Scalable K-Means++

Steven Wu

University of Pennsylvania

October 3, 2014

Paper by Bahmani, Moseley, Vattani, Kumar and Vassilvitskii
Clustering is very important!
Clustering is very important!

- Fundamental problem in data analysis and machine learning
- “Frequently asked interview question by Big Data tech firms” — Yaroslavtsev
Clustering is very important!

- Fundamental problem in data analysis and machine learning
- “Frequently asked interview question by Big Data tech firms” — Yaroslavtsev
Problem

$k$-means Objective

Let $X = \{x_1, \ldots, x_n\}$ be a set of points in $d$-dimensional space, find a set of centers $C = \{c_1, \ldots, c_k\}$ to minimize

$$
\sum_{x \in X} \min_{i \in [k]} \|x - c_i\|^2
$$
Problem

\textit{k}-means Objective

Let \( X = \{x_1, \ldots, x_n\} \) be a set of points in \( d \)-dimensional space, find a set of centers \( C = \{c_1, \ldots, c_k\} \) to minimize

\[
\sum_{x \in X} \min_{i \in [k]} \|x - c_i\|^2
\]

NP-Hard Problem
$k$-means algorithm

$k$-means (sometimes called Lloyd’s algorithm)

- **Initialization:** Start with a set of randomly chosen initial centers
- **Repeat:**
  - Assign each point to its nearest center;
  - Recompute the center given the point assignment
- **Until** convergence
In theory

$k$-means algorithm not very appealing

- **Efficiency**: run time can be exponential in the worst case
- **Quality**: final solution is locally optimal, but far away from global optimum
In practice

we♥it
Goal

A scalable $k$-means algorithm with

- theoretical guarantee
- good practical performance
Goal

A scalable $k$-means algorithm with

- theoretical guarantee
- good practical performance
Focus: Initialization

A better way to initialize the clustering dramatically changes the performance of the algorithm.
Potential Problem: Sensitive to Initialization
Potential Problem: Sensitive to Initialization
Potential Problem: Sensitive to Initialization

Stuck in local optimum

Photo credited to David Arthur
Intuition: Spread Out
**k-means++ Initialization**

- First center is selected *uniformly* at random from the data
- Subsequent centers: each point is selected with probability

\[
\frac{d^2(x, C)}{\sum_x d^2(x, C)}
\]
$k$-means++ Initialization

- First center is selected \textit{uniformly} at random from the data
- Subsequent centers: each point is selected with probability
  \[
  \frac{d^2(x, C)}{\sum_x d^2(x, C)}
  \]
  proportional to its contribution to the overall error given the previous selections:
Pros and Cons of $k$-means++

**Advantage**
The initialization step itself already obtains an $8 \log k$ approximation to OPT in expectation.

**Disadvantage**
Not scalable! Sequential nature: the choice of the next center depends on the current set of centers. $k$ passes over the data (think of $k = 1000$)
Pros and Cons of $k$-means++

**Advantage**
The initialization step itself already obtains an $8 \log k$ approximation to OPT in expectation.
(The Lloyd iterations would only make it better)
Pros and Cons of $k$-means++

**Advantage**

The initialization step itself already obtains an $8 \log k$ approximation to OPT in expectation. (The Lloyd iterations would only make it better)

**Disadvantage**

Not scalable!
Pros and Cons of $k$-means++

**Advantage**
The initialization step itself already obtains an $8 \log k$ approximation to OPT in expectation.
(The Lloyd iterations would only make it better)

**Disadvantage**
Not scalable!
Sequential nature: the choice of the next center depends on the current set of centers.
## Pros and Cons of $k$-means++

### Advantage

The initialization step itself already obtains an $8 \log k$ approximation to OPT in expectation.
(The Lloyd iterations would only make it better)

### Disadvantage

Not scalable!
Sequential nature: the choice of the next center depends on the current set of centers. $k$ passes over the data (think of $k = 1000$)
Desiderata

- Fewer number of iterations (sample more than 1 points each round)
- Provable approximation guarantee
First center $C$: sample a point uniformly at random
1. First center \( C \): sample a point uniformly at random
2. Initial cost \( \psi = \sum_x d^2(x, C) \)
$k$-means

