SM-A On the difficulty of the medoid problem

We construct an example showing that no general purpose algorithm exists to solve the medoid problem in $o(N^2)$. Consider an almost fully connected graph containing $N = 2m + 1$ nodes, where the graph is exactly m edges short of being fully connected: one node has $2m$ edges and the others have $2m - 1$ edges. The graph has $2m^2$ edges. With the shortest path metric, it is easy to see that the node with $2m$ edges is the medoid, hence the medoid problem is as difficult as finding the node with $2m$ edges. But, supposing that the edges are provided as an unsorted adjacency list, it is clearly an $O(m^2)$ task to determine which node has $2m$ edges as one must look at all edges until a node with $2m$ edges is found. Thus determining the medoid is $O(m^2)$ which is $O(N^2)$.

SM-B `KMEDS` pseudocode

Alg. 2 presents the `KMEDS` algorithm of Park and Jun (2009), with the novel initialisation of `KMEDS` on line 1. `KMEDS` is essentially `lloyd`, with medoids instead of means.

Algorithm 2 `KMEDS` for clustering data $\{x(1), \ldots, x(N)\}$ around K medoids

1: Set all distances $D(i,j) \leftarrow \|x(i) - x(j)\|$ and sums $S(i) \leftarrow \sum_{j \in \{1,\ldots,N\}} D(i,j)$
2: Initialise medoid indices as K indices minimising $f(i) = \sum_{j \in \{1,\ldots,N\}} D(i,j)/S(j)$
3: **while** Some convergence criterion has not been met **do**
4: Assign each element to the cluster whose medoid is nearest to the element
5: Update cluster medoids according to assignments made above
6: **end while**

SM-C `RAND`, `TOPRANK` and `TOPRANK2` pseudocode

We present pseudocode for the `RAND`, `TOPRANK` and `TOPRANK2` algorithms of Okamoto et al. (2008), and discuss the explicit and implicit constants.

SM-C.1 On the number of anchor elements in `TOPRANK` : the constant in $\Theta(N^{\frac{2}{3}} (\log N)^{\frac{1}{3}})$

Note that the number of anchor points used in `TOPRANK` does not affect the result that the medoid is w.h.p. returned. However, Okamoto et al. (2008) show that by choosing the size of the anchor set to be $q (\log N)^{\frac{1}{3}}$ for any q, the run time is guaranteed to be $\tilde{O}(N^{5/3})$. They do not suggest a specific q, the optimal q being dataset dependant. We choose $q = 1$.

Consider Figure 3 in Section 5.1 for example, where $q = 1$. Had q be chosen to be less than 1, the line $\texttt{ncomputed} = N^{2/3} \log^{1/3} N$ to which `TOPRANK` runs parallel for large N would be shifted up or down by $\log q$, however the N at which the transition from $\texttt{ncomputed} = N^{2/3} \log^{1/3} N$ to $\texttt{ncomputed} = N^{2/3} \log^{1/3} N$ takes place would also change.

SM-C.2 On the parameter α' in `TOPRANK` and `TOPRANK2`

The threshold τ in (2) is proportional to the parameter α'. In Okamoto et al. (2008), it is stated that α' should be some value greater than 1. Note that the smaller α' is, the lower the threshold is, and hence fewer the number of computed points is, thus $\alpha' = 1.00001$ would be a fair choice. We use $\alpha' = 1$ in our experiments, and observe that the correct medoid is returned in all experiments.

Personal correspondence with the authors of Okamoto et al. (2008) has brought into doubt the proof of the result that the medoid is w.h.p. returned for any α' where $\alpha' > 1$. In our most recent correspondence, the authors suggest that the w.h.p. result can be proven with the more conservative bound of $\alpha' > \sqrt{1.5}$. Moreover, we show in SM-D that $\alpha' > 1$ is good enough to return the medoid with probability $N^{-(\alpha'-1)}$, a probability which still tends to 0 as N grows large, but not a w.h.p. result. Please refer to SM-D for further details on our correspondence with the authors.

SM-C.3 On the parameters specific to TOPRANK2

In addition to α', TOPRANK2 requires two parameters to be set. The first is l_0, the starting anchor set size, and the second is q, the amount by which l should be incremented at each iteration. Okamoto et al. (2008) suggest taking l_0 to be the number of top ranked nodes required, which in our case would be $l_0 = k = 1$. However, in our experience this is too small as all nodes lie well within the threshold and thus when l increases there is no change to number below threshold, which makes the algorithm break out of the search for the optimal l too early. Indeed, l_0 needs to be chosen so that at least some points have energies greater than the threshold, which in our experiments is already quite large. We choose $l_0 = \sqrt{N}$, as any value larger than $N^{2/3}$ would make TOPRANK2 redundant to TOPRANK. The parameter q we take to be $\log N$ as suggested by Okamoto et al. (2008).

Algorithm 3 RAND for estimating energies of elements of set S (Eppstein and Wang, 2004).

$I \leftarrow$ random uniform sample from $\{1, \dots, N\}$
// Compute all distances from anchor elements (I), using Dijkstra's algorithm on graphs
for $i \in I$ **do**
  for $j \in \{1, \dots, N\}$ **do**
    $d(i, j) \leftarrow \|x(i) - x(j)\|$,
  end for
end for
// Estimate energies as mean distances to anchor elements
for $j \in \{1, \dots, N\}$ **do**
  $\hat{E}(j) \leftarrow \frac{N}{|I|(N-1)} \sum_{i \in I} d(i, j)$
end for
return $\hat{E}$

Algorithm 4 TOPRANK for obtaining top k ranked elements of S (Okamoto et al., 2008).

$l \leftarrow N^{\frac{2}{3}} (\log N)^{\frac{1}{3}}$ // Okamoto et al. (2008) state that l should be $\Theta((\log N)^{\frac{1}{3}})$, the choice of 1 as the constant is arbitrary (see comments in the text of Section SM-C.1).
Run RAND with uniform random I of size l to get $\hat{E}(i)$ for $i \in \{1, \dots, N\}$.
Sort $\hat{E}$ so that $\hat{E}[1] \leq \hat{E}[2] \leq \dots \leq \hat{E}[N]$
$\hat{\Delta} \leftarrow 2 \min_{i \in I} \max_{j \in \{1, \dots, N\}} \|x(i) - x(j)\|$ // where $\|x(i) - x(j)\|$ computed in RAND
$Q \leftarrow \left\{ i \in \{1, \dots, N\} \mid \hat{E}(i) \leq \hat{E}[k] + 2\alpha' \hat{\Delta} \sqrt{\frac{\log(n)}{l}} \right\}$.
Compute exact energies of all elements in Q and return the element with the lowest energy.

