Algorithms for Data Science: Lecture on Clustering

Barna Saha

1 Clustering

Given a set of points with a notion of distance between points, group the points into some number of clusters so that members of a cluster are "close" to each other, while members of different clusters are far. The problem of clustering is ubiquitous. We may want to cluster documents by topic they represent, we may want to cluster the moviegoers by the types of movies they like, or cluster genes by their pattern etc. The clustering may seem easy in low-dimensional Euclidean space, but becomes notoriously hard in higher dimensions and considering non-Euclidean distances.

Often, given a set of points in a space, we would like to cluster them in a way to optimize some goodness criteria that tells how closely knit the clusters are. There had been many such proposals of goodness criteria. Here we elaborate on three such popular clustering paradigms: k-center clustering, k-median clustering and k-means clustering.

2 k-center Clustering

In the k-center problem, the input is a set of vertices V with distances $d : V \times V \to \mathbb{R}$ between them, and the goal is to select k centers such that the maximum distance to the closest center is minimized over V.

The problem is NP-hard, and even cannot be approximated within any factor $\alpha \geq 1$ unless the distance d is a metric satisfying triangle inequality. Hence, we consider the Metric-k-center problem and develop an approximation algorithm.

Suppose there exists k centers in V such that the maximum distance that any vertex in V needs to travel to reach one of the centers is OPT. We will develop an algorithm in which each vertex will have to travel at most $2OPT$ instead, that is we will develop a 2-approximation algorithm for the metric-k center.

2.1 Greedy Algorithm

1. Pick any vertex $v \in V$ arbitrarily and declare it as the first center c_1.
2. For $i = 2$ to k do
   (a) Select the vertex from V that is farthest from the already chosen centers, that is maximizes the minimum distance to any of the chosen center–as the new ith center.

We will now show that this is a 2-approximation.

Theorem 1. *The greedy algorithm returns a 2-approximation for the k-center problem.*

Proof. Consider when the ℓth center c_ℓ is chosen. $\mathcal{C}_{\ell-1} = \{c_1, c_2, .., c_{\ell-1}\}$ denote the already chosen sets. It must happen that minimum distance from c_ℓ to $\mathcal{C}_{\ell-1}$ is lower than the minimum distance c_i to $\mathcal{C}_{i-1}$. If not, consider the following contradicting scenario.

Suppose minimum distance from c_ℓ to $\mathcal{C}_{\ell-1}$ is to c_j. Let $j \leq i-1$. Then while choosing the ith center, the algorithm will prefer c_ℓ over c_i as c_ℓ is farther than c_i from $\mathcal{C}_{i-1}$. So suppose

$j \geq i$. But then minimum distance of c_ℓ to $\mathcal{C}_{i-1}$ is only higher–so higher than c_i, and again the algorithm will prefer c_ℓ over c_i as c_ℓ is farther than c_i from $\mathcal{C}_{i-1}$. This gives a contradiction.

Now, after the k centers have been selected–consider the point $c_{k+1} \in V$ that is farthest from $\mathcal{C}_k$. If that distance is $\leq 2OPT$, we have a 2-approximation. So, suppose that distance is $> 2OPT$. That means by the previous argument, all pairwise distances between c_i and c_j, $i \neq j \in [1, k+1]$ is $> 2OPT$.

Then among these $k+1$ points, two must belong to the same optimum cluster and are thus within distance at most OPT from one of the optimum centers. By triangle inequality, then distance between them is at most $2OPT$ giving a contradiction. $\square$

3 k-median Clustering

The k-center clustering is highly susceptible to outliers. To overcome this the clustering objective one can try to minimize the sum of the distances to the minimum cluster center. This is known as the k-median clustering.

Hence, in the k-median clustering, the goal is to select k centers $c_1, c_2, .., c_k$ from $\mathcal{C}$ such that $\sum_{v \in V} \min_{i=1}^{k} d(v, c_i))$ is minimized.

Again the problem is NP-hard, and here we give a simple *local search* algorithm that can be proven to give a good approximation factor.

3.1 Local Search

The following is a very intuitive algorithm to solve the k-median problem.

1. Start with any set of k-vertices from V as your centers $\mathcal{C}$, let $Cost(\mathcal{C})$ denote the value of objective function for $\mathcal{C}$ as solution, that is $Cost(\mathcal{C}) = \sum_{v \in V} \min_{c \in \mathcal{C}} d(v, c)$.

2. If there exists $x \in V \setminus \mathcal{C}$ and $Y \in \mathcal{C}$ such that swapping X and Y improves the solution then swap them. That is if $Cost(\mathcal{C} - Y + X) < Cost(\mathcal{C})$ then $Swap(X, Y)$.

3. Else return.

It can be shown that the above local search algorithm converges and gives a solution within 5 times of the optimum cost. There are better local search algorithms known that instead of swapping one vertex at a time, consider multiple vertices swapping.

4 k-means Clustering

We now look at the k-means clustering which is one of the oldest and popular clustering algorithms. In the k-means problem, instead of minimizing the sum of distances to the nearest center, we try to minimize the sum of the squared distances to the nearest center. k-means clustering is defined over the $\mathbb{R}^d$ space for Euclidean distance. Moreover, the center does not need to be a point in V itself. Given a set of points, we know if we select their mean as the cluster center that will minimize the total squared distance! (why?)—which is the reason behind the choice of name.

Solving the problem exactly is NP-Hard. 25 years ago, Llyod proposed another simple local search algorithm which till date is the most popular algorithm for k-means clustering. A 2002 survey of data mining techniques states that it “is by far the most popular clustering algorithm used in scientific and industrial applications.”

