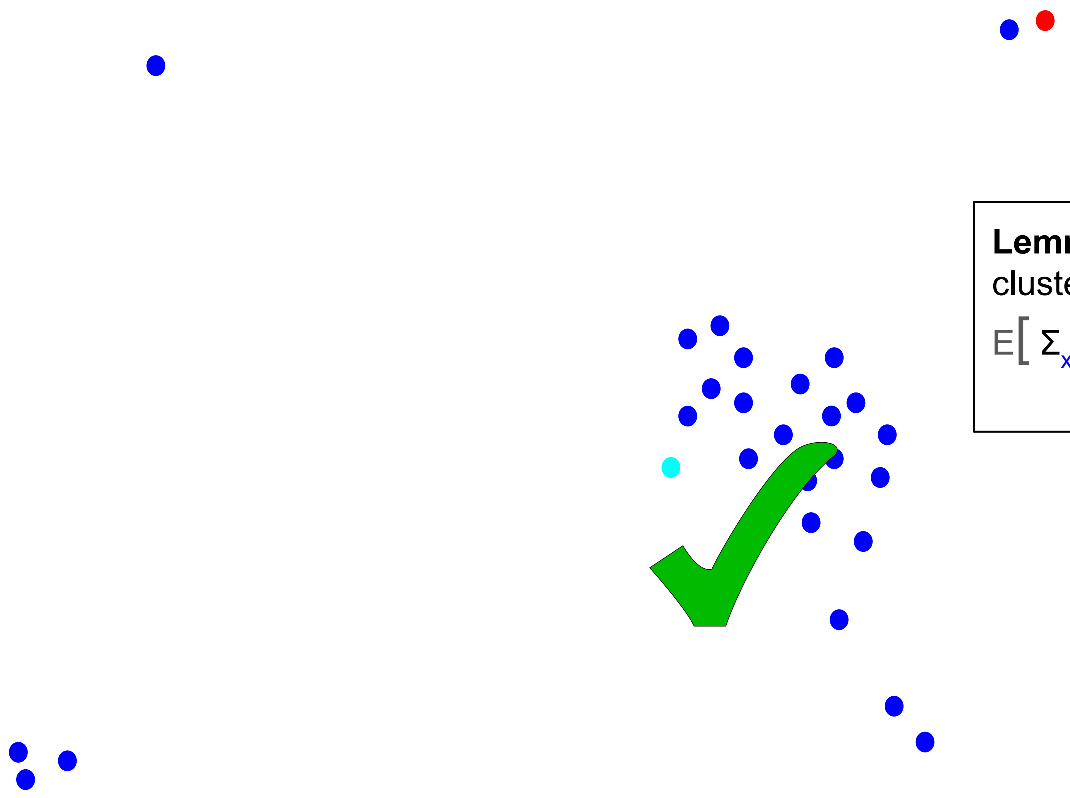
3: **for** $i \leftarrow 1, 2, \dots, k-1$ **do**

4: Sample $c_{i+1}^1, \dots, c_{i+1}^\ell \in X$ independently, sampling x with probability $\frac{\min_{c \in C_i} d(x,c)^2}{\sum_{x \in X} \min_{c \in C_i} d(x,c)^2}$.

5: Let $c_{i+1} = \cancel{\arg\min_{c \in \{c_{i+1}^1, \dots, c_{i+1}^\ell\}} \sum_{x \in X} \min_{c' \in C_i \cup \{c\}} d(x,c')^2}$ adversarially and set $C_{i+1} = C_i \cup \{c_{i+1}\}$.

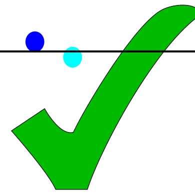
6: **return** $C := C_k$

$\Omega(k^{1-1/\ell})$ approximation for adversarial algorithm



Lemma: Condition on sampling c from some optimal cluster K. Then,

$$E\left[\sum_{x\in K}\min_{c'\in(C\cup c)} d(x,c')^2\right] \leq 8 \times \sum_{x\in K} d(x,\mu(K))^2$$



Algorithm 5 Greedy k-means++ seeding

Input: X, k, ℓ

Uniformly independently sample $c_1^1, \ldots, c_1^\ell \in X$;

Let $c_1 = \cancel{\arg\min_{c\in\{c_1^1,\ldots,c_1^\ell\}} \sum_{x\in X} d(x,c)^2}$ and set $C_1 = \{c_1\}$.

