k-means, k-means++

Barna Saha

March 8, 2016

K-Means: The Most Popular Clustering Algorithm

- k-means clustering problem is one of the oldest and most important problem.

K-Means: The Most Popular Clustering Algorithm

- k-means clustering problem is one of the oldest and most important problem.
- Given an integer k, and a set of data points in $\mathbb{R}^d$, the goal is to choose k *centers* so as to minimize ϕ, the sum of squared distance between each point and its closest center.

K-Means: The Most Popular Clustering Algorithm

- k-means clustering problem is one of the oldest and most important problem.
- Given an integer k, and a set of data points in $\mathbb{R}^d$, the goal is to choose k *centers* so as to minimize ϕ, the sum of squared distance between each point and its closest center.
- Solving this problem exactly is NP-Hard.

K-Means: The Most Popular Clustering Algorithm

- k-means clustering problem is one of the oldest and most important problem.
- Given an integer k, and a set of data points in $\mathbb{R}^d$, the goal is to choose k *centers* so as to minimize ϕ, the sum of squared distance between each point and its closest center.
- Solving this problem exactly is NP-Hard.
- 25 years ago Llyod proposed a simple *local search* algorithm that is still very widely used today.

K-Means: The Most Popular Clustering Algorithm

- k-means clustering problem is one of the oldest and most important problem.
- Given an integer k, and a set of data points in $\mathbb{R}^d$, the goal is to choose k *centers* so as to minimize ϕ, the sum of squared distance between each point and its closest center.
- Solving this problem exactly is NP-Hard.
- 25 years ago Llyod proposed a simple *local search* algorithm that is still very widely used today.
- A 2002 survey of data mining techniques states that it "is by far the most popular clustering algorithm used in scientific and industrial applications."

Llyod's Local Search Algorithm

1. Begin with k arbitrary centers, typically chosen uniformly at random from the data points.

ϕ is monotonically decreasing, which ensures no configuration is repeated.
Convergence could be slow: k^n possible clusterings.

Llyod's Local Search 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

ϕ is monotonically decreasing, which ensures no configuration is repeated.
Convergence could be slow: k^n possible clusterings.

Llyod's Local Search 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

ϕ is monotonically decreasing, which ensures no configuration is repeated.
Convergence could be slow: k^n possible clusterings.

Llyod's Local Search 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.

ϕ is monotonically decreasing, which ensures no configuration is repeated.
Convergence could be slow: k^n possible clusterings.

Llyod's Local Search 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.

ϕ is monotonically decreasing, which ensures no configuration is repeated.
Convergence could be slow: k^n possible clusterings.

Llyod's Local Search Algorithm

- Convergence could be slow: k^n possible clusterings.
- Lloyd's k-means algorithm has polynomial **smoothed** running time.

Illustration of k-means clustering

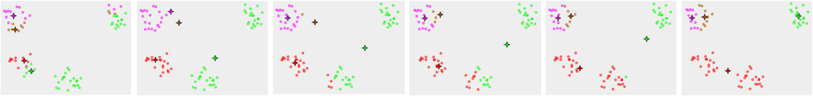
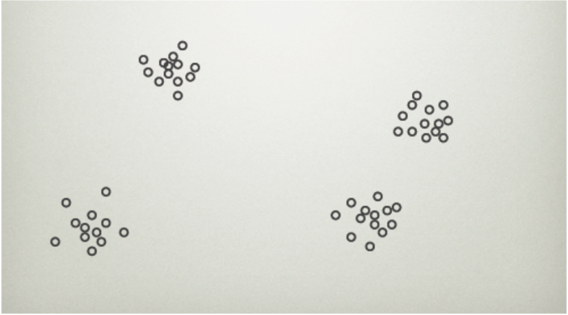


