Complementi di Algoritmi e Strutture Dati

k-Means

Docente: *Nicolò Cesa-Bianchi* versione 24 aprile 2025

These lecture notes are based on a set of slides written by Marco Bressan in 2023.

We consider the problem of partitioning a finite set $\mathcal{X} \subset \mathbb{R}^d$ of points in $k > 1$ clusters. Since we are in $\mathbb{R}^d$, we can use the Euclidean distance to measure the similarity between two points. We identify each cluster $i \in \{1, \dots, k\}$ with a center $\boldsymbol{c}_i \in \mathbb{R}^d$ (we do not require that these centers belong to $\mathcal{X}$) and we assign each point $\boldsymbol{x} \in \mathcal{X}$ to its closest center (with respect to the Euclidean distance).

The cost of a point in a clustering $\mathcal{C} = \{\boldsymbol{c}_1, \dots, \boldsymbol{c}_k\}$ is $\phi(\mathcal{C}, \boldsymbol{x}) = \min_{i=1,\dots,k} \|\boldsymbol{x} - \boldsymbol{c}_i\|^2$.

The cost of a clustering $\mathcal{C}$ is $\Phi(\mathcal{C}) = \sum_{\boldsymbol{x} \in \mathcal{X}} \phi(\mathcal{C}, \boldsymbol{x})$.

Note that each point pays the squared distance to its closest center. The optimal k-clustering $\mathcal{C}^*$ of $\mathcal{C}$ is any choice of centers that minimizes the cost,

$$\mathcal{C}^* = \operatorname*{argmin}_{\boldsymbol{c}_1, \dots, \boldsymbol{c}_k \in \mathbb{R}^d} \Phi(\boldsymbol{c}_1, \dots, \boldsymbol{c}_k)$$

Note that the optimal centers need not be unique. We use $\mathrm{OPT}(\mathcal{X})$ to denote the cost of $\mathcal{C}^*$. The k-means problem is, given $\mathcal{X}$ and k, that of finding any $\mathcal{C} \subset \mathbb{R}^d$ with $|\mathcal{C}| = k$ such that $\Phi(\mathcal{C}) = \mathrm{OPT}(\mathcal{X})$.

Note that k-means is trivial for $k = 1$, as there is a unique center $\boldsymbol{c}^*$ minimizing the cost which corresponds to the **centroid** of the set $\mathcal{X}$,

$$\boldsymbol{c}^* = \operatorname*{argmin}_{\boldsymbol{c} \in \mathbb{R}^d} \sum_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{c}\|^2 = \frac{1}{|\mathcal{X}|} \sum_{\boldsymbol{x} \in \mathcal{X}} \boldsymbol{x}$$

This can be shown by noticing that $F(\boldsymbol{c}) = \sum_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{c}\|^2$ is a convex function that is minimized when $\boldsymbol{c}$ is the centroid. This implies that the centers of $\mathcal{C}^*$ are the centroids of their corresponding clusters.

The k-means problem implicitly assumes that the points in $\mathcal{X}$ are sampled from k spherical Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_i, \sigma_i^2 I)$ for $i = 1, \dots, k$ whose means $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k$ are the centers and whose variances $\sigma_1^2, \dots, \sigma_k^2$ upper bounds the optimal cost,

$$\boldsymbol{\mu}_i = \operatorname*{argmin}_{\boldsymbol{c}} \mathbb{E}\big[\, \|\boldsymbol{X} - \boldsymbol{c}\|^2 \,\big] \quad \text{where} \quad \boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}_i, \sigma_i^2 I) \quad \text{and} \quad \mathbb{E}\big[\Phi(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k)\big] \le \sum_{i=1}^k \sigma_i^2$$

It turns out that the k-means problem in $\mathbb{R}^d$ is $\mathcal{NP}$-hard even for $k = 2$ (when $d = 2n$). As a consequence of this result, the best known exact algorithm for solving k-means is based on:

1. enumerating all $k^{|\mathcal{X}|}$ partitions of $\mathcal{X}$ in k elements,

2. computing the centroids $\mathcal{C} = \{\boldsymbol{c}_1, \dots, \boldsymbol{c}_k\}$ for the k elements of the partition,

3. computing the cost $\Phi(\mathcal{C})$ of the partition.