adversarially

for $i \leftarrow 1, 2, \ldots, k-1$ **do**

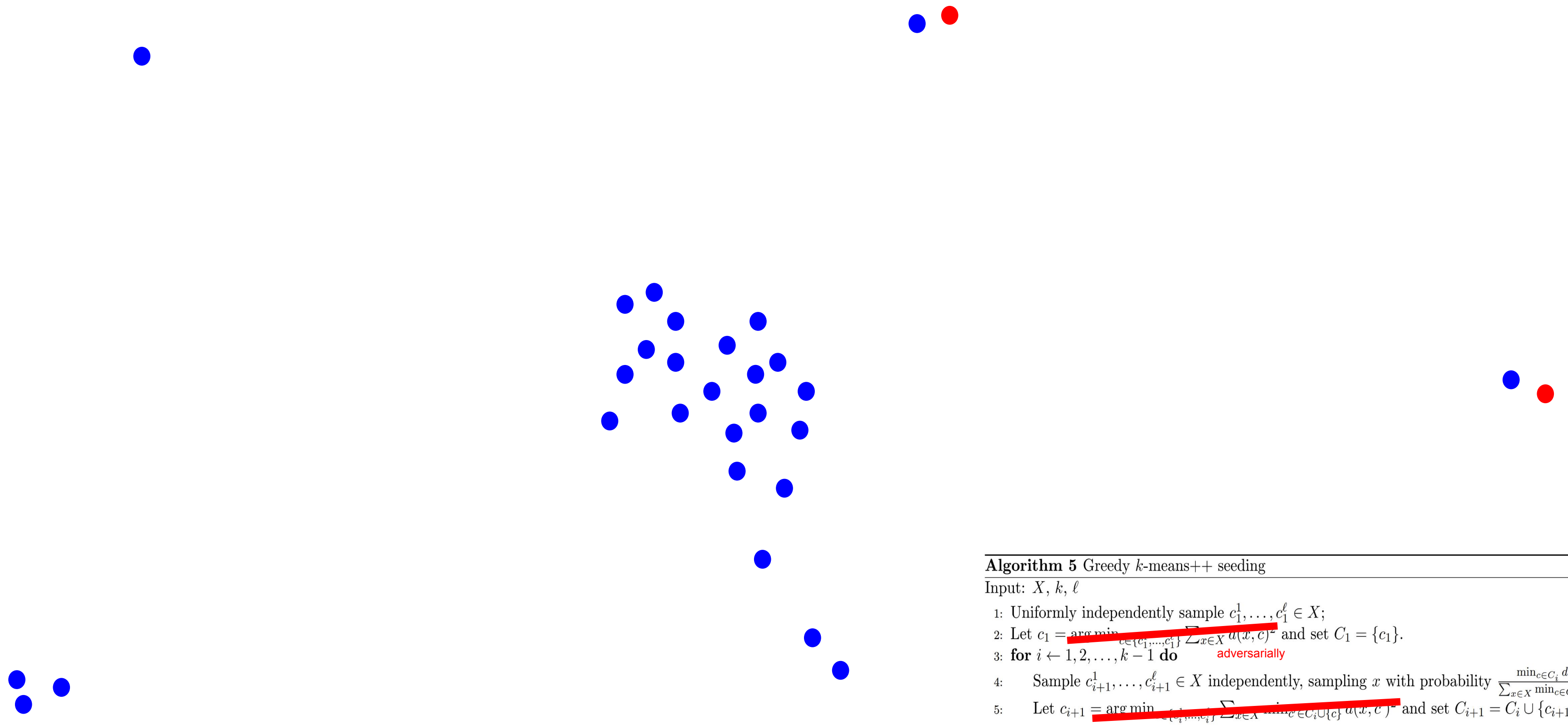
Sample $c_{i+1}^1, \ldots, c_{i+1}^\ell \in X$ independently, sampling x with probability $\frac{\min_{c\in C_i} d(x,c)^2}{\sum_{x\in X}\min_{c\in C_i} d(x,c)^2}$.

Let $c_{i+1} = \cancel{\arg\min_{c\in\{c_{i+1}^1,\ldots,c_{i+1}^\ell\}} \sum_{x\in X} \min_{c'\in C_i\cup\{c\}} d(x,c')^2}$ and set $C_{i+1} = C_i \cup \{c_{i+1}\}$.

adversarially

return $C := C_k$

$\Omega(k^{1-1/\ell})$ approximation for adversarial algorithm



Algorithm 5 Greedy k-means++ seeding

Input: X, k, ℓ

Uniformly independently sample $c_1^1, \dots, c_1^\ell \in X$;

Let $c_1 = \cancel{\arg\min_{c \in \{c_1^1, \dots, c_1^\ell\}} \sum_{x \in X} d(x,c)^2}$ adversarially and set $C_1 = \{c_1\}$.