Figure: Convergence to Local Optimum

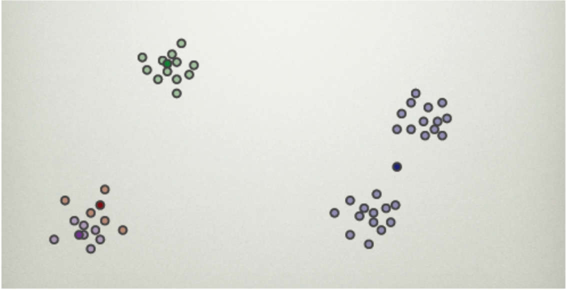
Drawbacks of k-means algorithm

- 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

k-means++

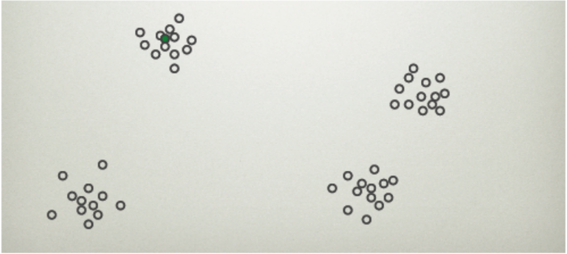


k-means++



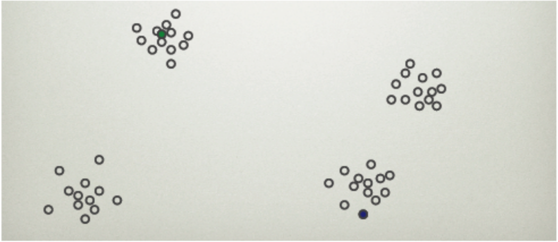
k-means++

- The first fix: *Select centers based on distance.*



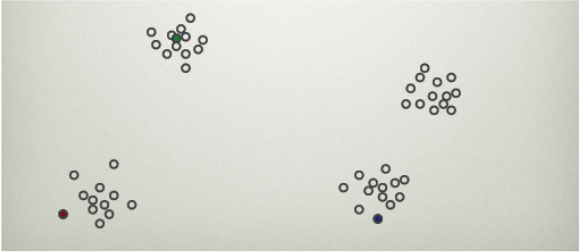
k-means++

- The first fix: *Select centers based on distance.*



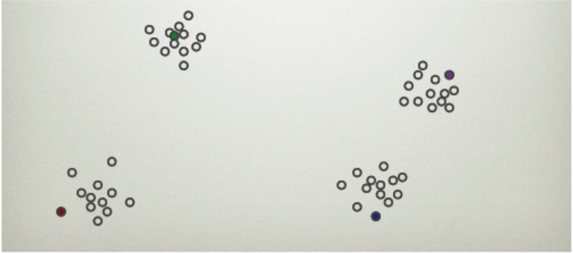
k-means++

- The first fix: *Select centers based on distance.*



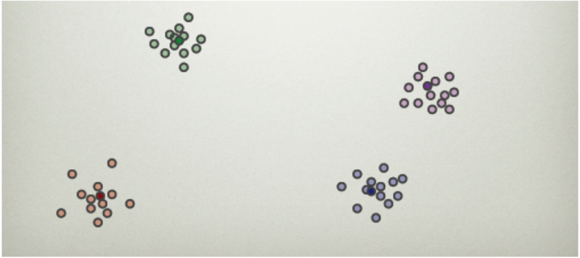
k-means++

- The first fix: *Select centers based on distance.*



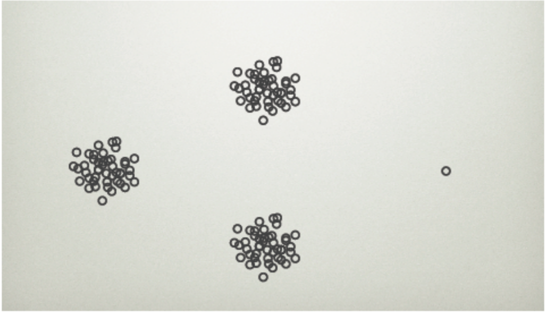
k-means++

- The first fix: *Select centers based on distance.*



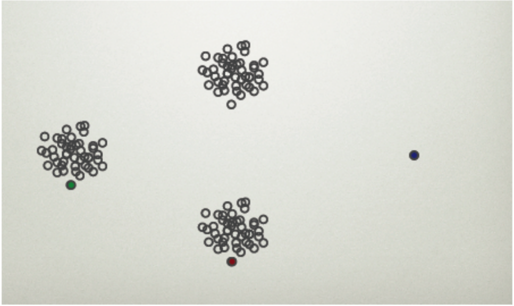
k-means++

- Sensitive to outliers.



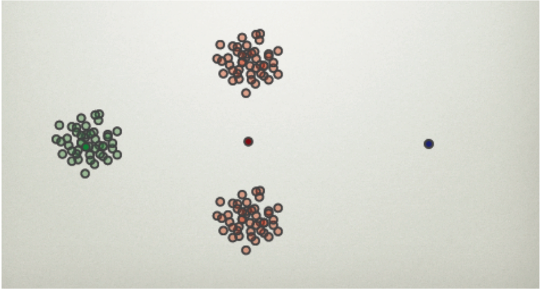
k-means++

- Sensitive to outliers.



k-means++

- Sensitive to outliers.



