Fast Construction of Similarity Graphs with Kernel Density Estimation: Supplementary Material

Anonymous Author(s)
Affiliation
Address
`email`

The supplementary material of the paper is organised as follows: we present the complete proof of Theorem 1 in Section A, and present some additional experimental results in Section B.

A Proof of Theorem 1

This section presents the complete proof of Theorem 1. To make the supplementary material self-contained, we first re-state the main theorem of our paper, and our presented algorithm is described in Algorithm 2.

Theorem 1. *Given a set of data points* $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ *as input, there is a randomised algorithm that constructs a sparse graph* G *of* X*, such that it holds with probability at least* $9/10$ *that*

1. *graph* G *has* $\widetilde{O}(n)$ *edges,*
2. *graph* G *has the same cluster structure as the fully connected similarity graph* K *of* X*; that is, if* K *has* k *well-defined clusters, then it holds that* $\rho_{\mathsf{G}}(k) = O(k \cdot \rho_{\mathsf{K}}(k))$ *and* $\lambda_{k+1}(\mathbf{N}_{\mathsf{G}}) = \Omega(\lambda_{k+1}(\mathbf{N}_{\mathsf{K}}))$[1].

The algorithm uses an approximate KDE *algorithm as a black-box, and has running time* $\widetilde{O}(T_{\mathsf{KDE}}(n, n, \epsilon))$*, where* $T_{\mathsf{KDE}}(n, n, \epsilon)$ *is the running time of solving the* KDE *problem for* n *data points up to a* $(1+\epsilon)$*-approximation.*

For simplicity, throughout the proof we always use K to stand for the fully connected similarity graph and G stands for the sparsifier constructed by Algorithm 2.

For each x_i in the input data, Algorithm 2 adds $L = 6C \cdot \log(n)/\lambda_{k+1}$ edges to the constructed graph G for some constant C. Let $y_{i,1}, \ldots, y_{i,L}$ be random variables which are equal to the L points sampled for x_i. Recall that by the SZ algorithm, the "ideal" sampling probability for x_j from x_i is

$$p_i(x_j) \triangleq \min\left\{ \frac{k(x_i, x_j)}{\deg_{\mathsf{K}}(x_i)} \cdot \frac{C\log(n)}{\lambda_{k+1}}, 1 \right\}.$$

We denote the true sampling probability under Algorithm 2 to be

$$\widetilde{p}_i(x_j) \triangleq \mathbb{P}\left[x_j \in \{y_{i,1}, \ldots y_{i,L}\}\right].$$

Finally, for each added edge, Algorithm 2 also computes an estimate of $p_i(x_j)$ which we denote

$$\widehat{p}_i(x_j) \triangleq \min\left\{ \frac{k(x_i, x_j)}{g(x_i)} \cdot \frac{C\log(n)}{\lambda_{k+1}}, 1 \right\}.$$

Similarly, we define

---

[1] The formal definition of $\rho_{\mathsf{G}}(k)$ and matrix $\mathbf{N}_G$ can be found in Section 2. We show in Section 3 that these two conditions guarantee that graph G and K have the same cluster structure.

Algorithm 1 SAMPLE

```
Input: set $S$ of $\{y_i\}$
       set $X$ of $\{x_i\}$
Output:
    $E = \{(y_i, x_j)$ for some $i$ and $j\}$
if $|X| = 1$ then
    return $S \times X$
else
    $X_1 = \{x_j : j < |X|/2\}$
    $X_2 = \{x_j : j \geq |X|/2\}$
    Compute $g_{X_1}(y_i)$ for all $i$ with a KDE algorithm
    Compute $g_{X_2}(y_i)$ for all $i$ with a KDE algorithm
    $S_1 = S_2 = \emptyset$
    for $y_i \in S$ do
        $r \sim \mathrm{Unif}[0, 1]$
        if $r \leq g_{X_1}(y_i)/(g_{X_1}(y_i) + g_{X_2}(y_i))$ then
            $S_1 = S_1 \cup \{y_i\}$
        else
            $S_2 = S_2 \cup \{y_i\}$
        end if
    end for
    return SAMPLE$(S_1, X_1) \cup$SAMPLE$(S_2, X_2)$
end if
```

Algorithm 2 FASTSIMILARITYGRAPH

```
Input: data point set $X = \{x_1, \ldots, x_n\}$
Output: similarity graph $\mathsf{G}$
$E = \emptyset$, $L = C \cdot \log n$
for $\ell \in [1, L]$ do
    $E = E \cup$ SAMPLE$(X, X)$
end for
Compute $g_{[1,n]}(x_i)$ for each $i$ with a KDE algorithm
for $(v_i, v_j) \in E$ do
    $\widehat{p}_i(j) = \min\left\{L \cdot k(x_i, x_j)/g_{[1,n]}(x_i), 1\right\}$
    $\widehat{p}_j(i) = \min\left\{L \cdot k(x_i, x_j)/g_{[1,n]}(x_j), 1\right\}$
    $\widehat{p}(i, j) = \widehat{p}_i(j) + \widehat{p}_j(i) - \widehat{p}_i(j) \cdot \widehat{p}_j(i)$
    Set $w(v_i, v_j) = k(x_i, x_j)/\widehat{p}(i, j)$
end for
return graph $\mathsf{G} = (X, E, w)$
```