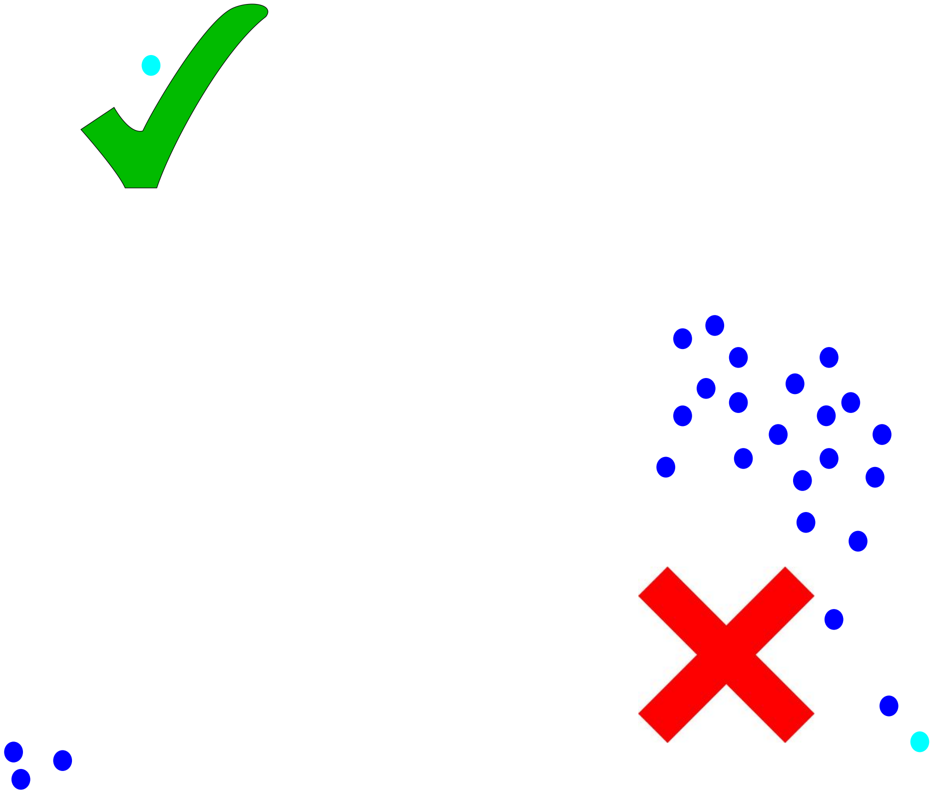
for $i \leftarrow 1, 2, \dots, k-1$ **do**

  Sample $c_{i+1}^1, \dots, c_{i+1}^\ell \in X$ independently, sampling x with probability $\frac{\min_{c \in C_i} d(x,c)^2}{\sum_{x \in X} \min_{c \in C_i} d(x,c)^2}$.

  Let $c_{i+1} = \cancel{\arg\min_{c \in \{c_i^1, \dots, c_i^\ell\}} \sum_{x \in X} \min_{c' \in C_i \cup \{c\}} d(x,c')^2}$ adversarially and set $C_{i+1} = C_i \cup \{c_{i+1}\}$.

return $C := C_k$

$\Omega(k^{1-1/\ell})$ approximation for adversarial algorithm



Lemma: Condition on sampling c from some optimal cluster K. Then,

$$E\left[\, \Sigma_{x \in K} \min_{c' \in (C \cup c)} d(x,c')^2 \,\right] \leq 8 \times \Sigma_{x \in K} d(x,\mu(K))^2$$

Algorithm 5 Greedy k-means++ seeding

Input: X, k, ℓ

Uniformly independently sample $c_1^1, \ldots, c_1^\ell \in X$;

Let $c_1 = \cancel{\arg\min_{c \in \{c_1^1, \ldots, c_1^\ell\}} \sum_{x \in X} d(x,c)^2}$ and set $C_1 = \{c_1\}$.

