k-means++: few more steps yield constant approximation

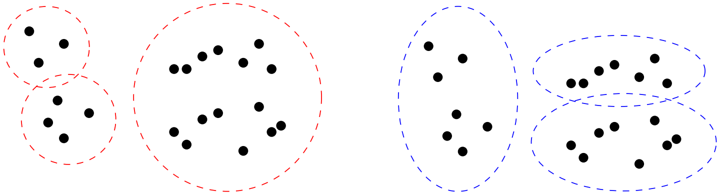
Davin Choo Christoph Grunau
Julian Portmann Václav Rozhoň

ETH Zürich

ICML 2020

Clustering

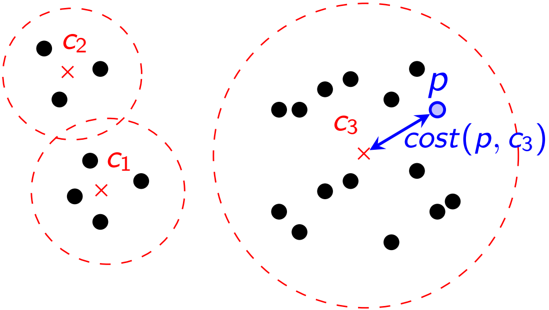
Given *unlabelled* d-dimensional data points $P = \{p_1, \ldots, p_n\}$, group *similar* ones together into k clusters



Which is a better clustering into $k = 3$ groups?

k-means metric

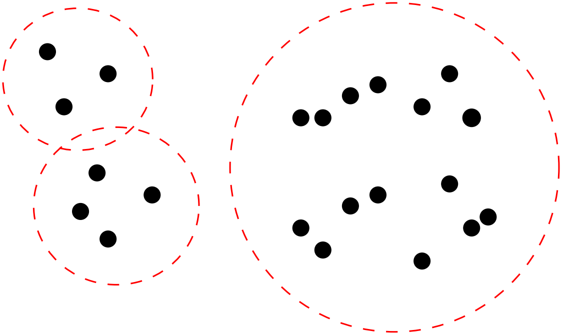
- Centers $C = \{c_1, \ldots, c_k\}$
- $cost(P, C) = \sum_{p \in P} \min_{c \in C} d(p, c)^2 = \sum_{p \in P} cost(p, C)$



- Restricting $C \subseteq P$ only loses a 2-factor in $cost(P, C)$
- NP-hard to find optimal solution [ADHP09, MNV09]

k-means metric

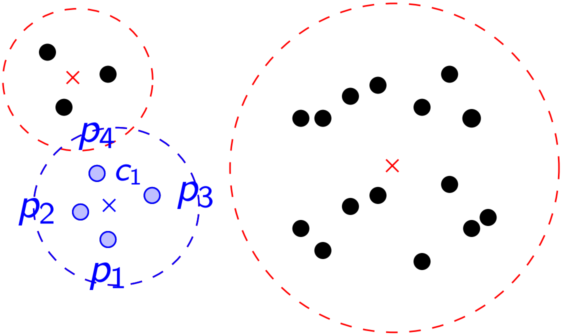
- Centers $C = \{c_1, \ldots, c_k\}$
- $cost(P, C) = \sum_{p \in P} \min_{c \in C} d(p, c)^2 = \sum_{p \in P} cost(p, C)$



- Given k clusters, optimal centers are the means/centroids

k-means metric

- Centers $C = \{c_1, \ldots, c_k\}$
- $cost(P, C) = \sum_{p \in P} \min_{c \in C} d(p, c)^2 = \sum_{p \in P} cost(p, C)$