SM-D On the proof that TOPRANK returns the medoid with high probability

Through correspondence with the authors of Okamoto et al. (2008), we have located a small problem in the proof that the medoid is returned w.h.p. for $\alpha' > 1$, the problem lying in the second inequality of Lemma 1. To arrive at this inequality, the authors have used the fact that for all i,

$$\mathbb{P}(E(i) \geq \hat{E}(i) + f(l) \cdot \Delta) \geq 1 - \frac{1}{2N^2}, \tag{8}$$

which is a simple consequence of the Hoeffding inequality as shown in Eppstein and Wang (2004). Essentially (8) says that, for a fixed node i, from which the mean distance to other nodes is $E(i)$, if one uniformly samples l distances to i and computes the mean $\hat{E}(i)$, the probability that $\hat{E}(i)$ is less than $E(i) + f(l)$ is greater than $1 - \frac{1}{2N^2}$.

The inequality (8) is true for a fixed node i. However, it no longer holds if i is selected to be the node with the lowest $\hat{E}(i)$. To illustrate this, suppose that $E(i) = 1$ for all i, and compute $\hat{E}(i)$ for all i. Let $\hat{E}^* = \arg\min_i \hat{E}(i)$. Now, we have a strong prior on $\hat{E}^*$ being significantly less than 1, and (8) no longer holds as a statement made about $\hat{E}^*$.

In personal correspondence, the authors show that the problem can be fixed by the use of an additional layer of union bounding, with a correction to be published (if not already done so at time of writing). However, the

Algorithm 5 TOPRANK2 for obtaining top k ranked elements of S (Okamoto et al., 2008).

// In Okamoto et al. (2008), it is suggested that l_0 be taken as k, which in the case of the medoid problem is 1. We have experimented with several choices for l_0, as discussed in the text.
$l \leftarrow l_0$
Run RAND with uniform random I of size l to get $\hat{E}(i)$ for $i \in \{1, \ldots, N\}$.
$\hat{\Delta} \leftarrow 2 \min_{i \in I} \max_{j \in \{1,\ldots,N\}} \|x(i) - x(j)\|$ // where $\|x(i) - x(j)\|$ computed in RAND
Sort $\hat{E}$ so that $\hat{E}[1] \leq \hat{E}[2] \leq \ldots \leq \hat{E}[N]$
$Q \leftarrow \left\{ i \in \{1, \ldots, N\} \mid \hat{E}(i) \leq \hat{E}[k] + 2\alpha' \hat{\Delta} \sqrt{\frac{\log(n)}{l}} \right\}$.
$g \leftarrow 1$
while g is 1 **do**
  $p \leftarrow |Q|$
  // The recommendation for q in Okamoto et al. (2008) is $\log(n)$, we follow the suggestion
  Increment I with q new anchor points
  Update $\hat{E}$ for all data according to new anchor points
  $l \leftarrow |I|$
  $\hat{\Delta} \leftarrow 2 \min_{i \in I} \max_{j \in \{1,\ldots,N\}} \|x(i) - x(j)\|$
  Sort $\hat{E}$ so that $\hat{E}[1] \leq \hat{E}[2] \leq \ldots \leq \hat{E}[N]$
  $Q \leftarrow \left\{ i \in \{1, \ldots, N\} \mid \hat{E}(i) \leq \hat{E}[k] + 2\alpha' \hat{\Delta} \sqrt{\frac{\log(n)}{l}} \right\}$
  $p' \leftarrow |Q|$
  if $p - p' < \log(n)$ **then**
    $g \leftarrow 0$
  end if
end while
Compute exact energies of all elements in Q and return the element with the lowest energy

additional layer of union bound requires a more conservative constraint on α', which is $\alpha' > 2$, although the authors propose that the w.h.p. result can be proven with $\alpha' > \sqrt{1.5}$ for N sufficiently large. We now present a small proof proving the w.h.p. result for $\alpha' > \sqrt{2}$ for N sufficiently large, with at the same time $\alpha' > 1$ guaranteeing that the medoid is returned with probability $O(N^{\alpha'-1})$.

SM-D.1 That the medoid is returned *with high probability* holds for $\alpha' > \sqrt{2}$ and that *with vanishing probability* it is returned for $\alpha' > 1$

Recall that we have N nodes with *energies* $E(1), \ldots, E(n)$. We wish to find the k lowest energy nodes (the original setting of Okamoto et al. (2008)). From Hoeffding's inequality we have,

$$\mathbb{P}(|E(i) - \hat{E}(i)| \geq \epsilon\Delta) \leq 2\exp\left(-l\epsilon^2\right). \tag{9}$$

Set the probability on the right hand side of 9 to be $2/N^{1+\beta}$, that is,

$$2\exp\left(-l\epsilon^2\right) = 2/N^{1+\beta},$$

which corresponds to

$$\epsilon = \sqrt{\left(\frac{1+\beta}{l}\right)\log(N)} \quad := \quad \tilde{f}(l).$$

Clearly $\sqrt{1+\beta}$ corresponds to α'. With this notation we have,

$$\mathbb{P}(|E(i) - \hat{E}(i)| \geq \tilde{f}(l)\Delta) \leq \frac{2}{N^{1+\beta}}. \tag{10}$$

Applying the union bound to (10) we have,

$$\mathbb{P}\left(\neg\left(\wedge_{i \in \{1,\ldots,N\}} |E(i) - \hat{E}(i)| \leq \tilde{f}(l)\Delta\right)\right) \leq \frac{2}{N^{\beta}}. \tag{11}$$

			$K = 10$		$K = \lceil\sqrt{N}\rceil$		$K = \lceil \frac{N}{10} \rceil$	
Dataset	N	d	μ_u/μ_{park}	σ_u/μ_{park}	μ_u/μ_{park}	σ_u/μ_{park}	μ_u/μ_{park}	σ_u/μ_{park}
gassensor	256	128	1.09	0.08	**0.90**	0.03	**0.83**	0.01
house16H	1927	17	1.01	0.02	**0.97**	0.01	**0.93**	0.01
S1	5000	2	1.05	0.05	**0.75**	0.01	**0.32**	0.01
S2	5000	2	1.04	0.07	**0.68**	0.01	**0.34**	0.00
S3	5000	2	1.03	0.05	**0.76**	0.01	**0.35**	0.00
S4	5000	2	1.02	0.03	**0.75**	0.01	**0.41**	0.01
A1	3000	2	**0.82**	0.03	**0.43**	0.01	**0.19**	0.00
A2	5250	2	**0.98**	0.03	**0.47**	0.01	**0.25**	0.00
A3	7500	2	**0.96**	0.02	**0.42**	0.02	**0.22**	0.00
thyroid	215	5	**0.95**	0.08	**0.97**	0.04	**0.93**	0.04
yeast	1484	8	1.00	0.02	**0.96**	0.02	**0.91**	0.02
wine	178	14	1.01	0.02	1.02	0.01	**0.98**	0.02
breast	699	9	**0.79**	0.03	**0.77**	0.02	**0.68**	0.02
spiral	312	3	1.03	0.03	**0.99**	0.02	**0.82**	0.03

Table 3: Comparing the initialisation scheme proposed in Park and Jun (2009) with random uniform initialisation for the `KMEDS` algorithm. The final energy using the deterministic scheme proposed in Park and Jun (2009) is μ_{park}. The mean over 10 random uniform initialisations is μ_u, and the corresponding standard deviation is σ_u. For small K ($K = 10$), the performances using the two schemes are comparable, while for larger K, it is clear that uniform initialisation performs much better on the majority of datasets.

