# A Sub-Quadratic Medoid Algorithm

James Newling & François Fleuret
{james.newling,francois.fleuret}@idiap.ch

## Set Medoid

Given set $\mathcal{S} = \{x(1), \dots, x(N)\}$, the *energy* of element $i \in \{1, \dots, N\}$ is,

$$E(i) = \frac{1}{N} \sum_{j \in \{1,\dots,N\}} \mathrm{dist}(x(i), x(j)).$$

The element in $\mathcal{S}$ with minimum energy is the *medoid*. The problem of finding the medoid arises in facility allocation, network analysis and clustering. In the general case, there is no sub-quadratic exact algorithm, but we present an $O(N^{3/2})$ algorithm in $\mathbb{R}^d$, which uses the triangle inequality to bound distances.

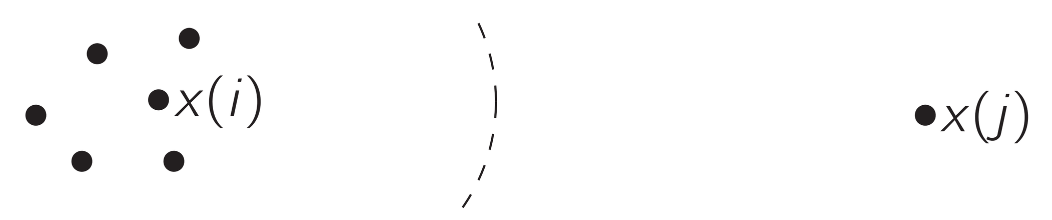
## Using the Triangle Inequality

When $E(i)$ is known, we can use

$$|E(i) - \mathrm{dist}(x(i), x(j))| \leq E(j)$$

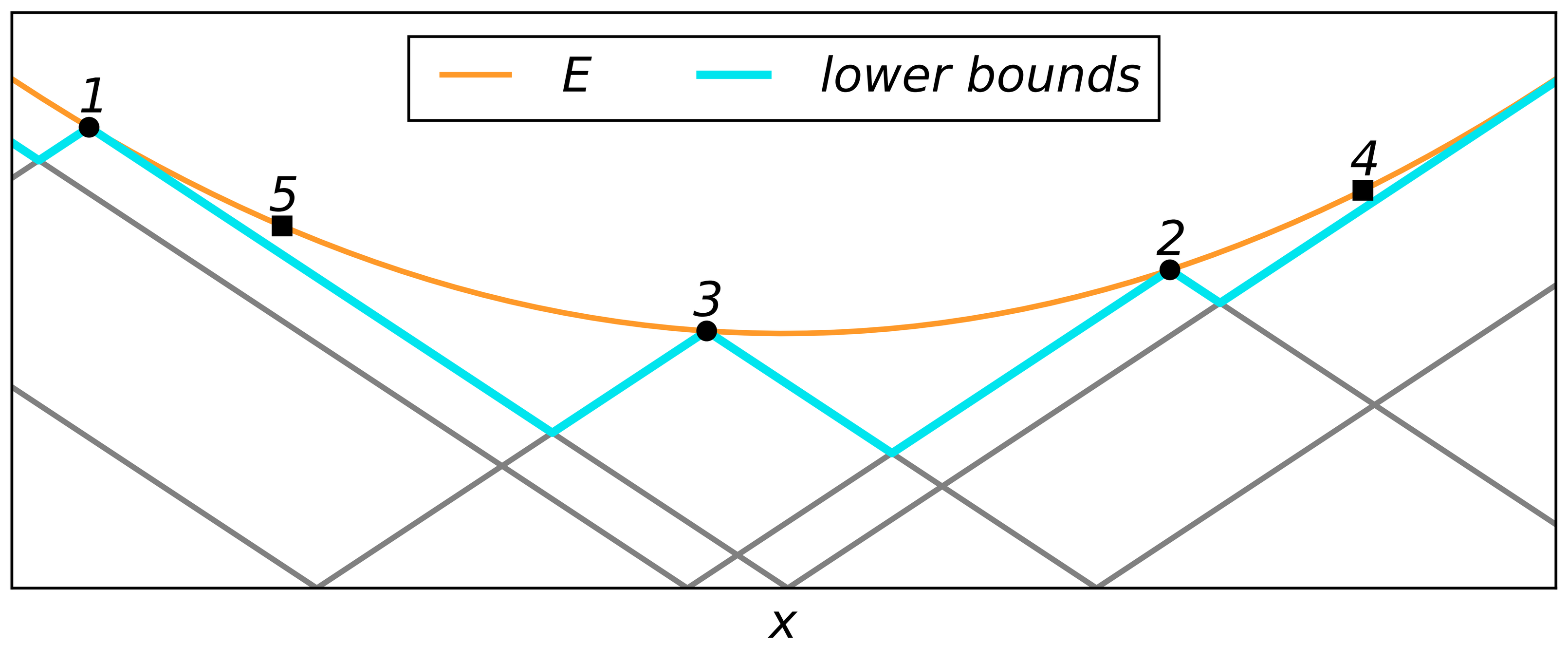
to eliminate $j$ as a medoid candidate, saving $N$ distance calculations.