- $p(x_i, x_j) = p_i(x_j) + p_j(x_i) - p_i(x_j)p_j(x_i)$,
- $\widetilde{p}(x_i, x_j) = \widetilde{p}_i(x_j) + \widetilde{p}_j(x_i) - \widetilde{p}_i(x_j)\widetilde{p}_j(x_i)$, and
- $\widehat{p}(x_i, x_j) = \widehat{p}_i(x_j) + \widehat{p}_j(x_i) - \widehat{p}_i(x_j)\widehat{p}_j(x_i)$.

Following the convention of [2], we use $p_i(x_j)$ to refer to the probability that a given edge is sampled *from the vertex* x_i and $p(x_i, x_j)$ is the probability that the given edge (x_i, x_j) is sampled at all by the algorithm. We use the same convention for $\widetilde{p}_i(x_j)$ and $\widehat{p}_i(x_j)$.

Before coming to the proof of Theorem 1, we first show a sequence of lemmas showing that these probabilities are all within a constant factor of each other.

Lemma 1. *For any point x_i, the probability that a given sampled neighbour $y_{i,l}$ is equal to x_j is given by*

$$\frac{k(x_i, x_j)}{2 \deg_{\mathsf{K}}(x_i)} \leq \mathbb{P}\left[y_{i,l} = x_j\right] \leq \frac{2k(x_i, x_j)}{\deg_{\mathsf{K}}(x_i)}.$$

Proof. Let $X = x_1, \ldots, x_n$ be the input data points for Algorithm 2, and $X_{[a,b]} = \{x_a, \ldots, x_b\}$ be the subset of X with indices between a and b. Then, in each recursive call to Algorithm 1, we are given a range $X_{[a,b]}$ as input and assign $y_{i,l}$ to one half of it: either $X_{[a,\lfloor b/2 \rfloor]}$ or $X_{[\lfloor b/2 \rfloor + 1, b]}$. By Algorithm 1, we have that the probability of assigning $y_{i,l}$ to $X_{[a,\lfloor b/2 \rfloor]}$ is

$$\mathbb{P}\left[y_{i,l} \in X_{[a,\lfloor b/2 \rfloor]} | y_{i,l} \in X_{[a,b]}\right] = \frac{g_{[a,\lfloor b/2 \rfloor]}(x_i)}{g_{[a,b]}(x_i)},$$

where we write $g_{[a,b]}$ rather than $g_{X_{[a,b]}}$ for clarity. By our guarantee on the performance of the KDE algorithm, we have that $g_{[a,b]}(x_i) \in (1 \pm \epsilon) \deg_{[a,b]}(x_i)$, where we define

$$\deg_{[a,b]}(x_i) \triangleq \sum_{j=a}^{b} k(x_i, x_j).$$

This gives

$$\left(\frac{1-\epsilon}{1+\epsilon}\right)\frac{\deg_{[a,\lfloor b/2\rfloor]}(x_i)}{\deg_{[a,b]}(x_i)} \leq \mathbb{P}\left[Y \in X_{[a,\lfloor b/2\rfloor]} | Y \in X_{[a,b]}\right] \leq \left(\frac{1+\epsilon}{1-\epsilon}\right)\frac{\deg_{[a,\lfloor b/2\rfloor]}(x_i)}{\deg_{[a,b]}(x_i)}. \tag{1}$$

Then, notice that we can write

$$\begin{aligned}\mathbb{P}\left[y_{i,l} = x_j\right] = \mathbb{P}\left[y_{i,l} = x_j | y_{i,l} \in X_{[a_1,b_1]}\right] \times \mathbb{P}\left[y_{i,l} \in X_{[a_1,b_1]} | y_{i,l} \in X_{[a_2,b_2]}\right] \\ \times \ldots \times \mathbb{P}\left[y_{i,l} \in X_{[a_k,b_k]} | y_{i,l} \in X_{[1,n]}\right],\end{aligned}$$

where each term corresponds to one level of recursion of Algorithm 1, and there are at most $\lceil \log_2(n) \rceil$ terms. Then, by (1), and noticing that the denominator and numerator of adjacent terms cancel out, we have

$$\left(\frac{1-\epsilon}{1+\epsilon}\right)^{\lceil \log_2(n)\rceil} \frac{k(x_i,x_j)}{\deg(x_i)} \leq \mathbb{P}\left[y_{i,l} = x_j\right] \leq \left(\frac{1+\epsilon}{1-\epsilon}\right)^{\lceil \log_2(n)\rceil} \frac{k(x_i,x_j)}{\deg(x_i)}$$

since $\deg_{[j,j]}(x_i) = k(x_i, x_j)$ and $\deg_{[1,n]}(x_i) = \deg(x_i)$.