Recall that we wish to obtain the k nodes with lowest energy. Denote by $r(j)$ the index of the node with the j'th lowest energy, so that

$$E(r(1)) \leq \ldots \leq E(r(j)) \leq \ldots \leq E(r(N)).$$

Denote by $\hat{r}(j)$ the index of the node with the j'th lowest estimated energy, so that

$$\hat{E}(\hat{r}(1)) \leq \ldots \leq \hat{E}(\hat{r}(j)) \leq \ldots \leq \hat{E}(\hat{r}(N)).$$

Now assume that for all i, it is true that $|E(i) - \hat{E}(i)| \leq \tilde{f}(l)$. Then consider, for $j \leq k$,

$$\begin{aligned} \hat{E}(\hat{r}(k)) - \hat{E}(r(j)) &= \underbrace{\Big(\hat{E}(\hat{r}(k)) - E(r(k))\Big)}_{\geq -\tilde{f}(l)\Delta} + \underbrace{\Big(E(r(k)) - E(r(j))\Big)}_{\geq 0} + \underbrace{\Big(E(r(j)) - \hat{E}(r(j))\Big)}_{\geq -\tilde{f}(l)\Delta}, \\ &\geq -2\tilde{f}(l)\Delta. \end{aligned} \tag{12}$$

The first bound in (12) is obtained by considering the most extreme case possible under the assumption, which is $\hat{E}(i) = a(E) - \tilde{f}(l)$ for all i. The second bound follows from $j \leq k$, and the third bound follows directly from the assumption. We thus have that, under the assumption,

$$\hat{E}(r(j)) \leq \hat{E}(\hat{r}(k)) + 2\tilde{f}(l)\Delta,$$

which says that all nodes of rank less than or equal to k have approximate energy less than $\hat{E}(\hat{r}(k)) + 2\tilde{f}(l)\Delta$. As the assumption holds with probability greater than $1 - 2/N^{\beta}$ by (11), we are done. Take $\beta = 1$ if you want the statement *with high probability*, that is

$$\epsilon = \sqrt{\frac{2\log(n)}{l}},$$

but for any $\beta > 0$, which corresponds to $\alpha' > 1$, the probability of failing to return the k lowest energy nodes tends to 0 as N grows.

SM-E On the initialisation of Park and Jun (2009)

In Table 3 we present the full results of the 48 experiments comparing the initialisation proposed in Park and Jun (2009) with simple uniform initialisation. The 14 datasets are all available from `https://cs.joensuu.fi/sipu/datasets/`.

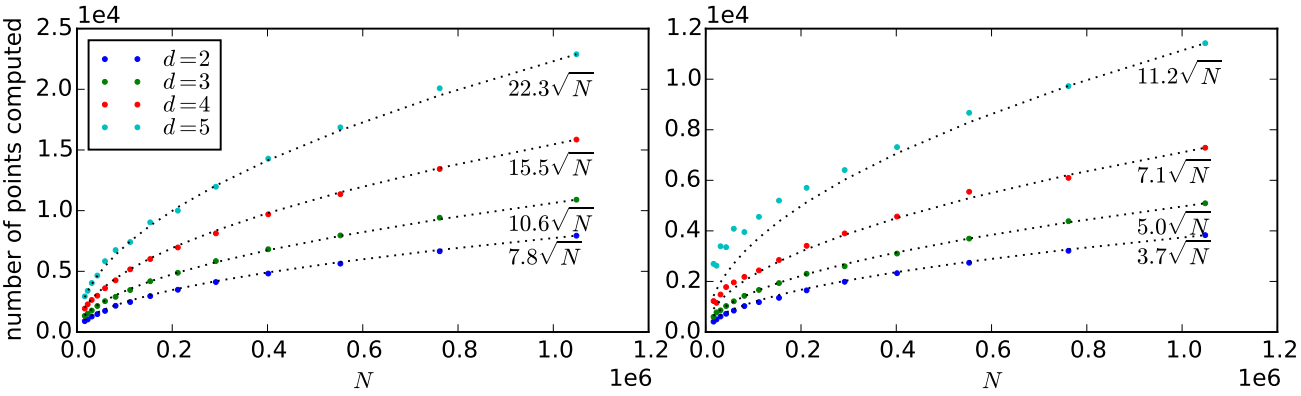

Figure 4: Number of points computed on simulated data. Points are drawn from $\mathcal{B}_d(0,1)$, for $d \in \{2,3,4,5\}$. On the left, points are drawn uniformly, while on the right, the density in $\mathcal{B}_d(0,(1/2)^{1/d})$ is 19× lower that in $\mathcal{A}_d(0,(1/2)^{1/2},1)$, where recall that $\mathcal{A}_d(x,r_1,r_2)$ denotes an annulus centred at x of inner radius r_1 and outer radius r_2. We observe a near perfect fit of the number of computed points to $\xi\sqrt{N}$ where the constant ξ depends on the dimension and the distribution (left and right). The number of computed points increases with dimension. The strong convexity constant of the distribution on the right is larger, corresponding to fewer distance calculations as predicted by Theorem 3.2.

SM-F Scaling with α, N, and dimension d

We perform more experiments to provide further validation of Theorem 3.2. In particular, we check how the number of computed elements scales with N, d and α. We generate data from a unit ball in various dimensions, according to two density functions with different strong convexity constants α. The first density function is uniform, so that the density everywhere in the ball is uniform. To sample from this distribution, we generate two random variables, $X_1 \sim \mathcal{N}_d(0,1)$ and $X_2 \sim U(0,1)$ and use

$$X_3 = X_1/\|X_1\| \cdot X_2^{\frac{1}{d}}, \tag{13}$$

as a sample from the unit ball $\mathcal{B}_d(0,1)$ with uniform distribution. The second distribution we consider has a higher density beyond radius $(1/2)^{1/d}$. Specifically, within this radius the density is 19× lower than beyond this radius. To sample from this distribution, we sample X_3 according to (13), and then points lying within radius $(1/2)^{1/d}$ are with probability 1/10 re-sampled uniformly beyond this radius.

The second distribution has a larger strong convexity constant α. To see this, note that the strong convexity constant at the center of the ball depends only on the density of the ball on its surface, that is at radius 1, as can be shown using an argument based on cancelling energies of internal points. As the density at the surface under distribution 2 is approximately twice that of under distribution 1, the change in energy caused by a small shift in the medoid is twice as large under distribution 2. Thus, according to Theorem 3.2, we expect the number of computed points to be larger under distribution 1 than under distribution 2. This is what we observe, as shown in Figure 4, where distribution 1 is on the left and distribution 2 is on the right.

In Figure 4 we observe a near perfect $N^{1/2}$ scaling of number of computed points. Dashed curves are exact $N^{1/2}$ relationships, while the coloured points are the observed number of computed points.