adversarially

for $i \leftarrow 1, 2, \ldots, k-1$ **do**

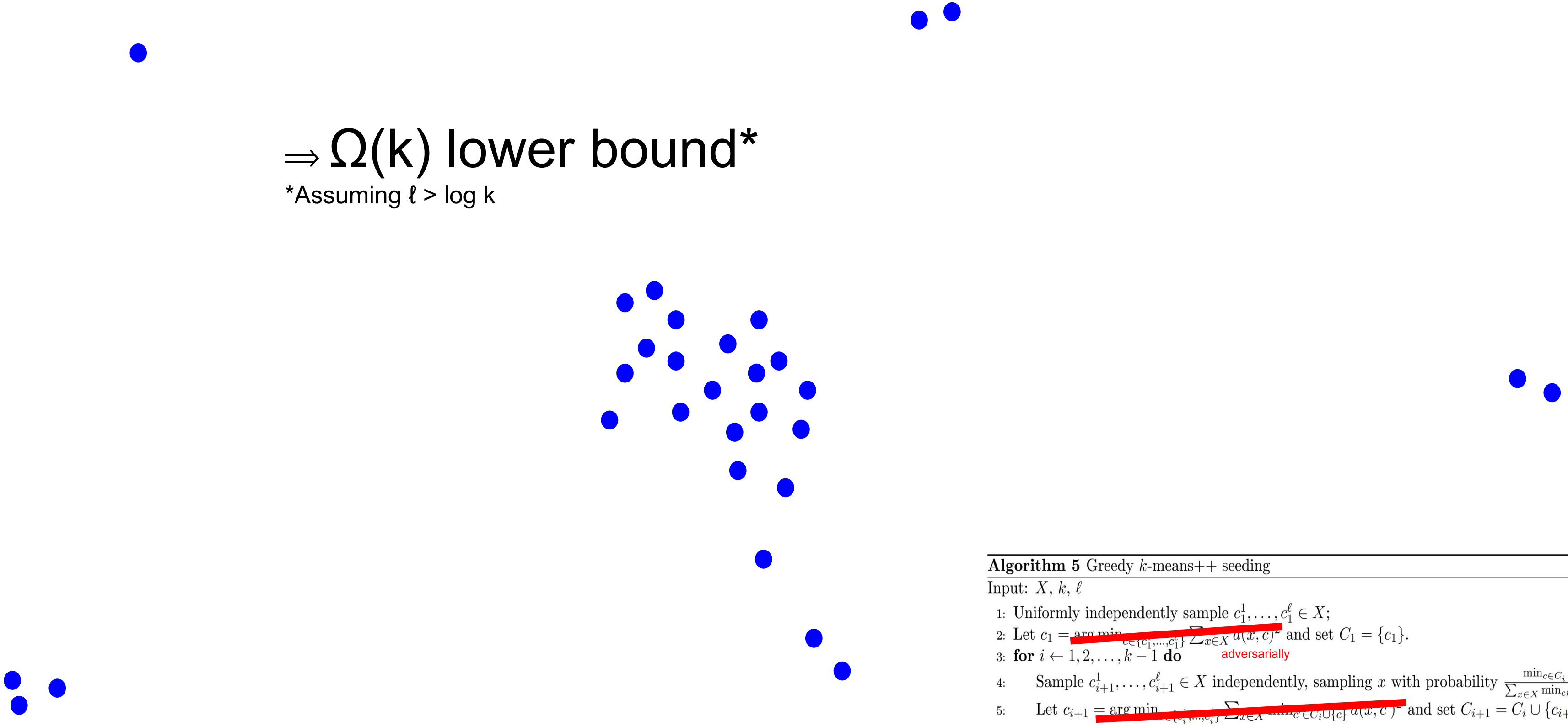
  Sample $c_{i+1}^1, \ldots, c_{i+1}^\ell \in X$ independently, sampling x with probability $\frac{\min_{c \in C_i} d(x,c)^2}{\sum_{x \in X} \min_{c \in C_i} d(x,c)^2}$.

  Let $c_{i+1} = \cancel{\arg\min_{c \in \{c_i^1, \ldots, c_i^\ell\}} \sum_{x \in X} \min_{c' \in C_i \cup \{c\}} d(x,c')^2}$ and set $C_{i+1} = C_i \cup \{c_{i+1}\}$.

adversarially

return $C := C_k$

$\Omega(k^{1-1/\ell})$ approximation for adversarial algorithm



$\Rightarrow \Omega(k)$ lower bound*

*Assuming $\ell > \log k$

Algorithm 5 Greedy k-means++ seeding

Input: X, k, ℓ

Uniformly independently sample $c_1^1, \dots, c_1^\ell \in X$;

Let $c_1 = \cancel{\arg\min_{c \in \{c_1^1, \dots, c_1^\ell\}} \sum_{x \in X} d(x,c)^2}$ adversarially and set $C_1 = \{c_1\}$.

