Euclidean k-Means with α-Center Proximity

Apoorv Vikram Singh
Joint-work with Amit Deshpande (Microsoft Research India)
and Anand Louis (Indian Institute of Science)

February 5, 2020

Clustering Definition?

- Intuitively, clustering is a task of grouping objects such that:
  1. Similar objects end up in the same group
  2. Dissimilar objects end up in different groups

Clustering Definition?

- Intuitively, clustering is a task of grouping objects such that:
  1. Similar objects end up in the same group
  2. Dissimilar objects end up in different groups
- Not clear how to come up with a more rigorous definition.

Clustering Definition?

- Intuitively, clustering is a task of grouping objects such that:
  1. Similar objects end up in the same group
  2. Dissimilar objects end up in different groups

- Not clear how to come up with a more rigorous definition.

Cluster sharing is a transitive relation.

Similarity need not be a transitive relation.

Clustering Definition?

- Intuitively, clustering is a task of grouping objects such that:
  1. Similar objects end up in the same group
  2. Dissimilar objects end up in different groups

- Not clear how to come up with a more rigorous definition.

Cluster sharing is a transitive relation.

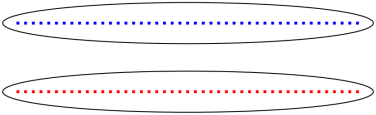
Similarity need not be a transitive relation.

$$x_1, x_2 \in C_1 \text{ and } x_2, x_3 \in C_1 \implies x_1, x_3 \in C_1$$

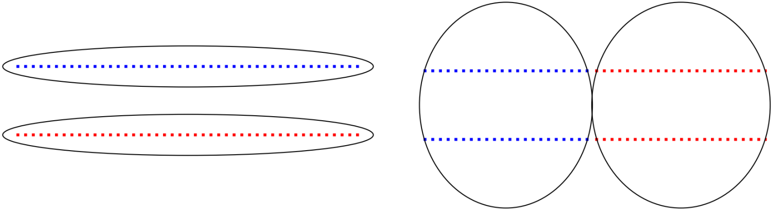
$$x_1 \; x_2 \; \text{-----------------} \; x_{n-1} \; x_n$$

- Another Problem: Lack of “ground-truth” clustering (there is no absolute success evaluation)

- Another Problem: Lack of “ground-truth” clustering (there is no absolute success evaluation)



- Another Problem: Lack of "ground-truth" clustering (there is no absolute success evaluation)



- Popular Approach: Define a cost-function over a parametrized set of possible partitions.
  - Goal: Find a partitioning that outputs minimum-cost clustering.
- Popular Cost-Functions: k-means, k-medians, k-centers, etc.

- Popular Approach: Define a cost-function over a parametrized set of possible partitions.
- Goal: Find a partitioning that outputs minimum-cost clustering.
- Popular Cost-Functions: k-means, k-medians, k-centers, etc.

- Today we will focus on k-means cost-function.

k-Means Clustering

For a k-clustering $C_1, ..., C_k$ of the data, k-means cost-function measures the squared distance between each point to the centroid $\mu(C_i)$ of its cluster:

$$\sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu(C_i)\|^2 .$$

Goal: Find a k-clustering such that the k-means cost is minimized.

Existing k-Means Result

- [Inaba et al., 1994]: PTAS when both k and dimension d are constant.
- [Kumar et al., 2004]: PTAS when k is a constant.
- [Cohen-Addad et al., 2016, Friggstad et al., 2016]: PTAS when d is a constant.
- [Ahmadian et al., 2017]: $6.375 + \varepsilon$-approximation algorithm for k-means.

Hardness of k-Means

- [Aloise et al., 2009] [Dasgupta, 2008] [Mahajan et al., 2012] Optimizing the k-means objective is NP-Hard in the worst case (even for $k = 2$ or $d = 2$).

- [Awasthi et al., 2015] There exists an $\varepsilon > 0$ such that it is NP-Hard to find a clustering which approximates the optimal k-means cost within a factor of $(1 + \varepsilon)$.

Lloyd's Algorithm

1. Start wit k-centers $\mu_1, ..., \mu_k$ chosen uniformly at random from the data.
2. Assign all the points to their closest center.
3. Update the centers to the centroid of all the points assigned to it.
4. Repeat until centers do not change.

