

Clustering

Anil Maheshwari

anil@scs.carleton.ca
School of Computer Science
Carleton University
Canada

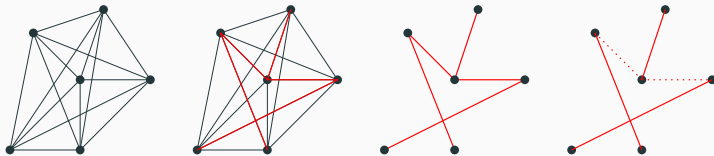
Introduction

Clustering Problem

Input: A set $X = \{x_1, \dots, x_n\}$ of n objects. For every pair $x_i, x_j \in X$, we have the distance $d(x_i, x_j) \geq 0$ such that $d(x_i, x_i) = 0$ and $d(x_i, x_j) = d(x_j, x_i)$.

Problem: Divide objects in X into k non-empty groups such that the *gap* between the groups is as large as possible. Distance between two groups is defined as the smallest distance between pair of points, where in the pair points belong to different groups.

1. Define a complete graph $G = (V = X, E)$, where each edge $e = (x_i, x_j)$ has a weight $d(x_i, x_j)$.
2. Construct a minimum spanning tree T of G
3. Delete $k - 1$ most expensive edges from T
4. Output the resulting k -connected components $C_1, \dots, C_k$



Claim

The components $C_1, \dots, C_k$ constitute a k -clustering of X that maximizes the gap.

K-Means Clustering Problem

Input: A set $X = \{x_1, \dots, x_n\}$ of n -points in $\mathbb{R}^d$. An integer $0 < k \leq n$.

Problem: Partition X into k clusters $C_1, \dots, C_k$. Points within a cluster should be *close* to each other compared to points outside the cluster.

Potential Function

Let $C_1, \dots, C_k$ be a k -clustering of X with centers $\mathcal{C} = \{c_1, \dots, c_k\}$, where $c_i \in \mathbb{R}^d$.

Define for all points $x \in X$, $D(x) = \min_{c \in \mathcal{C}} \|x - c\|$, i.e., the distance of x to the nearest center in $\mathcal{C}$.

Define the potential function $\Phi_{\mathcal{C}}(X) = \sum_{x \in X} D(x)^2$.

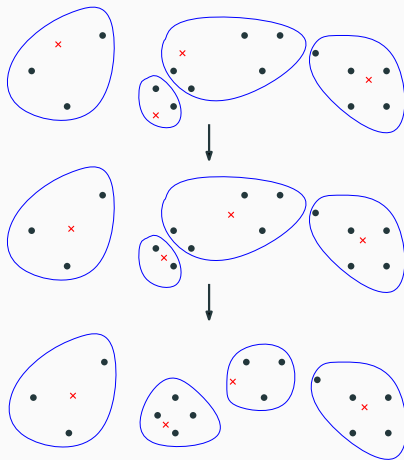
$\Phi_{\mathcal{C}}(X)$ = Sum of the squared distance between each point x in X to its nearest center in $\mathcal{C}$

Problem:

Given X and k , find k -centers $\mathcal{C}$ such that the corresponding clustering $C_1, \dots, C_k$ minimizes $\Phi_{\mathcal{C}}(X)$

1. Select Initial Centers: Arbitrary choose k -centers and initialize $\mathcal{C} = \{c_1, \dots, c_k\}$
2. Partition X : Compute sets $C_1, \dots, C_k$ with respect to centers in $\mathcal{C}$.
Point $x \in X$ is assigned to the cluster C_i if x 's nearest center in $\mathcal{C}$ is c_i .
3. Recompute Centers: For each $i \in \{1, \dots, k\}$, set c_i (the new cluster center) to be the center of the mass of points in C_i
4. Repeat Steps 2 & 3 till $\mathcal{C}$ no longer changes

An illustration of a Phase of Lloyd's Algorithm



Question 1: Does the algorithm terminate?

Question 2: Let $\mathcal{C}^*$ be an optimal clustering with potential $\Phi_{\mathcal{C}^*}(X)$.
Is the ratio $\frac{\Phi_{\mathcal{C}}(X)}{\Phi_{\mathcal{C}^*}(X)}$ bounded?

Question 1: Does the algorithm terminate?

Decrease in Potential

Each execution of Steps 2 & 3 decrease the value of the potential function.

Proof: We will use the following Lemma.