SM-G Proof of Theorem 3.2 (See page 5)

Theorem 3.2. *Let $\mathcal{S} = \{x(1),\dots,x(N)\}$ be a set of N elements in $\mathbb{R}^d$, drawn independently from probability distribution function f_X. Let the medoid of $\mathcal{S}$ be $x(m^*)$, and let $E(m^*) = E^*$. Suppose that there exist strictly positive constants ρ, δ_0 and δ_1 such that for any set size N with probability $1 - O(1/N)$*

$$x \in \mathcal{B}_d(x(m^*),\rho) \implies \delta_0 \leq f_X(x) \leq \delta_1, \tag{6}$$

where $\mathcal{B}_d(x,r) = \{x' \in \mathbb{R}^d \ : \ \|x'-x\| \leq r\}$. *Let* $\alpha > 0$ *be a constant (independent of* N*) such that with probability* $1 - O(1/N)$ *all* $i \in \{1,\dots,N\}$ *satisfy,*

$$x(i) \in \mathcal{B}_d(x(m^*),\rho) \implies E(i) - E^* \geq \alpha \|x(i) - x(m^*)\|^2. \tag{7}$$

Then, the expected number of elements computed by `trimed` *is* $O\left(\left(V_d[1]\delta_1 + d\left(\frac{4}{\alpha}\right)^d\right)N^{\frac{1}{2}}\right)$, *where* $V_d[1] = \pi^{\frac{d}{2}}/(\Gamma(\frac{d}{2}+1))$ *is the volume of* $\mathcal{B}_d(0,1)$.

Proof. We show that the assumptions made in Th. 3.2 validate the assumptions required in Thm SM-G.1. Firstly, if $e(i) > \rho$ then $e(i) \geq \alpha\rho^2 e(i) > \rho$, which follows from the convexity of the loss function and. Secondly, the existance of β follows from continuity of the gradient of the distance, combined with the existence of δ_1 (non-exploding). □

Theorem SM-G.1 (Main Theorem Expanded)**.** *Let* $\mathcal{S} = \{x(1),\dots,x(N)\} \subset \mathbb{R}^d$ *have medoid* $x(m^*)$ *with minimum energy* $E(m^*) = E^*$, *where elements in* $\mathcal{S}$ *are drawn independently from probability distribution function* f_X. *Let* $e(i) = \|x(i) - x(m^*)\|$. *Suppose that for* f_X *there exist strictly positive constants* $\alpha, \beta, \rho, \delta_0$ *and* δ_1 *satisfying,*

$$x \in \mathcal{B}_d(x(m^*),\rho) \implies \delta_0 \leq f_X(x) \leq \delta_1, \tag{14}$$

where $\mathcal{B}_d(x,r) = \{x' \in \mathbb{R}^d \ : \ \|x'-x\| \leq r\}$, *and that for any set size* N, *w.h.p. all* $i \in \{1,\dots,N\}$ *satisfy,*

$$E(i) - E^* \geq \begin{cases} \alpha e(i)^2 & \text{if} \quad e(i) \leq \rho, \\ \alpha\rho^2 & \text{if} \quad e(i) > \rho, \end{cases} \tag{15}$$

and,

$$E(i) - E^* \leq \beta e(i)^2 \quad \text{if} \quad e(i) \leq \rho. \tag{16}$$

Then the expected number of elements computed, which is to say not eliminated on line 4 of `trimed`, *is* $O\left(\left(V_d[1]\delta_1 + d\left(\frac{4}{\alpha}\right)^d\right)N^{\frac{1}{2}}\right)$, *where* $V_d[1] = \pi^{\frac{d}{2}}/(\Gamma(\frac{d}{2}+1))$ *is the volume of* $\mathcal{B}_d(0,1)$.

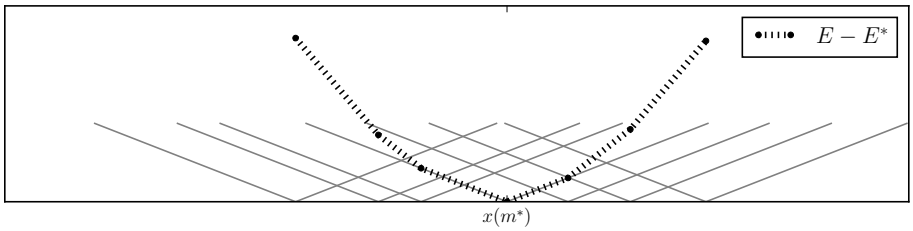

Figure 5: A sum of uniformly distributed cones is approximately quadratic.

Proof. We first show that the expected number of computed elements in $\mathcal{B}_d(x(m^*), N^{-\frac{1}{2d}})$ is $O(V_d[1]\delta_1 N^{\frac{1}{2}})$. When N is sufficiently large, $f_X(x) \leq \delta_1$ within $\mathcal{B}_d(x(m^*), N^{-\frac{1}{2d}})$. The expected number of samples in $\mathcal{B}_d(x(m^*), N^{-\frac{1}{2d}})$ is thus upper bounded by δ_1 multiplied by the volume of the ball. But the volume of a ball of radius $N^{-\frac{1}{2d}}$ in $\mathbb{R}^d$ is $V_d[1]N^{-\frac{1}{2}}$.

In Lemma SM-G.2 we use a packing argument to show that the number of computed elements in the annulus $\mathcal{A}_d(x(m^*), N^{-\frac{1}{2d}}, \infty)$ is $O\left(d\left(\frac{4}{\alpha}\right)^d N^{\frac{1}{2}}\right)$, but we there assume that the medoid index m^* is the first element in `shuffle`$(\{1,\dots,N\})$ on line 3 of `trimed` and thus that the medoid energy is known from the first iteration ($E^{cl} = E^*$). We now extend Lemma SM-G.2 to the case where the medoid is not the first element processed. We do this by showing that w.h.p. an element with energy very close to E^* has been computed after $N^{-\frac{1}{2}}$ iterations of `trimed`, and thus that the bounds on numbers of computed elements obtained using the packing arguments underlying Lemma SM-G.2 are all correct to within some small factor after $N^{-\frac{1}{2}}$ iterations.

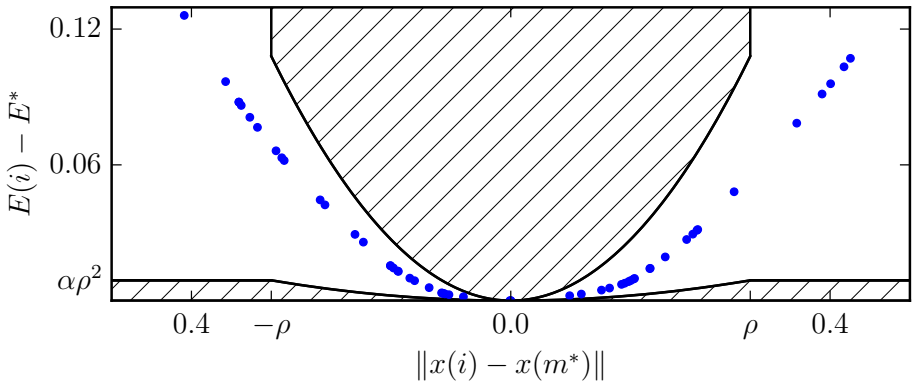

Figure 6: Illustrating the parameters α, β and ρ of Theorem 3.2. Here we draw $N = 101$ samples uniformly from $[-1, 1]$ and compute their energies, plotted here as the series of points. Theorem 3.2 states that their exists α, β and ρ such that irrespective of N, all energies (points) will lie in the envelope (non-hatched region).