4.1 Llyod's Algorithm

1. Begin with k arbitrary centers, typically chosen uniformly at random from the data points.
2. Assign each point to its nearest cluster center
3. Recompute the new cluster centers as the center of mass of the points assigned to the clusters
4. Repeat Steps 1-3 until the process converges.

It can be shown that the objective function is monotonically decreasing, which ensures no configuration is repeated. However, the convergence could be slow: k^n possible clusterings.

Moreover, the algorithm can converge to a local minima without any guarantee on the objective function value.

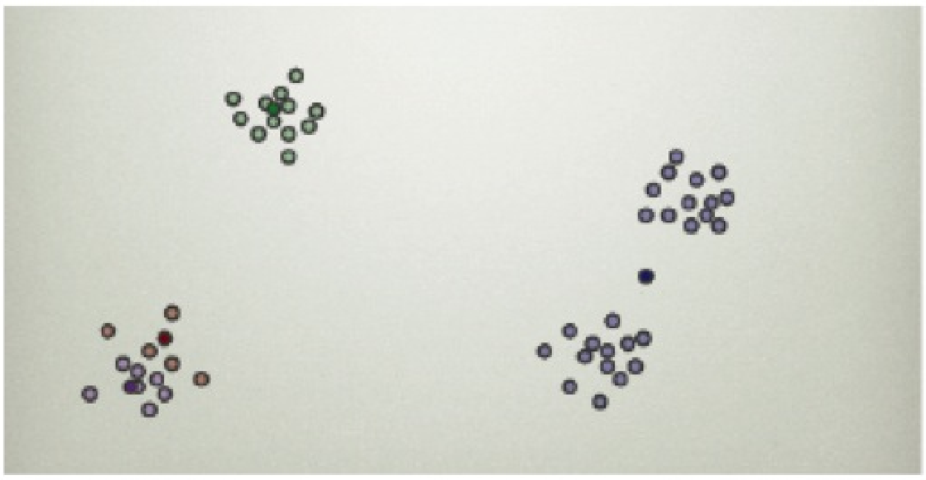

Figure 1: k-means get stuck in a local minima

4.2 Drawbacks of k-means clustering

- Euclidean distance is used as a metric and variance is used as a measure of cluster scatter.
- The number of clusters k is an input parameter: an inappropriate choice of k may yield poor results. That is why, when performing k-means, it is important to run diagnostic checks for determining the number of clusters in the data set.
- Convergence to a local minimum may produce results far away from optimal clustering

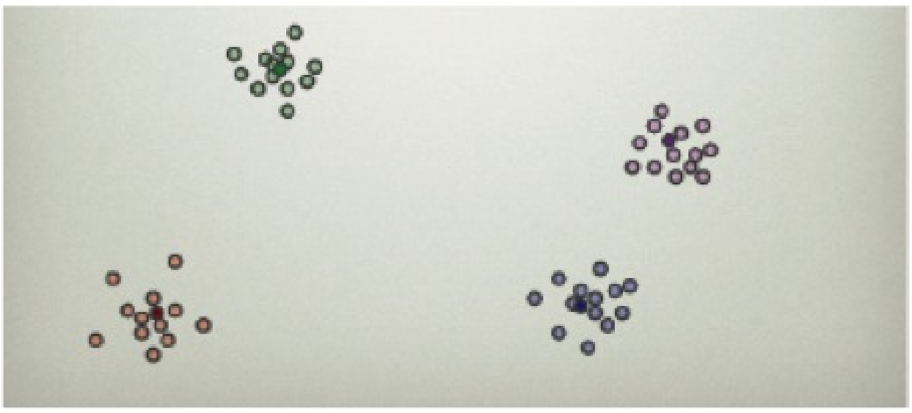

Figure 2: Alternative to select initial k clusters in k-means

4.3 k-means++

By selecting initial centers wisely k-means++ generalizes k-means and ensures the converged local minima has an objective value close to the actual optimum objective.

To overcome the scenario depicted in Figure 1, one can try the following fix, where we always select the center that is farthest away from the currently opened centers.

However, such an assignment will be highly susceptive to outliers. For example, see Figure 3.

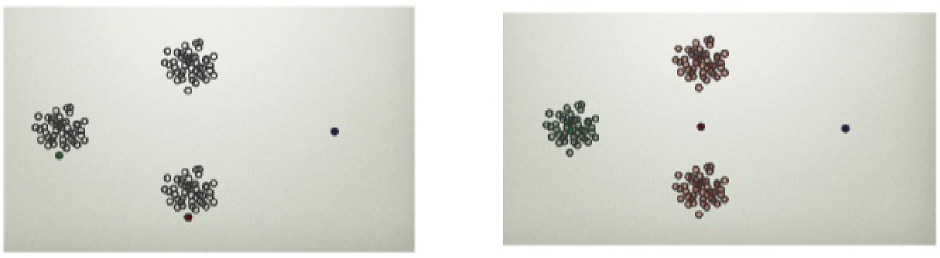

Figure 3: Susceptibility to Outliers

k-means++ interpolates between selecting centers randomly and selecting centers only by distance. Specifically, let $D(x)$ denote the minimum distance of x to the nearest chosen cluster centers. Then select x as a new center with probability $\frac{D(x)^2}{\sum_{v\in V} D(v)^2}$.

It can be shown by this small change k-means++ returns a solution which is within an $O(\log k)$-factor of the optimum objective value.