Lemma 1

Consider a set of points S . Let m^* denote the center of mass of S . Let z be an arbitrary point. Define $\Delta(S, z) = \sum_{x \in S} d(x, z)^2$. Then

$$\Delta(S, z) = \Delta(S, m^*) + |S|d(m^*, z)^2$$

Corollary 1

If S is a single cluster with initial center z , then moving the cluster center to m^* reduces the potential as $\Delta(S, z) - \Delta(S, m^*) = |S|d(m^*, z)^2 \geq 0$

Lemma 1

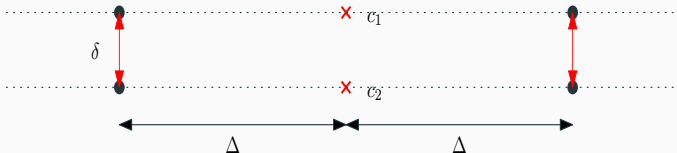
$\Delta(S, z) = \Delta(S, m^*) + |S|d(m^*, z)^2$, where m^* is center of mass of S and z is an arbitrary point

Proof: Assume we are in 2-dimensions. Let $z = (z_x, z_y)$ and $S = \{p_1, \dots, p_n\}$, where $p_i = (x_i, y_i)$.

1. $m^* = \left(\frac{\sum x_i}{n}, \frac{\sum y_i}{n} \right)$
2. $\Delta(S, z) = \sum_{p \in S} d(p, z)^2 = \sum_{p=(x_i, y_i) \in S} ((x_i - z_x)^2 + (y_i - z_y)^2)$
3. $\Delta(S, m^*) = \sum_{p \in S} d(p, m^*)^2 = \sum_i \left(x_i - \frac{\sum x_i}{n} \right)^2 + \sum_i \left(y_i - \frac{\sum y_i}{n} \right)^2$
4. $\begin{aligned} \Delta(S, z) - \Delta(S, m^*) &= nz_x^2 + nz_y^2 - 2z_x \sum x_i - 2z_y \sum y_i + n \left(\frac{\sum x_i}{n} \right)^2 + n \left(\frac{\sum y_i}{n} \right)^2 \\ &= n \left[z_x^2 + z_y^2 - 2z_x \frac{\sum x_i}{n} - 2z_y \frac{\sum y_i}{n} + \left(\frac{\sum x_i}{n} \right)^2 + \left(\frac{\sum y_i}{n} \right)^2 \right] \\ &= |S|d(m^*, z)^2 \end{aligned}$

Question 2: Is $\frac{\Phi_C(X)}{\Phi_{C^*}(X)}$ bounded?

Let $\Phi_{C^*}(X)$ be the potential of an optimal clustering and let $\Phi_C(X)$ be the potential of the clustering returned by Lloyd's heuristic.



Competitive Ratio

Competitive ratio of Lloyd's heuristic is unbounded, i.e. $\frac{\Phi_C(X)}{\Phi_{C^*}(X)} \rightarrow \infty$

How to choose initial centers?

Question: How to choose initial centers so that we are guaranteed to have some bounded competitive ratio with respect to optimum?

k-means++ Algorithm

Let $D(x)$ = Shortest distance from x to the nearest center among the current set of centers.

***k*-Means++ Algorithm:**

Step 1: Choose an initial cluster center c_1 uniformly at random from X .

Step 2: (Randomization Step) Choose the next center c_i by selecting a point $x \in X$ with probability $\frac{D(x)^2}{\sum_{y \in X} D(y)^2}$

Step 3: Repeat Step 2 till k centers are chosen

Step 4: Execute Lloyd's Heuristic by choosing $\{c_1, \dots, c_k\}$ as the initial centers

1. In the Randomization Step, the points of X that are farther from the currently chosen centers have a higher chance of being selected.
2. The algorithm is $8(\ln k + 2)$ -competitive. Let the k -centers returned by k -means++ algorithm be $\mathcal{C}$. Then, $E[\Phi_{\mathcal{C}}(X)] \leq 8(\ln k + 2)\Phi_{\mathcal{C}^*}(X)$.
3. Claim of $8(\ln k + 2)$ -competitiveness holds for the clustering obtained after Step 3. Step 4 may further improve the potential and the ratio.
4. Proof is not easy. Consider clusters of an optimal solution $\mathcal{C}^*$. The authors show
 - The algorithm is 2-competitive w.r.t. the points in the optimal cluster from where the first center c_1 is chosen by the algorithm
 - The algorithm is 8-competitive in all those clusters of optimal from which the algorithm chooses a center.
 - If $\mathcal{C}$ doesn't have centers from some of the clusters of the optimal, then the algorithm is $8(\ln k + 2)$ -competitive.

k -Means++ Algorithm:

- Step 1:** Choose an initial cluster center c_1 uniformly at random from X .
- Step 2: (Randomization Step)** Choose the next center c_i by selecting a point $x \in X$ with probability $\frac{D(x)^2}{\sum_{y \in X} D(y)^2}$
- Step 3:** Repeat Step 2 till k centers are chosen
- Step 4:** Execute Lloyd's Heuristic by choosing $\{c_1, \dots, c_k\}$ as the initial centers

Theorem (Arthur and Vassilvitskii 2007)

The k -means++ algorithm is $8(\ln k + 2)$ -competitive