The probability of a sample lying within radius $N^{-\frac{2}{3d}}$ of $x(m^*)$ is $\Omega(\delta_0 N^{-\frac{2}{3}})$, and so the probability that none of the first $N^{\frac{1}{2}}$ samples lies within radius $N^{-\frac{2}{3d}}$ is $O((1-\delta_0 N^{-\frac{2}{3d}})^{N^{\frac{1}{2}}})$ which is $O(\frac{1}{N})$. Thus w.h.p. after $N^{\frac{1}{2}}$ iterations of `trimed`, E^{cl} is within $\beta N^{-\frac{4}{3d}}$ of E^*, which means that the radii of the balls used in the packing argument are overestimated by at most a factor $N^{-\frac{1}{3d}}$. Thus w.h.p. the upper bounds obtained with the packing argument are correct to within a factor $1 + N^{-\frac{1}{3}}$. The remaining $O(\frac{1}{N})$ cases do not affect the expectation, as we know that no more than N elements can be computed. □

Lemma SM-G.2 (Packing beyond the vanishing radius)**.** *If we assume* (15) *from Theorem 3.2 and that the medoid index m^* is the first element processed by* `trimed`*, then the number of elements computed in $\mathcal{A}_d(x(m^*), N^{-\frac{1}{2d}}, \infty)$ is $O\left(d\left(\frac{4}{\alpha}\right)^d N^{\frac{1}{2}}\right)$.*

Proof. Follows from Lemmas SM-G.3 and SM-G.4. □

Lemma SM-G.3 (Packing from the vanishing radius $N^{-\frac{1}{d}}$ to ρ)**.** *If we assume* (15) *from Theorem 3.2 and that the medoid index m^* is the first element processed in* `trimed`*, then the number of computed elements in $\mathcal{A}(x(m^*), N^{-\frac{1}{2d}}, \rho)$ is $O(d\left(\frac{4}{\alpha}\right)^d N^{\frac{1}{2}})$.*

Proof. According to Assumption 15, an element at radius $r < \rho$ has surplus energy at least αr^2. This means that, assuming that the medoid has already been computed, an element computed at radius r will be surrounded by an exclusion zone of radius αr^2 in which no element will subsequently be computed. We will use this fact to upper bound the number of computed elements in $\mathcal{A}(x(m^*), N^{-\frac{1}{2d}}, \rho)$, firstly by bounding the number in an annulus of inner radius r and width αr^2, that is the annulus $\mathcal{A}_d(x(m^*), r, r + \alpha r^2)$, and then summing over concentric rings of this form which cover $\mathcal{A}(x(m^*), N^{-\frac{1}{2d}}, \rho)$. Recall that the number of computed elements in $\mathcal{A}_d(x(m^*), r, r + \alpha r^2)$ is denoted by $N_c(x(m^*), r, r + \alpha r^2)$.

We use Lemma SM-G.5 to bound $N_c(x(m^*), r, r+\alpha r^2)$,

$$\begin{aligned}
N_c(x(m^*), r, r+\alpha r^2) &\leq (d+1)^2 \left(\frac{4}{\sqrt{3}}\right)^d \frac{\alpha r^2 (r+\alpha r^2)^{d-1}}{(\alpha r^2)^d} \\
&\leq (d+1)^2 \left(\frac{4}{\sqrt{3}}\right)^d \left(1+\frac{1}{\alpha r}\right)^{d-1} \\
&\leq (d+1)^2 \left(\frac{4}{\sqrt{3}}\right)^d \left(\max\left(2, \frac{2}{\alpha r}\right)\right)^{d-1} \\
&\leq (d+1)^2 \left(\frac{4}{\sqrt{3}}\right)^d \left(\max\left(2^{d-1}, \left(\frac{2}{\alpha r}\right)^{d-1}\right)\right) \\
&\leq (d+1)^2 \left(\frac{4}{\sqrt{3}}\right)^d \left(2^{d-1} + \left(\frac{2}{\alpha r}\right)^{d-1}\right) \\
&\leq (d+1)^2 \left(\frac{8}{\sqrt{3}}\right)^d + (d+1)^2 \left(\frac{8}{\sqrt{3}}\right)^d \left(\frac{1}{\alpha r}\right)^{d-1}
\end{aligned}$$

Let $r_0 = N^{-\frac{1}{2d}}$ and $r_{i+1} = r_i + \alpha r_i^2$, and let T be the smallest index i such that $r_i \leq \rho$. With this notation in hand, we have

$$N_c(x(m^*), N^{-\frac{1}{2d}}, \rho) \leq \sum_{i=0}^{T} N_c(x(m^*), r_i, \alpha r_i + r_i^2).$$

The summation on the right-hand side can be upper-bounded by an integral. Using that the difference between r_i and r_{i+1} is αr_i^2, we need to divide terms in the sum by αr_i^2 when converting to an integral. Doing this, we obtain,

$$\begin{aligned}
N_c(x(m^*), N^{-\frac{1}{2d}}, \rho) &\leq \int_{N^{-\frac{1}{2d}}}^{\rho+\alpha\rho^2} N_c(x(m^*), r, \alpha r^2) dr \\
&\leq \text{const } + (d+1)^2 \left(\frac{8}{\sqrt{3}}\right)^d \left(\frac{1}{\alpha}\right)^d \int_{N^{-\frac{1}{2d}}}^{\infty} r^{-(1+d)} dr \\
&\leq \text{const } + (d+1) \left(\frac{4}{\alpha}\right)^d N^{\frac{1}{2}}.
\end{aligned}$$

This completes the proof, and provides the hidden constant of complexity as $(d+1)\left(\frac{4}{\alpha}\right)^d$. Thus larger values for α should result in fewer computed elements in the annulus $\mathcal{A}_d(x(m^*), r, r+\alpha r^2)$, which makes sense given that large values of α imply larger surplus energies and thus larger elimination zones. □

Lemma SM-G.4 (Packing beyond ρ)**.** *If we assume* (15) *from Theorem 3.2 and that the medoid index m^* is the first element processed by* `trimed`*, then the number of computed elements in $\mathcal{A}_d(x(m^*), \rho, \infty)$ is less than $(1 + 4E^*/(\alpha\rho^2))^d$.*

Proof. Recall that we at assuming $m^* = 1$, that is that the medoid is the first element processed in `trimed`. All elements beyond radius $2E^*$ are eliminated by type 1 eliminations (Figure 1), which provides the first inequality below. Then, as the excess energy is at least $\epsilon = \alpha\rho^2$ for all elements beyond radius ρ of $x(m^*)$, we apply Lemma SM-G.8 with $\epsilon = \alpha\rho^2/2$ to obtain the second inequality below,

$$\begin{aligned}
N_c(m(x), \rho, \infty) &\leq N_c(m(x), \rho, 2E^*) \\
&\leq \frac{(2E^* + \frac{1}{2}\alpha\rho^2)^d}{(\frac{1}{2}\alpha\rho^2)^d} \\
&\leq \left(1 + \frac{4E^*}{\alpha\rho^2}\right)^d.
\end{aligned}$$

