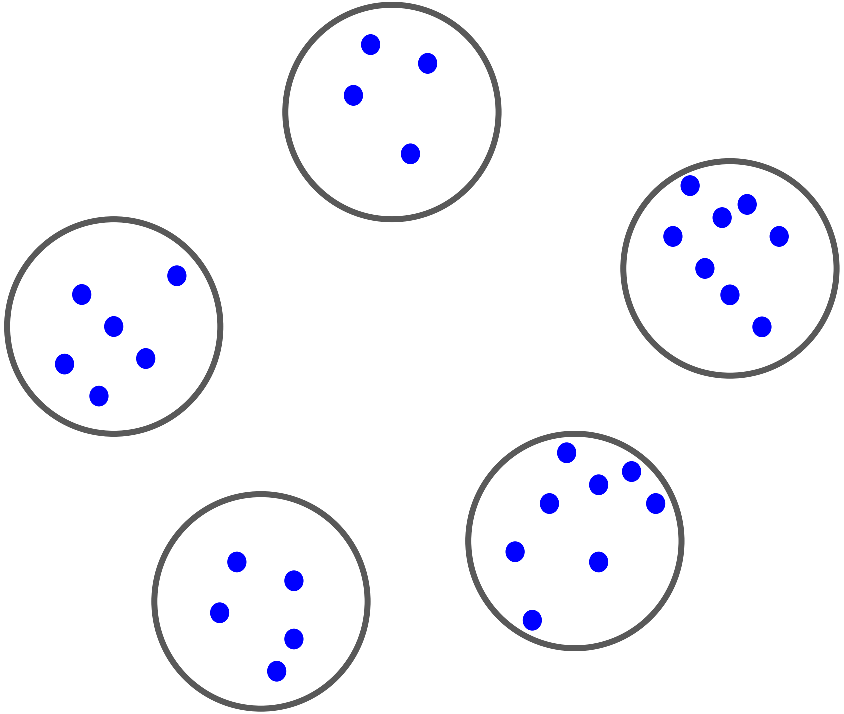
Improved analysis of an algorithm of Lattanzi and Sohler

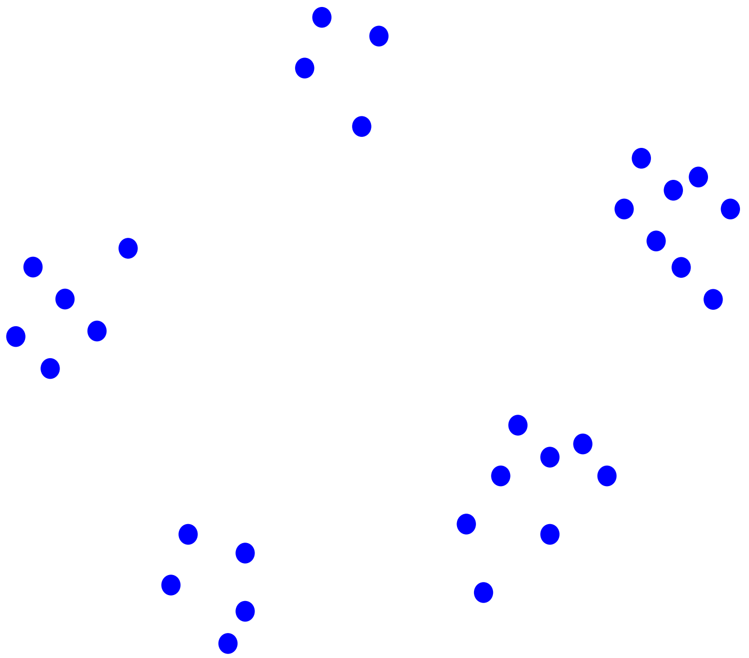
Davin Choo, Christoph Grunau, Julian Portmann, Václav Rozhoň

k-means clustering



clustering: I have bunch of points, say in R^d, and want to cluster them so that close points are together.

k-means clustering



find k "centers" so as to minimize

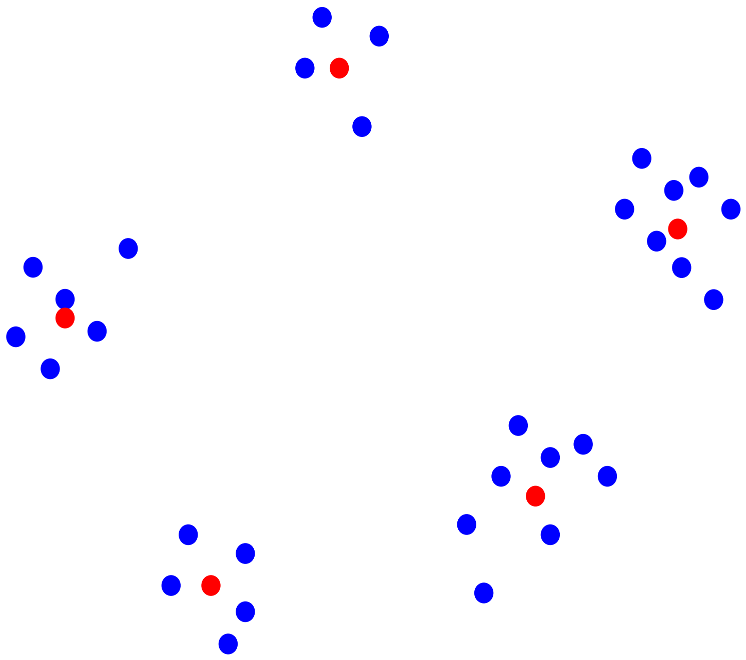
$$\sum_{p \in P} \min_{c \in C} d(p,c)^2$$

sum over input points

closest center

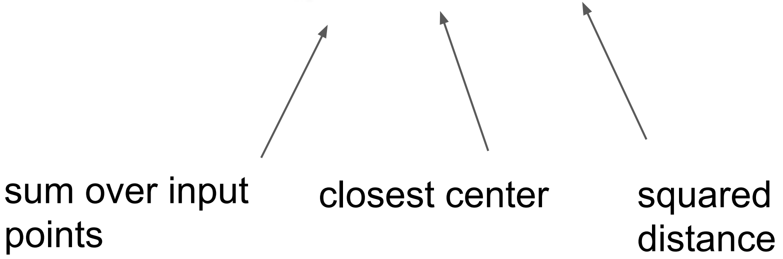
squared distance

k-means clustering

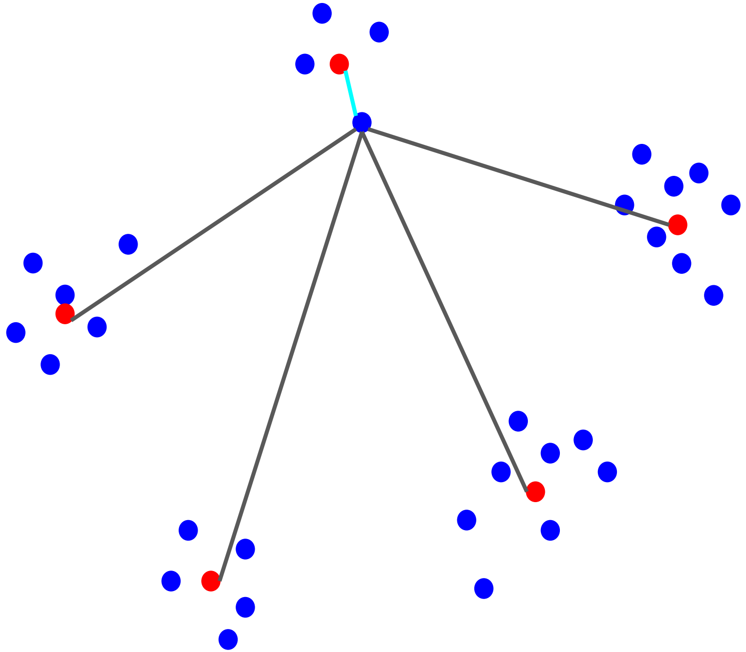


find k "centers" so as to minimize

$$\sum_{p \in P} \min_{c \in C} d(p,c)^2$$



k-means clustering

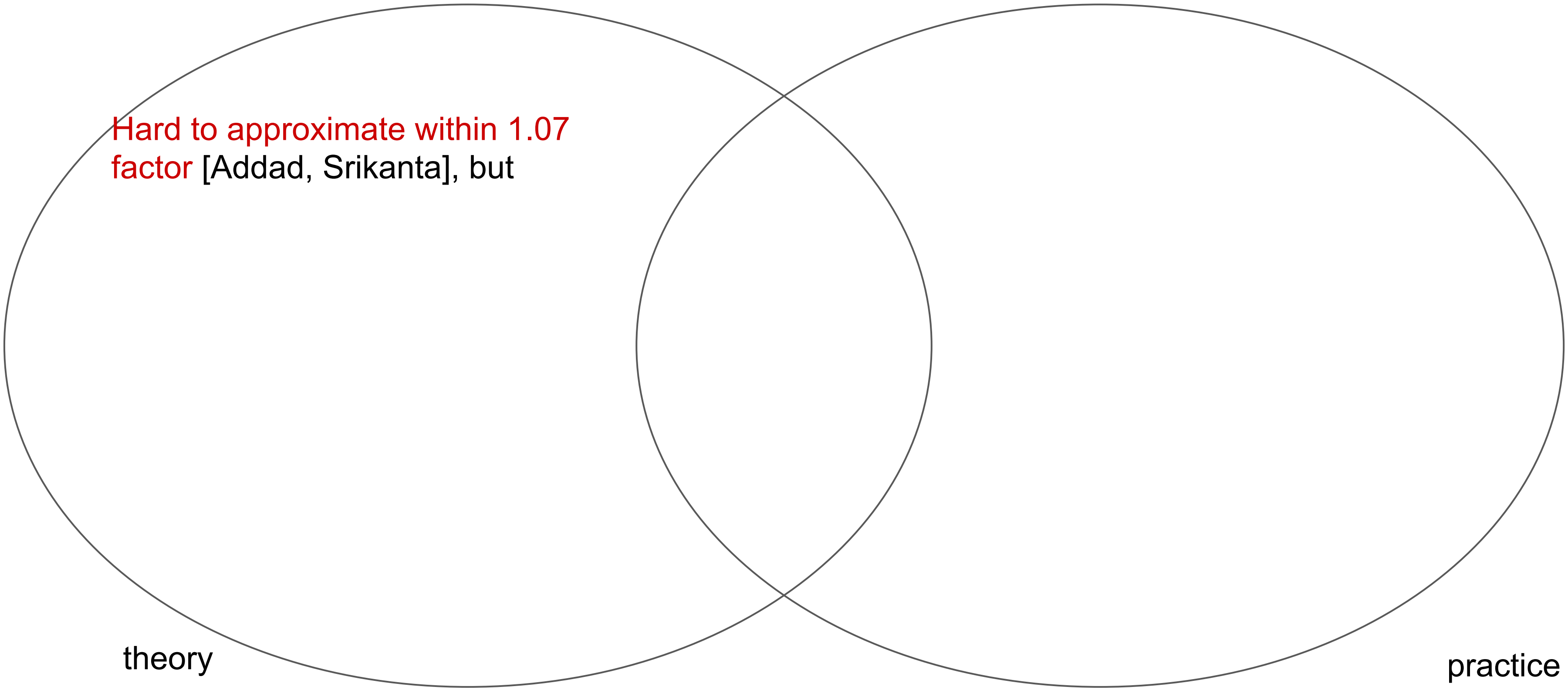


find k "centers" so as to minimize

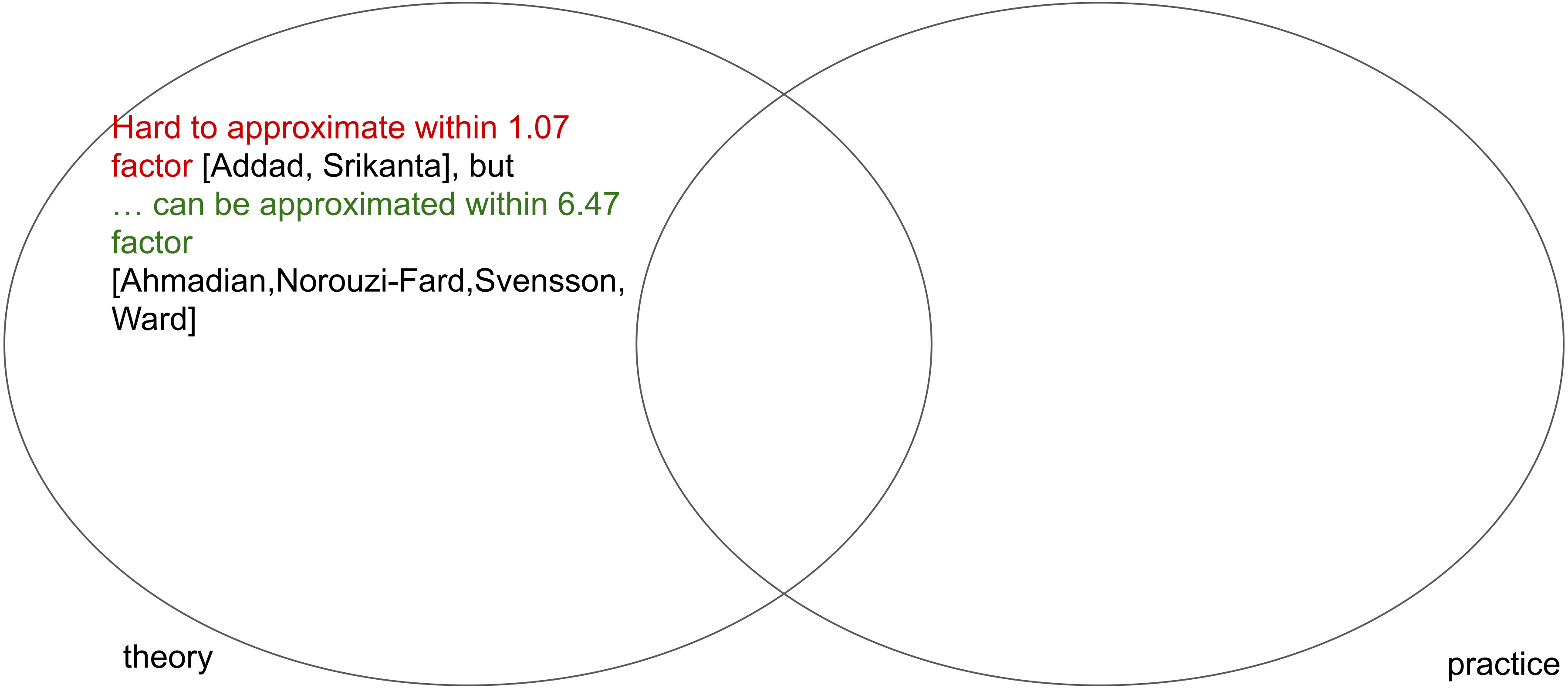
$$\sum_{p \in P} \min_{c \in C} d(p,c)^2$$