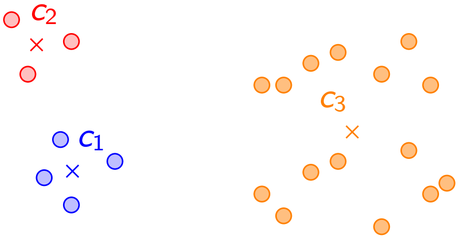


- Given k clusters, optimal centers are the means/centroids

$$\text{e.g.} \quad c_1 = \frac{1}{4}\left[p_1 + p_2 + p_3 + p_4\right]$$

k-means metric

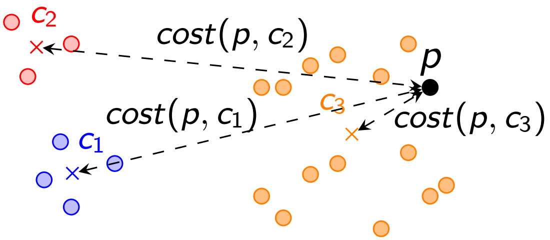
- Centers $C = \{c_1, \ldots, c_k\}$
- $cost(P, C) = \sum_{p \in P} \min_{c \in C} d(p, c)^2 = \sum_{p \in P} cost(p, C)$



- Given k clusters, optimal centers are the means/centroids
- Given k centers, optimal cluster assignment is closest center

k-means metric

- Centers $C = \{c_1, \ldots, c_k\}$
- $cost(P, C) = \sum_{p \in P} \min_{c \in C} d(p, c)^2 = \sum_{p \in P} cost(p, C)$



- Given k clusters, optimal centers are the means/centroids
- Given k centers, optimal cluster assignment is closest center

Lloyd's algo. [Llo82]: Heuristic alternating minimization

Given k initial centers (Remark: centers not necessarily from P)

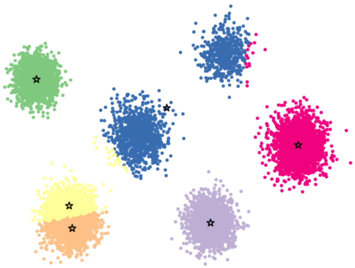
Optimal assignment $\longleftrightarrow$ Optimal clustering

(Animation works only for PDF readers like Adobe Acrobat Reader)

Lloyd's algo. [Llo82]: Heuristic alternating minimization

Given k initial centers (Remark: centers not necessarily from P)

$$\text{Optimal assignment} \longleftrightarrow \text{Optimal clustering}$$

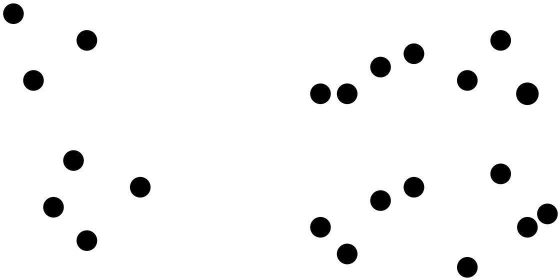


- Lloyd's algorithm *never* worsens $cost(P, C)$ but has no performance guarantees (local minimas)
- One way to get theoretic guarantees:

Seed with provably good initial centers

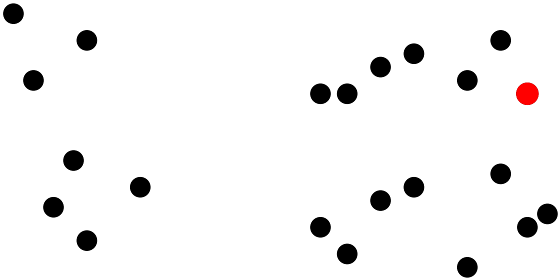
k-means++ initialization [AV07]

- Chooses k points from P: $\mathcal{O}(\log k)$ apx. (in expectation)
- 1^{st} center chosen uniformly at random from P