1. First center $C$: sample a point uniformly at random
2. Initial cost $\psi = \sum_x d^2(x, C)$
3. for $O(\log \psi)$ times do
   - $C' \leftarrow$ sample each point $x \in X$ independently with probability
     $$p_x = \frac{\ell \cdot d^2(x, C)}{\sum_x d^2(x, C)}$$
   - $C \leftarrow C \cup C'$
**$k$-means**

1. First center $C$: sample a point uniformly at random
2. Initial cost $\psi = \sum_x d^2(x,C)$
3. **for** $O(\log \psi)$ times **do**
   - $C' \leftarrow$ sample each point $x \in X$ independently with probability
     
     \[
     p_x = \frac{\ell \cdot d^2(x,C)}{\sum_x d^2(x,C)}
     \]
   - $C \leftarrow C \cup C'$
4. For $x \in C$, let $w_x$ be the number of points belonging to this center
5. Recluster the **weighted** points in $C$ into $k$ clusters
Number of points

Number of intermediate centers?

- Oversampling factor $\ell = \Theta(k)$.
- Expected number of points in $C$: $\ell \log \psi$
• An interpolation between Lloyd and K-means++

Number of iterations ($R$)

$R=k$: Simulating K-means++ ($l=1$) $\rightarrow$ Strong guarantee

Small $R$: K-means $\mid |$ $\rightarrow$ Can it possibly give any guarantees?

$R=0$: Lloyd $\rightarrow$ No guarantees

Photo credited to Bahmani
Theoretical Guarantee

Theorem

If an $\alpha$-approximation is used in the last step, then $k$-means obtains a solution that is an $O(\alpha)$-approximation to $k$-means.

For example, we could use $k$-means++ and get $O(\log k)$-approximation.
The theoretical guarantee states that if an $\alpha$-approximation is used in the last step, then $k$-means|| obtains a solution that is an $O(\alpha)$-approximation to $k$-means. For example, we could use $k$-means++ and get $O(\log k)$-approximation.
Sketch of the Analysis

**Theorem**

If $\Psi$ and $\Psi'$ are the costs of the clustering at the beginning and end of an iteration, and $OPT$ is the cost of the optimum clustering

$$\mathbb{E}[\Psi'] \leq O(OPT) + \frac{k}{e\ell} \Psi.$$
Consider a cluster $A$ in OPT

$$A = \{a_1, \ldots, a_T\}$$

Centroid $c_A = \frac{1}{|T|} \sum a_t$

Increasing order of their distance to $c_A$
Consider a cluster $A$ in OPT

$$A = \{a_1, \ldots, a_T\}$$

Centroid $c_A = \frac{1}{|T|} \sum a_t$

Increasing order of their distance to $c_A$

Prior to our iteration, we have $C$ and let

$$\phi(C) = \sum_x d^2(x, C); \phi_A(C) = \sum_a d^2(a, C)$$
Consider a cluster $A$ in OPT

$$A = \{a_1, \ldots, a_T\}$$

Centroid $c_A = \frac{1}{|T|} \sum a_t$

Increasing order of their distance to $c_A$

Prior to our iteration, we have $C$ and let

$$\phi(C) = \sum_{x} d^2(x, C); \phi_A(C) = \sum_{a} d^2(a, C)$$

Let $p_t = \ell d^2(a_t, C) / \phi(C)$ be the probability of selecting $a_t$
Consider a cluster $A$ in OPT

$$A = \{a_1, \ldots, a_T\}$$
Centroid $c_A = \frac{1}{|T|} \sum a_t$
Increasing order of their distance to $c_A$