sum over input points — closest center — squared distance

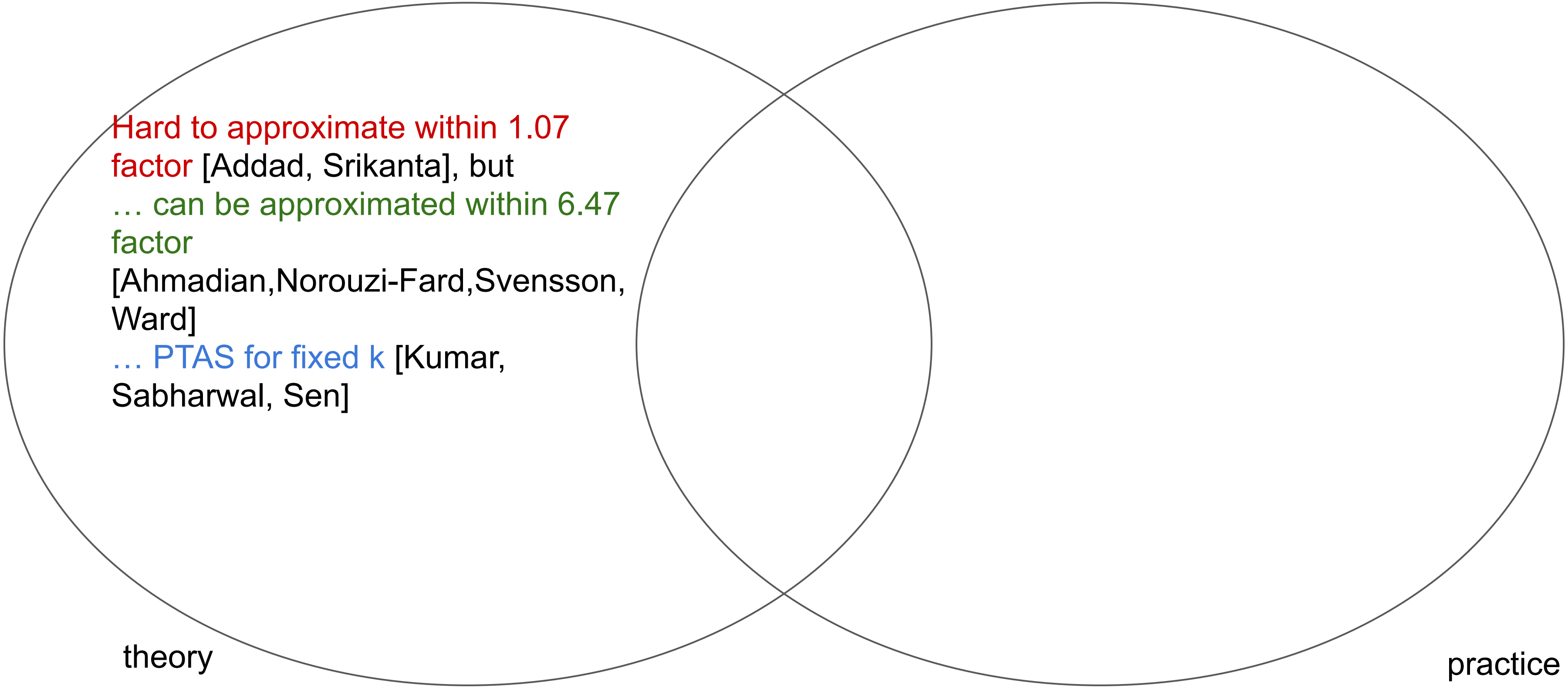
k-means: theory and practice



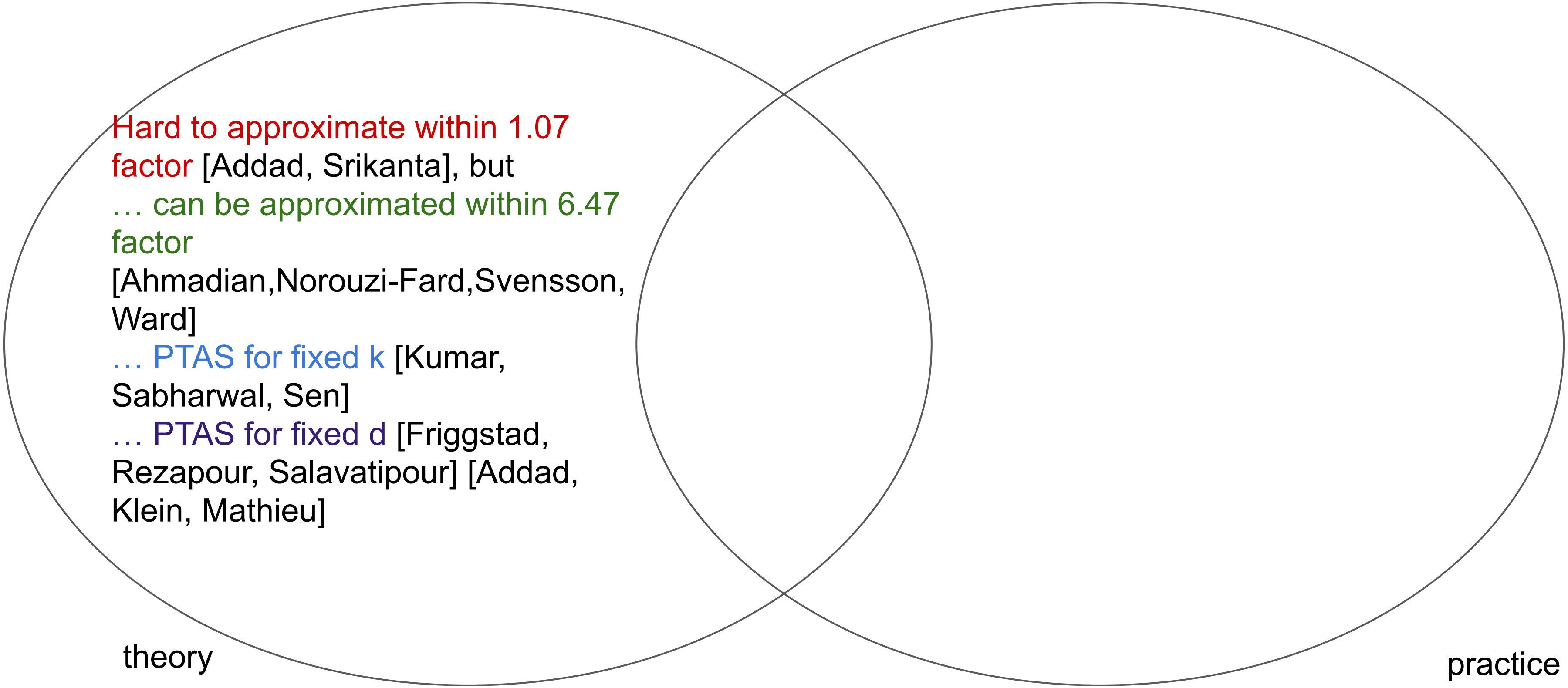
k-means: theory and practice



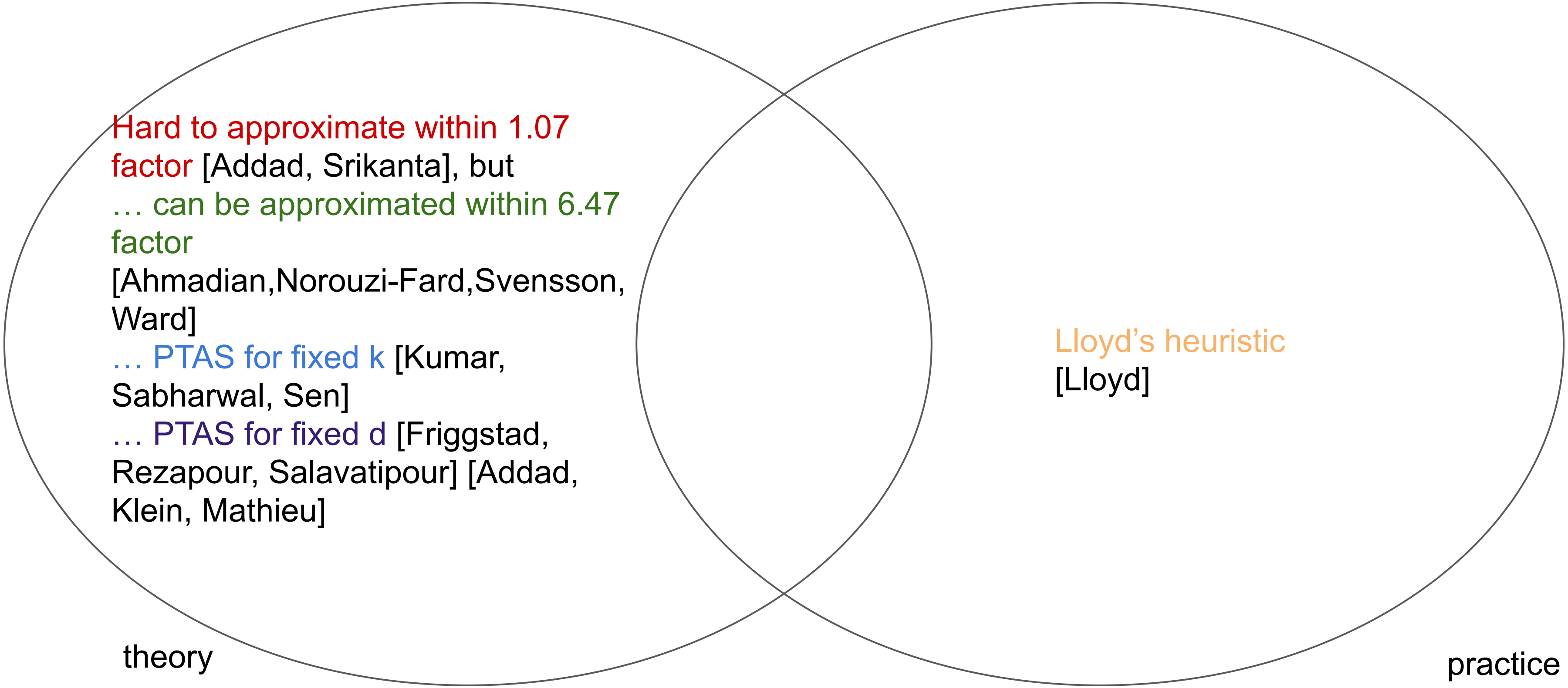
k-means: theory and practice

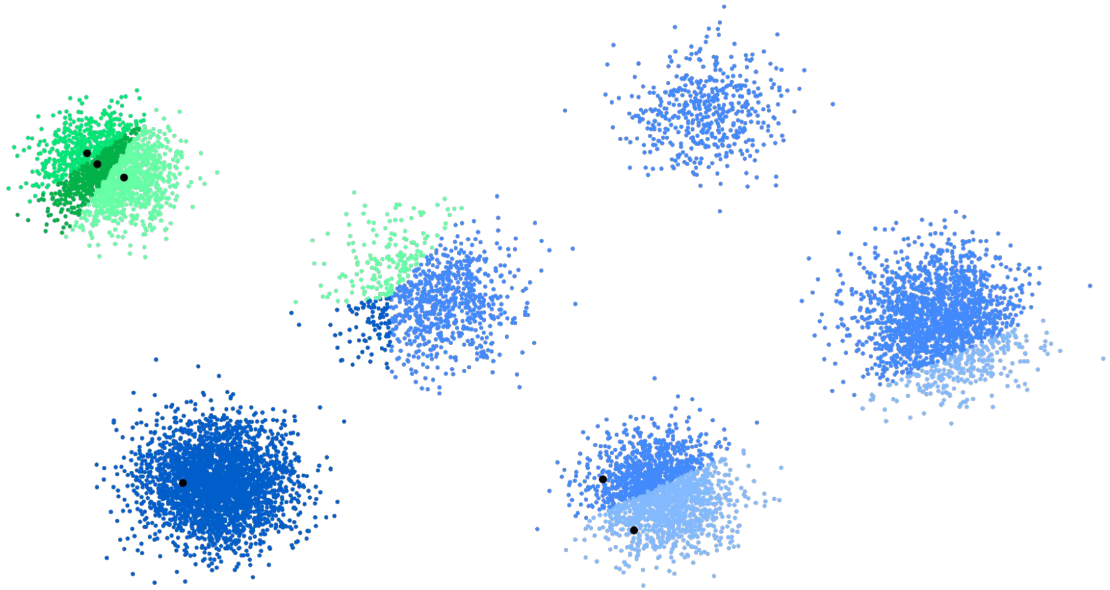


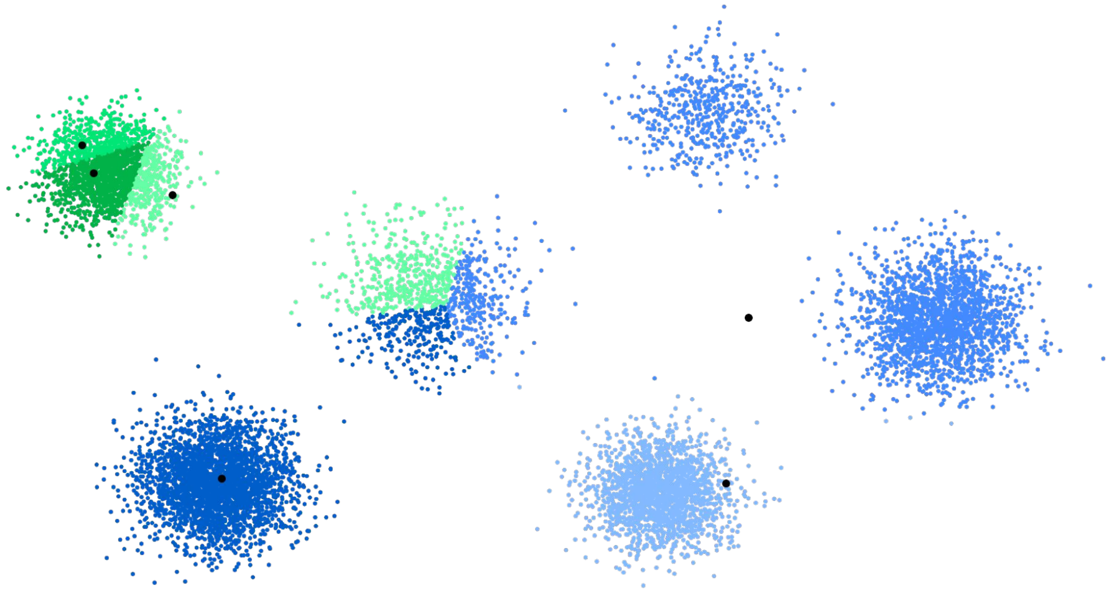
k-means: theory and practice

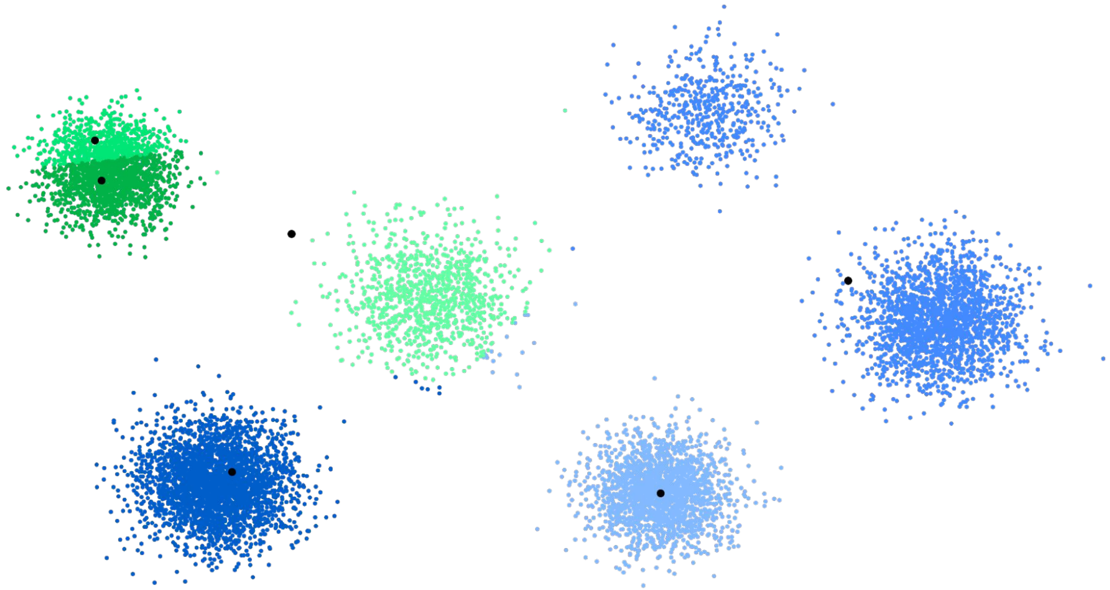


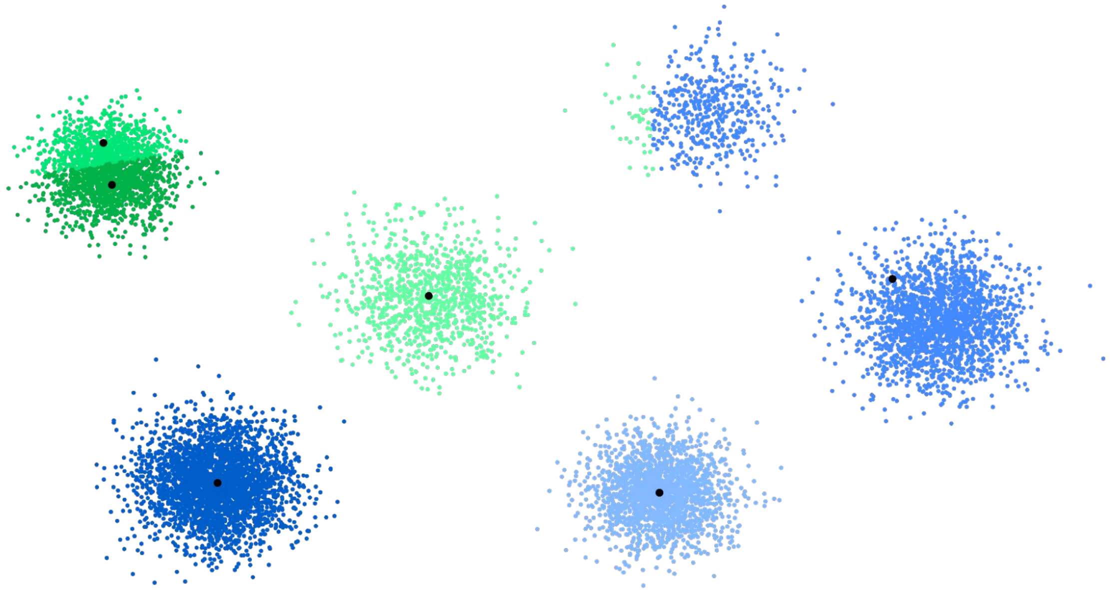