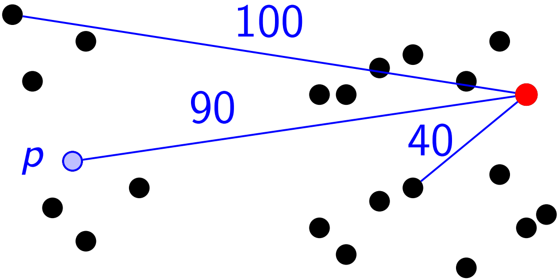
k-means++ initialization [AV07]

- Chooses k points from P: $\mathcal{O}(\log k)$ apx. (in expectation)
- 1^{st} center chosen uniformly at random from P



k-means++ initialization [AV07]

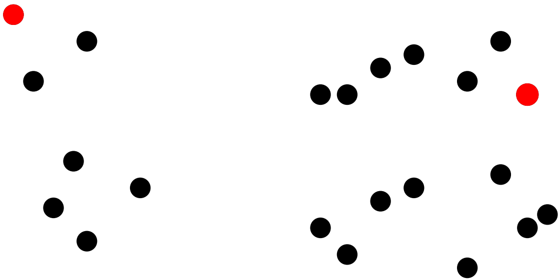
- Chooses k points from P: $\mathcal{O}(\log k)$ apx. (in expectation)
- 1^{st} center chosen uniformly at random from P
- D^2-sampling: $\Pr[p] = \frac{cost(p,C)}{\sum_{p \in P} cost(p,C)}$ ⟵ Cost to centers C (updated at each step)



$cost(p, C) = 90$

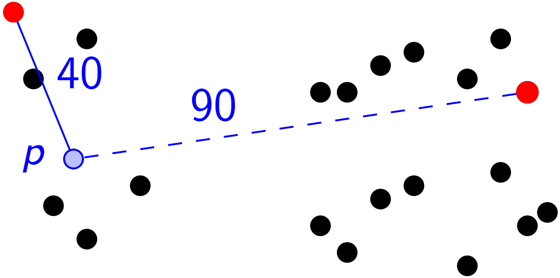
k-means++ initialization [AV07]

- Chooses k points from P: $\mathcal{O}(\log k)$ apx. (in expectation)
- 1^{st} center chosen uniformly at random from P
- D^2-sampling: $\Pr[p] = \frac{cost(p,C)}{\sum_{p \in P} cost(p,C)}$ ← Cost to centers C (updated at each step)



k-means++ initialization [AV07]

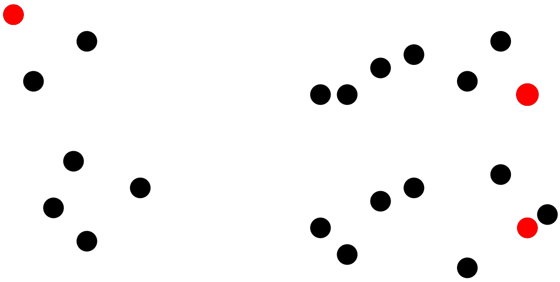
- Chooses k points from P: $\mathcal{O}(\log k)$ apx. (in expectation)
- 1^{st} center chosen uniformly at random from P
- D^2-sampling: $\Pr[p] = \frac{cost(p,C)}{\sum_{p \in P} cost(p,C)}$ ⟵ Cost to centers C (updated at each step)



$cost(p, C) = 40$

k-means++ initialization [AV07]

- Chooses k points from P: $\mathcal{O}(\log k)$ apx. (in expectation)
- 1^{st} center chosen uniformly at random from P
- D^2-sampling: $\Pr[p] = \frac{cost(p,C)}{\sum_{p \in P} cost(p,C)}$ ⟵ Cost to centers C (updated at each step)



- Practically efficient: $\mathcal{O}(dnk)$ running time
- Exist instances where running k-means++ yield $\Omega(\log k)$ apx. with high probability in k [BR13, BJA16]

What is known?

- Lloyd's algorithm [Llo82]

Practice

Theory

What is known?