Lloyd's Algorithm

1. Start wit k-centers $\mu_1, ..., \mu_k$ chosen uniformly at random from the data.
2. Assign all the points to their closest center.
3. Update the centers to the centroid of all the points assigned to it.
4. Repeat until centers do not change.

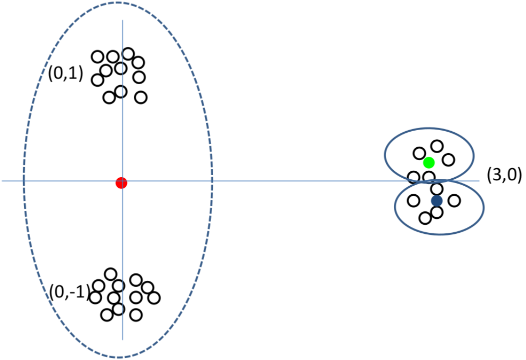
- Motivation: Parallel Axis Theorem (for a single cluster):

$$\sum_{x \in C} \left\| x - \mu' \right\|^2 = \sum_{x \in C} \left\| x - \mu(C) \right\|^2 + |C| \left\| \mu' - \mu(C) \right\|^2.$$

Lloyd's can be bad

- Cost of clustering generated by Lloyd's algorithm can be arbitrarily bad.
- Known worst-case instances where the Lloyd's algorithm can take exponentially many iterations to converge to a local optimum.
  - [Arthur et al., 2011] Lloyd's algorithm has a smoothed running time polynomial in n.

Lloyd's algorithm behaving bad



In practice ...

- Clustering algorithms like the Lloyd's algorithm, k-means++ algorithm works well on real-world data-sets.

- This dichotomy between the theoretical intractability and the empirical observations has lead to the CDNM hypothesis:

  Clustering is Difficult only when it does Not Matter.

  [Daniely, Linial, and Saks, 2012]

Lloyd's Guarantee

- [Arthur and Vassilvitskii, 2007] Initializing using D^2-sampling followed by Lloyd's iteration gives a $\mathcal{O}(\log k)$-approximation.

- [Kumar and Kannan, 2010] For separable data, centers given by a constant factor approximation to k-means on a "sketch" of data, followed by Lloyd's iteration gives an exact solution.

- [Chaudhuri et al., 2009] Lloyd's algorithm work well for mixtures of two Gaussians.

Why does clustering work well on real-world data?

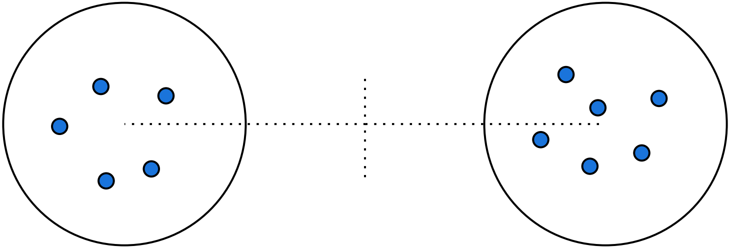
- In most real-world data, the underlying “ground-truth” clustering is unambiguous and is “stable” under small perturbations of the data.

Why does clustering work well on real-world data?

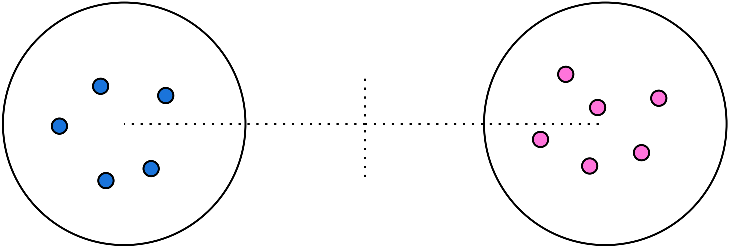
- In most real-world data, the underlying “ground-truth” clustering is unambiguous and is “stable” under small perturbations of the data.
- This kind of phenomenon has lead to the study of “beyond worst-case analysis” in the TCS community.

