Approximate Clustering with Same-Cluster Queries

Ragesh Jaiswal

CSE, IIT Delhi

ITCS Talk, January 11, 2018

[Joint work with Nir Ailon (Technion), Anup Bhattacharya (IITD), and Amit Kumar (IITD)]

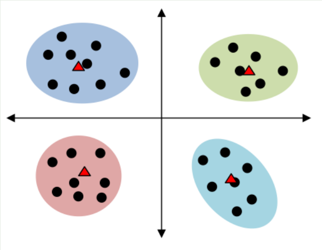
k-means Clustering

Problem (k-means)

Given n points $X \subset \mathbb{R}^d$, and an integer k, find k points $C \subset \mathbb{R}^d$ (called centers) such that the sum of squared Euclidean distance of each point in X to its closest center in C is minimized. That is, the following cost function is minimized:

$$\Phi(C, X) = \sum_{x \in X} \min_{c \in C} \left(||x - c||^2 \right)$$

Example: $k = 4$, $d = 2$



- Lower bounds:
    - The problem is NP-hard when $k \geq 2, d \geq 2$ [Das08, MNV12, Vat09].
    - Theorem [ACKS15]: There is a constant $\epsilon > 0$ such that it is NP-hard to approximate the k-means problem to a factor better than $(1 + \epsilon)$.

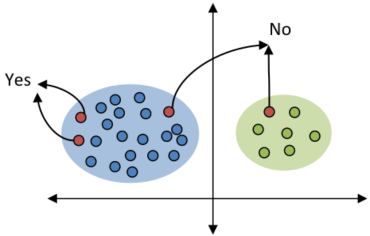
- Lower bounds:
  - The problem is NP-hard when $k \geq 2, d \geq 2$ [Das08, MNV12, Vat09].
  - Theorem [ACKS15]: There is a constant $\epsilon > 0$ such that it is NP-hard to approximate the k-means problem to a factor better than $(1 + \epsilon)$.
- Upper bounds: There are various approximation algorithms for the k-means problem.

Citation	**Approx. factor**	**Running Time**
[AV07]	$O(\log k)$	polynomial time
[KMN+02]	$9 + \epsilon$	polynomial time
[KSS10, JKY15, FMS07]	$(1 + \epsilon)$	$O\left(nd \cdot 2^{\tilde{O}(k/\epsilon)}\right)$

- Various results of "*beyond worst-case*" flavour have been attempted in the context of the k-means and clustering problems in general.
  - Mixture of Gaussians.
  - Clustering under separation assumptions on the dataset. The working philosophy is that a dataset is clusterable only when it satisfies some separation.
    - ORSS separation [ORSS13]
    - BBG approximate stability [BBG13]
    - ...

- "*Beyond worst-case*"
  - Mixture of Gaussians.
  - Clustering under separation.
  - Clustering in semi-supervised setting where the clustering algorithm is allowed to make "*queries*" during its execution.

- "*Beyond worst-case*"
    - Mixture of Gaussians.
    - Clustering under separation.
    - Clustering in semi-supervised setting where the clustering algorithm is allowed to make "*queries*" during its execution.
        - Semi-Supervised Active Clustering (SSAC) [AKBD16]: The clustering algorithm is given the dataset $X \subset \mathbb{R}^d$ and integer k (as in the classical setting) and it can make same-cluster queries.



- SSAC framework: Same-cluster queries.
  - A limited number of such queries (or some weaker version) may be feasible in certain settings.
  - So, understanding the power and limitations of this idea may open interesting future directions.

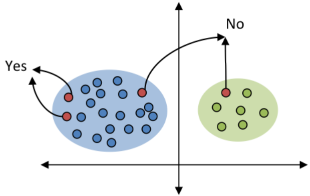


Figure: SSAC framework: same-cluster queries

- Clearly, we can output optimal clustering using $O(n^2)$ same-cluster queries. Can we cluster using fewer queries?
- The following result is already known for the SSAC setting.

Theorem (Informally stated theorem from [AKBD16])

There is a randomised algorithm that runs in time $O(kn\log n)$ and makes $O(k^2\log k + k\log n)$ same-cluster queries and returns the optimal clustering for a dataset that satisfies some separation guarantee.

Theorem (Informally stated theorem from [AKBD16])

There is a randomised algorithm that runs in time $O(kn \log n)$ and makes $O(k^2 \log k + k \log n)$ same-cluster queries and returns the optimal clustering for a dataset that satisfies some separation guarantee.

- A few things to note about the above result:
  - This is an exact clustering result.
  - The result holds given that the input datasets satisfies a separation guarantee.
  - Finally, the number of same-cluster queries is not independent of the data size n.