□

Lemma SM-G.5 (Annulus packing). *For $0 \leq r$ and $0 < \epsilon \leq w$. If*

$$\mathcal{X} \subset \mathcal{A}_d(0, r, r+w),$$

where

$$\forall x \in \mathcal{X}, \mathcal{B}_d(x, \epsilon) \cup \mathcal{X} = \{x\}, \tag{17}$$

then,

$$|\mathcal{X}| \leq (d+1)^2 \left(\frac{4}{\sqrt{3}}\right)^d \frac{w\,(r+w)^{d-1}}{\epsilon^d}.$$

Proof. The condition (17) implies,

$$\forall x, x' \in \mathcal{X} \times \mathcal{X}, \mathcal{B}\left(x, \frac{\epsilon}{2}\right) \cup \mathcal{B}\left(x', \frac{\epsilon}{2}\right) = \emptyset. \tag{18}$$

Using that $\epsilon \in (0, w]$ and Lemma SM-G.6, one can show that for all $x \in \mathcal{A}(0, r, r+w)$,

$$\text{volume}\left(\mathcal{B}\left(x, \frac{\epsilon}{2}\right) \cap \mathcal{A}(0, r, r+w)\right) > \frac{1}{d+1}\left(\frac{3}{4}\right)^{\frac{d}{2}} V_d\left[\frac{\epsilon}{2}\right] \tag{19}$$

Combining (18) with (19) we have,

$$\text{volume}\left(\bigcup_{x \in \mathcal{X}} \mathcal{B}\left(x, \frac{\epsilon}{2}\right) \cap \mathcal{A}(0, r, r+w)\right) > \frac{V_d\,[1]}{d+1}\left(\frac{\sqrt{3}}{4}\right)^d |\mathcal{X}|\epsilon^d. \tag{20}$$

Letting $S_d\,[\epsilon]$ denote the surface area of a $\mathcal{B}(0, \epsilon)$, it is easy to see that

$$\text{volume}\left(\mathcal{A}(0, r, r+w)\right) < S_d\,[1]\, w\,(r+w)^{d-1}. \tag{21}$$

Combining (20) with (21) we get,

$$\frac{V_d\,[1]}{d+1}\left(\frac{\sqrt{3}}{4}\right)^d |\mathcal{X}|\epsilon^d < S_d\,[1]\, w\,(r+w)^{d-1}.$$

which combined with the fact that

$$\begin{aligned}\frac{S_d\,[1]}{V_d\,[1]} &= \left(\frac{\frac{dV_d}{dr}}{V_d}\right)_{r=1} \\ &= d,\end{aligned}$$

provides us with,

$$|\mathcal{X}| \leq (d+1)^2 \left(\frac{4}{\sqrt{3}}\right)^d \frac{w\,(r+w)^{d-1}}{\epsilon^d}.$$

□

Lemma SM-G.6 (Volume of ball intersection). *For $x_0, x_1 \in \mathbb{R}^d$ with $\|x_0 - x_1\| = 1$,*

$$\frac{\text{volume}\left(\mathcal{B}_d\left(x_0, 1\right) \cap \mathcal{B}_d\left(x_1, 1\right)\right)}{\text{volume}\left(\mathcal{B}_d\left(x_0, 1\right)\right)} \geq \frac{1}{d+1}\left(\frac{3}{4}\right)^{\frac{d}{2}}.$$

Proof. Let $V_d[r]$ denote the volume of $\mathcal{B}_d(0,r)$. It is easy to see that,

$$\begin{aligned}
\text{volume}\left(\mathcal{B}_d(x_0,1)\cap\mathcal{B}_d(x_1,1)\right) &= 2\int_0^{\frac{1}{2}} V_{d-1}\left[\sqrt{x(2-x)}\right]dx \\
&\geq 2\int_0^{\frac{1}{2}} V_{d-1}\left[\sqrt{\frac{3}{2}x}\right]dx \\
&\geq 2V_{d-1}[1]\int_0^{\frac{1}{2}}\left(\frac{3}{2}x\right)^{\frac{d-1}{2}}dx \\
&\geq 2V_{d-1}[1]\left(\frac{3}{2}\right)^{\frac{d-1}{2}}\left(\frac{2}{d+1}\right)\left(\frac{1}{2}\right)^{\frac{d+1}{2}} \\
&\geq V_{d-1}[1]\left(\frac{3}{2}\right)^{\frac{d-1}{2}}\left(\frac{2}{d+1}\right)\left(\frac{1}{2}\right)^{\frac{d-1}{2}} \\
&\geq V_{d-1}[1]\left(\frac{3}{4}\right)^{\frac{d-1}{2}}\left(\frac{2}{d+1}\right).
\end{aligned}$$

Using that $\dfrac{V_{d-1}[1]}{V_d[1]} > \dfrac{1}{\sqrt{\pi}}$, we divide the intersection volume through by $V_d[1]$ to obtain,

$$\begin{aligned}
\frac{\text{volume}\left(\mathcal{B}_d(x_0,1)\cap\mathcal{B}_d(x_1,1)\right)}{\text{volume}\left(\mathcal{B}_d(x_0,1)\right)} &\geq \left(\frac{3}{4}\right)^{\frac{d-1}{2}}\left(\frac{2}{\sqrt{\pi}(d+1)}\right) \\
&\geq \frac{1}{d+1}\left(\frac{3}{4}\right)^{\frac{d}{2}}
\end{aligned}$$

□

Lemma SM-G.7 (Packing balls in a ball)**.** *The number of non-intersecting balls of radius ϵ which can be packed into a ball of radius r in $\mathbb{R}^d$ is less than $\left(\frac{r}{\epsilon}\right)^d$*

Proof. The technique used here is a loose version of that used in proving Lemma SM-G.5. The volume of $\mathcal{B}_d(0,\epsilon)$ is a factor $(r/\epsilon)^d$ smaller than that of $\mathcal{B}_d(0,r)$. As the balls of radius ϵ are non-overlapping, the volume of their union is simply the sum of their volumes. The result follow from the fact that the union of the balls of radius ϵ is contained within the ball of radius r. □

Lemma SM-G.8 (Packing points in a ball)**.** *Given $\mathcal{X}\subset\mathcal{B}_d(0,r)$ such that no two elements of $\mathcal{X}$ lie within a distance of ϵ of each other, $|\mathcal{X}| < \left(\frac{2r+\epsilon}{\epsilon}\right)^d$.*

Proof. As no two elements lie within distance ϵ of each other, balls of radius $\epsilon/2$ centred at elements are non-intersecting. As each of the balls of radius $\epsilon/2$ centred at elements of $\mathcal{X}$ lies entirely within $\mathcal{B}_d(0,r+\epsilon/2)$, we can apply Lemma (SM-G.7), arriving at the result. □

SM-H Pseudocode for `trikmeds`

In Alg. (6) we present `trikmeds`. It is decomposed into algorithms for initialisation (7), updating medoids (8), assigning data to clusters (9) and updating bounds on the `trimed` derived bounds (10). Table 4 summarised all of the variables used in `trikmeds`.