- Lloyd's algorithm [Llo82]

Practice

- Best known approximation factor [ANFSW19]: 6.357
- PTAS for fixed k [KSS10]
- PTAS for fixed d [CAKM19, FRS19]
- Local search [KMN$^+$04]: $(9 + \epsilon)$-approximation in poly-time

Theory

What is known?

Practice

- Lloyd's algorithm [Llo82]
- k-means++ [AV07]: $\mathcal{O}(\log k)$ apx. in $\mathcal{O}(dnk)$ time
- LocalSearch++ [LS19]: $\mathcal{O}(1)$ apx. in $\mathcal{O}(dnk^2 \log\log k)$ time

Theory

- Best known approximation factor [ANFSW19]: 6.357
- PTAS for fixed k [KSS10]
- PTAS for fixed d [CAKM19, FRS19]
- Local search [KMN$^+$04]: $(9+\epsilon)$-approximation in poly-time

What is known?

- Lloyd's algorithm [Llo82]

Practice

- k-means++ [AV07]: $\mathcal{O}(\log k)$ apx. in $\mathcal{O}(dnk)$ time
- LocalSearch++ [LS19]: $\mathcal{O}(1)$ apx. in $\mathcal{O}(dnk^2 \log\log k)$ time

- Best known approximation factor [ANFSW19]: 6.357
- PTAS for fixed k [KSS10]
- PTAS for fixed d [CAKM19, FRS19]
- Local search [KMN$^+$04]: $(9+\epsilon)$-approximation in poly-time

Theory

- Bi-criteria approximation [Wei16, ADK09]:
  $\mathcal{O}(1)$-approximation with $\mathcal{O}(k)$ cluster centers

What is known?

Practice

- Lloyd's algorithm [Llo82]
- k-means++ [AV07]: $\mathcal{O}(\log k)$ apx. in $\mathcal{O}(dnk)$ time
- LocalSearch++ [LS19]: $\mathcal{O}(1)$ apx. in $\mathcal{O}(dnk^2 \cancel{\log\log k})$ time

Theory

- Best known approximation factor [ANFSW19]: 6.357
- PTAS for fixed k [KSS10]
- PTAS for fixed d [CAKM19, FRS19]
- Local search [KMN$^+$04]: $(9+\epsilon)$-approximation in poly-time

- Bi-criteria approximation [Wei16, ADK09]: $\mathcal{O}(1)$-approximation with $\mathcal{O}(k)$ cluster centers
- This work: $\mathcal{O}(dnk^2)$ running time, $\mathcal{O}(1)$ approximation

Outline of talk

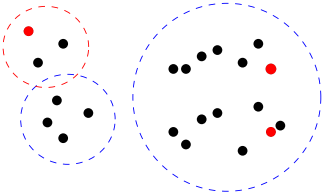
- What we have discussed
  - Clustering as a motivation
  - Lloyd's heuristic and k-means++ initialization
  - Prior work

Outline of talk

- What we have discussed
  - Clustering as a motivation
  - Lloyd's heuristic and k-means++ initialization
  - Prior work
- What's next
  - Idea of bi-criteria algorithm and notion of settledness
  - Idea of local search
  - LocalSearch++: combining k-means++ with local search
  - Key idea behind how we tighten analysis of LocalSearch++

Bi-criteria [Wei16, ADK09] and settledness

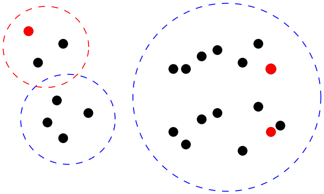
- "Balls into bins" process
  - k bins: Optimal k-clustering of points defined by OPT_k
  - $\mathcal{O}(k)$ balls: Sampled points in C