For the lower bound, we have that

$$\left(\frac{1-\epsilon}{1+\epsilon}\right)^{\lceil \log_2(n)\rceil} \geq (1-2\epsilon)^{\lceil \log_2(n)\rceil} \geq 1 - 3\log_2(n)\epsilon \geq 1/2,$$

where the final inequality follows by the fact that $\epsilon \leq 1/(6\log_2(n))$.

For the upper bound, we similarly have

$$\left(\frac{1+\epsilon}{1-\epsilon}\right)^{\lceil \log_2(n)\rceil} \leq (1+3\epsilon)^{\lceil \log_2(n)\rceil} \leq \exp\left(3\lceil \log_2(n)\rceil\epsilon\right) \leq \mathrm{e}^{2/3} \leq 2,$$

where the first inequality follows since $\epsilon < 1/6$. □

The next lemma shows that Algorithm 2 samples each edge with approximately the correct probability.

Lemma 2. *For every i and $j \neq i$, we have*

$$\frac{9}{10} p_i(x_j) \leq \widetilde{p}_i(x_j) \leq 12 p_i(x_j).$$

Proof. Algorithm 2 samples $6C\log(n)/\lambda_{k+1}$ neighbours of x_i, and Lemma 1 guarantees that the chosen neighbour is equal to x_j with probability roughly proportional to $k(x_i,x_j)/\deg(x_i)$.

Let $Y = \{y_{i,1}, \ldots, y_{i,L}\}$ be the neighbours of x_i sampled by Algorithm 2, where $L = 6C\log(n)/\lambda_{k+1}$. Then,

$$\mathbb{P}\left[x_j \in Y\right] = 1 - \prod_{l=1}^{L}\left(1 - \mathbb{P}\left[y_{i,l} = x_j\right]\right) \geq 1 - \left(1 - \frac{k(x_i,x_j)}{2\deg(x_i)}\right)^{L} \geq 1 - \exp\left(-L \cdot \frac{k(x_i,x_j)}{2\deg(x_i)}\right)$$

The proof proceeds by case distinction.

Case 1: $p_i(x_j) \leq 0.9$. In this case, we have,

$$\mathbb{P}\left[x_j \in Y\right] \geq 1 - \exp\left(-6p_i(x_j)/2\right) \geq p_i(x_j).$$

Case 2: $p_i(x_j) > 0.9$. In this case, we have

$$\mathbb{P}\left[x_j \in Y\right] \geq 1 - \exp\left(-\frac{9 \cdot 6}{20}\right) \geq \frac{9}{10},$$

which completes the proof of the lower bound of $\widetilde{p}(x_j)$.

For the upper bound, we have

$$\mathbb{P}\left[x_j \in Y\right] \leq 1 - \left(1 - \frac{2k(x_i,x_j)}{\deg(x_i)}\right)^{L} \leq \frac{2k(x_i,x_j)}{\deg(x_i)} \cdot L = \frac{12Ck(x_i,x_j)}{\deg(x_i)}\frac{\log(n)}{\lambda_{k+1}},$$

from which the statement follows. □

An immediate corollary of Lemma 2 is as follows.

Corollary 1. *For all x_i and x_j, it holds that*

$$\frac{8}{10}p(x_i,x_j) \leq \widetilde{p}(x_i,x_j) \leq 144 p(x_i,x_j)$$

and

$$\frac{1}{216}\widetilde{p}(x_i,x_j) \leq \widehat{p}(x_i,x_j) \leq \frac{30}{16}\widetilde{p}(x_i,x_j).$$

We now come to the proof of Theorem 1. It is important to note that although some of the analysis is parallel to that of [2], our analysis is more involved since we need to carefully take into account the error introduced by the approximate KDE algorithm which changes the edge sampling probabilities slightly. We also need to be careful since the edges of G are not sampled independently in our algorithm.

The proof makes use of the following concentration inequalities.