k-means: theory and practice



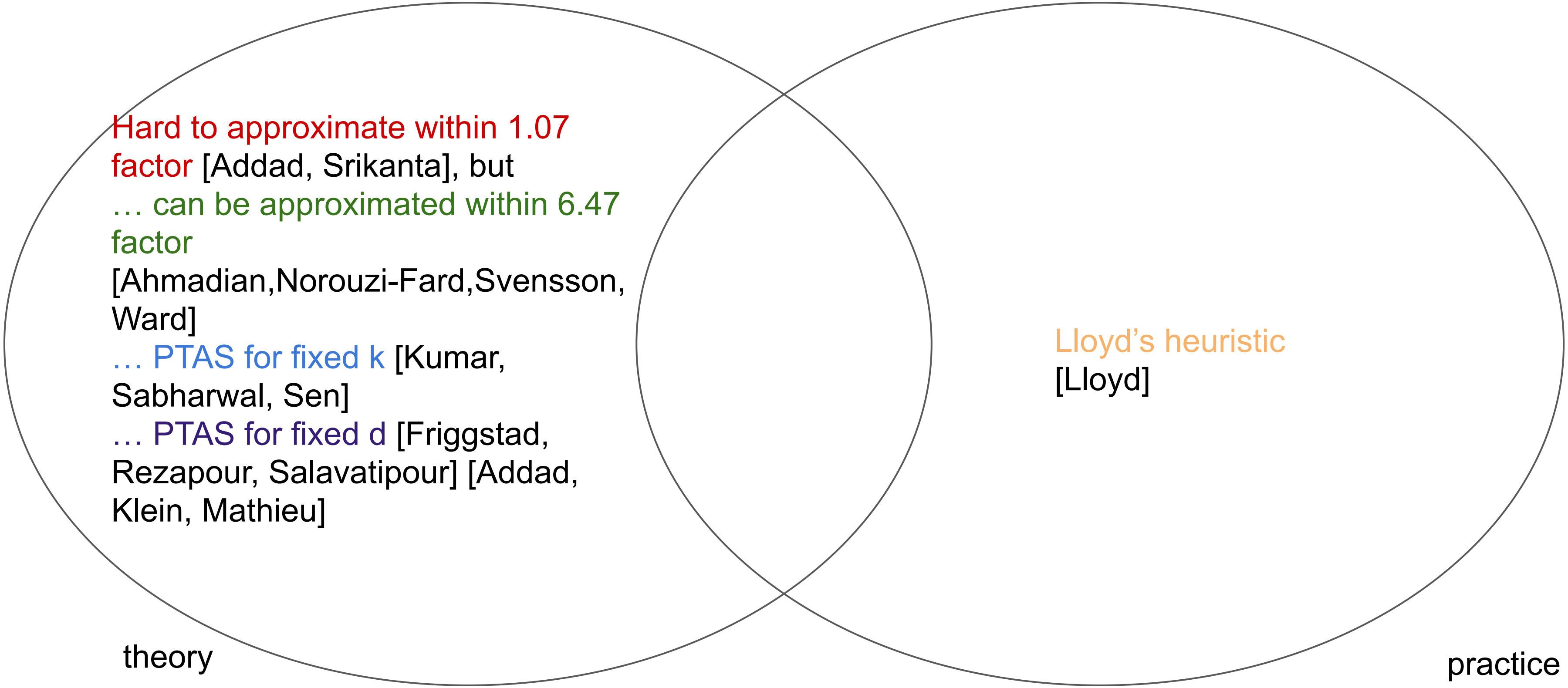




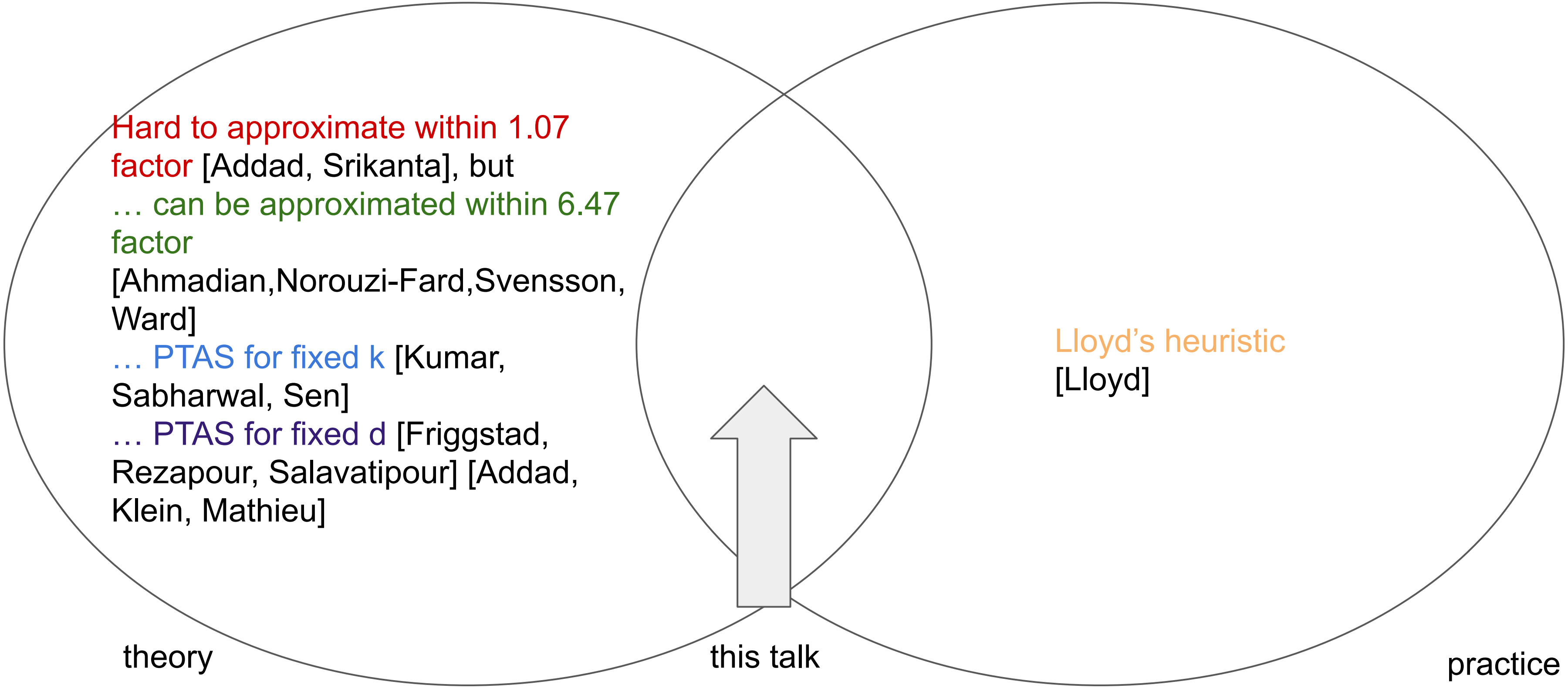




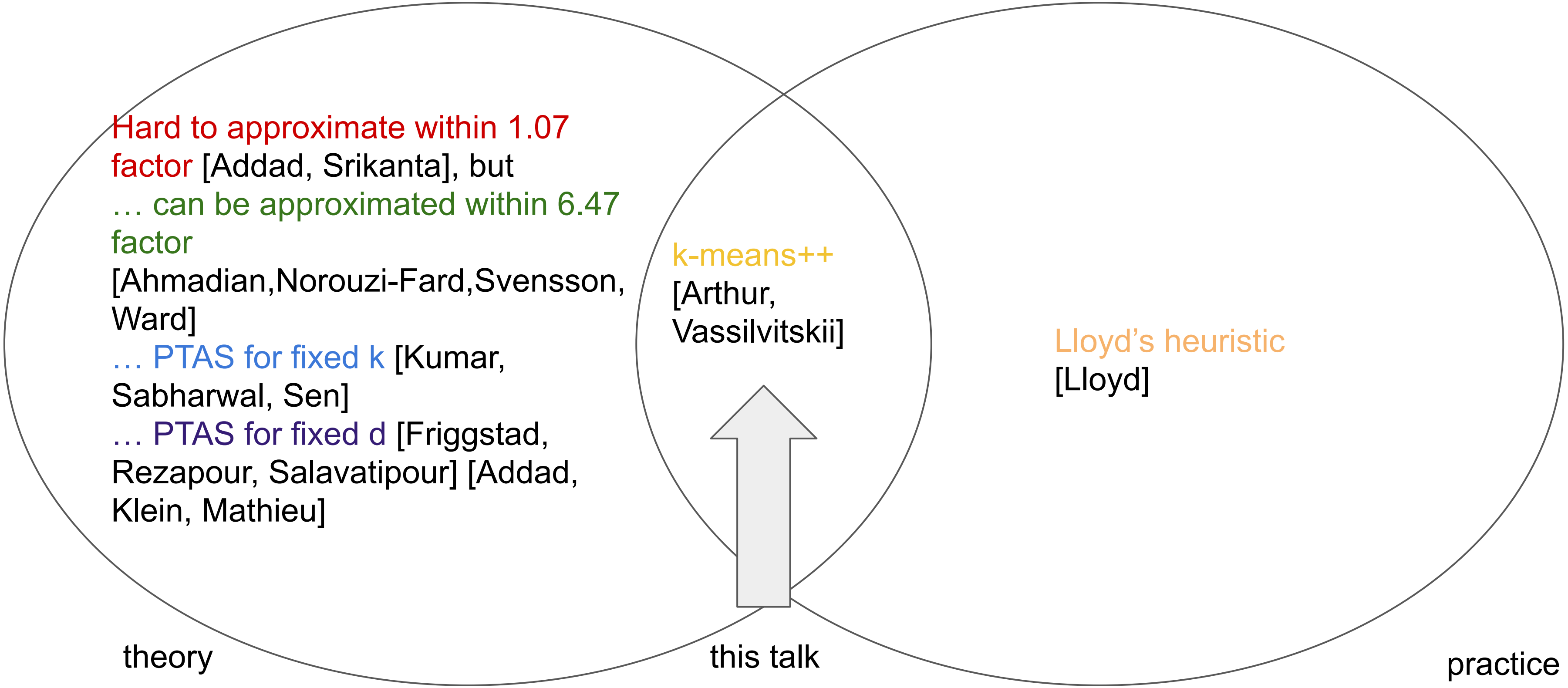
k-means: theory and practice



k-means: theory and practice



k-means: theory and practice



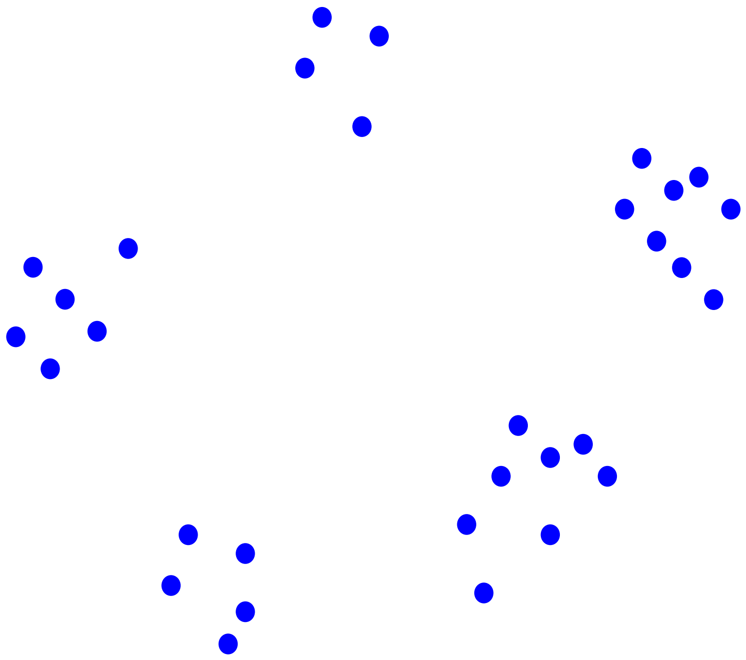
Outline

- Explain k-means++
- Explain its improved variant by Lattanzi and Sohler
- Tighter analysis of Lattanzi-Sohler's algorithm