The following algorithm is the most popular heuristic solver for k-means.

Algoritmo 1 Lloyd's Algorithm

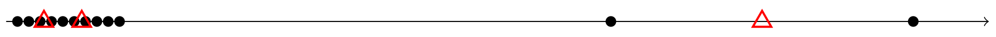
```
Input: Finite set of points $\mathcal{X} \subset \mathbb{R}^d$, integer $1 < k < |\mathcal{X}|$.
1: Draw $k$ points $\boldsymbol{c}_1, \dots, \boldsymbol{c}_k$ u.a.r. from $\mathcal{X}$
2: repeat
3:     for $\boldsymbol{x} \in \mathcal{X}$ do
4:         Assign $\boldsymbol{x}$ to cluster $C_i$ where $i = \operatorname*{argmin}_{j=1,\dots,k} \|\boldsymbol{x} - \boldsymbol{c}_j\|^2$
5:     end for
6:     for $i = 1, \dots, k$ do
7:         $\boldsymbol{c}_i = \frac{1}{|C_i|} \sum_{\boldsymbol{x} \in X_i} \boldsymbol{x}$                ▷ $\boldsymbol{c}_i$ is the centroid of $C_i$
8:     end for
9: until $\boldsymbol{c}_1, \dots, \boldsymbol{c}_k$ remain unchanged
Output: $\boldsymbol{c}_1, \dots, \boldsymbol{c}_k$
```

The per-iteration runnning time of Lloyd's algorithm is $\mathcal{O}(nkd)$. One can use random projections to map $\mathcal{X}$ to $\mathbb{R}^N$ with $N = \Theta(\ln n)$, while blowing up OPT by at most a constant factor. This reduces the running time of each iteration of Lloyd's algorithm to $\mathcal{O}(nk \ln n)$. unfortunately, the worst-case number of iterations of the algorithm is $2^{\Omega(\sqrt{n})}$.

Although Lloyd's algorithm works well in practice, it does not approximate OPT to within any constant factor, as shown by the next result.

Teorema 1 *For any $a > 1$ there exist 1-dimensional instances $\mathcal{X} \subset \mathbb{R}$ of 3-means where Lloyd's algorithm returns a cluster $\mathcal{C}$ such that $\Phi(\mathcal{C}) \geq a\,\mathrm{OPT}$ with probability arbitrarily close to 1.*

DIMOSTRAZIONE. Pick $a > 1$ and let $\mathcal{X}$ of size n be such that $n - 2$ points are evenly spaced in the $[0, 1]$ unit segment and the two remaining points (the outliers) are placed at $2\sqrt{an}$ and $3\sqrt{an}$.



The probability that Lloyd's algorithm does not draw both outliers as initial centers is computed as follows: there are $\binom{n}{3}$ ways of choosing three points in a set of n points. There are $n - 2$ ways of choosing three points when two of which are the outliers. Hence the probability of not drawing both outliers is

$$p_n = 1 - \frac{n-2}{\binom{n}{3}} = 1 - \frac{(n-3)!\,6(n-2)}{n!} = 1 - \frac{6}{n(n-1)}$$

Consider the bad event that Lloyd's algorithm initially draws at most one outlier. Conditioned on this event, Lloyd's algorithm terminates with at least two centers in the $[0, 1]$ segment and at most one center at $\frac{3}{2}\sqrt{an}$. The cost $\Phi(\mathcal{C})$ of this clustering $\mathcal{C}$ is at least $\frac{an}{2}$, while the cost of the optimal cluster (two centers located at the outliers and the remaining center at 1/2) is OPT $= \frac{n-2}{4}$. Therefore, $\Phi(\mathcal{C})/\mathrm{OPT} = \Omega(a)$ As $n \to \infty$, we have $p_n \to 1$, implying that the bad event occurs with arbitrarily high probability. □

We now show that, if the centers are moved in Lloyd's algorithm, then Φ strictly decreases and it can do so for at most $\mathcal{O}(k^n)$ times (the number of possible partitions of $\mathcal{X}$ with $|\mathcal{X}| = n$ in k elements).