Lemma 3 (Bernstein's Inequality [1])**.** *Let $X_1,\ldots,X_n$ be independent random variables such that $|X_i| \leq M$ for any $i \in \{1,\ldots,n\}$. Let $X = \sum_{i=1}^n X_i$, and $R = \sum_{i=1}^n \mathbb{E}\left[X_i^2\right]$. Then, it holds that*

$$\mathbb{P}\left[|X - \mathbb{E}[X]| \geq t\right] \leq 2\exp\left(-\frac{t^2}{2(R + Mt/3)}\right).$$

Lemma 4 (Matrix Chernoff Bound [3])**.** *Consider a finite sequence $\{X_i\}$ of independent, random, PSD matrices of dimension d that satisfy $\|X_i\| \leq R$. Let $\mu_{\min} \triangleq \lambda_{\min}(\mathbb{E}[\sum_i X_i])$ and $\mu_{\max} \triangleq \lambda_{\max}(\mathbb{E}[\sum_i X_i])$. Then, it holds that*

$$\mathbb{P}\left[\lambda_{\min}\left(\sum_i X_i\right) \leq (1-\delta)\mu_{\min}\right] \leq d\left(\frac{\mathrm{e}^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mu_{\min}/R}$$

for $\delta \in [0,1]$, and

$$\mathbb{P}\left[\lambda_{\max}\left(\sum_i X_i\right) \geq (1+\delta)\mu_{\max}\right] \leq d\left(\frac{\mathrm{e}^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu_{\max}/R}$$

forr $\delta \geq 0$.

Proof of Theorem 1. We first show that the degrees of all nodes in the similarity graph K are preserved with high probability in the sparsifier G. We follow SZ and consider only the edges with $p_i(j) < 1$ since we can assume with high probability that the other edges are included in the sparsifier G. For any node x_i, and let $y_{i,1},\ldots,y_{i,L}$ be the neighbours of x_i sampled by Algorithm 2.

We fix an arbitrary node x_i, and let $Y_1,\ldots,Y_L$ be random variables defined by

$$Y_l \triangleq \frac{k(x_i,y_{i,l})}{\widehat{p}(x_i,y_{i,l})}.$$

For each $j \neq i$, we define the random variable $Z_{j,i}$ by

$$Z_{j,i} \triangleq \begin{cases} \frac{k(x_i,x_j)}{\widehat{p}(x_i,x_j)} & \text{if } x_i \in \{y_{j,1}\ldots y_{j,L}\}, \\ 0 & \text{otherwise.} \end{cases}$$

Then, we can write

$$\deg_{\mathsf{G}}(x_i) = \sum_{l=1}^{L} Y_l + \sum_{j\neq i} Z_{j,i}.$$

We have

$$\begin{aligned}\mathbb{E}\left[\deg_{\mathsf{G}}(x_i)\right] &= \sum_{l=1}^{L}\mathbb{E}[Y_l] + \sum_{j\neq i}\mathbb{E}[Z_{j,i}] \\ &= \sum_{l=1}^{L}\sum_{j\neq i}\mathbb{P}[y_{i,l} = x_j]\cdot\frac{k(x_i,x_j)}{\widehat{p}(x_i,x_j)} + \sum_{j\neq i}\widetilde{p}_j(i)\cdot\frac{k(x_i,x_j)}{\widehat{p}(x_i,x_j)}.\end{aligned}$$

By Lemmas 1 and 2 and Corollary 1, we have

$$\begin{aligned}
\mathbb{E}\left[\deg_{\mathsf{G}}(x_i)\right] &\geq \sum_{j\neq i} \frac{k(x_i,x_j)}{\widehat{p}(x_i,x_j)} \left(\frac{L\cdot k(x_i,x_j)}{2\deg(x_i)} + \widetilde{p}_j(x_i)\right) \\
&\geq \sum_{j\neq i} \frac{k(x_i,x_j)}{4\cdot \widehat{p}(x_i,x_j)} \left(\widetilde{p}_i(x_j) + \widetilde{p}_j(x_i)\right) \\
&\geq \sum_{j\neq i} \frac{2\cdot k(x_i,x_j)}{15} = \frac{2\cdot \deg_{\mathsf{K}}(x_i)}{15}.
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
\mathbb{E}\left[\deg_{\mathsf{G}}(x_i)\right] &\leq \sum_{j\neq i} \frac{k(x_i,x_j)}{\widehat{p}(x_i,x_j)} \left(\frac{2\cdot L\cdot k(x_i,x_j)}{\deg(x_i)} + \widetilde{p}_j(x_i)\right) \\
&\leq \sum_{j\neq i} \frac{40\cdot k(x_i,x_j)}{3\cdot \widehat{p}(x_i,x_j)} \left(\widetilde{p}_i(x_j) + \widetilde{p}_j(x_i)\right) \\
&\leq \sum_{j\neq i} 5760\cdot k(x_i,x_j) = 5760\cdot \deg_{\mathsf{K}}(x_i).
\end{aligned}$$