- Extension of their algorithm to a similar problem (if time allows)

k-means++

Practice: fast seeding for Lloyd’s, better than random seeding

Theory: expected O(log k) approximation guarantee

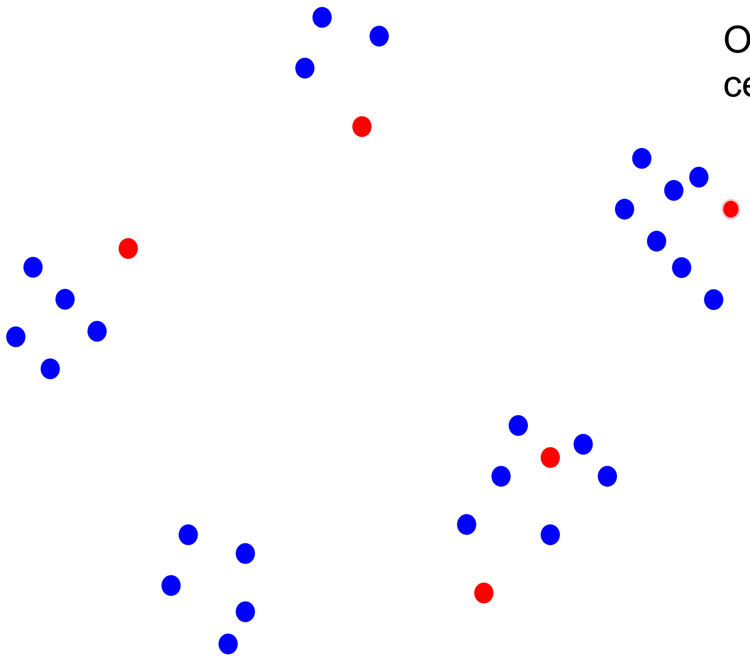


k-means++

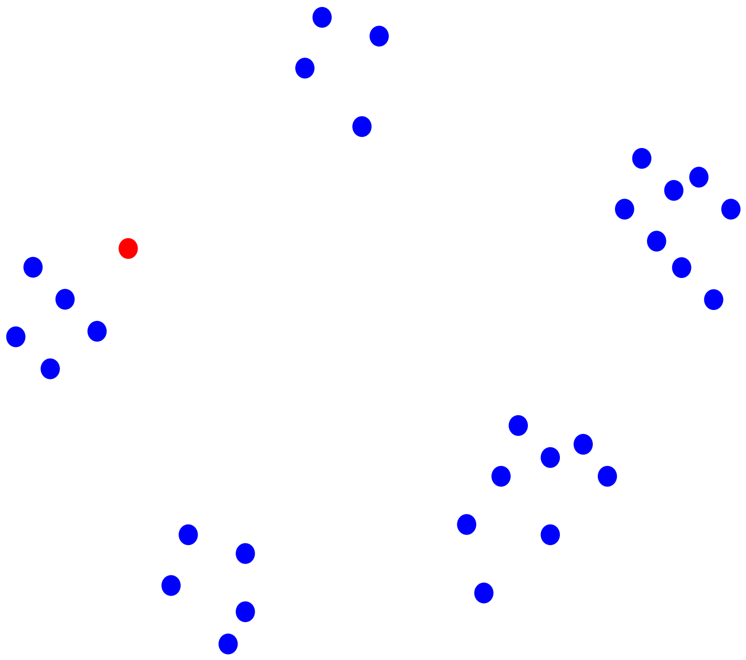
Practice: fast seeding for Lloyd's, better than random

Theory: expected O(log k) approximation guarantee

Outputs a set of centers that are subset of the input points (the centers then define clusters)

