Greedy Sampling for Clustering in the Presence of Outliers

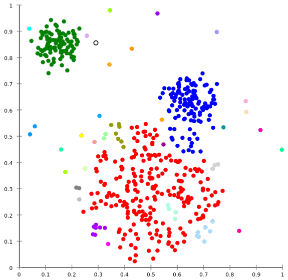
Aditya Bhaskara, Sharvaree Vadgama and Hong Xu

NeurIPS 2019

Overview

Clustering

One of the fundamental tasks in data analysis



Tale of many formulations: k-means, k-center, k-median, hierarchical clustering, ...

Focus of the paper: clustering when data has **outliers**

Definitions: k-center

Problem

Given points X in a metric space, find a set C with k "centers" so as to minimize

$$\max_{u \in X} d(u, C) \qquad \left[\text{Recall: } d(u,C) := \min_{c \in C} d(u,c)\right]$$

- Find least r so that every point in X is dist $\leq r$ from some point in C
- Gonzales algorithm. "Furthest point traversal". Iteratively add point in X furthest from current centers
- Known to be a factor 2 approximation

Definitions: k-means

Problem

Given points X in a metric space, find a set C with k "centers" so as to minimize

$$\sum_{u \in X} d(u,C)^2 \qquad \left[\text{Recall: } d(u,C) := \min_{c \in C} d(u,c)\right]$$

- Classic problem; best approximation factor $\approx 6.357..$
- Lot of literature on heuristics. Lloyd's "k-means" algorithm (unfortunately no guarantees)
- k-means++: decent worst-case bound of $O(\log n)$, good initializer for Lloyd's ["Smooth" analog of furthest point traversal: add points w.p. $\propto d(u,C)^2$]

What if we have outliers?

What if some of the data points are outliers?

Suppose outliers are far away from true clusters..

- Furthest point traversal can be really bad! (only picks outliers)
- k-means++ places most of prob mass on outliers

Greedy sampling algorithms simple & effective, but **not robust**

Main result: Simple modifications of these algorithms lead to *guarantees* when data has outliers

Formulations

Clustering with outliers

Suppose the input $X = X_{\text{in}} \cup X_{\text{out}}$ (unknown partition), and suppose $|X_{\text{out}}| \leq z$, for some parameter z. Given X, find partition $X = X'_{\text{in}} + X'_{\text{out}}$ with $|X'_{\text{out}}| \leq z$, s.t. k-clustering objective on X'_{in} is comparable to objective on X_{in}

Note. In some sense the *gold standard*

Common relaxations in applications – *bi-criteria*

- May be fine to regard some more points as outliers ($X'_{\text{in}} = O(z)$)
- Might also be OK to return $> k$ centers

Prior work: robust clustering

1. Very well studied problem (given ubiquity of clustering)
2. k-center with outliers classic problem
3. k-means/median – only bi-criteria known until recently
4. Recent result [Krishnaswamy et al. 2018]: can obtain constant factor approximation (no loss in k, z)

Problem solved? Yes in theory, but algorithms complicated;
Can iterative greedy methods be made robust?

Main results: k-center

Algorithm: robust furthest point traversal

1. Guess r (optimum value), initizalize $S = \emptyset$
2. For k iterations: add $u \in X$ to S, where u is a *random* point in $X \setminus B(S, r)$

I.e., add *random point* not-too-close to current set

Theorems – bi-criteria guarantees

- Given dataset X and bound z on #(outliers), algorithm obtains 2-approx to objective, and violates constraint on z by a factor $(\log n)$
- If allowed to pick ck centers, we get 2-approx to objective, violate bound on z by factor $(c + 1)/c$

Main Results: k-means

Algorithm: **thresholded** k-means++

For k iterations: add $u \in X$ to S, with probability

$$p_u \propto \min\{\beta, d(u, S)^2\}$$

Theorems – bi-criteria guarantees

For appropriate choice of β, we have

- Set of centers obtained give $O(\log n)$ approximation to k-means objective, while violating bound on z by factor $O(\log n)$
- If allowed to pick ck centers, we get $O(1)$ approximation to objective, with $\approx (1+c)/c$ violation in bound on z

Remarks

- Algorithms simple modifications of original greedy methods
- Theorems generalize the “non-robust” versions
- Trade-offs between #(centers) and violation of z