Theorem (Informally stated theorem from [AKBD16])

There is a randomised algorithm that runs in time $O(kn \log n)$ and makes $O(k^2 \log k + k \log n)$ same-cluster queries and returns the optimal clustering for a dataset that satisfies some separation guarantee.

- A few things to note about the above result:
  - This is an exact clustering result.
  - The result holds given that the input datasets satisfies a separation guarantee.
  - Finally, the number of same-cluster queries is not independent of the data size n.
- Our contributions (informal):
  - We extend the theory to the approximation setting while removing the separation requirement.
  - We give bounds on the number of same-cluster queries which interestingly is independent of data size n.
  - We extend our results to a faulty-query setting where the answers to same-cluster queries may be incorrect. This is a more reasonable setting.

- Our contributions (informal):
  - We extend the theory to the approximation setting while removing the separation requirement.
  - We give bounds on the number of same-cluster queries which interestingly is independent of data size n.
  - We extend our results to a faulty-query setting where the answers to same-cluster queries may be incorrect. This is a more reasonable setting.

Theorem (Main result)

Let $0 < \varepsilon < 1/2$. There is a randomised query algorithm that returns a $(1+\varepsilon)$ approximate clustering for any given dataset. The algorithm runs in time $O(nd \cdot poly(k/\varepsilon))$ makes $poly(k/\varepsilon)$ same-cluster queries.

- Our contributions (informal):
  - We extend the theory to the approximation setting while removing the separation requirement.
  - We give bounds on the number of same-cluster queries which interestingly is independent of data size n.
  - We extend our results to a faulty-query setting where the answers to same-cluster queries may be incorrect. This is a more reasonable setting.

Theorem (Main result)

Let $0 < \varepsilon < 1/2$. There is a randomised query algorithm that returns a $(1 + \varepsilon)$ approximate clustering for any given dataset. The algorithm runs in time $O(nd \cdot poly(k/\varepsilon))$ makes $poly(k/\varepsilon)$ same-cluster queries.

Theorem (Main result - query lower bound)

If ETH holds, then there exists a constant $c > 1$ such that any c-approximation query algorithm that runs in time $poly(n, k, d)$ makes at least $k/polylog(k)$ same-cluster queries.

- Our contributions (informal):
  - We extend the theory to the approximation setting while removing the separation requirement.
  - We give bounds on the number of same-cluster queries which interestingly is independent of data size n.
  - We extend our results to a faulty-query setting where the answers to same-cluster queries may be incorrect. This is a more reasonable setting.

Theorem (Main result)

Let $0 < \varepsilon < 1/2$. There is a randomised query algorithm that returns a $(1+\varepsilon)$ approximate clustering for any given dataset. The algorithm runs in time $O(nd \cdot poly(k/\varepsilon))$ makes $poly(k/\varepsilon)$ same-cluster queries.

- The above result can be extended to a setting where the response to every same-cluster query is incorrect with probability at most $q < 1/2$.

Theorem (Main result - query lower bound)

If ETH holds, then there exists a constant $c > 1$ such that any c-approximation query algorithm that runs in time $poly(n, k, d)$ makes at least $k/polylog(k)$ same-cluster queries.

Main ideas for Query Algorithm

Lemma ([IKI94])

Let S be a set of s point sampled independently from any given point set $X \subset \mathbb{R}^d$ uniformly at random. Then for any $\delta > 0$, the following holds with probability at least $(1 - \delta)$:

$$\Phi(\Gamma(S), X) \leq \left(1 + \frac{1}{\delta \cdot s}\right) \cdot \Phi(\Gamma(X), X), \;\; \textit{where } \Gamma(X) = \frac{\sum_{x \in X} x}{|X|}$$

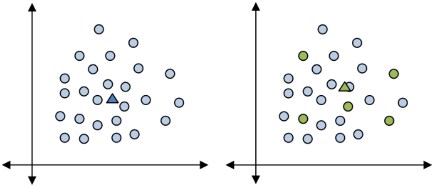
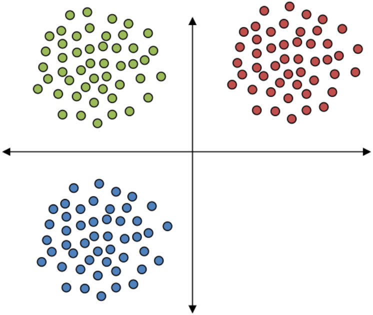


Figure: The cost w.r.t. the centroid (blue triangle) of all points (blue dots) is close to the cost w.r.t. the centroid (green triangle) of a few randomly chosen points (green dots).