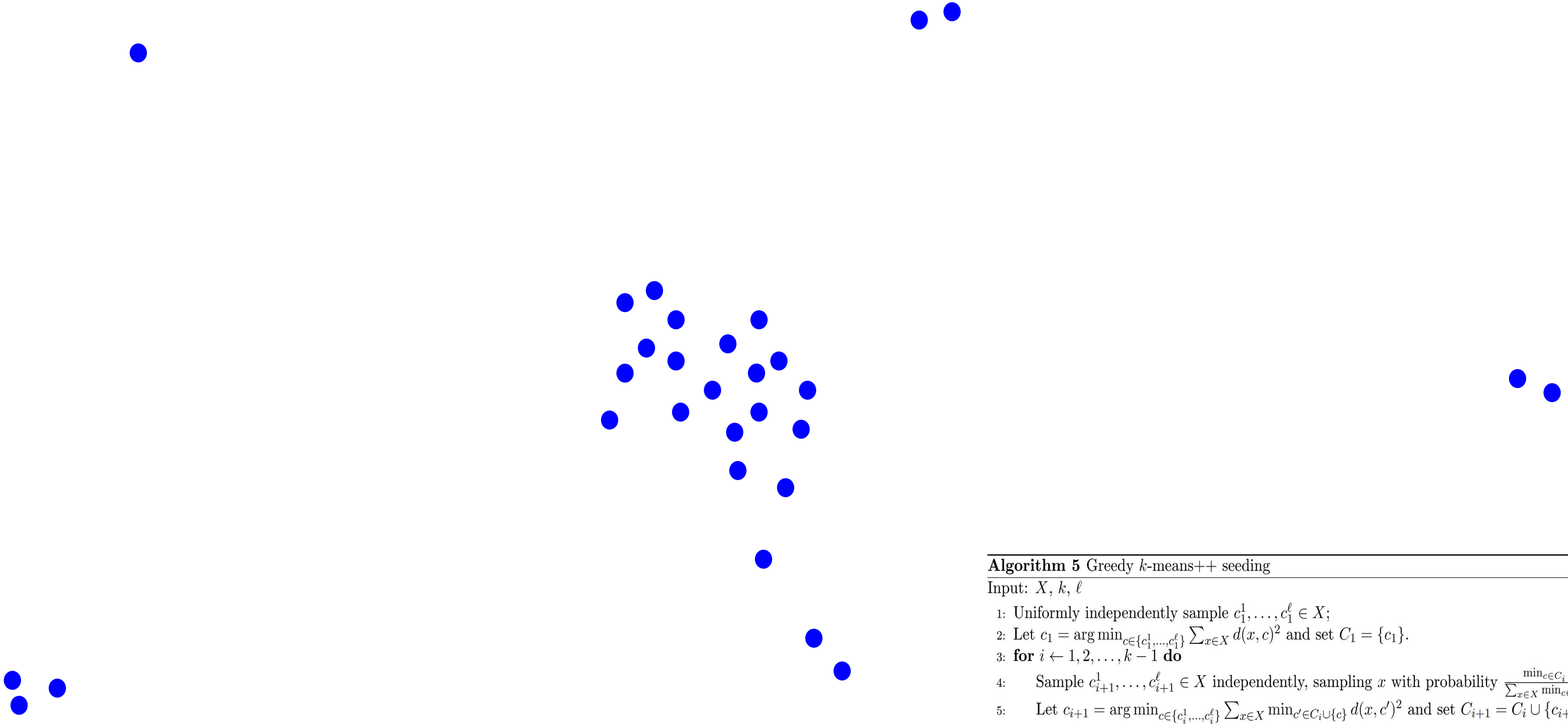
for $i \leftarrow 1, 2, \dots, k-1$ **do**

  Sample $c_{i+1}^1, \dots, c_{i+1}^\ell \in X$ independently, sampling x with probability $\frac{\min_{c \in C_i} d(x,c)^2}{\sum_{x \in X} \min_{c \in C_i} d(x,c)^2}$.

  Let $c_{i+1} = \cancel{\arg\min_{c \in \{c_i^1, \dots, c_i^\ell\}} \sum_{x \in X} \min_{c' \in C_i \cup \{c\}} d(x,c')^2}$ adversarially and set $C_{i+1} = C_i \cup \{c_{i+1}\}$.

return $C := C_k$

But what about greedy?