Intuition of a “stable” instance

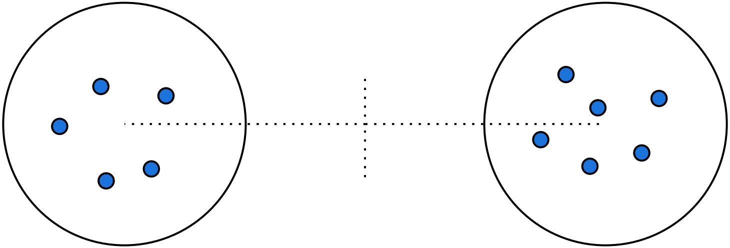
Formalizing the Intuition



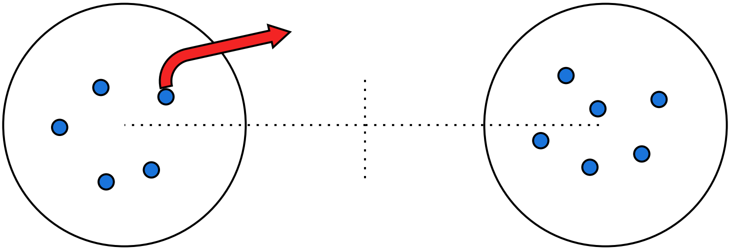
Formalizing the Intuition



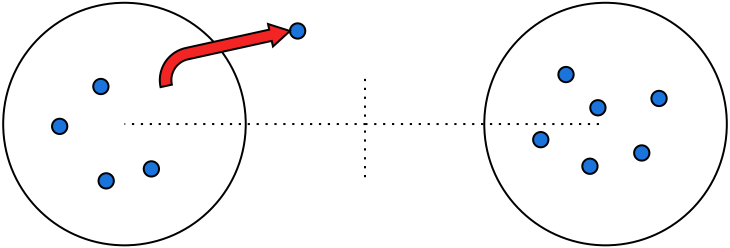
Formalizing the Intuition



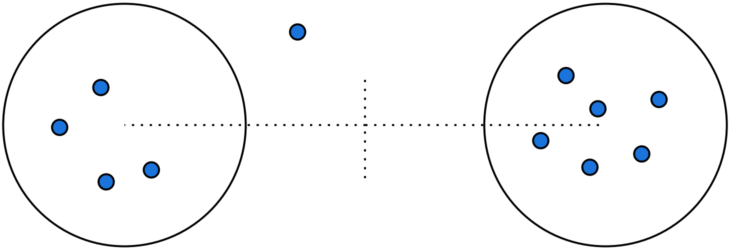
Formalizing the Intuition



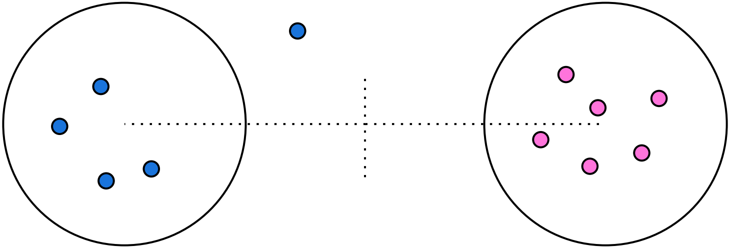
Formalizing the Intuition



Formalizing the Intuition



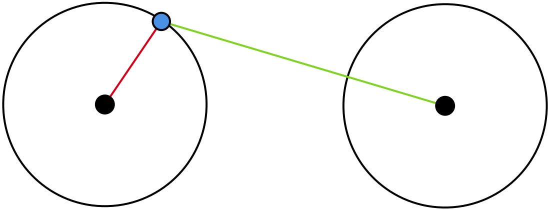
Formalizing the Intuition



Formally: Center Proximity

- Proposed by [Awasthi et al., 2012].
- A clustering $C_1, \ldots, C_k$ with centers $\mu_1, \ldots, \mu_k$ is called α-center proximal if

$$\forall x \in C_i,\ \boldsymbol{\alpha} \left\| x - \mu_i \right\| < \left\| x - \mu_j \right\|,\ i \neq j.$$