- Proofs based on *potential function* arguments

Outline of Proofs: k-center

The key step is to define the appropriate potential function. To this end, let w_t denote the number of times that one of the outliers was added to the set S in the first t iterations. I.e., $w_t = |X_{\text{out}} \cap S_t|$. The potential we consider is now:

$$\Psi_t := \frac{w_t |\mathcal{F}_t \cap X_{\text{in}}|}{n_t}. \tag{1}$$

Lemma

Let S_t be any set of centers chosen in the first t iterations, for some $t \geq 0$. We have

$$\mathbb{E}_{t+1}\left[\Psi_{t+1} - \Psi_t \mid S_t\right] \leq \frac{z}{n_t}.$$

Outline of Proofs: k-means

For any set of centers C, we define

$$\tau(x, C) = \min\left(d(x, C)^2, \frac{\beta \cdot \textsc{Opt}}{z}\right) \tag{2}$$

The key to the analysis is the observation that instead of attempting to bound the k-means objective, it suffices to bound the quantity $\sum_{x \in X} \tau(x, S_\ell)$.

Outline of Proofs: k-means

Lemma

Let C be a set of centers, and suppose that $\tau(X, C) \leq \alpha \cdot \text{OPT}$. Then we can partition X into X'_{in} and X'_{out} such that

1. $\sum_{x \in X'_{in}} d(x, C)^2 \leq \alpha \cdot \text{OPT}$, *and*
2. $|X'_{out}| \leq \frac{\alpha z}{\beta}$.

Theorem

Running T-kmeans++ *for k iterations outputs a set S_k that satisfies*

$$\mathbb{E}[\tau(X, S_k)] \leq (\beta + O(1)) \log k \cdot \text{OPT}.$$

Outline of Proofs: k-means

Theorem

Consider running T-kmeans++ *for* $\ell = (1+c)k$ *iterations, where* $c > 0$ *is a constant. Then for any* $\delta > 0$, *with probability* $\geq \delta$, *the set* S_ℓ *satisfies*

$$\tau(X, S_\ell) \leq \frac{(\beta + 64)(1+c)\textsc{Opt}}{(1-\delta)c}.$$

Experiments

K-center experiments on synthetic data

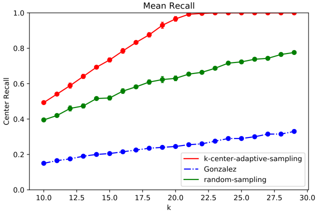


Figure: Cluster recall for the three algorithms, when $k = 20$, $z = 100$ and $n = 10120$. The x axis shows the number of clusters we pick.

Experiments

K-means experiments on synthetic data

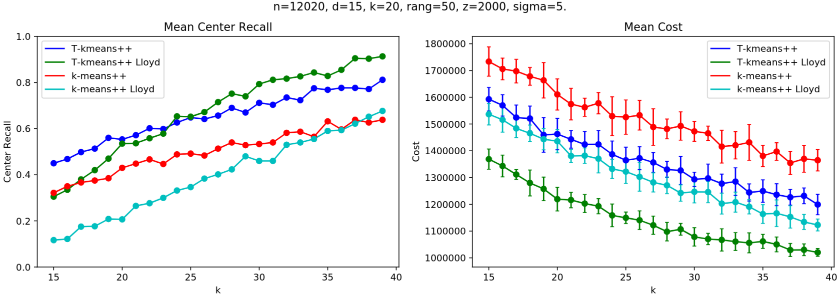


Figure: The empirical cluster recall for the T-kmeans++ algorithm compared to prior heuristics. Here $k = 20, z = 2000, n = 12020$. The x axis shows the number of clusters we pick.

Experiments

K-means experiments on real datasets wherein 2.5% of data is corrupted.

Dataset	k	KM recall	TKM recall	KM objective	TKM objective
NIPS	10	0.960	0.977	4173211	4167724
	20	0.939	0.973	4046443	4112852
	30	0.924	0.978	3956768	4115889
Skin	10	0.619	0.667	7726552	7439527
	20	0.642	0.690	5936156	5637427
	30	0.630	0.690	5164635	4853001
MNIST	10	0.975	0.977	159129783	148848993
	20	0.969	0.974	154588753	142313226
	30	0.968	0.976	150851200	139026059

Table showing outlier recall for KM (k-means++) and TKM (T-kmeans++) along with the k-means cost.