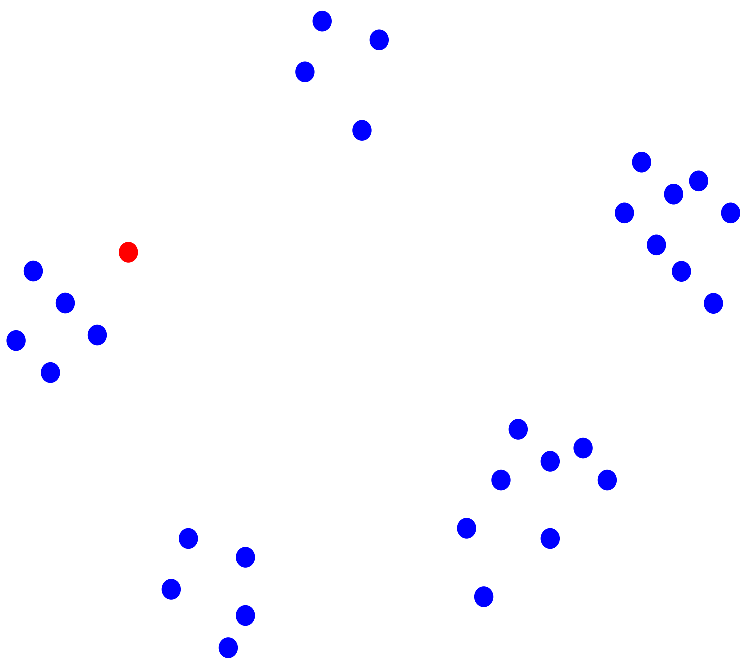


k-means++



First center: uniformly at random

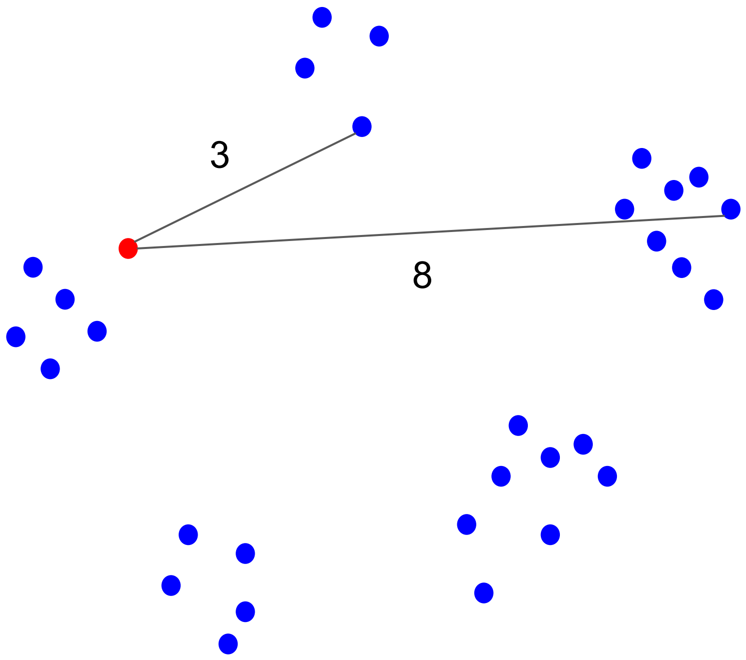
k-means++



First center: uniformly at random

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

k-means++

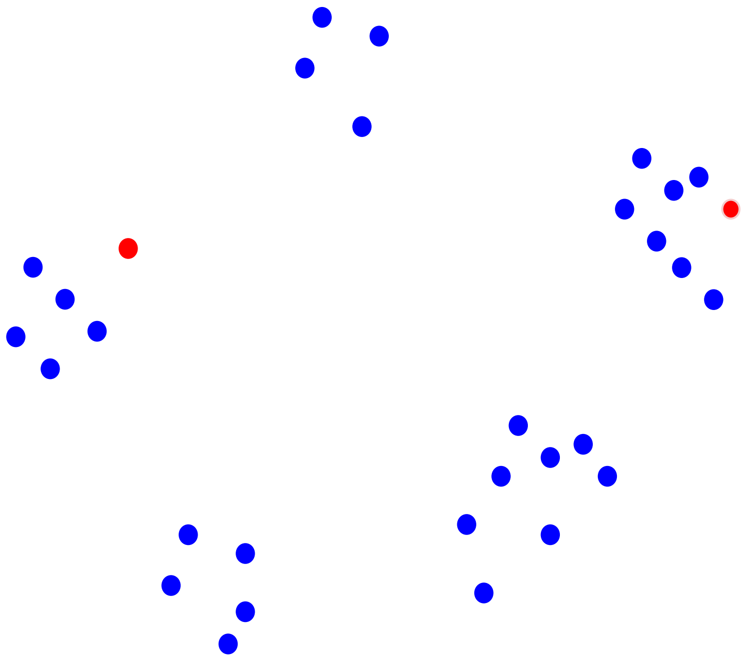


First center: uniformly at random

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

$$\sum_{p \in P} \min_{c \in C} d(p, c)^2$$

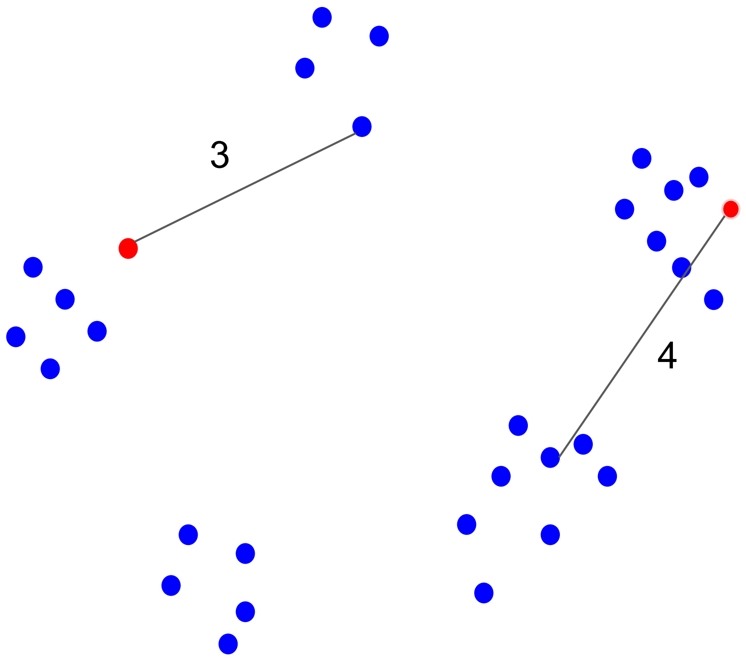
k-means++



First center: uniformly at random

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

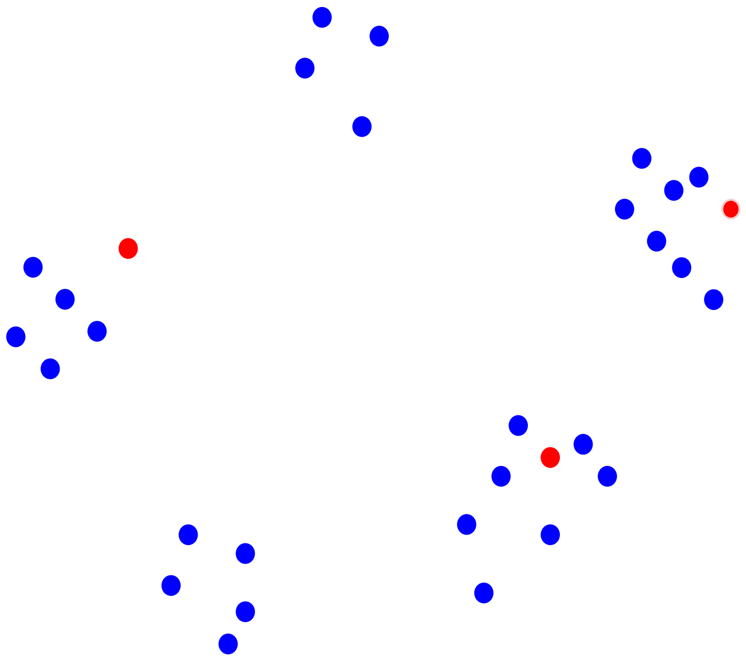
k-means++



First center: uniformly at random

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

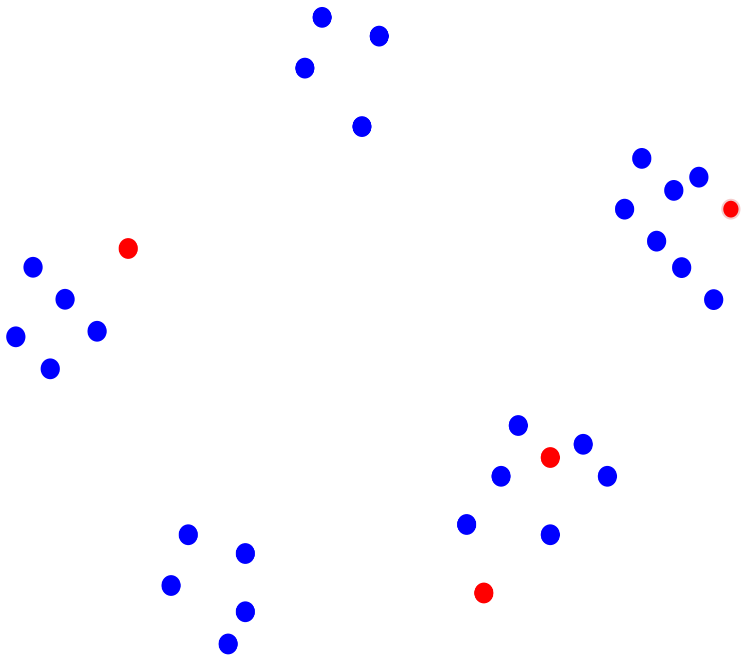
k-means++



First center: uniformly at random

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

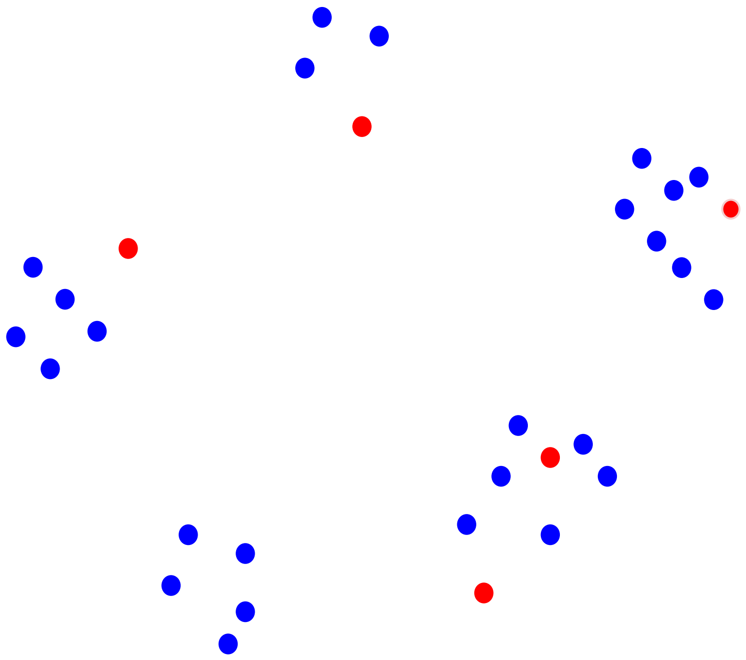
k-means++



First center: uniformly at random

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

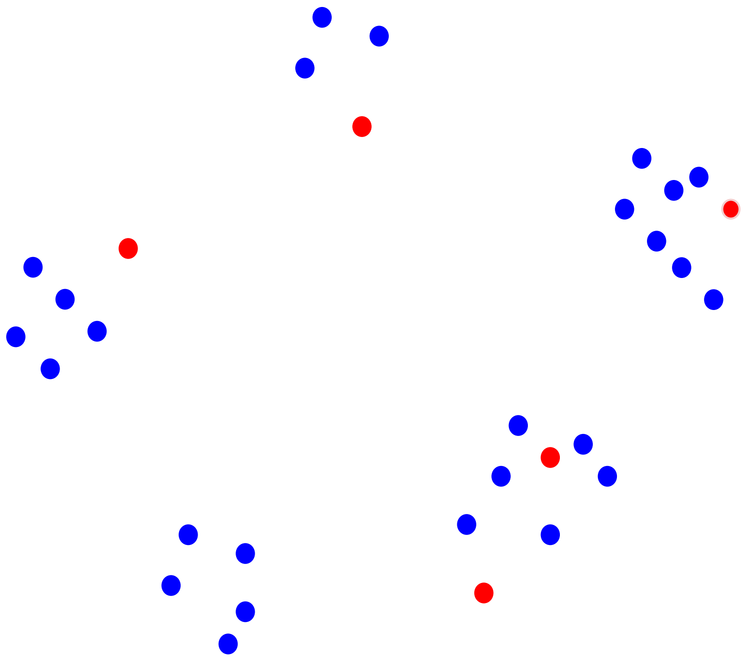
k-means++



First center: uniformly at random

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

k-means++



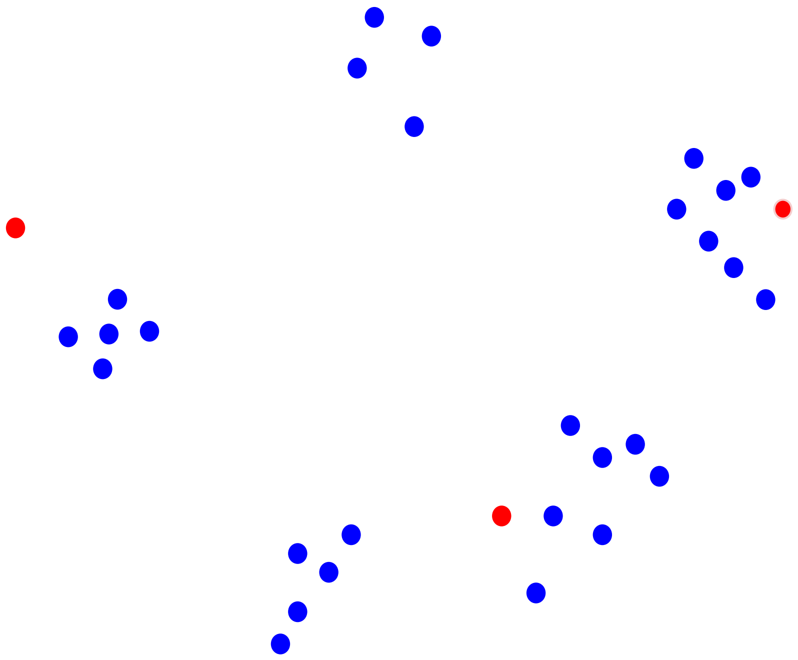
First center: uniformly at random

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

Looks like alright heuristic, but why does it give O(log k) approximation?

k-means++: bicriteria

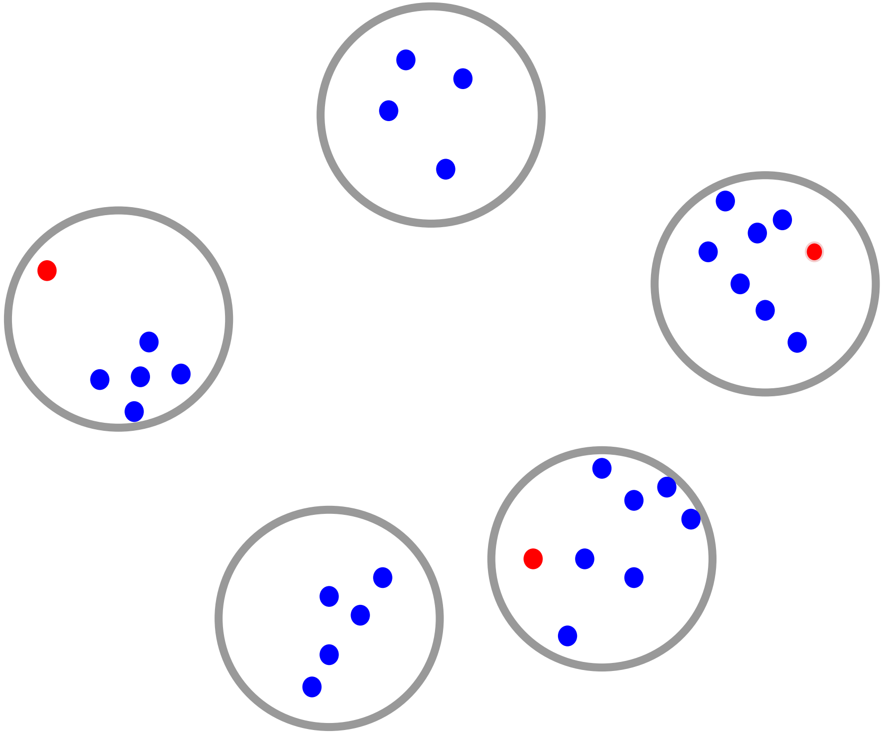
Sampling O(k) centers yields O(1) approximation to optimal solution on k centers. [Aggarwal, Deshpande, Kannan]



k-means++: bicriteria

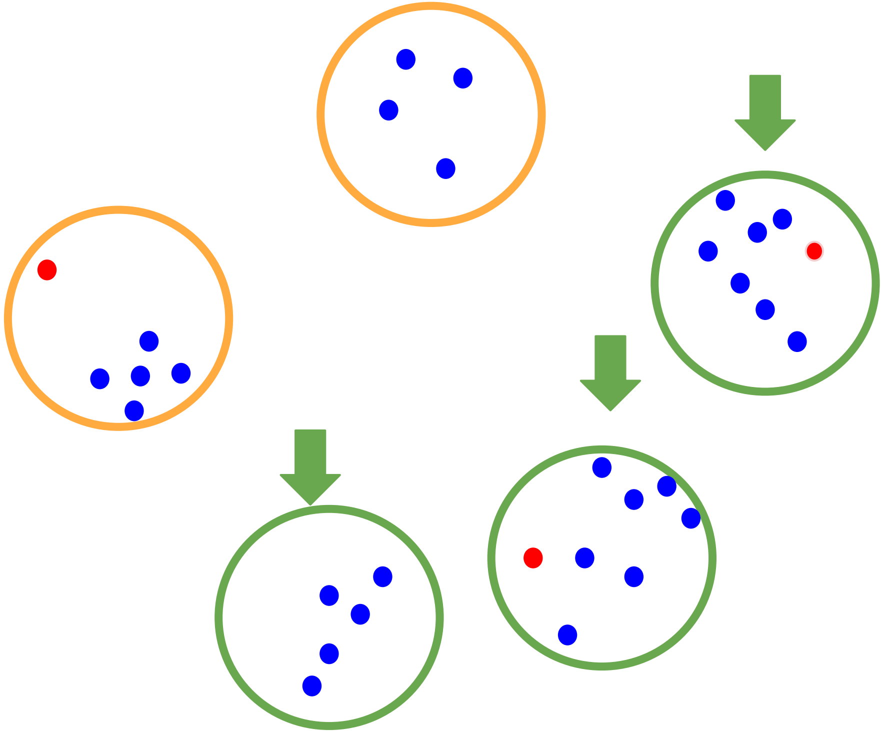
Sampling O(k) centers yields O(1) approximation to optimal solution on k centers. [Aggarwal, Deshpande, Kannan]

“balls into bins”:
A new center is sampled from given cluster proportional to the cost of the cluster.