- A cluster Q is *settled* if $cost(Q, C) \leq 10 \cdot cost(Q, OPT_k)$

Bi-criteria [Wei16, ADK09] and settledness

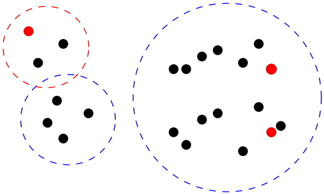
- "Balls into bins" process
  - k bins: Optimal k-clustering of points defined by OPT_k
  - $\mathcal{O}(k)$ balls: Sampled points in C



- A cluster Q is *settled* if $cost(Q, C) \leq 10 \cdot cost(Q, OPT_k)$
- Can show (with constant success probabilities):
  - If not yet 20-apx., D^2-sampling chooses from unsettled cluster
  - If sample p from unsettled cluster Q, adding p makes Q settled

Bi-criteria [Wei16, ADK09] and settledness

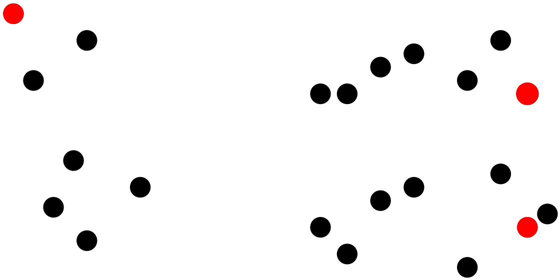
- "Balls into bins" process
  - k bins: Optimal k-clustering of points defined by OPT_k
  - $\mathcal{O}(k)$ balls: Sampled points in C



- A cluster Q is *settled* if $cost(Q, C) \leq 10 \cdot cost(Q, OPT_k)$
- Can show (with constant success probabilities):
  - If not yet 20-apx., D^2-sampling chooses from unsettled cluster
  - If sample p from unsettled cluster Q, adding p makes Q settled
- After $\mathcal{O}(k)$ samples, $cost(P, C) \leq 20 \cdot cost(P, OPT_k)$

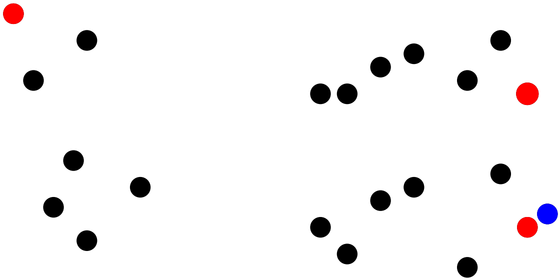
Local search [KMN+04]

- Initialize arbitrary k points $\rightarrow C$
- Repeat
  - Pick arbitrary point $p \in P$
  - If $\exists q \in C$ such that $cost(P, C \setminus \{q\} \cup \{p\})$ improves cost, swap



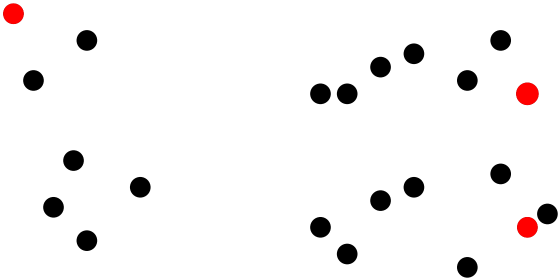
Local search [KMN+04]

- Initialize arbitrary k points $\to C$
- Repeat
  - Pick arbitrary point $p \in P$
  - If $\exists q \in C$ such that $cost(P, C \setminus \{q\} \cup \{p\})$ improves cost, swap



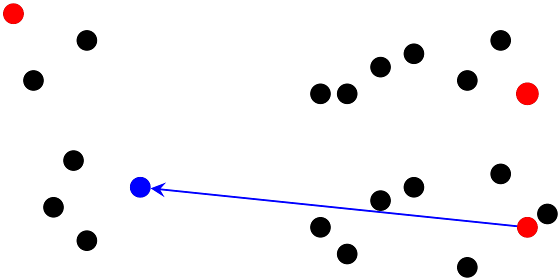
Local search [KMN+04]