Value of α in real-world data

- $\alpha \approx 1.12$

Dataset	$\alpha \geqslant 1.04$	$\alpha \geqslant 1.06$	$\alpha \geqslant 1.08$	$\alpha \geqslant 1.1$	$\alpha \geqslant 1.12$
Wine (k++)	1	0.994	0.989	0.989	0.978
Wine (k++ - pruned)	1	1	1	1	1
Wine (GT)	1	0.994	0.989	0.989	0.978
Wine (GT - pruned)	1	1	1	1	1
Iris (k++)	0.993	0.993	0.993	0.98	0.98
Iris (k++ - pruned)	1	1	1	1	1
Iris (GT)	0.993	0.993	0.987	0.987	0.98
Iris (GT - pruned)	1	1	1	1	1
Banknote Auth. (k++)	0.989	0.985	0.98	0.976	0.97
Banknote Auth. (k++ - pruned)	0.999	0.999	0.998	0.997	0.992
Banknote Auth. (GT)	0.989	0.985	0.98	0.976	0.97
Banknote Auth. (GT - pruned)	0.999	0.999	0.998	0.997	0.992

- $\alpha \approx 1.025$

Dataset	$\alpha \geqslant 1.017$	$\alpha \geqslant 1.019$	$\alpha \geqslant 1.021$	$\alpha \geqslant 1.023$	$\alpha \geqslant 1.025$
Letter Rec. (k++)	0.966	0.962	0.957	0.952	0.948
Letter Rec. (k++ - pruned)	0.995	0.994	0.994	0.994	0.994
Letter Rec. (GT)	0.964	0.96	0.954	0.949	0.945
Letter Rec. (GT - pruned)	0.995	0.994	0.994	0.994	0.993

Previous Result

[Angelidakis et al., 2017] Can cluster in polynomial time if

1. $\alpha \geq 2$.
2. The clustering giving the optimal cost solution must be α-center proximal.

Comments about existing results

- $\alpha \geq 2$ is unrealistic. Real-world data doesn't satisfy that.
- [Ben-David, 2018] Clustering giving the optimal-cost solution need not be the most "stable" clustering.
  - All previous works assume that the optimal-cost clustering is the most stable clustering.
- In practice, people don't care about the optimal-cost solution. They look for the "ground-truth" clustering.

“Desirable” properties of ground-truth clustering

- The ground-truth clustering must be the most stable clustering, i.e., the clustering with the maximum value of α.

“Desirable” properties of ground-truth clustering

- The ground-truth clustering must be the most stable clustering, i.e., the clustering with the maximum value of α.
- The clusters must be roughly balanced, i.e., the ratio of the size of largest cluster to the size of smallest cluster must be a constant.

Our Algorithmic Result

- Aim: Given a value of α, output a k-clustering such that the clustering is α-center proximal. Moreover, the clusters must be roughly balanced.

Theorem

Suppose there exists a k-clustering with roughly-balanced clusters which is α-center proximal. Our algorithm can output such a clustering with constant probability in time $\mathcal{O}\left(nd2^{poly(k/(\alpha-1))}\right)$.

Our Algorithmic Result

- Aim: Given a value of α, output a k-clustering such that the clustering is α-center proximal. Moreover, the clusters must be roughly balanced.

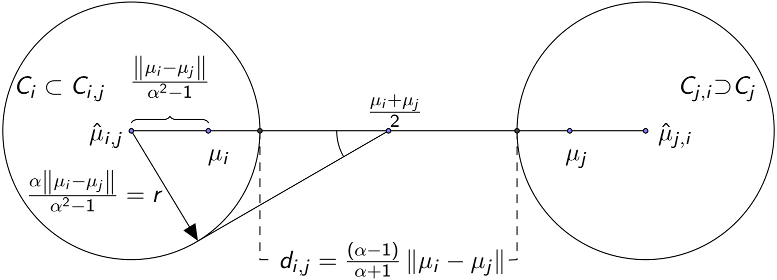
Theorem

Suppose there exists a k-clustering with roughly-balanced clusters which is α-center proximal. Our algorithm can output such a clustering with constant probability in time $\mathcal{O}\left(nd2^{poly(k/(\alpha-1))}\right)$.

- Comment: In real-world data the value of α is not known. We can iterate over the values of α.