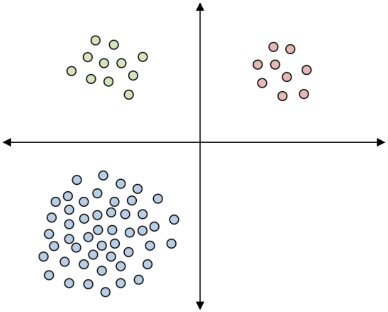
Query Algorithm
Main idea

- Easy case: The optimal clusters have roughly the same size.

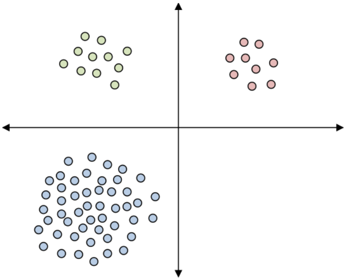


- The query algorithm samples $poly(k/\varepsilon)$ points uniformly from the dataset and uses same-cluster queries to partition them into subsets of optimal clusters.
- The mean of the partitions will be good centers using [IKI94] lemma since each partition contains $\Omega(1/\varepsilon)$ points.

- The query algorithm samples $poly(k/\varepsilon)$ points uniformly from the dataset and uses same-cluster queries to partition them into subsets of optimal clusters.
- The mean of the partitions will be good centers using [IKI94] lemma since each partition contains $\Omega(1/\varepsilon)$ points.
- The above idea fails if some clusters are small compared to other clusters as below.



- Difficult (general) case: Some clusters are small compared to other clusters.



- Main idea: After finding the first center using uniform sampling find subsequent centers using D^2-sampling.
  - D^2-sampling: Biased sampling that gives preference to points that are far from the already chosen centers.

Other Results

- Query lower bounds:
  - The query lower bound is obtained by showing that any approximation algorithm whose running time has polynomial dependence on n and d will have exponential dependence on k.
  - $ETH \rightarrow PCP \rightarrow$ VC in triangle-free graphs $\overset{\text{[ACKS15]}}{\rightarrow}$ k-means.
- Faulty-query setting:
  - Crucially uses results about recovering clusters in Stochastic Block Model.
- k-means++ variant:
  - The ideas also gives very simple constant factor approximation algorithms in the SSAC framework using the well-known k-means++ algorithm.

Future Directions

- Future directions:
  - Gap in query upper and lower bounds.
  - Faulty-query setting.

References I

Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop, *The hardness of approximation of euclidean k-means*, CoRR **abs/1502.03316** (2015).

Hassan Ashtiani, Shrinu Kushagra, and Shai Ben-David, *Clustering with same-cluster queries*, Advances in neural information processing systems, 2016, pp. 3216–3224.

David Arthur and Sergei Vassilvitskii, *k-means++: the advantages of careful seeding*, Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (Philadelphia, PA, USA), SODA '07, Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

Maria-Florina Balcan, Avrim Blum, and Anupam Gupta, *Clustering under approximation stability*, J. ACM **60** (2013), no. 2, 8:1–8:34.

Sanjoy Dasgupta, *The hardness of k-means clustering*, Tech. Report CS2008-0916, Department of Computer Science and Engineering, University of California San Diego, 2008.

Dan Feldman, Morteza Monemizadeh, and Christian Sohler, *A PTAS for k-means clustering based on weak coresets*, Proceedings of the twenty-third annual symposium on Computational geometry (New York, NY, USA), SCG '07, ACM, 2007, pp. 11–18.

Mary Inaba, Naoki Katoh, and Hiroshi Imai, *Applications of weighted voronoi diagrams and randomization to variance-based k-clustering: (extended abstract)*, Proceedings of the tenth annual symposium on Computational geometry (New York, NY, USA), SCG '94, ACM, 1994, pp. 332–339.

Ragesh Jaiswal, Mehul Kumar, and Pulkit Yadav, *Improved analysis of D^2-sampling based PTAS for k-means and other clustering problems*, Information Processing Letters **115** (2015), no. 2.

References II

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*, Proc. 18th Annual Symposium on Computational Geometry, 2002, pp. 10–18.

Amit Kumar, Yogish Sabharwal, and Sandeep Sen, *Linear-time approximation schemes for clustering problems in any dimensions*, J. ACM **57** (2010), no. 2, 5:1–5:32.

Meena Mahajan, Prajakta Nimbhorkar, and Kasturi Varadarajan, *The planar k-means problem is NP-hard*, Theoretical Computer Science **442** (2012), no. 0, 13 – 21, Special Issue on the Workshop on Algorithms and Computation (WALCOM 2009).

Rafail Ostrovsky, Yuval Rabani, Leonard J. Schulman, and Chaitanya Swamy, *The effectiveness of lloyd-type methods for the k-means problem*, J. ACM **59** (2013), no. 6, 28:1–28:22.

Andrea Vattani, *The hardness of k-means clustering in the plane*, Tech. report, Department of Computer Science and Engineering, University of California San Diego, 2009.

Thank you