k-means++: bicriteria

cluster is settled = we pay ≤10 times more than what OPT pays for that cluster

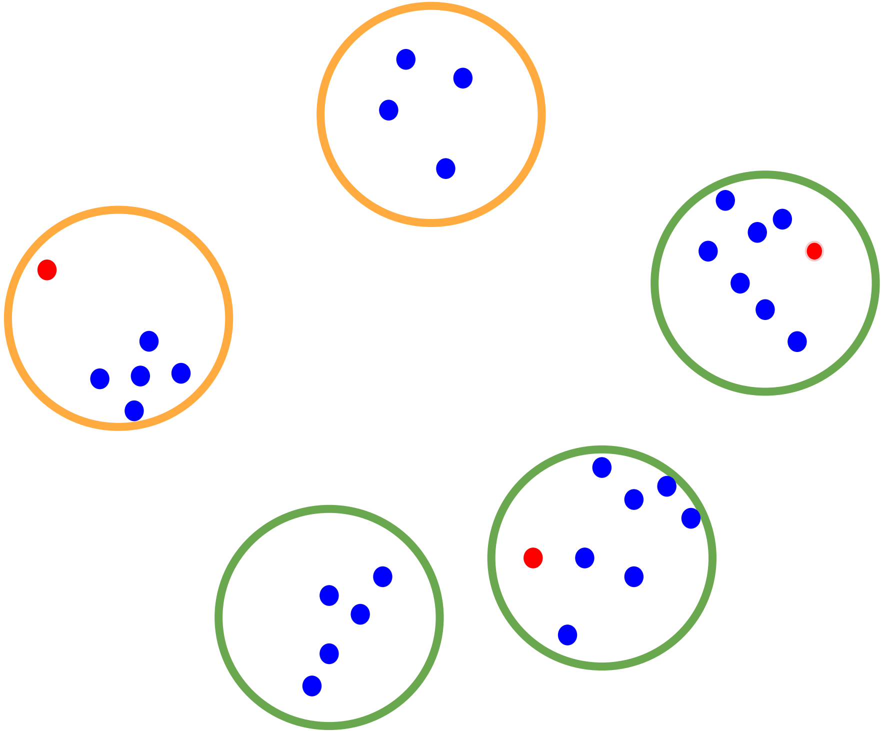


Sampling O(k) centers yields O(1) approximation to optimal solution on k centers. [Aggarwal, Deshpande, Kannan]

“balls into bins”:
A new center is sampled from given cluster proportional to the cost of the cluster.

k-means++: bicriteria

cluster is settled = we pay ≤10 times more than what OPT pays for that cluster



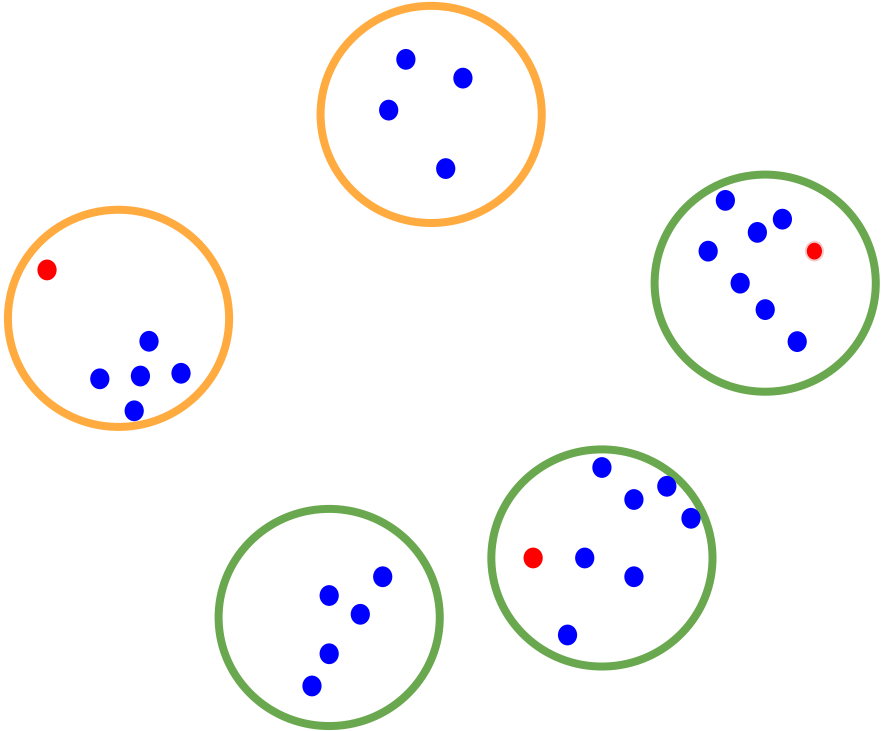
Sampling $O(k)$ centers yields $O(1)$ approximation to optimal solution on k centers. [Aggarwal, Deshpande, Kannan]

"balls into bins":
A new center is sampled from given cluster proportional to the cost of the cluster.

If current solution is ≥20 approximation of OPT, with ≥1/2 probability we sample a point from an unsettled cluster.

k-means++: bicriteria

cluster is settled = we pay ≤10 times more than what OPT pays for that cluster



Sampling O(k) centers yields O(1) approximation to optimal solution on k centers. [Aggarwal, Deshpande, Kannan]

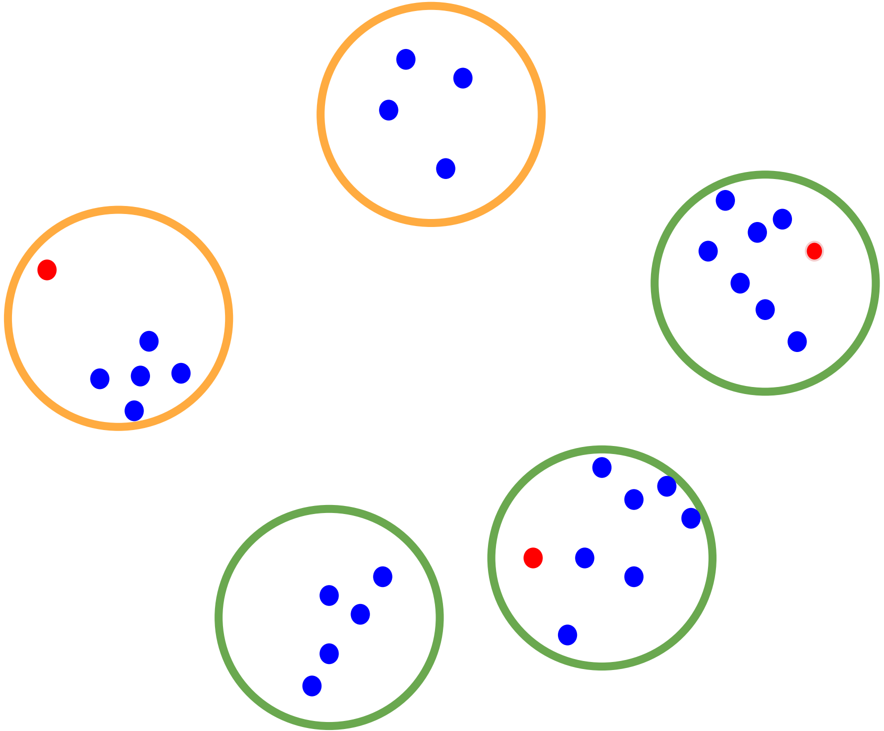
“balls into bins”:
A new center is sampled from given cluster proportional to the cost of the cluster.

If current solution is ≥20 approximation of OPT, with ≥1/2 probability we sample a point from an unsettled cluster.

Turns out that if we sample from any cluster, with 1/10 probability we make it settled.

k-means++: bicriteria

cluster is settled = we pay ≤10 times more than what OPT pays for that cluster



Sampling O(k) centers yields O(1) approximation to optimal solution on k centers. [Aggarwal, Deshpande, Kannan]

“balls into bins”:
A new center is sampled from given cluster proportional to the cost of the cluster.

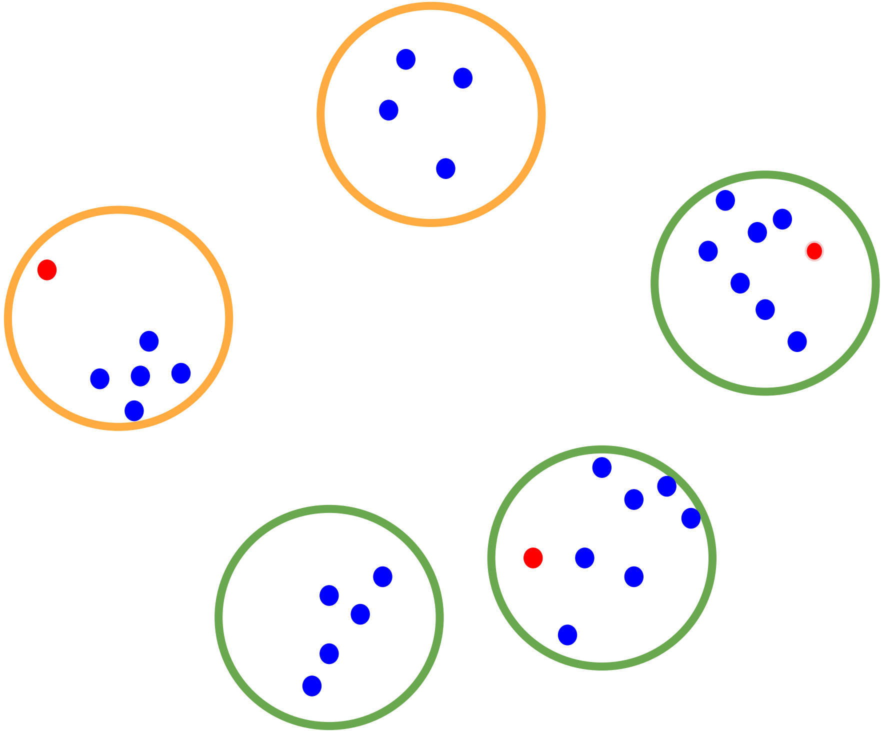
If current solution is ≥20 approximation of OPT, with ≥1/2 probability we sample a point from an unsettled cluster.

Turns out that if we sample from any cluster, with 1/10 probability we make it settled.

=> Each step makes at least one unsettled cluster settled with constant probability.

k-means++: bicriteria

cluster is settled = we pay ≤10 times more than what OPT pays for that cluster



Sampling O(k) centers yields O(1) approximation to optimal solution on k centers. [Aggarwal, Deshpande, Kannan]

"balls into bins":
A new center is sampled from given cluster proportional to the cost of the cluster.

If current solution is ≥20 approximation of OPT, with ≥1/2 probability we sample a point from an unsettled cluster.

Turns out that if we sample from any cluster, with 1/10 probability we make it settled.

=> Each step makes at least one unsettled cluster settled with constant probability.

After O(k) steps, we are done whp :-)

Outline

- Explain k-means++
- Explain its improved variant by Lattanzi and Sohler
- Tighter analysis of Lattanzi-Sohler's algorithm
- Extension of their algorithm to a similar problem (if time allows)

Algorithm of Lattanzi and Sohler (Local search ++)

k-means++:

- Sampling k centers yields O(log k) approximation
- Sampling O(k) centers yields O(1) approximation

Algorithm of Lattanzi and Sohler (Local search ++)

k-means++:

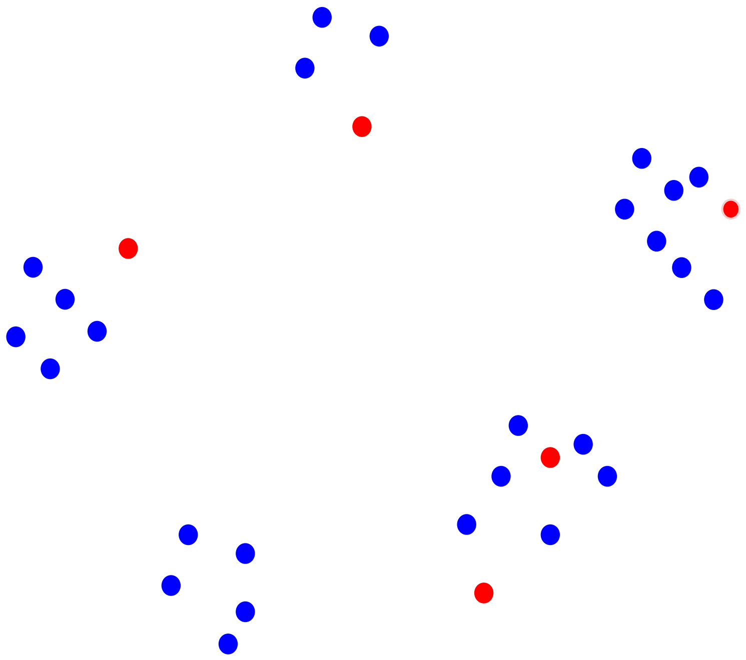
- Sampling k centers yields O(log k) approximation
- Sampling O(k) centers yields O(1) approximation

Lattanzi-Sohler:

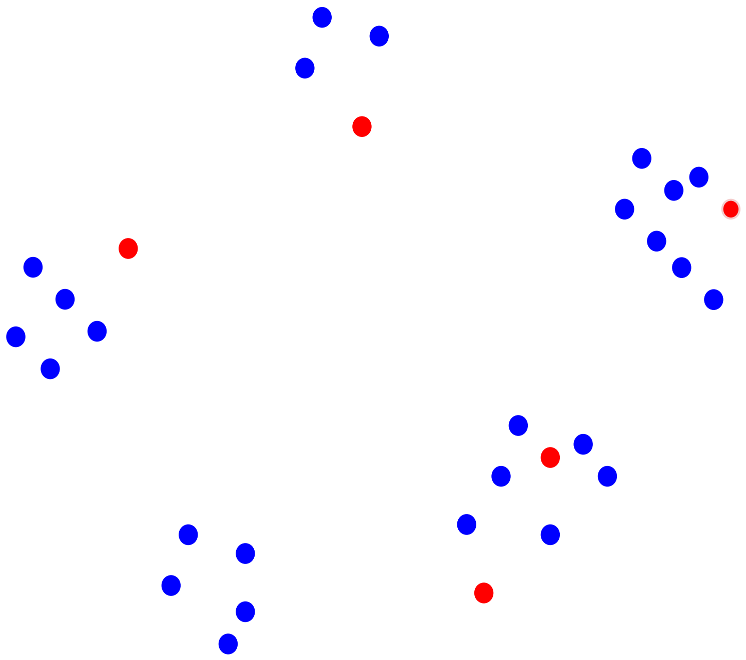
- sample k centers and yields O(1) approximation

