**K-medoids for K-means seeding**

James Newling & Françoise Fleuret
(james.newling,francois.fleuret)@idiap.ch

**Abstract**

We present a scheme for K-means seeding, which results in a 3% geometric mean reduction in K-means loss as compared to vanilla K-means++ seeding, on 16 publicly available datasets. It is based on the CLARANS K-medoids algorithm of Ng and Han (1994).

**K-medoids**

Given samples \( \mathcal{X} = \{x(1), \ldots, x(N)\} \), function \( \text{dist} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+ \) and monotonic energy function \( \psi : \mathbb{R}^+ \rightarrow \mathbb{R}^+ \), find \( C \subset \{1, \ldots, N\} \) where \( |C| = K \) to minimize

\[
E = \sum_{i=1}^{N} \min_{j \in C \setminus \{i\}} \|x(i) - x(j)\|^2.
\]

It has applications in clustering sequences, graph vertices, sparse and dense vectors, etc. Two popular algorithms are,

- MEDLLOYD (Hastie et al. (2001), Park and Jun (2009)): like Lloyd’s algorithm, but centroids are replaced by medoids
- CLARANS (Ng and Han (1994)): random swaps between centers and non-centers are proposed, and only accepted if \( E(C) \) decreases.

**K-means and K-means seeding**

The K-means task is to find K centers, \( \{C(1), \ldots, C(K)\} \), not necessarily elements of \( \{x(1), \ldots, x(N)\} \), to minimize

\[
E = \sum_{i=1}^{N} \min_{k \in \{1,\ldots,K\}} \|x(i) - C(k)\|^2.
\]

In the popular LLOYD algorithm centers are initialized or seeded as a subset of \( \mathcal{X} \). Good seeding is critical to avoid poor local minima. Most seeding algorithms attempt to minimize initial energy (K-means++, Bradley-Fayad, etc.). Minimizing seeding energy is the special case of K-medoids with

\[
\text{dist}(x(i), x(j)) = \|x(i) - x(j)\|_2 \quad \text{and} \quad \psi(v) = v^2.
\]

This motivates the use of other popular and well-established K-medoids algorithms for K-means seeding.

**Acknowledgements**

This work was supported by HASLERSTIFTUNG

---

**The CLARANS K-medoids algorithm**

The algorithm iteratively proposes swapping a medoid \( x(i) \) with a non-medoid \( x(i') \). Only energy reducing swaps are implemented.

1. Initialize center indices \( C \subset \{1, \ldots, N\} \), where \( |C| = K \)
2. \( E \leftarrow \sum_{i=1}^{N} \min_{j \in C} \psi(\text{dist}(x(i), x(j))) \)
3. while stopping criterion false do
4. sample \( i \in C \) and \( i' \in \{1, \ldots, N\} \setminus C \)
5. \( E' \leftarrow \sum_{i=1}^{N} \min_{j \in C \cup \{i'\}} \psi(\text{dist}(x(i), x(j))) \)
6. if \( E' < E \) then
7. \( C \leftarrow C \setminus \{i\} \cup \{i'\} \)
8. \( E \leftarrow E' \)
9. end if
10. end while

**Five routes to K-means local minima**


**Accelerating the CLARANS algorithm**

There are many more evaluations (line 5) than implementations. Assuming balanced clusters, and that \( \text{dist} \) satisfies the triangle inequality, we present a technique where evaluation is \( O(N/K) \), and implementation is \( O(N) \). It requires recording,

- for non-centers (such as \( x(1) \) below), distance to nearest \( (d_1) \) and second nearest \( (d_2) \) centers (as in Ng and Han),
- for centers (such as \( x(2) \) below), maximum over cluster of \( d_1 \) and \( d_2 \) (\( R_1 \) and \( R_2 \) respectively), and distances to all centers.

**Results**

(Below) An experiment with 16 RNA datasets, \( N = 16 \times 10^4 \), \( d = 8 \) and \( K = 4 \times 10^2 \). With 50 runs seeded with K-medoids++ (red), and several runs with CLARANS in between K-means and K-medoids++ (blue). The best run without CLARANS has 6% higher \( E \).

---

**On github** Sequences, sparse vectors, various \( \text{dist}, \phi, \text{etc.} \)