case 1:




case 2:


The technique is effective when $x(i)$ and $x(j)$ are far away (case 1) or nearby (case 2). We keep lower bounds on all energies, updating them whenever distances are computed. Below, bounds for $x(4)$ and $x(5)$, obtained from distances to $x(1), x(2)$ and $x(3)$, eliminate them as medoid candidates.




## Proposed Algorithm

1: $l(i) \leftarrow 0$ for all $i$
2: $m^{cl}, E^{cl} \leftarrow -1, \infty$ the (c)urrent (l)owests
3: **for** $i \in \texttt{shuffle}(\{1, \dots, N\})$ **do**
4:   **if** $l(i) < E^{cl}$ **then**
5:     **for** $j \in \{1, \dots, N\}$ **do**
6:       $d(j) \leftarrow \mathrm{dist}(x(i), x(j))$
7:     **end for**
8:     $l(i) \leftarrow \frac{1}{N} \sum_{j=1}^{N} d(j)$
9:     **if** $l(i) < E^{cl}$ **then**
10:       $m^{cl}, E^{cl} \leftarrow i, l(i)$
11:     **end if**
12:     **for** $j \in \{1, \dots, N\}$ **do**
13:       $l(j) \leftarrow \max(l(j), |l(i) - d(j)|)$
14:     **end for**
15:   **end if**
16: **end for**

## Theoretical Results

**Theorem 3.1** *The algorithm finds the medoid.*

*proof summary.* Using the triangle inequality one can show that lower bounds remain consistent at line 13.

**Theorem 3.2** *Assume $\mathcal{S} = \{x(1), \dots, x(N)\} \subset \mathbb{R}^d$ are drawn independently from p.d.f. $f_X$. Let the medoid of $\mathcal{S}$ be $x(m^*)$, with $E(m^*) = E^*$. Suppose there exist strictly positive constants $\rho, \delta_0$ and $\delta_1$ such that for all N, with probability $1 - O(1/N)$*

$$\|x - x(m^*)\| < \rho \implies \delta_0 \leq f_X(x) \leq \delta_1,$$

*Let $\alpha > 0$ be a Lipschitz constant (independent of N) such that with probability $1 - O(1/N)$ all $i \in \{1, \dots, N\}$ satisfy,*

$$\|x(i) - x(m^*)\| < \rho \implies E(i) - E^* \geq \alpha \|x(i) - x(m^*)\|^2.$$

*Then the expected number of computed elements is*

$$O\left(V_d \delta_1 N^{\frac{1}{2}} + d \left(\frac{4}{\alpha}\right)^d N^{\frac{1}{2}}\right),$$