In order to prove a concentration bound on this degree estimate, we would like to apply the Bernstein inequality for which we need to bound

$$\begin{aligned}
R &= \sum_{l=1}^{L} \mathbb{E}\left[Y_l^2\right] + \sum_{j\neq i} \mathbb{E}\left[Z_{j,i}^2\right] \\
&= \sum_{l=1}^{L}\sum_{j\neq i} \mathbb{P}\left[y_{i,l} = x_j\right] \frac{k(x_i,x_j)^2}{\widehat{p}(x_i,x_j)^2} + \sum_{j\neq i} \widetilde{p}_j(i) \frac{k(x_i,x_j)^2}{\widehat{p}(x_i,x_j)^2} \\
&\leq \sum_{j\neq i} \frac{40\cdot k(x_i,x_j)^2}{3\cdot \widehat{p}(x_i,x_j)^2} \left(\widetilde{p}_i(j) + \widetilde{p}_j(i)\right) \\
&\leq \sum_{j\neq i} 5760\cdot \frac{k(x_i,x_j)^2}{\widehat{p}(x_i,x_j)} \\
&\leq \sum_{j\neq i} 6720\cdot \frac{k(x_i,x_j)^2}{p_i(x_j)} \\
&= \sum_{j\neq i} 6720\cdot \frac{k(x_i,x_j)\cdot \deg_{\mathsf{K}}(x_i)\cdot \lambda_{k+1}}{C\log(n)} \\
&\leq \frac{6720\cdot \deg_{\mathsf{K}}(x_i)^2\cdot \lambda_{k+1}}{C\log(n)}.
\end{aligned}$$

Then, by applying Bernstein's inequality we have for any constant C_2 that

$$\begin{aligned}
\mathbb{P}\left[\left|\deg_{\mathsf{G}}(x_i) - \mathbb{E}[\deg_{\mathsf{G}}(x_i)]\right| \geq \frac{1}{C_2}\deg_{\mathsf{K}}(x_i)\right] &\leq 2\cdot \exp\left(-\frac{\deg_{\mathsf{K}}(x_i)^2/C_2^2}{\frac{6720\deg_{\mathsf{K}}(x_i)^2\lambda_{k+1}}{C\log(n)} + \frac{1}{6}\frac{\deg_{\mathsf{K}}(x_i)^2\lambda_{k+1}}{C_2 C\log(n)}}\right) \\
&\leq 2\exp\left(-\frac{C\cdot \log(n)}{6721\cdot \lambda_{k+1}\cdot C_2^2}\right) \\
&= o(1/n).
\end{aligned}$$

Therefore, by taking C to be sufficiently large and by the union bound, it holds with high probability that the degree of all the nodes in G are preserved up to a constant factor. For the remainder of the proof, we assume that this is the case. Note in particular that this implies $\mathrm{vol}_{\mathsf{G}}(S) = \Theta(\mathrm{vol}_{\mathsf{K}}(S))$ for any subset $S \subseteq V$.

Next, we prove it holds for G that $\phi_{\mathsf{G}}(S_i) = O\left(k \cdot \phi_{\mathsf{K}}(S_i)\right)$ for any $1 \leq i \leq k$, where $S_1, \ldots, S_k$ form an optimal clustering in K.

By the definition of $Z_{i,j}$, it holds for any $1 \leq i \leq k$ that

$$\begin{aligned}
\mathbb{E}\left[w_{\mathsf{G}}(S_i, V \setminus S_i)\right] &= \mathbb{E}\left[\sum_{j \in S_i} \sum_{l \notin S_i} Z_{l,j} + Z_{j,l}\right] \\
&= \sum_{j \in S_i} \sum_{l \notin S_i} \frac{k(x_j, x_l)}{\widehat{p}(x_j, x_l)} \left(\widetilde{p}(x_j, x_l) + \widetilde{p}(x_l, x_j)\right) \\
&= O\left(w_{\mathsf{K}}(S_i, V \setminus S_i)\right)
\end{aligned}$$

where the last line follows by Corollary 1. By Markov's inequality and the union bound, with constant probability it holds for all $i = 1, \ldots, k$ that

$$w_{\mathsf{G}}(S_i, V \setminus S_i) = O(k \cdot w_{\mathsf{K}}(S_i, V \setminus S_i)).$$

Therefore, it holds with constant probability that

$$\rho_{\mathsf{G}}(k) \leq \max_{1 \leq i \leq k} \phi_{\mathsf{G}}(S_i) = \max_{1 \leq i \leq k} O(k \cdot \phi_{\mathsf{K}}(S_i)) = O(k \cdot \rho_{\mathsf{K}}(k)).$$

Next, we prove that $\lambda_{k+1}(\mathbf{N}_{\mathsf{G}}) = \Omega(\lambda_{k+1}(\mathbf{N}_{\mathsf{K}}))$. Let $\overline{\mathbf{N}}_{\mathsf{K}}$ be the projection of $\mathbf{N}_{\mathsf{K}}$ on its top $n-k$ eigenspaces, and notice that $\overline{\mathbf{N}}_{\mathsf{K}}$ can be written

$$\overline{\mathbf{N}}_{\mathsf{K}} = \sum_{i=k+1}^{n} \lambda_i f_i f_i^{\intercal}$$

where $f_1, \ldots, f_n$ are the eigenvectors of $\mathbf{N}_{\mathsf{K}}$. Let $\overline{\mathbf{N}}_{\mathsf{K}}^{-1/2}$ be the square root of the pseudoinverse of $\overline{\mathbf{N}}_{\mathsf{K}}$.

