$K$-medoids for $K$-means seeding

James Newling & François Fleuret

Machine Learning Group, Idiap Research Institute

November 28th, 2017
The standard $K$-means pipeline

First step: Seeding. Second step: Lloyd’s algorithm

simulated data
$K = 12^2, N = 25K$

uniform

$LLOYD$

$E = 0.105$

$LLOYD$

$E = 0.072$
The standard *K*-means pipeline (+CLARANS)

simulated data

$K = 12^2, N = 25K$

uniform

*K*-means++

CLARANS

LLOYD

$E = 0.032$

$E = 0.072$

$E = 0.105$

$E = 0.032$
Talk Outline

1) $K$-medoids, $K$-means++ and LLOYD

2) The CLARANS algorithm of Ng and Han (1994), algorithmic complexity and improvements

3) Results
K-medoids problem

Input:
- \( N \) samples \( \mathcal{X} = \{x(1), \ldots, x(N)\} \)
- Dissimilarity function \( dis : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} \)

Task:
- Find \( K \) indices \( c(1), \ldots, c(K) \in \{1, \ldots, N\} \) to minimize

\[
E = \sum_{i=1}^{N} \min_{k=1:K} dis(x(i), x(c(k))).
\]

- NP-hard.
$K$-means++ seeding

- Arthur and Vassilvitskii (2007)
- A $K$-medoids algorithm for

$$diss(x(i), x(i')) = \|x(i) - x(i')\|^2.$$  

1: select $c(1)$ uniformly from $\{1, \ldots, N\}$
2: **for** $k = 2 : K$ **do**
3: select $c(k) = i$ with prob $\sim \min_{k' < k} diss(x(i), x(c(k')))$$
4: **end for**

- Provides $8 \ln K + 2$ approximation bound to optimal $K$-means solution in expectation
LLOYD and MEDLLOYD

LLOYD for $K$-means:

1. $C(k) \leftarrow x(c(k))$ for $k \in \{1, \ldots, K\}$
2. while not converged do
3. for $i = 1 \rightarrow N$ set $a(i) = \arg\min_{k=1:K} \|x(i) - C(k)\|
4. for $k = 1 \rightarrow K$ set $C(k) \leftarrow \frac{\sum_{i:a(i)=k} x(i)}{\|i : a(i) = k\|}$
5. end while
LLOYD and MEDLLOYD

LLOYD for $K$-means:

1. $C(k) \leftarrow x(c(k))$ for $k \in \{1, \ldots, K\}$
2. while not converged do
3. for $i = 1 \rightarrow N$ set $a(i) = \arg\min_{k=1:K} \|x(i) - C(k)\|$
4. for $k = 1 \rightarrow K$ set $C(k) \leftarrow \frac{\sum_{i:a(i)=k} x(i)}{\|i : a(i) = k\|}$
5. end while

MEDLLOYD for $K$-medoids similar, with the constraint that centers are always samples,

$$C(k) \leftarrow x \left( \arg\min_{i:a(i)=k} \sum_{i':a(i')=k} \|x(i) - x(i')\|^2 \right)$$
A very simple $K$-medoids algorithm:

Randomly propose swaps between 1 center and 1 non-center, accept if $E$ decreases.
CLARANS
Demonstration

- center
  - non-center
1) propose a swap
CLARANS
Demonstration

1) propose a swap
2) evaluate energy
CLARANS
Demonstration

reject the swap

1) propose a swap
2) evaluate energy
3) implement or reject
1) propose a swap
1) propose a swap
2) evaluate energy
accept the swap

1) propose a swap
2) evaluate energy
3) implement or reject
1) propose a swap
2) evaluate energy
3) implement or reject

implement the swap
Advantages of CLARANS over MEDLLOYD (and LLOYD) are,

- updates are *not local*
- assignments and centers change simultaneously

Easy to show that,

- \( \{ \text{local minima of CLARANS} \} \subseteq \{ \text{local minima of MEDLLOYD} \}. \)
Ng and Han use $N^2$ dissimilarity matrix, infeasible now

Useful to distinguish between evaluate and implement steps:

1. **while** stopping condition is false **do**
2. randomly select center and non-center
3. **evaluate** proposal energy
4. **if** proposal energy lower **then**
5. implement proposed swap
6. **end if**
7. **end while**
We present different levels of optimization:

1. For all samples, record distances to nearest and second nearest centers ($d_1$ and $d_2$ respectively).
2. Also record for all clusters maximum $d_1$ and $d_2$, and inter-center distances. ($\triangle$)

Assuming largest cluster is $O(N/K)$,

<table>
<thead>
<tr>
<th></th>
<th>1.</th>
<th>2.</th>
</tr>
</thead>
<tbody>
<tr>
<td>evaluate</td>
<td>$O(N)$</td>
<td>$O(\frac{N}{K})$</td>
</tr>
<tr>
<td>implement</td>
<td>$O(N)$</td>
<td>$O(N)$</td>
</tr>
</tbody>
</table>

3. Terminate evaluation early if swap unpromising.
Empirical speed-up

\( N = 10^6/2, \ K = 10^3/2 \), data from 4-d Gaussian with \( I \) covariance.

\begin{align*}
\text{time [s]} & \quad 0:28 \quad 0:30 \quad 0:32 \quad 0:34 \quad 0:36 \\
\text{number of distance calculations} & \quad 10^9
\end{align*}
CLARANS for $K$-means seeding

Stopping criterion

Two possible stopping criteria

- Ng and Han stop after $R$ consecutive swap rejections
- Can stop after $S$ implementations (swap accepts)

We prefer a time based criterion

- If first seeding (with $K$-means++) takes $T_0$, stop after $\eta T_0$. 
Experiment 1

Subset of RNA dataset, $d = 8$, $N = 16 \times 10^4$, $K = 400$.

50 runs of $K$-means++ → LLOYD, and several with CLARANS. Number with CLARANS chosen so total times equal. Comparing best solutions, using CLARANS results in 6% lower $E$. 
Results Summary

Summary of experiments. Each point is an experiment with same setup as previous slide, horizontal position is reduction in $E$ obtained using CLARANS.

Dataset dimensions range in $d : 2 \rightarrow 90$, $N : 1484 \rightarrow 488565$. 
From conclusion of *K*-means++ paper of Arthur and Vassilvitskii (2007):

“Also, experiments showed that *k*-means++ generally performed better if it selected several new centers during each iteration, and then greedily chose the one that decreased $E$ as much as possible. Unfortunately, our proofs do not carry over to this scenario.”
Clustering library: zentas

\((\triangle): \text{diss}(x(i), x(i')) = \psi(\text{dist}(x(i), x(i'))), \) where

- \(\psi: \mathbb{R} \rightarrow \mathbb{R}\) is non-decreasing
- \(\text{dist}: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}\) satisfies the \(\triangle\)-inequality

Our software \texttt{zentas} implements accelerated CLARANS for different metrics. Levenshtein for sequence data, \(l_0, l_1, \ldots, l_\infty\) for sparse/dense vectors. Also fast \(K\)-means++, LLOYD, many others.
Conclusion

We discussed CLARANS, and how

- to accelerate it
- it improves seeding for $K$-means
- it is a versatile clustering algorithm in its own right.
The end

james.newling@idiap.ch