Algorithm of Lattanzi and Sohler (Local search ++)

Run k-means++ (for k steps)



Algorithm of Lattanzi and Sohler (Local search ++)



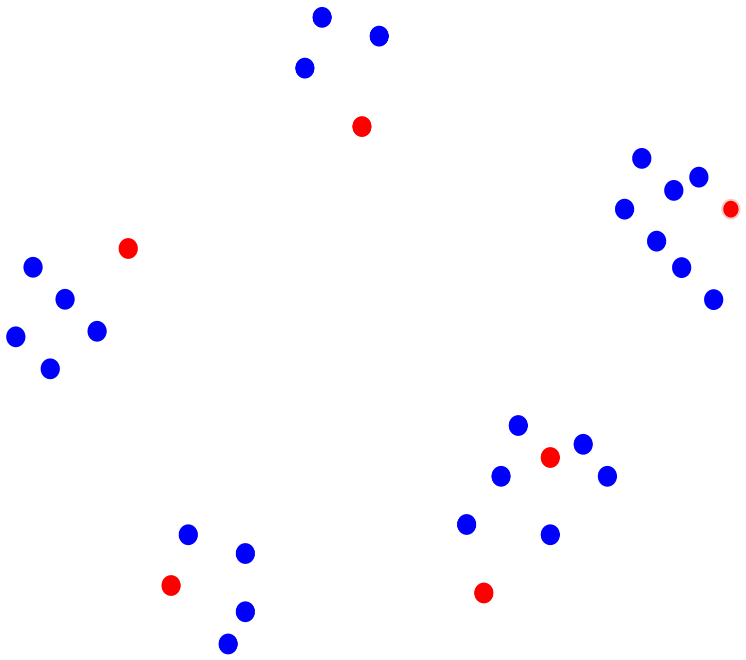
Run k-means++ (for k steps)

Then repeat the following:

- sample k+1th point as in k-means++
- go over your k+1 points and take out the one whose removal increases the cost the least

$$\sum_{p \in P} \min_{c \in C} d(p,c)^2$$

Algorithm of Lattanzi and Sohler (Local search ++)



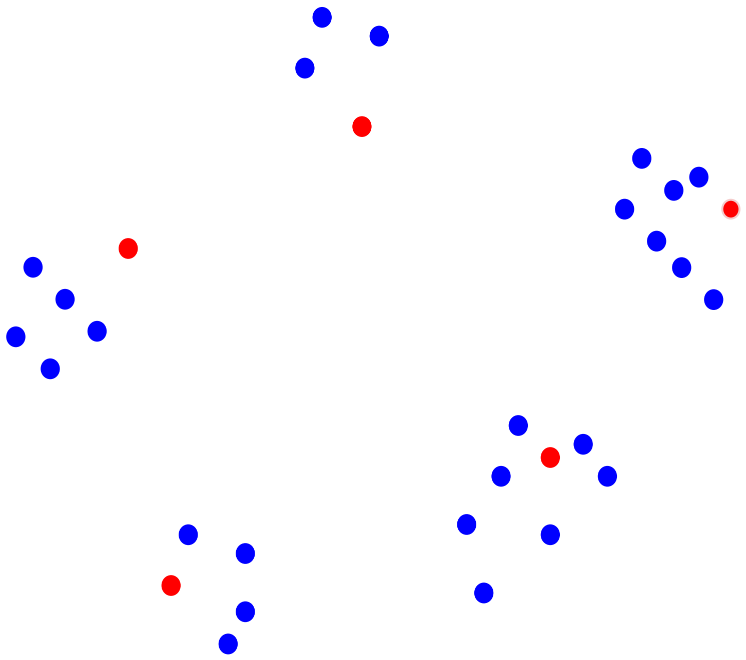
Run k-means++ (for k steps)

Then repeat the following:

- sample k+1th point as in k-means++
- go over your k+1 points and take out the one whose removal increases the cost the least

$$\sum_{p \in P} \min_{c \in C} d(p, c)^2$$

Algorithm of Lattanzi and Sohler (Local search ++)



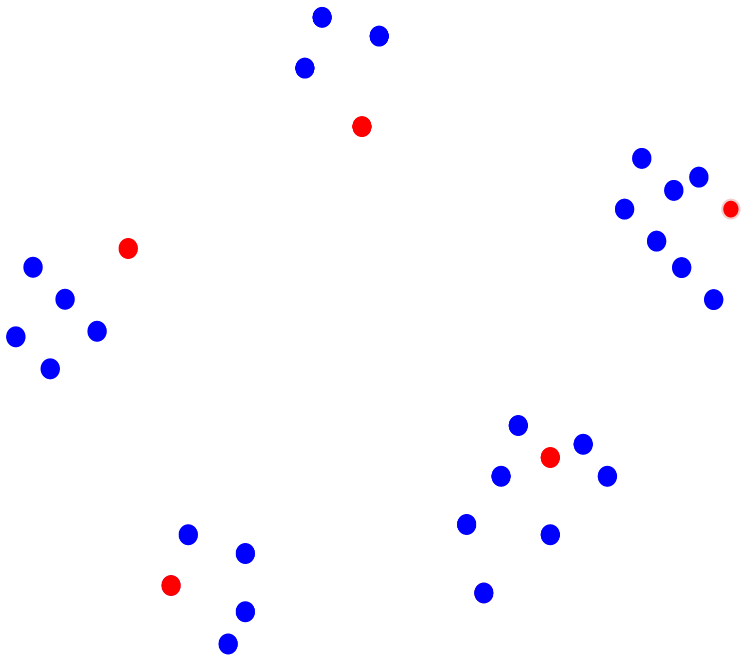
Run k-means++ (for k steps)

Then repeat the following:

- sample k+1th point as in k-means++
- go over your k+1 points and take out the one whose removal increases the cost the least

$$\sum_{p \in P} \min_{c \in C} d(p,c)^2$$

Algorithm of Lattanzi and Sohler (Local search ++)



Run k-means++ (for k steps)

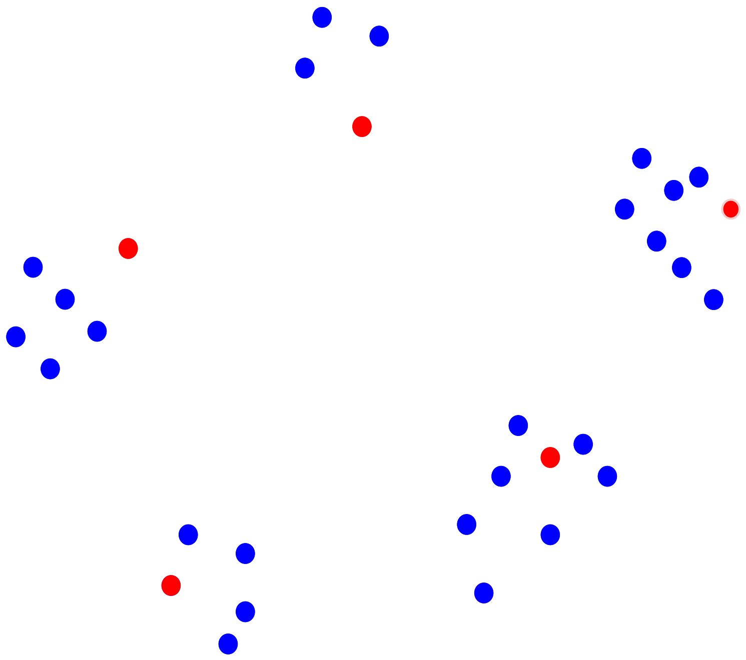
Then repeat the following:

- sample k+1th point as in k-means++
- go over your k+1 points and take out the one whose removal increases the cost the least

Theorem (LS): repeat O(k loglog k) times and you get O(1) approximation.

$$\sum_{p \in P} \min_{c \in C} d(p, c)^2$$

Algorithm of Lattanzi and Sohler (Local search ++)



Run k-means++ (for k steps)

Then repeat the following:

- sample k+1th point as in k-means++
- go over your k+1 points and take out the one whose removal increases the cost the least

Theorem (LS): repeat O(k loglog k) times and you get O(1) approximation.
Theorem (CGPR): actually, εk steps suffice for O(1/ε^3) approximation.

$$\sum_{p \in P} \min_{c \in C} d(p, c)^2$$

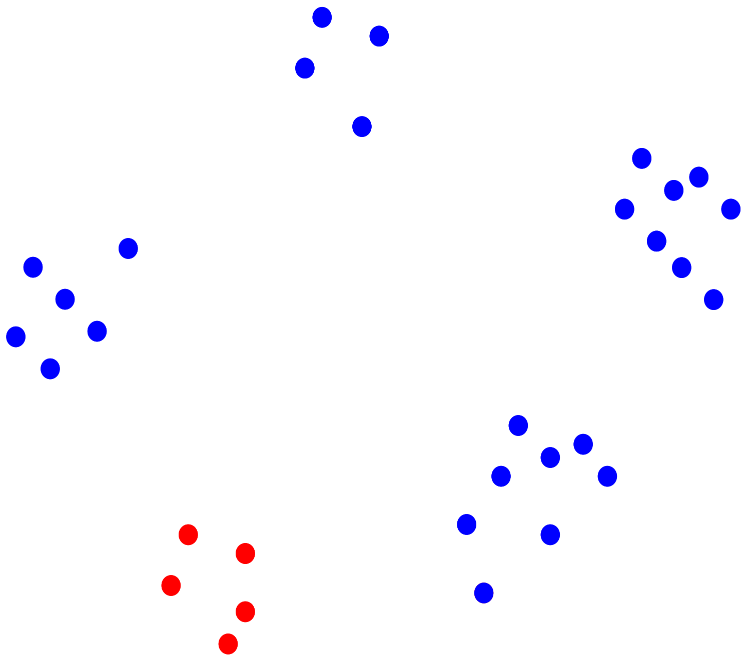
Analysis: intuition

Theorem ("local search", Kanungo et al): If we start with any set of k centers and try to "swap" any input points with any center in each step, we achieve O(1) approximation in polynomial time.

Different intuition based bicriteria guarantees: just sampling without removals gets O(1) approximation.

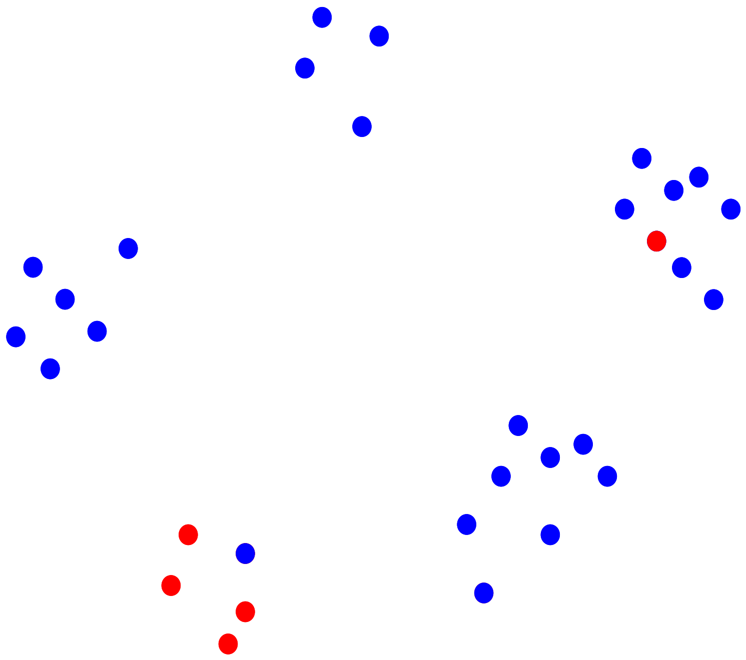
Analysis: one step

LS: cost of solution decreases multiplicatively by $1-\Theta(1/k)$ with constant probability



Analysis: one step

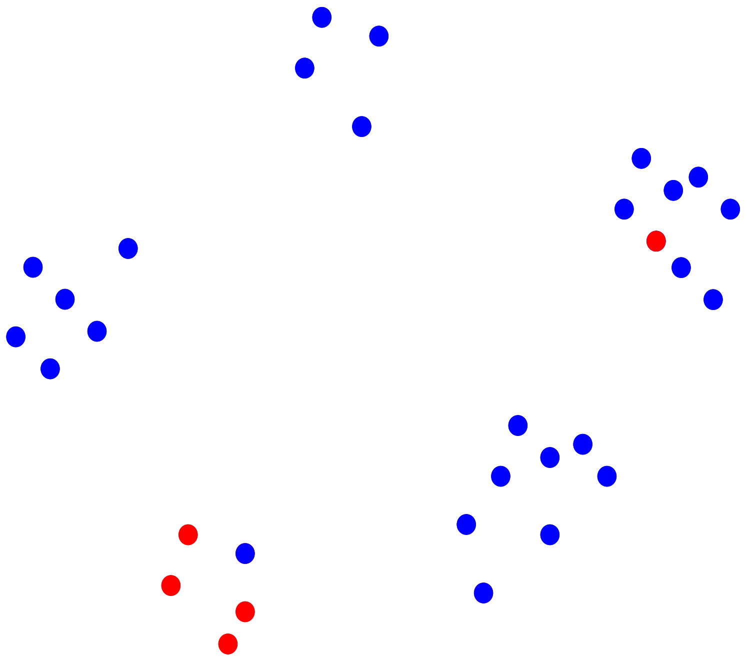
LS: cost of solution decreases multiplicatively by 1-Θ(1/k) with constant probability