We prove that the top $n-k$ eigenvalues of $\mathbf{N}_{\mathsf{K}}$ are preserved, which implies that $\lambda_{k+1}(\mathbf{N}_{\mathsf{K}}) = \Theta(\lambda_{k+1}(\mathbf{N}_{\mathsf{G}}))$. To prove this, for each data point x_i and sample $1 \leq l \leq L$, we define a random matrix $X_{i,l} \in \mathbb{R}^{n \times n}$ by

$$X_{i,l} = w_{\mathsf{G}}(x_i, y_{i,l}) \cdot \overline{\mathbf{N}}_{\mathsf{K}}^{-1/2} b_e b_e^{\intercal} \overline{\mathbf{N}}_{\mathsf{K}}^{-1/2},$$

where $b_e = \chi_{x_i} - \chi_{y_{i,l}}$ is the edge indicator vector. Notice that

$$\sum_{i=1}^{n} \sum_{l=1}^{L} X_{i,l} = \sum_{\text{sampled edges } e=(x_i,x_j)} w_{\mathsf{G}}(x_i, x_j) \overline{\mathbf{N}}_{\mathsf{K}}^{-1/2} b_e b_e^{\intercal} \overline{\mathbf{N}}_{\mathsf{K}}^{-1/2} = \overline{\mathbf{N}}_{\mathsf{K}}^{-1/2} \mathbf{N}_{\mathsf{G}}^{'} \overline{\mathbf{N}}_{\mathsf{K}}^{-1/2}$$

where

$$\mathbf{N}_{\mathsf{G}}^{'} = \sum_{\text{sampled edges } e=(x_i,x_j)} w_{\mathsf{G}}(x_i, x_j) b_e b_e^{\intercal}$$

is the Laplacian matrix of G normalised with respect to the degrees of the nodes in K. We prove that, with high probability, the top $n-k$ eigenvectors of $\mathbf{N}_{\mathsf{G}}^{'}$ and $\mathbf{N}_{\mathsf{K}}$ are approximately the same. Then, we show the same for $\mathbf{N}_{\mathsf{G}}$ and $\mathbf{N}_{\mathsf{G}}^{'}$ which implies that $\lambda_{k+1}(\mathbf{N}_{\mathsf{G}}) = \Omega(\lambda_{k+1}(\mathbf{N}_{\mathsf{K}}))$.

We begin by looking at the first moment of the expression above:

$$\begin{aligned}
\lambda_{\min}\left(\mathbb{E}\left[\sum_{i=1}^{n} \sum_{l=1}^{L} X_{i,l}\right]\right) &= \lambda_{\min}\left(\sum_{i=1}^{n} \sum_{l=1}^{L} \sum_{j \neq i} \mathbb{P}\left[y_{i,l} = x_j\right] \frac{k(x_i, x_j)}{\widehat{p}(x_i, x_j)} \cdot \overline{\mathbf{N}}_{\mathsf{K}}^{-1/2} b_e b_e^{\intercal} \overline{\mathbf{N}}_{\mathsf{K}}^{-1/2}\right) \\
&\geq \lambda_{\min}\left(\sum_{i=1}^{n} \sum_{j \neq i} \frac{\widetilde{p}_i(x_j)}{4} \frac{k(x_i, x_j)}{\widehat{p}(x_i, x_j)} \cdot \overline{\mathbf{N}}_{\mathsf{K}}^{-1/2} b_e b_e^{\intercal} \overline{\mathbf{N}}_{\mathsf{K}}^{-1/2}\right) \\
&\geq \lambda_{\min}\left(\frac{2}{15} \cdot \overline{\mathbf{N}}_{\mathsf{K}}^{-1/2} \mathbf{N}_{\mathsf{K}} \overline{\mathbf{N}}_{\mathsf{K}}^{-1/2}\right) = \frac{2}{15}.
\end{aligned}$$

Similarly,

$$
\begin{aligned}
\lambda_{\max}\left(\mathbb{E}\left[\sum_{i=1}^{n}\sum_{l=1}^{L} X_{i,l}\right]\right) &= \lambda_{\max}\left(\sum_{i=1}^{n}\sum_{l=1}^{L}\sum_{j\neq i}\mathbb{P}\left[y_{i,l}=x_j\right]\frac{k(x_i,x_j)}{\widehat{p}(x_i,x_j)}\cdot\overline{\mathbf{N}}_{\mathsf{K}}^{-1/2} b_e b_e^{\intercal}\overline{\mathbf{N}}_{\mathsf{K}}^{-1/2}\right)\\
&\leq \lambda_{\max}\left(\sum_{i=1}^{n}\sum_{j\neq i}\frac{120\cdot\widetilde{p}_i(x_j)}{9}\frac{k(x_i,x_j)}{\widehat{p}(x_i,x_j)}\cdot\overline{\mathbf{N}}_{\mathsf{K}}^{-1/2} b_e b_e^{\intercal}\overline{\mathbf{N}}_{\mathsf{K}}^{-1/2}\right)\\
&\leq \lambda_{\max}\left(5760\cdot\overline{\mathbf{N}}_{\mathsf{K}}^{-1/2}\mathbf{N}_{\mathsf{K}}\overline{\mathbf{N}}_{\mathsf{K}}^{-1/2}\right) = 5760.
\end{aligned}
$$