Lemma 2 *If in any iteration some center is moved, then Φ decreases strictly.*

Dimostrazione. The proof makes use of the following fact. For any finite $C \subset \mathbb{R}^d$ and for any $\boldsymbol{c} \in \mathbb{R}^d$,

$$\sum_{\boldsymbol{x}\in C} \|\boldsymbol{x} - \boldsymbol{c}\|^2 = \sum_{\boldsymbol{x}\in C} \|\boldsymbol{x} - \boldsymbol{\mu}\|^2 + |C| \; \|\boldsymbol{c} - \boldsymbol{\mu}\|^2 \tag{1}$$

where $\boldsymbol{\mu}$ is the centroid of C. Let $C_1, \dots, C_k, \boldsymbol{c}_1, \dots, \boldsymbol{c}_k$ the clusters and the centers at the beginning of an iteration (Line 2) and let $C'_1, \dots, C'_k, \boldsymbol{c}'_1, \dots, \boldsymbol{c}'_k$ be the clusters and the centers at the end of an iteration (Line 9). Let

$$\psi(C_1, \dots, C_k, \boldsymbol{c}_1, \dots, \boldsymbol{c}_k) = \sum_{i=1}^{k} \sum_{\boldsymbol{x}\in C_i} \|\boldsymbol{x} - \boldsymbol{c}_i\|^2$$

Note that $\psi(C_1, \dots, C_k, \boldsymbol{c}_1, \dots, \boldsymbol{c}_k) \geq \psi(C'_1, \dots, C'_k, \boldsymbol{c}_1, \dots, \boldsymbol{c}_k)$ since Line 4 assigns each point to its nearest center. Now, if $\boldsymbol{c}'_i \neq \boldsymbol{c}_i$ for some i, then

$$\psi(C'_1, \dots, C'_k, \boldsymbol{c}_1, \dots, \boldsymbol{c}_k) > \psi(C'_1, \dots, C'_k, \boldsymbol{c}'_1, \dots, \boldsymbol{c}'_k)$$

To see that, recalling that $\boldsymbol{c}'_i$ is the centroid of C'_i (Line 7),

$$\sum_{\boldsymbol{x}\in C'_i} \|\boldsymbol{x} - \boldsymbol{c}_i\|^2 = \sum_{\boldsymbol{x}\in C'_i} \left\|\boldsymbol{x} - \boldsymbol{c}'_i\right\|^2 + |C'| \; \left\|\boldsymbol{c}_i - \boldsymbol{c}'_i\right\|^2 > \sum_{\boldsymbol{x}\in C'_i} \left\|\boldsymbol{x} - \boldsymbol{c}'_i\right\|^2$$

where we used (1) in the first step and $\boldsymbol{c}_i \neq \boldsymbol{c}'_i$ in the second step. Hence,

$$\Phi(C_1, \dots, C_k) = \psi(C_1, \dots, C_k, \boldsymbol{c}_1, \dots, \boldsymbol{c}_k) > \psi(C'_1, \dots, C'_k, \boldsymbol{c}'_1, \dots, \boldsymbol{c}'_k) = \Phi(C'_1, \dots, C'_k)$$

concluding the proof. □

This immediately implies the following result.

Teorema 3 *Lloyd's algorithm terminates on any input $(\mathcal{X}, k)$ after at most $k^{|\mathcal{X}|}$ iterations.*

Dimostrazione. Note that Φ is a function of the current clustering $\{C_1, \dots, C_k\}$, which can take on at most k^n distinct values. Moreover, Lloyd's algorithm does not terminate only if the current iteration changed the clustering. Since Φ can only decrease when the clustering is changed, the algorithm must terminate after at most k^n iterations. □