K-means++: The Advantages of Careful Seeding

Sergei Vassilvitskii
David Arthur
(Stanford University)
Given $n$ points in $\mathbb{R}^d$ split them into $k$ similar groups.
Given $n$ points in $\mathbb{R}^d$ split them into $k$ similar groups.

This talk: k-means clustering:

Find $k$ centers, $C$ that minimize

$$\sum_{x \in X} \min_{c \in C} \|x - c\|^2$$
Why Means?

Objective: Find \( k \) centers, \( C \) that minimize

\[
\sum_{x \in X} \min_{c \in C} \|x - c\|_2^2
\]

For one cluster: Find \( y \) that minimizes

\[
\sum_{x \in X} \|x - y\|_2^2
\]

Easy! \( y = \frac{1}{|X|} \sum_{x \in X} x \)
Lloyd’s Method: k-means

Initialize with random clusters
Lloyd’s Method: k-means

Assign each point to nearest center
Lloyd’s Method: k-means

Recompute optimum centers (means)
Lloyd’s Method: k-means

Repeat: Assign points to nearest center
Lloyd’s Method: k-means

Repeat: Recompute centers
Lloyd’s Method: k-means

Repeat...
Lloyd’s Method: k-means

Repeat...Until clustering does not change
Analysis

How good is this algorithm?

Finds a local optimum

That is potentially arbitrarily worse than optimal solution
Approximating k-means

- Mount et al.: $9 + \epsilon$ approximation in time $O(n^3/\epsilon^d)$
- Har Peled et al.: $1 + \epsilon$ in time $O(n + k^{k+2\epsilon^{-2d}} \log^k (n/\epsilon))$
- Kumar et al.: $1 + \epsilon$ in time $2^{(k/\epsilon)^{O(1)}} nd$
**Approximating k-means**

- Mount et al.: $9 + \epsilon$ approximation in time $O(n^3/\epsilon^d)$
- Har Peled et al.: $1 + \epsilon$ in time $O(n + k^{k+2}\epsilon^{-2dk} \log^k(n/\epsilon))$
- Kumar et al.: $1 + \epsilon$ in time $2^{(k/\epsilon)^{O(1)}} nd$

Lloyd’s method:

- Worst-case time complexity: $2^{\Omega(\sqrt{n})}$
- Smoothed complexity: $n^{O(k)}$
Approximating K-Means

- Mount et al.: $9 + \epsilon$ approximation in time $O(n^3/\epsilon^d)$
- Har Peled et al.: $1 + \epsilon$ in time $O(n + k^{k+2} \epsilon^{-2dk} \log^k(n/\epsilon))$
- Kumar et al.: $1 + \epsilon$ in time $2^{(k/\epsilon)^{O(1)}} nd$

Lloyd’s method:

For example, Digit Recognition dataset (UCI):

$n = 60,000 \quad d = 600$

Convergence to a local optimum in 60 iterations.
**Challenge**

Develop an approximation algorithm for k-means clustering that is competitive with the k-means method in speed and solution quality.

Easiest line of attack: focus on the initial center positions.

Classical k-means: pick $k$ points at random.
K-means on Gaussians
K-MEANS ON GAUSSIANS
Easy Fix

Select centers using a furthest point algorithm (2-approximation to k-Center clustering).
Easy Fix

Select centers using a furthest point algorithm (2-approximation to k-Center clustering).
Easy Fix

Select centers using a furthest point algorithm (2-approximation to k-Center clustering).
Easy Fix

Select centers using a furthest point algorithm (2-approximation to k-Center clustering).
Select centers using a furthest point algorithm (2-approximation to k-Center clustering).
Sensitive to Outliers
Sensitive to Outliers
Sensitive to Outliers
Interpolate between the two methods:

Let $D(x)$ be the distance between $x$ and the nearest cluster center. Sample proportionally to $(D(x))^\alpha = D^\alpha(x)$

Original Lloyd’s: $\alpha = 0$

Furthest Point: $\alpha = \infty$

k-means++: $\alpha = 2$

Contribution of $x$ to the overall error
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.
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:

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

\[
= 2 \sum_{a \in A} \|a - \bar{A}\|^2 = 2\phi^*(A)
\]
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:

$$D(b_0) \leq D(b) + \|b - b_0\|$$
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
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:
\[
E[\phi(B)] = \sum_{b_0 \in B} \frac{D^2(b_0)}{\sum_{b \in B} D^2(b)} \cdot \sum_{b \in B} \min(D(b), \|b - b_0\|)^2 \]
\[
\leq \frac{4}{|B|} \sum_{b_0 \in B} \sum_{b \in B} \|b - b_0\|^2 = 8\phi^*(B) \]
If clusters are well separated, and we always pick a center from a new optimal cluster, the algorithm is $8$-competitive.
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.
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.
Experiments

Tested on several datasets:

Synthetic
- 10k points, 3 dimensions

Cloud Cover [UCI Repository]
- 10k points, 54 dimensions

Color Quantization
- 16k points, 16 dimensions

Intrusion Detection [KDD Cup]
- 500k points, 35 dimensions
Typical Run

KM++ v. KM v. KM-Hybrid

Stage Error

LLOYD
HYBRID
KM++
## Experiments

### Total Error

<table>
<thead>
<tr>
<th></th>
<th>k-means</th>
<th>km-Hybrid</th>
<th>k-means++</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic</td>
<td>0.016</td>
<td>0.015</td>
<td>0.014</td>
</tr>
<tr>
<td>Cloud Cover</td>
<td>$6.06 \times 10^5$</td>
<td>$6.02 \times 10^5$</td>
<td>$5.95 \times 10^5$</td>
</tr>
<tr>
<td>Color</td>
<td>741</td>
<td>712</td>
<td>670</td>
</tr>
<tr>
<td>Intrusion</td>
<td>$32.9 \times 10^3$</td>
<td>$-$</td>
<td>$3.4 \times 10^3$</td>
</tr>
</tbody>
</table>

### Time:

k-means++ 1% slower due to initialization.
Final Message

Friends don’t let friends use k-means.
Thank You

Any Questions?