- Initialize arbitrary k points $\to C$
- Repeat
  - Pick arbitrary point $p \in P$
  - If $\exists q \in C$ such that $cost(P, C \setminus \{q\} \cup \{p\})$ improves cost, swap



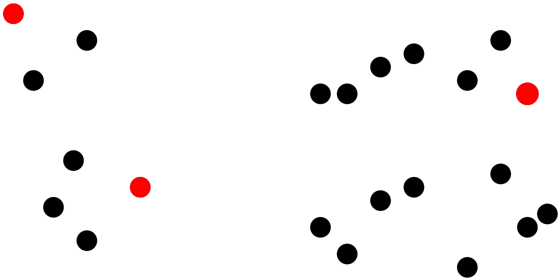
Local search [KMN+04]

- Initialize arbitrary k points $\rightarrow C$
- Repeat
  - Pick arbitrary point $p \in P$
  - If $\exists q \in C$ such that $cost(P, C \setminus \{q\} \cup \{p\})$ improves cost, swap



Local search [KMN+04]

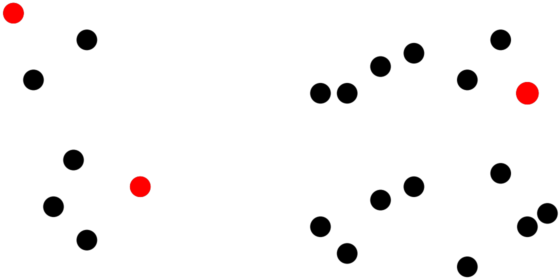
- Initialize arbitrary k points $\rightarrow C$
- Repeat
  - Pick arbitrary point $p \in P$
  - If $\exists q \in C$ such that $cost(P, C \setminus \{q\} \cup \{p\})$ improves cost, swap



- Polynomial number of iterations $\rightarrow \mathcal{O}(1)$ approximation

LocalSearch++ [LS19]

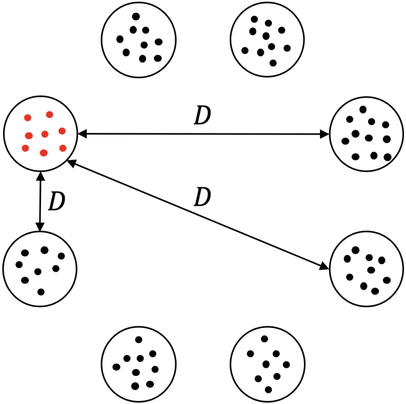
- Initialize ~~arbitrary~~ k points $\rightarrow C$ from output of k-means++
- Repeat
  - Pick ~~arbitrary~~ point $p \in P$ using D^2-sampling
  - If $\exists q \in C$ such that $cost(P, \{p\} \cup C \setminus \{q\})$ improves cost, swap



- $\mathcal{O}(k \log \log k)$ ~~Polynomial~~ number of iterations $\rightarrow \mathcal{O}(1)$ approximation

LocalSearch++ [LS19]: One step of analysis

- Lemma: In each step, cost decrease by factor of $1 - \Theta\left(\frac{1}{k}\right)$ with constant probability



LocalSearch++ [LS19]: One step of analysis

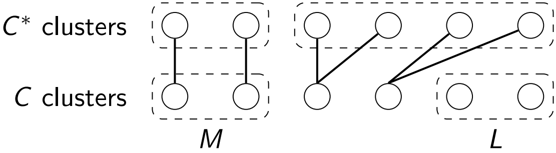
- Lemma: In each step, cost decrease by factor of $1 - \Theta\left(\frac{1}{k}\right)$ with constant probability
- Implication: After $\mathcal{O}(k)$ steps, approximation factor halves

k-means++ is $\mathcal{O}(\log k)$ apx. in expectation

$$\underbrace{\mathcal{O}(\log k)\text{-apx.} \xrightarrow{\mathcal{O}(k)\text{ steps}} \mathcal{O}\left(\frac{\log k}{2}\right)\text{-apx.} \xrightarrow{\mathcal{O}(k)\text{ steps}} \cdots \rightarrow \mathcal{O}\left(\frac{\log k}{2^r}\right) = \mathcal{O}(1)\text{-apx.}}_{r = \mathcal{O}(\log\log k)\text{ phases, totaling } \mathcal{O}(k \log\log k)\text{ steps}}$$