Prior to our iteration, we have $C$ and let

$$\phi(C) = \sum_x d^2(x, C); \phi_A(C) = \sum_a d^2(a, C)$$

Let $p_t = \ell d^2(a_t, C)/\phi(C)$ be the probability of selecting $a_t$

For any $1 \leq t \leq T$,

$$q_t = p_t \prod_{j=1}^{t-1} (1 - p_j)$$
Consider a cluster $A$ in OPT

$$A = \{a_1, \ldots, a_T\}$$

Centroid $c_A = \frac{1}{|T|} \sum a_t$

Increasing order of their distance to $c_A$

Prior to our iteration, we have $C$ and let

$$\phi(C) = \sum_x d^2(x, C); \phi_A(C) = \sum_a d^2(a, C)$$

Let $p_t = \ell d^2(a_t, C) / \phi(C)$ be the probability of selecting $a_t$

For any $1 \leq t \leq T$,

$$q_t = p_t \prod_{j=1}^{t-1} (1 - p_j)$$
Either assign all points in some selected $a_t$, or stick with the original

$$s_t = \min \left\{ \phi_A, \sum_{a \in A} \|a - a_t\|^2 \right\}$$
Either assign all points in some selected $a_t$, or stick with the original

$$s_t = \min \left\{ \phi_A, \sum_{a \in A} \|a - a_t\|^2 \right\}$$

$$\mathbb{E} \left[ \phi_A(C \cup C') \right] \leq \sum_t q_t s_t + q_{T+1} \phi_A(C)$$

where $q_{T+1}$ is the probability no point in $A$ is selected.
Either assign all points in some selected $a_t$, or stick with the original

$$s_t = \min \left\{ \phi_A, \sum_{a \in A} \|a - a_t\|^2 \right\}$$

$$\mathbb{E} \left[ \phi_A(C \cup C') \right] \leq \sum_t q_t s_t + q_{T+1} \phi_A(C)$$

where $q_{T+1}$ is the probability no point in $A$ is selected.
Plug in $p_t = p$ (the case in which all points are far from $C$ and they are tightly clustered)
Either assign all points in some selected $a_t$, or stick with the original

$$s_t = \min \left\{ \phi_A, \sum_{a \in A} \|a - a_t\|^2 \right\}$$

$$\mathbb{E} \left[ \phi_A(C \cup C') \right] \leq \sum_{t} q_t s_t + q_{T+1} \phi_A(C)$$

where $q_{T+1}$ is the probability no point in $A$ is selected.

Plug in $p_t = p$ (the case in which all points are far from $C$ and they are tightly clustered)

$q_t = p(1 - p)^t$
\[ s'_t = \sum_{a \in A} \| a - a_t \|^2 \]

\{ s'_t \} is an increasing sequence.
\[ s'_t = \sum_{a \in A} ||a - a_t||^2 \]

\{ s'_t \} is an increasing sequence.

\[ \sum_t q_t s_t \leq \sum_t q_t s'_t \leq \frac{1}{T} \left( \sum_t q_t \cdot \sum_t s'_t \right) = \left( \sum_t q_t \cdot \frac{1}{T} \sum_t s'_t \right) = \left( \sum_t q_t \right) 2\phi_A^* \]
Finally

\[ E[\phi_A(C \cup C')] \leq (1 - q_{T+1})2\phi_A^* + q_{T+1}\phi_A(C) \]
Parallel Implementation

Lloyd’s iteration: easy to implement as long as we can store the set $C$ among all mappers

1. First center $C$: sample a point uniformly at random
Parallel Implementation

1. First center $C$: sample a point uniformly at random
2. Initial cost $\psi = \sum_x d^2(x, C)$ (reducer simply adds)
Parallel Implementation

1. First center $C$: sample a point uniformly at random
2. Initial cost $\psi = \sum_x d^2(x, C)$ (reducer simply adds)
3. for $O(\log \psi)$ times do
   - $C' \leftarrow$ sample each point $x \in X$ independently with probability (mapper independently sample)
     \[ p_x = \frac{\ell \cdot d^2(x, C)}{\sum_x d^2(x, C)} \]
   - $C \leftarrow C \cup C'$
## Experimental Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Clustering Cost Right After Initialization</th>
<th>Clustering Cost After Lloyd Convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>NA</td>
<td>22,000</td>
</tr>
<tr>
<td>K-means++</td>
<td>430</td>
<td>65</td>
</tr>
<tr>
<td>K-means</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**GAUSSMIXTURE:** 10,000 points in 15 dimensions  
K=50  
Costs scaled down by $10^4$

Photo credited to Bahmani
Scalable K-Means++

Steven Wu

University of Pennsylvania

October 3, 2014

Paper by Bahmani, Moseley, Vattani, Kumar and Vassilvitskii