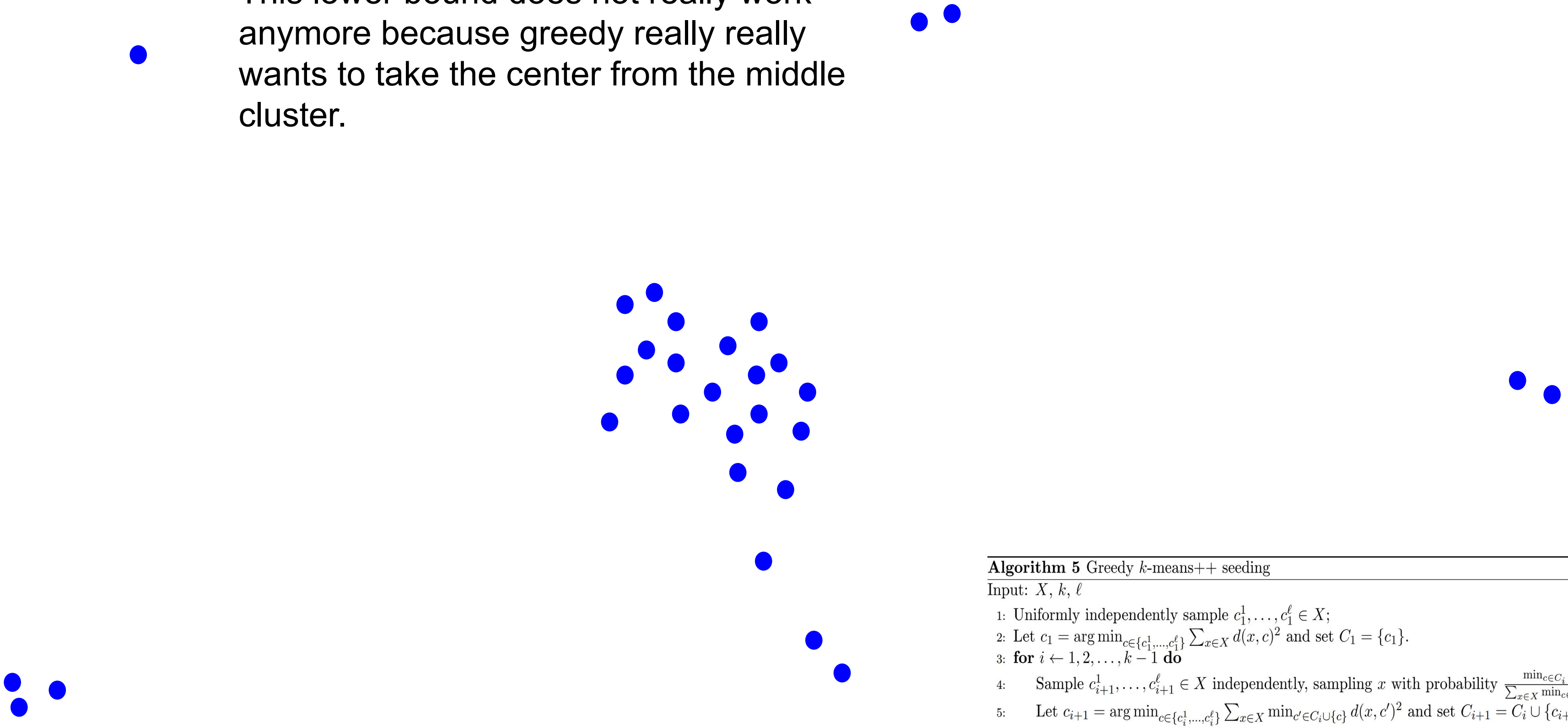
Algorithm 5 Greedy k-means++ seeding

Input: X, k, ℓ

Uniformly independently sample $c_1^1, \ldots, c_1^\ell \in X$;
Let $c_1 = \arg\min_{c \in \{c_1^1, \ldots, c_1^\ell\}} \sum_{x \in X} d(x, c)^2$ and set $C_1 = \{c_1\}$.
for $i \leftarrow 1, 2, \ldots, k-1$ **do**
    Sample $c_{i+1}^1, \ldots, c_{i+1}^\ell \in X$ independently, sampling x with probability $\frac{\min_{c \in C_i} d(x,c)^2}{\sum_{x \in X} \min_{c \in C_i} d(x,c)^2}$.
    Let $c_{i+1} = \arg\min_{c \in \{c_i^1, \ldots, c_i^\ell\}} \sum_{x \in X} \min_{c' \in C_i \cup \{c\}} d(x, c')^2$ and set $C_{i+1} = C_i \cup \{c_{i+1}\}$.
return $C := C_k$

But what about greedy?

This lower bound does not really work anymore because greedy really really wants to take the center from the middle cluster.



Algorithm 5 Greedy k-means++ seeding

Input: X, k, ℓ

Uniformly independently sample $c_1^1, \ldots, c_1^\ell \in X$;

Let $c_1 = \arg\min_{c \in \{c_1^1, \ldots, c_1^\ell\}} \sum_{x \in X} d(x,c)^2$ and set $C_1 = \{c_1\}$.

for $i \leftarrow 1, 2, \ldots, k-1$ **do**

  Sample $c_{i+1}^1, \ldots, c_{i+1}^\ell \in X$ independently, sampling x with probability $\frac{\min_{c \in C_i} d(x,c)^2}{\sum_{x \in X} \min_{c \in C_i} d(x,c)^2}$.

  Let $c_{i+1} = \arg\min_{c \in \{c_i^1, \ldots, c_i^\ell\}} \sum_{x \in X} \min_{c' \in C_i \cup \{c\}} d(x,c')^2$ and set $C_{i+1} = C_i \cup \{c_{i+1}\}$.

return $C := C_k$

But what about greedy?

Main technical lemma for greedy k-means++

Lemma: For every cluster in OPT, the expected number of points sampled from this cluster until covered is $O(\ell^2 \log^2 k)$.

But what about greedy?

Lemma: For every cluster in OPT, the expected number of points sampled from this cluster until covered is $O(\ell^2 \log^2 k)$.

Corollary: The approximation ratio of greedy k-means++ is

$$O(\log k)\, O(\ell)\, O(\ell^2 \log^2 k) = O(\ell^3 \log^3 k).$$

But what about greedy?

Lemma: For every cluster in OPT, the expected number of points sampled from this cluster until covered is $O(\ell^2 \log^2 k)$.

Corollary: The approximation ratio of greedy k-means++ is

$$O(\log k)\, O(\ell)\, O(\ell^2 \log^2 k) = O(\ell^3 \log^3 k).$$

original k-means++
analysis

But what about greedy?

Lemma: For every cluster in OPT, the expected number of points sampled from this cluster until covered is $O(\ell^2 \log^2 k)$.

Corollary: The approximation ratio of greedy k-means++ is

$$O(\log k)\, O(\ell)\, O(\ell^2 \log^2 k) = O(\ell^3 \log^3 k).$$

original k-means++ analysis

sampling from a "covered" cluster is ℓ times more probable

But what about greedy?

Lemma: For every cluster in OPT, the expected number of points sampled from this cluster until covered is $O(\ell^2 \log^2 k)$.

Corollary: The approximation ratio of greedy k-means++ is

$$O(\log k)\, O(\ell)\, O(\ell^2 \log^2 k) = O(\ell^3 \log^3 k).$$

original k-means++ analysis

sampling from a "covered" cluster is ℓ times more probable

Lemma

But what about greedy?