When there are no distance bounds, the location of the bottleneck in terms of distance calculations depends on N/K^2. If $N/K \gg K$, the bottleneck lies in updating medoids, which can be improved through the strategy used in `trimed`. If $N/K \ll K$, the bottleneck lies in assigning elements to clusters, which is effectively handled through the approach of Elkan (2003).

Algorithm 6 `trikmeds`

```
initialise()
while not converged do
  update-medoids()
  assign-to-clusters()
  update-sum-bounds()
end while
```

Algorithm 7 `initialise`

```
// Initialise medoid indices, uniform random sample without replacement (or otherwise)
{m(1), ..., m(K)} ← uniform-no-replacement({1, ..., N})
for k = 1 : K do
  // Initialise medoid and set cluster count to zero
  c(k) ← x(m(k))
  v(k) ← 0
  // Set sum of in-cluster distances to medoid to zero
  s(k) ← 0
end for
for i = 1 : N do
  for k = 1 : K do
    // Tightly initialise lower bounds on data-to-medoid distances
    l_c(i, k) ← ‖x(i) − c(k)‖
  end for
  // Set assignments and distances to nearest (assigned) medoid
  a(i) ← arg min_{k∈{1,...,K}} l_c(i, k)
  d(i) ← l_c(i, a(i))
  // Update cluster count
  v(a(i)) ← v(a(i)) + 1
  // Update sum of distances to medoid
  s(a(i)) ← s(a(i)) + d(i)
  // Initialise lower bound on sum of in-cluster distances to x(i) to zero
  l_s(i) ← 0
end for
V(0) ← 0
for k = 1 : K do
  // Set cumulative cluster count
  V(k) ← V(k − 1) + v(k)
  // Initialise lower bound on in-cluster sum of distances to be tight for medoids
  l_s(m(k)) ← s(k)
end for
// Make clusters contiguous
contiguate()
```

Algorithm 8 `update-medoids`

```
for k = 1 : K do
  for i = V(k−1) : V(k) − 1 do
    // If the bound test cannot exclude i as m(k)
    if l_s(i) < s(k) then
      // Make l_s(i) tight by computing and cumulating all in-cluster distances to x(i),
      l_s(i) ← 0
      for i' = V(k−1) : V(k) − 1 do
        d̃(i') ← ‖x(i) − x(i')‖
        l_s(i) ← l_s(i) + d̃(i')
      end for
      // Re-perform the test for i as candidate for m(k), now with exact sums. If i is the new best candidate,
      update some cluster information
      if l_s(i) < s(k) then
        s(k) ← l_s(i)
        m(k) ← i
        for i' = V(k−1) : V(k) − 1 do
          d(i') ← ‖x(i) − x(i')‖
        end for
      end if
      // Use computed distances to i to improve lower bounds on sums for all samples in cluster k (see Figure
      X)
      for i' = V(k−1) : V(k) − 1 do
        l_s(i') ← max (l_s(i'), |d̃(i')v(k) − l_s(i)|)
      end for
    end if
  end for
  // If the medoid of cluster k has changed, update cluster information
  if m(k) ≠ V(k−1) then
    p(k) ← ‖c(k) − x(m(k))‖
    c(k) ← x(m(k))
  end if
end for
```

Algorithm 9 `assign-to-clusters`

```
// Reset variables monitoring cluster fluxes,
for k = 1 : K do
    // the number of arrivals to cluster k,
    Δ_{n-in}(k) ← 0
    // the number of departures from cluster k,
    Δ_{n-out}(k) ← 0
    // the sum of distances to medoid k of samples which leave cluster k
    Δ_{s-out}(k) ← 0
    // the sum of distances to medoid k of samples which arrive in cluster k
    Δ_{s-in}(k) ← 0
end for
for i = 1 : N do
    // Update lower bounds on distances to medoids based on distances moved by medoids
    for k = 1 : K do
        l(i,k) = l(i,k) − p(k)
    end for
    // Use the exact distance of current assignment to keep bound tight (might save future calcs)
    l(i,a(i)) = d(i)
    // Record current assignment and distance
    a_old = a(i)
    d_old = d(i)
    // Determine nearest medoid, using bounds to eliminate distance calculations
    for k = 1 : K do
        if l(i,k) < d(i) then
            l(i,k) ← ‖x(i) − c(k)‖
            if l(i,k) < d(i) then
                a(i) = k
                d(i) = l(i,k)
            end if
        end if
    end for
    // If the assignment has changed, update statistics
    if a_old ≠ a(i) then
        v(a_old) = v(a_old) − 1
        v(a(i)) = v(a(i)) + 1
        l_s(i) = 0
        Δ_{n-in}(a(i)) = Δ_{n-in}(a(i)) + 1
        Δ_{n-out}(a_old) = Δ_{n-out}(a_old) + 1
        Δ_{s-in}(a(i)) = Δ_{s-in}(a(i)) + d(i)
        Δ_{s-out}(a_old) = Δ_{s-out}(a_old) + d_old
    end if
end for
// Update cumulative cluster counts
for k = 1 : K do
    V(k) ← V(k − 1) + v(k)
end for
contiguate()
```

Table 4: Table Of Notation For `trikmeds`

N	:	number of training samples
i	:	index of a sample, $i \in \{1, \dots, N\}$
$x(i)$	:	sample i
K	:	number of clusters
k	:	index of a cluster, $k \in \{1, \dots, K\}$
$m(k)$	:	index of current medoid of cluster k, $m(k) \in \{1, \dots, N\}$
$c(k)$	:	current medoid of cluster k, that is $c(k) = x(m(k))$
$n_1(i)$	:	cluster index of centroid nearest to $x(i)$
$a(i)$	:	cluster to which $x(i)$ is currently assigned
$d(i)$	:	distance from $x(i)$ to $c(a(i))$
$v(k)$	:	number of samples assigned to cluster k
$V(k)$	:	number of samples assigned to a cluster of index less than $k+1$
$l_c(i,k)$	:	lowerbound on distance from $x(i)$ to $m(k)$
$l_s(i)$	:	lowerbound on $\sum_{i':a(i')=a(i)} \|x(i') - x(i)\|$
$p(k)$	:	distance moved (teleported) by $m(k)$ in last update
$s(k)$	:	sum of distances of samples in cluster k to medoid k

Algorithm 10 `update-sum-bounds`

for $k = 1 : K$ **do**
  // Obtain absolute and net fluxes of energy and count, for cluster k
  $\mathcal{J}_s^{abs}(k) = \Delta_{s-in}(k) + \Delta_{s-out}(k)$
  $\mathcal{J}_s^{net}(k) = \Delta_{s-in}(k) - \Delta_{s-out}(k)$
  $\mathcal{J}_n^{abs}(k) = \Delta_{n-in}(k) + \Delta_{n-out}(k)$
  $\mathcal{J}_n^{net}(k) = \Delta_{n-in}(k) - \Delta_{n-out}(k)$
  for $i = V(k-1) : V(k) - 1$ **do**
    // Update the lower bound on the sum of distances
    $l_s(i) \leftarrow l_s(i) - \min(\mathcal{J}_s^{abs}(k) - \mathcal{J}_n^{net}(k)d(i), \mathcal{J}_n^{abs}(k)d(i) - \mathcal{J}_s^{net}(k))$
  end for
end for