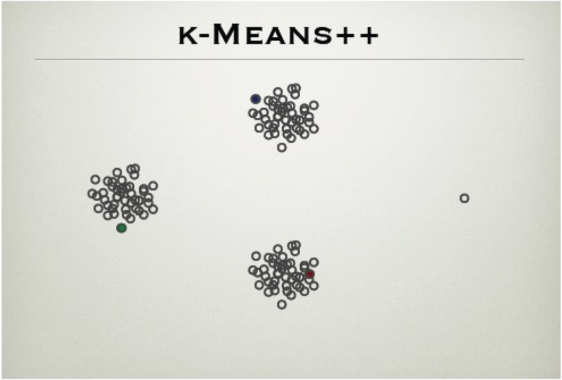
k-means++

- Interpolate between the two methods:

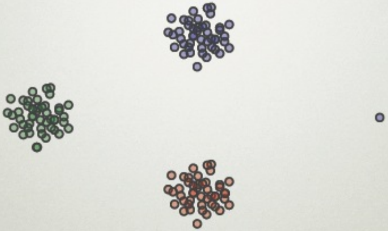
  Let $D(x)$ be the distance between x and the nearest cluster center. Sample x as a cluster center proportionately to $(D(x))^2$.

k-means++ returns clustering $\mathcal{C}$ which is $\log k$-competitive.

In the class we will show that if the cluster centers are chosen from each optimal cluster then k-means++ is 8-competitive.

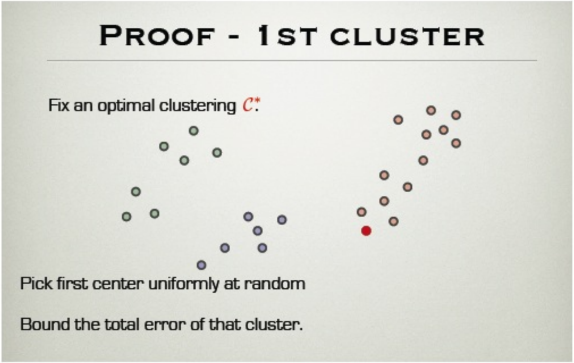

k-Means++

k-Means++



Theorem: k-means++ is $\Theta(\log k)$ approximate in expectation.

Ostrovsky et al. [06]: Similar method is $O(1)$ approximate under some data distribution assumptions.


PROOF - 1ST CLUSTER
Fix an optimal clustering C^*.
Pick first center uniformly at random
Bound the total error of that cluster.

Proof - 1st cluster

Let A be the cluster.

Each point $a_0 \in A$ equally likely to be the chosen center.

Expected Error:

$$E[\phi(A)] = \sum_{a_0 \in A} \frac{1}{|A|} \sum_{a \in A} \|a - a_0\|^2$$

$$= 2 \sum_{a \in A} \|a - \bar{A}\|^2 \quad = 2\phi^*(A)$$

Proof - Other Clusters

Suppose next center came from a new cluster in OPT.

Bound the total error of that cluster.

Other Clusters

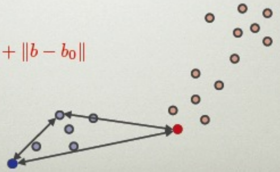
Let B be this cluster, and b_0 the point selected.

Then:

$$E[\phi(B)] = \sum_{b_0 \in B} \frac{D^2(b_0)}{\sum_{b \in B} D^2(b)} \cdot \sum_{b \in B} \min(D(b), \|b - b_0\|)^2$$

Key step:

$$D(b_0) \leq D(b) + \|b - b_0\|$$



Cont.

For any b: $D^2(b_0) \leq 2D^2(b) + 2\|b - b_0\|^2$

Avg. over all b: $D^2(b_0) \leq \frac{2}{|B|}\sum_{b \in B} D^2(b) + \frac{2}{|B|}\sum_{b \in B} \|b - b_0\|^2$

Same for all b_0

Cost in uniform sampling

Cont.

For any b: $D^2(b_0) \leq 2D^2(b) + 2\|b - b_0\|^2$

Avg. over all b: $D^2(b_0) \leq \frac{2}{|B|}\sum_{b\in B} D^2(b) + \frac{2}{|B|}\sum_{b\in B}\|b - b_0\|^2$

Recall:

$$E[\phi(B)] = \sum_{b_0\in B} \frac{D^2(b_0)}{\sum_{b\in B} D^2(b)} \cdot \sum_{b\in B} \min(D(b), \|b - b_0\|)^2$$

$$\leq \frac{4}{|B|}\sum_{b_0\in B}\sum_{b\in B}\|b - b_0\|^2 \quad = 8\phi^*(B)$$

Wrap Up

If clusters are well separated, and we always pick a center from a new optimal cluster, the algorithm is 8- competitive.

Intuition: if no points from a cluster are picked, then it probably does not contribute much to the overall error.

Formally, an inductive proof shows this method is $\Theta(\log k)$ competitive.