Proof Sketch

- $\alpha \|x - \mu_i\| < \|x - \mu_j\| \implies \left\| x - \frac{\alpha^2 \mu_i - \mu_j}{(\alpha^2 - 1)} \right\|^2 < \frac{\alpha^2 \|\mu_i - \mu_j\|^2}{(\alpha^2 - 1)}.$

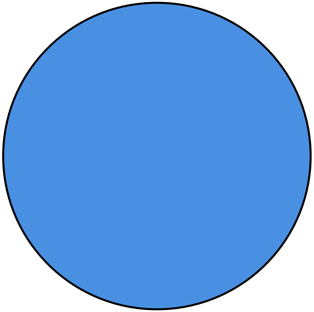


- All clusters lie in a bounded radius.
- Can make an error in estimating the mean.

Theorem (Sampling)

Sample points uniformly at random from a cluster of bounded radius. Mean of the sample is close to the cluster mean.

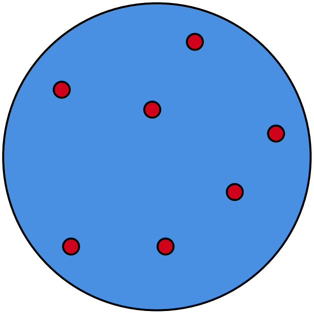
- Remark: Closeness depends on the number of points sampled. It is independent of the dimension d.



Theorem (Sampling)

Sample points uniformly at random from a cluster of bounded radius. Mean of the sample is close to the cluster mean.

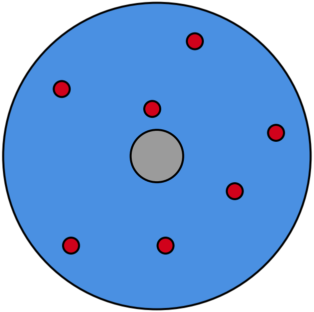
- Remark: Closeness depends on the number of points sampled. It is independent of the dimension d.



Theorem (Sampling)

Sample points uniformly at random from a cluster of bounded radius. Mean of the sample is close to the cluster mean.

- Remark: Closeness depends on the number of points sampled. It is independent of the dimension d.



Algorithm

1. Sample poly$(k/(\alpha - 1))$ points uniformly at random.
   - Since the desired clusters are roughly balanced, we get points from all the clusters.

Algorithm

1. Sample poly$(k/(\alpha - 1))$ points uniformly at random.
   - Since the desired clusters are roughly balanced, we get points from all the clusters.
2. Go over all k-partitions of the sampled points and estimate the k means.
   - At least one partitioning corresponds to the actual clustering, and one set the means is close to the true set of means.

Algorithm

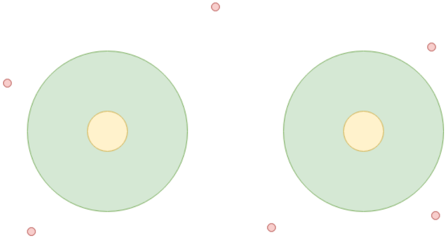
1. Sample poly$(k/(\alpha - 1))$ points uniformly at random.
   - Since the desired clusters are roughly balanced, we get points from all the clusters.
2. Go over all k-partitions of the sampled points and estimate the k means.
   - At least one partitioning corresponds to the actual clustering, and one set the means is close to the true set of means.
3. Cluster according to the sampled means, and output the lowest-cost α-center proximal clustering.

Class of Outliers

Let Z be the set of outliers, and suppose we know $|Z|$.

Definition

For $x \in C_i, \alpha \|x - \mu_i\| < \|x - \mu_j\|$. Moreover, for $x \in C_i$ and $z \in Z$, we have $\alpha \|x - \mu_i\| < \|z - \mu_j\|$.

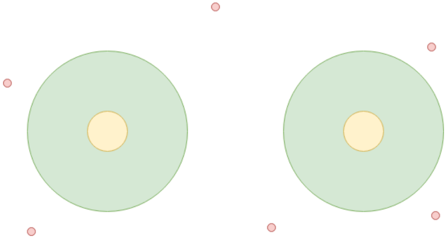


Class of Outliers

Let Z be the set of outliers, and suppose we know $|Z|$.

Definition

For $x \in C_i, \alpha \|x - \mu_i\| < \|x - \mu_j\|$. Moreover, for $x \in C_i$ and $z \in Z$, we have $\alpha \|x - \mu_i\| < \|z - \mu_j\|$.



- Algorithm essentially remains the same. Go over $k + 1$ partitions of the sampled points and remove the farthest $|Z|$ points after clustering.