SM-I Datasets

- *Birch1*, *Birch2* : Synthetic 2-D datasets available from `https://cs.joensuu.fi/sipu/datasets/`
- *Europe* : Border map of Europe available from `https://cs.joensuu.fi/sipu/datasets/`
- *U-Sensor Net* : Undirected 2-D graph data. Points drawn uniformly from unit square, with an undirected edge connecting points when the distance between them is less than $1.25\sqrt{N}$
- *D-Sensor Net* : Directed 2-D graph data. Points drawn uniformly from unit square, with directed edge connecting points when the distance between them is less than $1.45\sqrt{N}$, direction chosen at random.
- *Europe rail* : The European rail network, the shapefile is available at `http://www.mapcruzin.com/free-europe-arcgis-maps-shapefiles.htm`. We extracted edges from the shapefile using `networkx` available at `https://networkx.github.io/`.

Algorithm 11 `contiguate`

// This function performs an in place rearrangement over of variables a, d, l, x and m
// The permutation applied to a, d, l and x has as result a sorting by cluster,
// $a(i) = k$ if $i \in \{V(k-1), V(k)\}$ for $k \in \{1, \dots, K\}$
// and moreover that the first element of each cluster is the medoid,
// $m(k) = V(k-1)$ for $k \in \{1, \dots, K\}$

- *Pennsylvania road* The road network of Pennsylvania, the edge list is available directly from `https://snap.stanford.edu/data/`
- *Gnutella* Peer-to-peer network data, available from `https://snap.stanford.edu/data/`
- *MNIST (0)* The '0's in the MNIST training dataset.
- *Conflong* The conflongdemo data is available from `https://cs.joensuu.fi/sipu/datasets/`
- *Colormo* The colormoments data is available at `http://archive.ics.uci.edu/ml/datasets/Corel+Image+Features`
- *MNIST50* The MNIST dataset, projected into 50-dimensions using a random projection matrix where each of the 784×50 elements in the matrix is i.i.d. $\mathcal{N}(0, 1)$.
- *S1, S2, S3, S4, A1, A2, A3* All of these synthetic datasets are available from `https://cs.joensuu.fi/sipu/datasets/`.
- *thyroid, yeast, wine, breast, spiral* All of these real world datasets are available from `https://cs.joensuu.fi/sipu/datasets/`.

SM-J Scaling with dimension of TOPRANK and TOPRANK2

Recall the assumption (3) made for the `TOPRANK` and `TOPRANK2` algorithms. The assumption states that as one approaches the minimum energy E^* from above, the density of elements decreases. In other words, the lowest energy elements stand out from the rest and are not bunched up with very similar energies.

Consider the case where elements are points in $\mathbb{R}^d$. Suppose that the density f_X of points around the medoid is bounded by $0 < \rho_0 \leq f_X \leq \rho_1$, and that the energy grows quadratically in radius about the medoid. Then, as the number of points at radius ϵ is $O(\epsilon^{d-1})$, the density (by energy) of points at radius ϵ is $O(\epsilon^{d-2})$. Thus for $d = 1$ the assumption for `TOPRANK` and `TOPRANK` does not hold, which results in poor performance for $d = 1$. For $d = 2$, the assumption holds, as the density (by energy) of points is constant. For $d \geq 2$, as d increases the energy distribution becomes more and more favourable for `TOPRANK` and `TOPRANK2`, as the low ranking elements become more and more distinct with low energies becoming less probable. This explains the observation that `TOPRANK` scales well with dimension in Figure 3.

SM-K Example where geometric median is a poor approximation of medoid

There is no guarantee that the geometric median is close to the set medoid. Moreover, the element in $\mathcal{S}$ which is nearest to $g(\mathcal{S})$ is not necessarily the medoid, as illustrated in the following example. Suppose $S = \{x(1), \ldots, x(20)\} \subset \mathbb{R}^2$, with $x(i) = (0, 1)$ for $i \in \{1, \ldots, 9\}$, $x(i) = (0, -1)$ for $i \in \{10, \ldots, 18\}$, $x(19) = (1/2, 0)$ and $x(20) = (-1/2, 0)$. The geometric median is $(0, 0)$ and the nearest points to the geometric median, $x(19)$ and $x(20)$ have energy $1 + 18\sqrt{3}/2 \approx 16.6$. However, points $\{x(1), \ldots, x(18)\}$ have energy $2\sqrt{3}/2 + 9 = 10.7$. Thus by choosing a point in $\mathcal{S}$ which is nearest to the geometric median, one is choosing the element with the highest energy, the opposite of the medoid.

Note the above example appears to violate the assumptions required for $O(N^{3/2})$ convergence of `trimed`, as it requires that the probability density function vanishes at the distribution median. Indeed, in $\mathbb{R}^d$ it is the case that if the $O(N^{3/2})$ assumptions are satisfied, the set medoid converges to the geometric median, and so the geomteric median is a good approximation. We stress however that the geometric median is only relevant in vector spaces.

SM-L Miscellaneous

Figure 7 illustrates the idea behind algorithm `trimed`, comments in the caption.

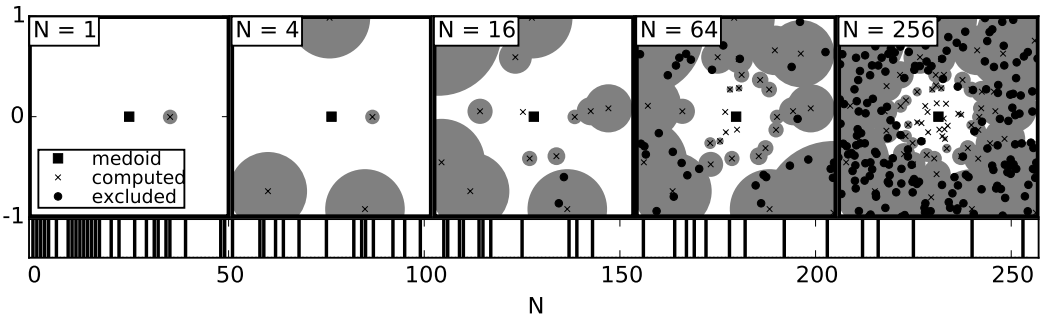

Figure 7: Eliminating samples as potential medoids using only type 1 elimination, where we assume that the medoid and its energy E^* are known, and so the radius of the exclusion ball of an element x is $E(x) - E^*$. Uniformly sampling from $[-1, 1] \times [-1, 1]$, energies are computed only if the sample drawn does not lie in the exclusion zone (union of balls). If the energy at x is computed, the exclusion zone is augmented by adding $\mathcal{B}_d(x, E(x) - E^*)$. Top left to right: the distribution of samples which are computed and excluded. Bottom: the times at which samples are computed. We prove that probability of computation at time n is $O(n^{-\frac{1}{2}})$.