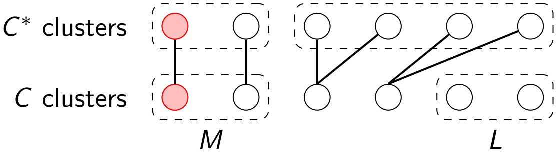
LocalSearch++ [LS19]: Bounding cost decrease

- Match OPT centers $c^* \in C^*$ to candidate centers $c \in C$



LocalSearch++ [LS19]: Bounding cost decrease

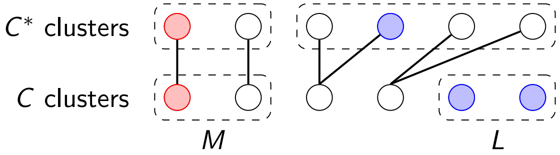
- Match OPT centers $c^* \in C^*$ to candidate centers $c \in C$



- If "D^2-sampled left side" $\rightarrow$ swap with paired $c \in C$

LocalSearch++ [LS19]: Bounding cost decrease

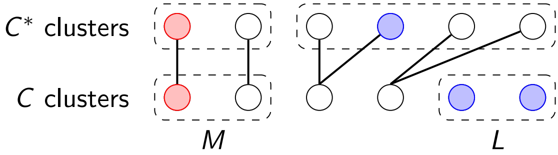
- Match OPT centers $c^* \in C^*$ to candidate centers $c \in C$



- If "D^2-sampled left side" $\rightarrow$ swap with paired $c \in C$
- If "D^2-sampled right side" $\rightarrow$ swap with "best" lonely $c \in C$

LocalSearch++ [LS19]: Bounding cost decrease

- Match OPT centers $c^* \in C^*$ to candidate centers $c \in C$



- If "D^2-sampled left side" $\rightarrow$ swap with paired $c \in C$
- If "D^2-sampled right side" $\rightarrow$ swap with "best" lonely $c \in C$
- Can show: Good probability to D^2-sample a point such that updating centers sufficiently decreases cost

Structural insight: Few bad clusters

- Cluster Q is β-settled if $cost(Q, C) \leq (\beta + 1) \cdot cost(Q, OPT)$
- Informal propositions
  - If current clustering is α-approximate,
    - There are $\mathcal{O}\left(\frac{k}{\sqrt[3]{\alpha}}\right)$ $\sqrt[3]{\alpha}$-unsettled clusters
    - D^2-sampling samples a point from an $\sqrt[3]{\alpha}$-unsettled cluster Q; Adding this point to C makes Q $\sqrt[3]{\alpha}$-settled

Structural insight: Few bad clusters

- Cluster Q is β-settled if $cost(Q, C) \leq (\beta + 1) \cdot cost(Q, OPT)$
- Informal propositions
  - If current clustering is α-approximate,
    - There are $\mathcal{O}\left(\frac{k}{\sqrt[3]{\alpha}}\right)$ $\sqrt[3]{\alpha}$-unsettled clusters
    - D^2-sampling samples a point from an $\sqrt[3]{\alpha}$-unsettled cluster Q; Adding this point to C makes Q $\sqrt[3]{\alpha}$-settled
  - In each step, cost decrease by factor of $1 - \Theta\left(\frac{\sqrt[3]{\alpha}}{k}\right)$

k-means++ with Markov: $\alpha \leq \exp(k^{0.1})$ with high probability in k

$$\underbrace{\alpha\text{-apx.} \xrightarrow[]{\mathcal{O}\left(\frac{k}{\sqrt[3]{\alpha}}\right)\text{ steps}} \frac{\alpha}{2}\text{-apx.} \xrightarrow[]{\mathcal{O}\left(\frac{k}{\sqrt[3]{\alpha/2}}\right)\text{ steps}} \cdots \longrightarrow \mathcal{O}\left(\frac{1}{\epsilon^3}\right)\text{-apx.}}_{\mathcal{O}(\log \alpha)\text{ phases, totalling } \epsilon k \text{ steps}}$$