Lower Bound

- [Ben-David and Reyzin, 2014] NP-Hard to cluster for $\alpha < 2$ in general metrics.
- $\exists \alpha, \varepsilon$ such that it is NP-Hard to find a clustering which approximate the optimal α-center proximal Euclidean k-means, where the clusters are roughly balanced, to a factor better than $(1 + \varepsilon)$.
- $\exists \alpha$ for which we can construct an instance such that the total number of optimal, balanced α-center proximal clusterings are $2^{\text{poly}(k/(\alpha-1))}$.

Discussions

- We show results for unbalanced clusters as well.
- Can be extended to k-median, or to any objective where the approximate centers of a cluster can be decided by sampling uniformly at random.
- Can be adapted to a setting with "same-cluster queries", with $\mathcal{O}\left(k^4 \log k/(\alpha - 1)\right)$ queries, in time $\mathcal{O}\left(ndk\right)$.
- This kind of technique is very general, and can be extended to other problems like cost-balanced clustering, topic modelling, fair clustering, etc.

References I

Ahmadian, S., Norouzi-Fard, A., Svensson, O., and Ward, J. (2017).
Better guarantees for k-means and euclidean k-median by primal-dual algorithms.
In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 61–72.

Aloise, D., Deshpande, A., Hansen, P., and Popat, P. (2009).
Np-hardness of euclidean sum-of-squares clustering.
Machine Learning, 75(2):245–248.

Angelidakis, H., Makarychev, K., and Makarychev, Y. (2017).
Algorithms for stable and perturbation-resilient problems.
In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 438–451. ACM.

Arthur, D., Manthey, B., and Röglin, H. (2011).
Smoothed analysis of the k-means method.
J. ACM, 58(5).

Arthur, D. and Vassilvitskii, S. (2007).
K-means++: The advantages of careful seeding.
In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035. Society for Industrial and Applied Mathematics.

Awasthi, P., Blum, A., and Sheffet, O. (2012).
Center-based clustering under perturbation stability.
Information Processing Letters, 112(1):49 – 54.

Awasthi, P., Charikar, M., Krishnaswamy, R., and Sinop, A. K. (2015).
The Hardness of Approximation of Euclidean k-Means.
In *31st International Symposium on Computational Geometry (SoCG 2015)*, volume 34 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 754–767. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

References II

Ben-David, S. (2018).
Clustering - what both theoreticians and practitioners are doing wrong.
In *Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI Publications.

Ben-David, S. and Reyzin, L. (2014).
Data stability in clustering: A closer look.
Theoretical Computer Science, 558:51 – 61.
Algorithmic Learning Theory.

Chaudhuri, K., Dasgupta, S., and Vattani, A. (2009).
Learning mixtures of gaussians using the k-means algorithm.
CoRR, abs/0912.0086.

Cohen-Addad, V., Klein, P. N., and Mathieu, C. (2016).
Local search yields approximation schemes for k-means and k-median in euclidean and minor-free metrics.
In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 353–364.

Dasgupta, S. (2008).
The hardness of k-means clustering.

Friggstad, Z., Rezapour, M., and Salavatipour, M. R. (2016).
Local search yields a ptas for k-means in doubling metrics.
In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 365–374.

Inaba, M., Katoh, N., and Imai, H. (1994).
Applications of weighted voronoi diagrams and randomization to variance-based k-clustering: (extended abstract).
In *Proceedings of the Tenth Annual Symposium on Computational Geometry*, SCG '94, pages 332–339. ACM.

References III

Kumar, A. and Kannan, R. (2010).
Clustering with spectral norm and the k-means algorithm.
In *51th Annual IEEE Symposium on Foundations of Computer Science (FOCS) 2010, October 23-26, 2010*, pages 299–308.

Kumar, A., Sabharwal, Y., and Sen, S. (2004).
A simple linear time $(1+\varepsilon)$-approximation algorithm for k-means clustering in any dimensions.
In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 454–462.

Mahajan, M., Nimbhorkar, P., and Varadarajan, K. (2012).
The planar k-means problem is np-hard.
Theoretical Computer Science, 442:13 – 21.
Special Issue on the Workshop on Algorithms and Computation (WALCOM 2009).