Additionally, for any i and l, we have that

$$
\begin{aligned}
\|X_{i,l}\| &\leq w_{\mathsf{G}}(x_i, y_{i,l})\cdot b_e^{\intercal}\overline{\mathbf{N}}_{\mathsf{K}}^{-1/2}\overline{\mathbf{N}}_{\mathsf{K}}^{-1/2} b_e\\
&= \frac{k(x_i,y_{i,l})}{\widehat{p}(x_i,y_{i,l})}\cdot b_e^{\intercal}\overline{\mathbf{N}}_{\mathsf{K}}^{-1} b_e\\
&\leq \frac{k(x_i,y_{i,l})}{\widehat{p}(x_i,y_{i,l})}\cdot\frac{1}{\lambda_{k+1}}\|b_e\|^2\\
&\leq \frac{2(1+\epsilon)\lambda_{k+1}}{C\log(n)\left(\frac{1}{\deg_{\mathsf{K}}(u)}+\frac{1}{\deg_{\mathsf{K}}(y_{i,l})}\right)}\cdot\frac{1}{\lambda_{k+1}}\left(\frac{1}{\deg_{\mathsf{K}}(x_i)}+\frac{1}{\deg_{\mathsf{K}}(y_{i,l})}\right)\\
&\leq \frac{2(1+\epsilon)}{C\log(n)}.
\end{aligned}
$$

Now, we apply the matrix Chernoff bound and have that

$$
\mathbb{P}\left[\lambda_{\max}\left(\sum_{i=1}^{n}\sum_{l=1}^{L} X_{i,l}\right)\geq 8640\right]\leq n\left(\frac{e^{1/2}}{(1+1/2)^{3/2}}\right)^{2C\log(n)/15(1+\epsilon)} = O(1/n^c)
$$

for some constant c. The other side of the matrix Chernoff bound gives us that

$$
\mathbb{P}\left[\lambda_{\min}\left(\sum_{i=1}^{n}\sum_{l=1}^{L} X_{i,l}\right)\leq 1/15\right]\leq O(1/n^c).
$$

Combining these, with probability $1-O(1/n^c)$ it holds for any non-zero $x\in\mathbb{R}^n$ in the space spanned by $f_{k+},\ldots,f_n$ that

$$
\frac{x^{\intercal}\overline{\mathbf{N}}_{\mathsf{K}}^{-1/2}\mathbf{N}_{\mathsf{G}}^{'}\overline{\mathbf{N}}_{\mathsf{K}}^{-1/2}x}{x^{\intercal}x}\in\{1/15, 8640\}.
$$

By setting $y=\overline{\mathbf{N}}_{\mathsf{K}}^{-1/2}x$, we can rewrite this as

$$
\frac{y^{\intercal}\mathbf{N}_{\mathsf{G}}^{'}y}{y^{\intercal}\overline{\mathbf{N}}_{\mathsf{K}}^{1/2}\overline{\mathbf{N}}_{\mathsf{K}}^{1/2}y} = \frac{y^{\intercal}\mathbf{N}_{\mathsf{G}}^{'}y}{y^{\intercal}\overline{\mathbf{N}}_{\mathsf{K}}y} = \frac{y^{\intercal}\mathbf{N}_{\mathsf{G}}^{'}y}{y^{\intercal}y}\frac{y^{\intercal}y}{y^{\intercal}\overline{\mathbf{N}}_{\mathsf{K}}y}\in\{1/240, 1280\}.
$$

Since $\dim(\mathrm{span}\{f_{k+1},\ldots,f_n\}) = n-k$, we have shown that there exist $n-k$ orthogonal vectors whose Rayleigh quotient with respect to $\mathrm{N}_{\mathsf{G}}^{'}$ is $\Omega(\lambda_{k+1}(\mathbf{N}_{\mathsf{K}}))$. By the Courant-Fischer Theorem, we have $\lambda_{k+1}(\mathbf{N}_{\mathsf{G}}^{'}) = \Omega(\lambda_{k+1}(\mathbf{N}_{\mathsf{K}}))$.