Lemma: For every cluster in OPT, the expected number of points sampled from this cluster until covered is $O(\ell^2 \log^2 k)$.

Corollary: The approximation ratio of greedy k-means++ is

$$O(\log k)\, O(\ell)\, O(\ell^2 \log^2 k) = O(\ell^3 \log^3 k).$$

(Almost) Matching Lower bound: Combining

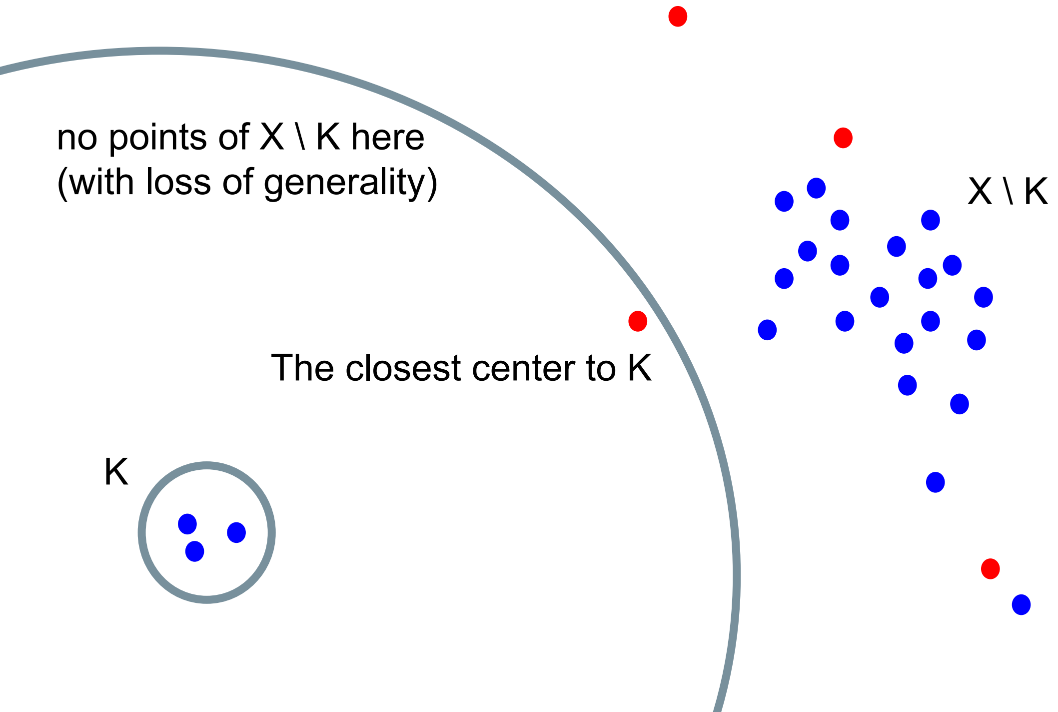
1. the k-means++ lower bound,
2. a version of the $\Omega(k^{1-1/\ell})$ lower bound.

(3)

Where is $O(\ell^2 \log^2 k)$ coming from?

(very fast if at all)

Why there are only $\log^2(k)$ samples from the same cluster? ($\ell = 2$)