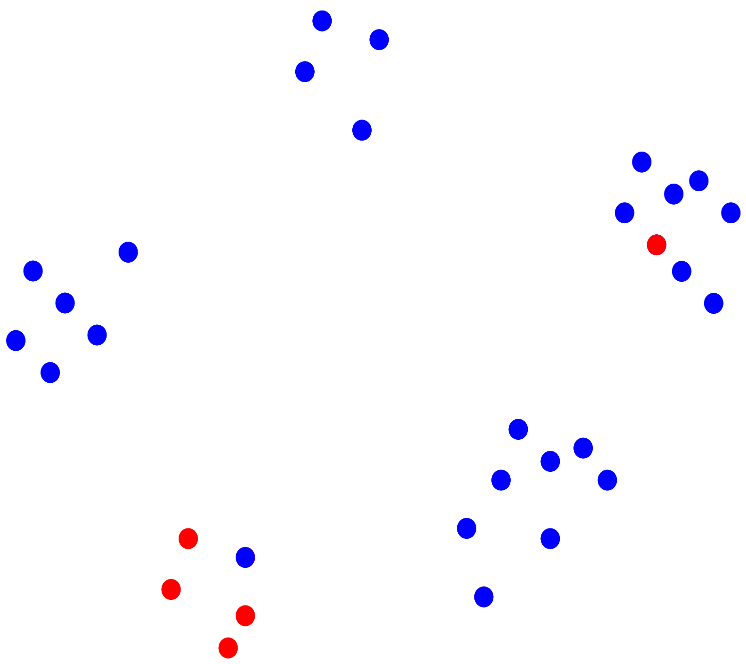
Analysis: one step

LS: cost of solution decreases multiplicatively by $1-\Theta(1/k)$ with constant probability

Hence, after $O(k)$ steps the approximation decrease from $\log(k)$ to $\log(k)/2$



Analysis: one step

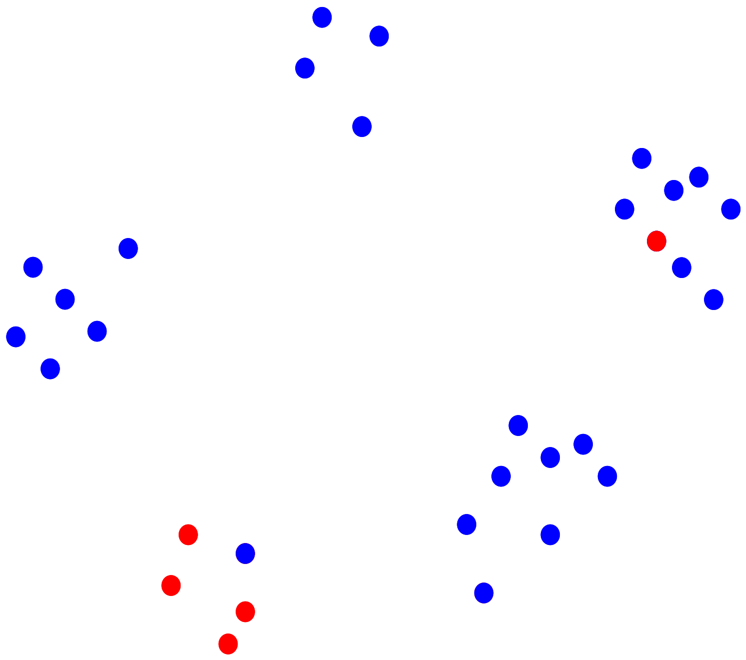


LS: cost of solution decreases multiplicatively by 1-Θ(1/k) with constant probability

Hence, after O(k) steps the approximation decrease from log(k) to log(k)/2

after O(k) more steps from log(k)/2 to log(k)/4
… after O(k loglog(k)) steps we are down to constant

Analysis: one step



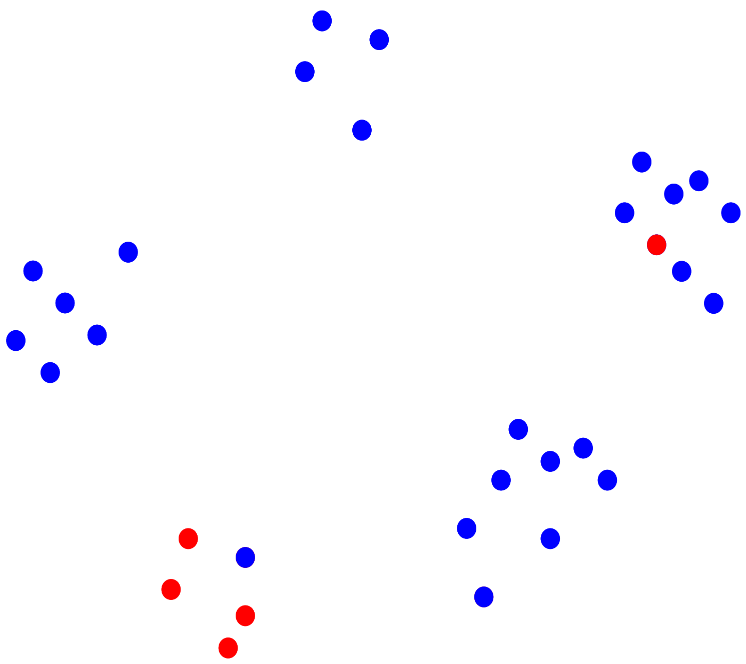
LS: cost of solution decreases multiplicatively by $1-\Theta(1/k)$ with constant probability

Hence, after $O(k)$ steps the approximation decrease from $\log(k)$ to $\log(k)/2$

after $O(k)$ more steps from $\log(k)/2$ to $\log(k)/4$
… after $O(k \log\log(k))$ steps we are down to constant

we cannot improve

Analysis: one step



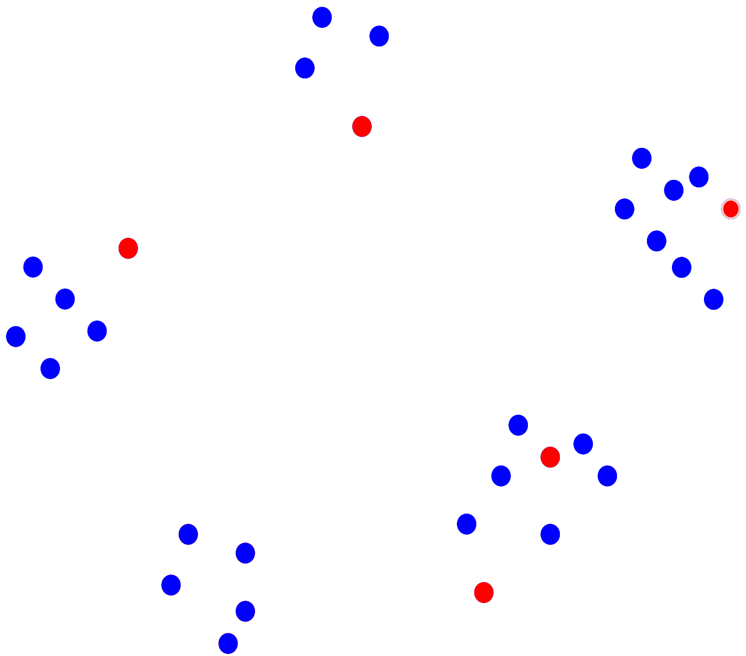
LS: cost of solution decreases multiplicatively by $1-\Theta(1/k)$ with constant probability

Hence, after $O(k)$ steps the approximation decrease from $\log(k)$ to $\log(k)/2$

after $O(k)$ more steps from $\log(k)/2$ to $\log(k)/4$
… after $O(k \log\log(k))$ steps we are down to constant

we cannot improve or can we?

Analysis: one step



LS: cost of solution decreases multiplicatively by $1-\Theta(1/k)$ with constant probability

Hence, after $O(k)$ steps the approximation decrease from $\log(k)$ to $\log(k)/2$

after $O(k)$ more steps from $\log(k)/2$ to $\log(k)/4$
… after $O(k \log\log(k))$ steps we are down to constant

we cannot improve or can we?

LS: cost of solution decreases multiplicatively by $1-\Theta(1/l)$ if the cost is "concentrated" just on l "unsettled" clusters

Outline

- Explain k-means++
- Explain its improved variant by Lattanzi and Sohler
- Tighter analysis of Lattanzi-Sohler's algorithm
- Extension of their algorithm to a similar problem (if time allows)

Analysis: few bad clusters

Proposition (CGPR): Suppose the current clustering is $\geq\alpha$-approximation of optimum. Then, $O(k/\sqrt[3]{\alpha})$ clusters are not $\sqrt[3]{\alpha}$-settled.

Analysis: few bad clusters

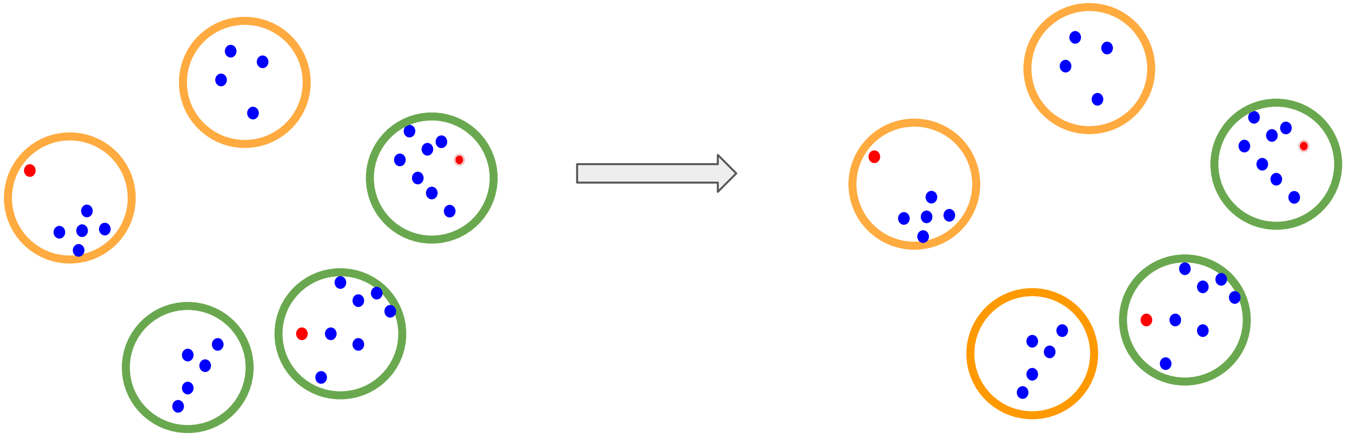
Proposition (CGPR): Suppose the current clustering is $\geq\alpha$-approximation of optimum. Then, $O(k/\sqrt[3]{\alpha})$ clusters are not $\sqrt[3]{\alpha}$-settled.

cluster A is β-settled: we pay at most β times more for A then what optimum pays.

Analysis: few bad clusters

Proposition (CGPR): Suppose the current clustering is $\geq\alpha$-approximation of optimum. Then, $O(k/\sqrt[3]{\alpha})$ clusters are not $\sqrt[3]{\alpha}$-settled.

cluster A is β-settled: in our set of centers C, there is $c \in A$ and it "certifies" we pay at most β times more for A then what optimum pays.



Analysis: O(k) steps

Proposition (CGPR): Suppose the current clustering is $\geq\alpha$-approximation of optimum. Then, $O(k/\sqrt[3]{\alpha})$ clusters are not $\sqrt[3]{\alpha}$-settled.

Fact (LS): Improvement of one step is $(1 - 1/l) = (1 - \sqrt[3]{\alpha}/k)$

Corollary: Hence, after $O(k/\sqrt[3]{\alpha})$ steps the approximation factor drops to $\alpha/2$ and after $O(k/\sqrt[3]{(\alpha/2)})$ steps drops to $\alpha/4$ … after $O(k)$ steps we have constant approximation.

Analysis: technical part

Proposition (CGPR): Suppose the current clustering is $\geq\alpha$-approximation of optimum. Then, $O(k/\sqrt[3]{\alpha})$ clusters are not $\sqrt[3]{\alpha}$-settled.

Fact: Suppose the current clustering is $\geq\alpha$-approximation of optimum. Then, with probability $1-1/\sqrt[3]{\alpha}$ we sample a new point from $\sqrt[3]{\alpha}$-unsettled cluster and make it $\sqrt[3]{\alpha}$-settled

Corollary: after kmeans++, there are $O(k/\sqrt[3]{\alpha})$ $\sqrt[3]{\alpha}$-unsettled clusters.

Corollary: in each local search step, the number of $\sqrt[3]{\alpha}$-unsettled clusters increments by ≤ 1 with probability $\leq 1/\sqrt[3]{\alpha}$ => after $O(k)$ steps still only $O(k/\sqrt[3]{\alpha})$ $\sqrt[3]{\alpha}$-unsettled clusters.

Outline

- Explain k-means++
- Explain its improved variant by Lattanzi and Sohler
- Tighter analysis of Lattanzi-Sohler’s algorithm
- Extension of their algorithm to a similar problem (if time allows)

Extension to k-means with outliers

Select a subset of z "outliers" and output k centers that optimize the k-means cost on the remaining vertices.

Bhaskara et al.: There is k-means based algorithm that gives $O(\log k)$ approximation, but only if it is allowed to output $O(z * \log k)$ many outliers.

Lattanzi-Sohler: $O(1)$ approximation with $O(z)$ outliers.

One more trick and more careful analysis (Grunau, R): $O(1/\varepsilon)$ approximation with $(1+\varepsilon)z$ outliers.

Also can be extended to k-center with outliers.

Summary

The trick of Lattanzi and Sohler enables you to turn bicriteria approximation in true approximation (for incremental sampling based algorithms).

The analysis of Lattanzi-Sohler algorithm can be improved if you use that
“in k-means++, most of the clusters are well approximated even if the cost is high”.