Summary

- Improved analysis of LocalSearch++
  - Simple algorithm: k-means++, then local search
  - Theoretic guarantees: ϵk local search steps yield $\mathcal{O}\left(\frac{1}{\epsilon^3}\right)$-apx.
  - Practical algorithm: Can yield $\sim$ 15% improvements compared to without any local search steps [LS19]

Summary

- Improved analysis of LocalSearch++
  - Simple algorithm: k-means++, then local search
  - Theoretic guarantees: ϵk local search steps yield $\mathcal{O}\left(\frac{1}{\epsilon^3}\right)$-apx.
  - Practical algorithm: Can yield $\sim 15\%$ improvements compared to without any local search steps [LS19]
- Structural analysis of clusters
  - Go beyond worst-case analysis of k-means++
  - After k-means++,
    - Few clusters are unsettled
    - Most clusters are "well-approximated"
    - A few steps of local search can fix this

References I

Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat.
NP-hardness of Euclidean sum-of-squares clustering.
Machine learning, 75(2):245–248, 2009.

Ankit Aggarwal, Amit Deshpande, and Ravi Kannan.
Adaptive sampling for k-means clustering.
In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 15–28. Springer, 2009.

Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward.
Better Guarantees for k-Means and Euclidean k-Median by Primal-Dual Algorithms.
SIAM Journal on Computing, 0(0):FOCS17–97, 2019.

References II

David Arthur and Sergei Vassilvitskii.
k-means++: The advantages of careful seeding.
In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

Anup Bhattacharya, Ragesh Jaiswal, and Nir Ailon.
Tight lower bound instances for k-means++ in two dimensions.
Theoretical Computer Science, 634:55–66, 2016.

Tobias Brunsch and Heiko Röglin.
A bad instance for k-means++.
Theoretical Computer Science, 505:19–26, 2013.

References III

Vincent Cohen-Addad, Philip N Klein, and Claire Mathieu.
Local search yields approximation schemes for k-means and k-median in euclidean and minor-free metrics.
SIAM Journal on Computing, 48(2):644–667, 2019.

Zachary Friggstad, Mohsen Rezapour, and Mohammad R Salavatipour.
Local search yields a ptas for k-means in doubling metrics.
SIAM Journal on Computing, 48(2):452–480, 2019.

Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu.
A local search approximation algorithm for k-means clustering.
Computational Geometry, 28(2-3):89–112, 2004.

References IV

Amit Kumar, Yogish Sabharwal, and Sandeep Sen.
Linear-time approximation schemes for clustering problems in any dimensions.
Journal of the ACM (JACM), 57(2):1–32, 2010.

Stuart Lloyd.
Least squares quantization in PCM.
IEEE transactions on information theory, 28(2):129–137, 1982.

Silvio Lattanzi and Christian Sohler.
A better k-means++ Algorithm via Local Search.
In *International Conference on Machine Learning*, pages 3662–3671, 2019.

References V

Meena Mahajan, Prajakta Nimbhorkar, and Kasturi Varadarajan.
The planar k-means problem is NP-hard.
In *International Workshop on Algorithms and Computation*, pages 274–285. Springer, 2009.

Dennis Wei.
A constant-factor bi-criteria approximation guarantee for k-means++.
In *Advances in Neural Information Processing Systems*, pages 604–612, 2016.