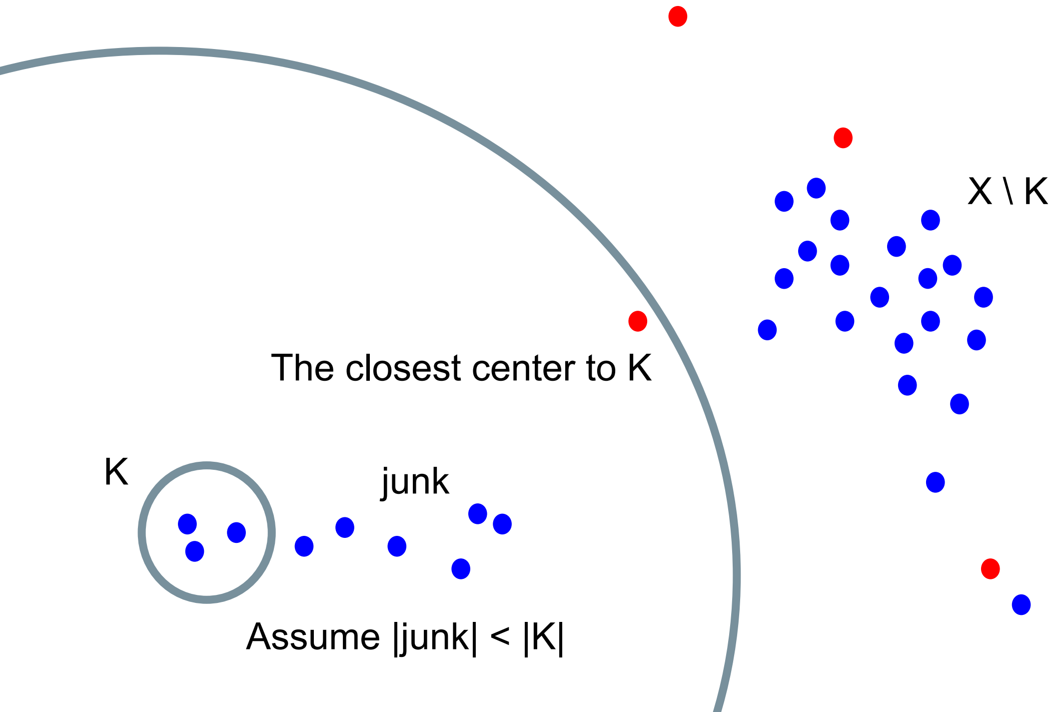


- WLOG, we always have:

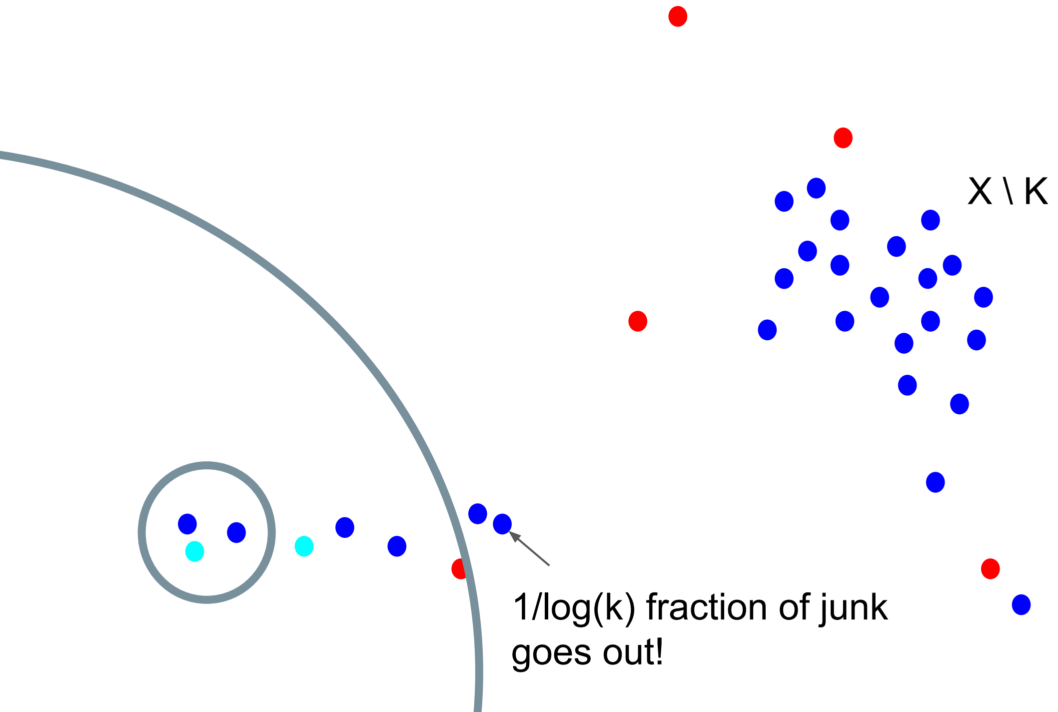
  cost(X \ K)/k ≤ cost(K) ≤ cost(X \ K)*k

- also WLOG, the cost drop by taking points in X \ K is at least cost(K).
- Thus, in expectation we sample 1 point from K during cost(X \ K) dropping by 2 factor
- Hence, we sample only log(k) points from K!

Why there are only $\log^2(k)$ samples from the same cluster? ($\ell = 2$)



Why there are only $\log^2(k)$ samples from the same cluster? ($\ell = 2$)



Summary

- *greedy k-means*++ is still “well-behaved”.
- But I view it as a small miracle – for such a simple algorithm, its analysis is surprisingly subtle.
- A theoretical justification for the greedy rule?

Algorithm 5 Greedy k-means++ seeding

Input: X, k, ℓ

Uniformly independently sample $c_1^1, \ldots, c_1^\ell \in X$;

Let $c_1 = \arg\min_{c \in \{c_1^1, \ldots, c_1^\ell\}} \sum_{x \in X} d(x,c)^2$ and set $C_1 = \{c_1\}$.

for $i \leftarrow 1, 2, \ldots, k-1$ **do**

  Sample $c_{i+1}^1, \ldots, c_{i+1}^\ell \in X$ independently, sampling x with probability $\frac{\min_{c \in C_i} d(x,c)^2}{\sum_{x \in X} \min_{c \in C_i} d(x,c)^2}$.

  Let $c_{i+1} = \arg\min_{c \in \{c_i^1, \ldots, c_i^\ell\}} \sum_{x \in X} \min_{c' \in C_i \cup \{c\}} d(x,c')^2$ and set $C_{i+1} = C_i \cup \{c_{i+1}\}$.

return $C := C_k$