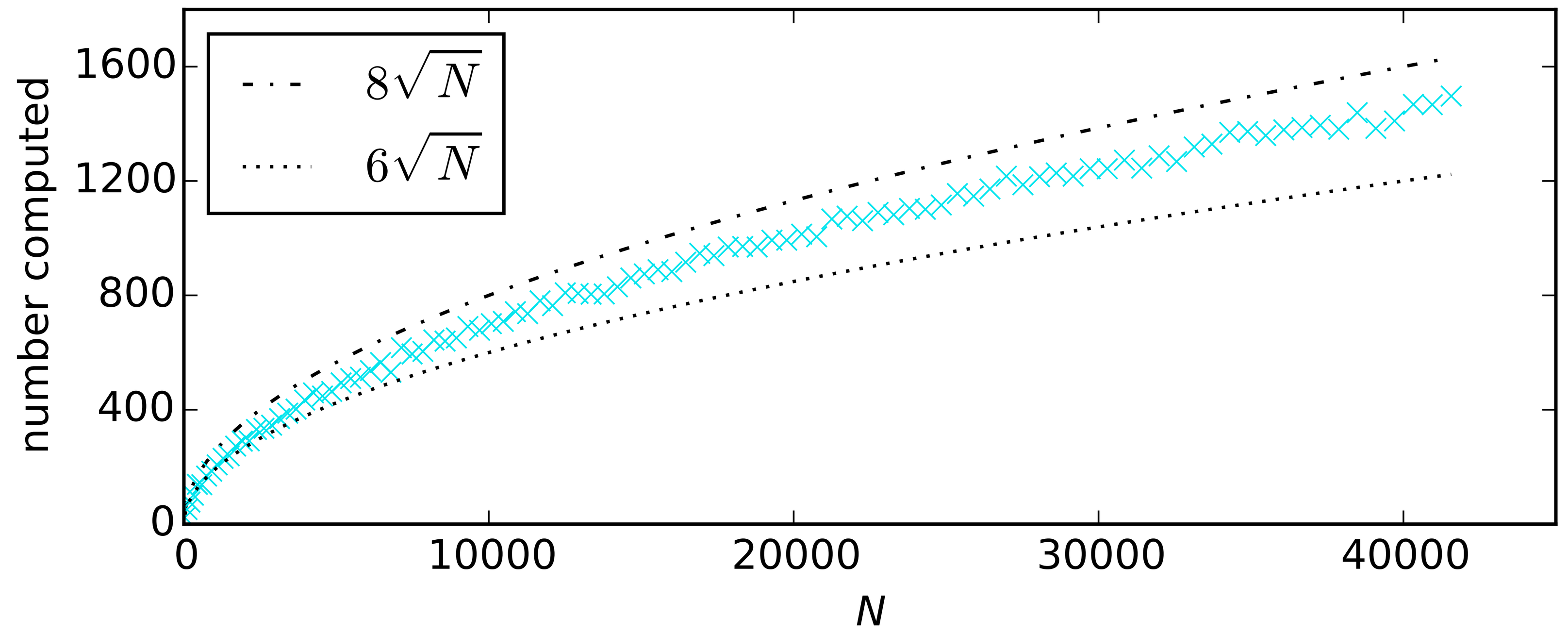
*where $V_d$ is the volume of a unit hypersphere in $\mathbb{R}^d$.*

*proof summary.* Case 1 eliminates far away elements. Case 2 creates *elimination balls*, the number of which beyond radius $N^{-1/2d}$ is bounded volumetrically (first term above). The expected number of sampled elements within radius $N^{-1/2d}$ is the second term.

## Previous Works

In 1-D Quickselect is $O(N)$. A related problem is finding the *geomteric median*: the point in a vector space which minimises energy. The most closely related algorithm to ours is TOPRANK of Okamoto et al. (2008), which estimates distances, and has complexity $\tilde{O}(N^{5/3})$.

## Experimental Results




Above: experimental validation of Theorem 3.2, where the number of computed elements is $O(N^{1/2})$. Samples are points drawn uniformly from $[0, 1]^2$

| | | | TOPRANK | Proposed Alg. |
|---|---|---|---|---|
| dataset | type | $N$ | $\hat{n}$ | $\hat{n}$ |
| Birch 1 | 2-d | $1.0 \times 10^5$ | 57944 | **2180** |
| Birch 2 | 2-d | $1.0 \times 10^5$ | 66062 | **2208** |
| Europe | 2-d | $1.6 \times 10^5$ | 176095 | **2862** |
| U-Sensor Net | u-graph | $3.6 \times 10^5$ | 113838 | **1593** |
| D-Sensor Net | d-graph | $3.6 \times 10^5$ | 99896 | **1372** |
| Penn. road | u-graph | $1.1 \times 10^6$ | 216390 | **2633** |
| Europe Rail | u-graph | $4.6 \times 10^4$ | 35913 | **518** |
| Gnutella | d-graph | $6.3 \times 10^3$ | 7043 | **6328** |
| MNIST | 784-d | $6.7 \times 10^3$ | 7472 | **6514** |

Above: The mean number of computed elements ($\hat{n}$) over 10 runs using TOPRANK and our proposed algorithm. Our algorithm displays good performance on spatial network data using the shortest path distance, but performs poorly on social network data (Gnutella) and in high-dimensions (MNIST), although TOPRANK does too.

## Acknowledgements

This work was sponsored by HASLERSTIFTUNG