It only remains to show that $\lambda_{k+1}(\mathbf{N}_{\mathsf{G}}) = \Omega(\lambda_{k+1}(\mathbf{N}_{\mathsf{G}}^{'}))$, which implies that $\lambda_{k+1}(\mathbf{N}_{\mathsf{G}}) = \Omega(\lambda_{k+1}(\mathbf{N}_{\mathsf{K}}))$. By the definition of $\mathbf{N}_{\mathsf{G}}^{'}$, we have that $\mathbf{N}_{\mathsf{G}} = \mathbf{D}_{\mathsf{G}}^{-1/2}\mathbf{D}_{\mathsf{K}}^{1/2}\mathbf{N}_{\mathsf{G}}^{'}\mathbf{D}_{\mathsf{K}}^{1/2}\mathbf{D}_{\mathsf{G}}^{-1/2}$. Therefore, for any $x\in\mathbb{R}^n$ and $y=\mathbf{D}_{\mathsf{K}}^{1/2}\mathbf{D}_{\mathsf{G}}^{-1/2}x$, it holds that

$$
\frac{x^{\intercal}\mathbf{N}_{\mathsf{G}}x}{x^{\intercal}x} = \frac{y^{\intercal}\mathbf{N}_{\mathsf{G}}^{'}y}{x^{\intercal}x} = \Omega\left(\frac{y^{\intercal}\mathbf{N}_{\mathsf{G}}^{'}y}{y^{\intercal}y}\right),
$$

where the final guarantee follows from the fact that the degrees in G are preserved up to a constant factor. The conclusion of the theorem follows by the Courant-Fischer Theorem.

Finally, we bound the running time of Algorithm 2 which is dominated by the recursive calls to Algorithm 1. We note that although the number of nodes doubles at each level of the recursion tree (visualised in Figure 4), the total number of samples S and data points X remain constant for each level of the tree. Then, since the running time of the KDE algorithm is superadditive, the total running time of the KDE algorithms at level i of the tree is

$$\begin{aligned} T_i &= \sum_{j=1}^{2^i} T_{\mathsf{KDE}}(|S_{i,j}|, |X_{i,j}|, \epsilon) \\ &\leq T_{\mathsf{KDE}}\left(\sum_{j=1}^{2^i} |S_{i,j}|, \sum_{j=1}^{2^i} |X_{i,j}|, \epsilon\right) = T_{\mathsf{KDE}}(|S|, |X|, \epsilon). \end{aligned}$$

Since there are $O(\log_2(n))$ levels of the tree, the total running time of Algorithm 1 is $\widetilde{O}(T_{\mathsf{KDE}}(|S|, |X|, \epsilon))$ which completes the proof. $\square$

B Additional Experimental Results

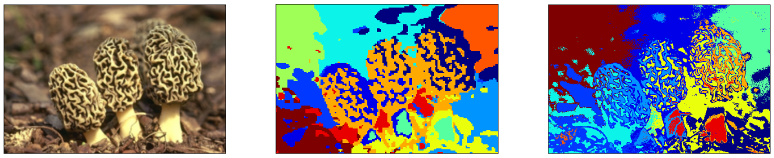
In this section, we include in Figures 1 and 2 some additional examples of the performance of the six spectral clustering algorithms on the BSDS image segmentation dataset. Due to the quadratic memory requirement of the SKLEARN GK algorithm, it cannot be used on the full-resolution image. Therefore, we present its results on each image downsampled to 20,000 pixels. For every other algorithm, we show the results on the full-resolution image. In every case, we find that our algorithm is able to identify more refined detail of the image when compared with the alternative algorithms.

References

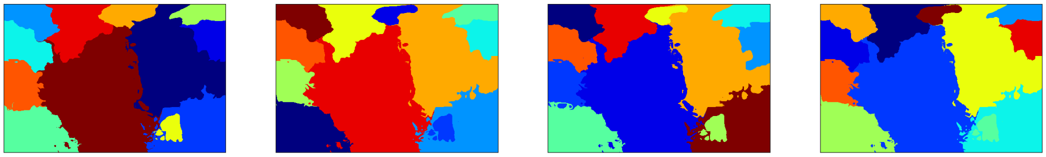
[1] Fan Chung and Linyuan Lu. Concentration inequalities and martingale inequalities: a survey. *Internet mathematics*, 3(1):79–127, 2006.

[2] He Sun and Luca Zanetti. Distributed graph clustering and sparsification. *ACM Transactions on Parallel Computing*, 6(3):17:1–17:23, 2019.

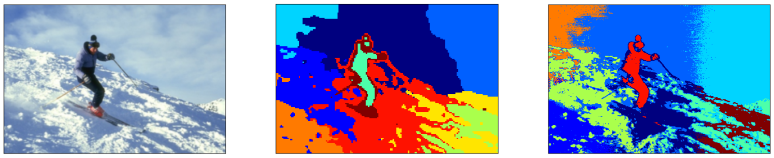
[3] Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.



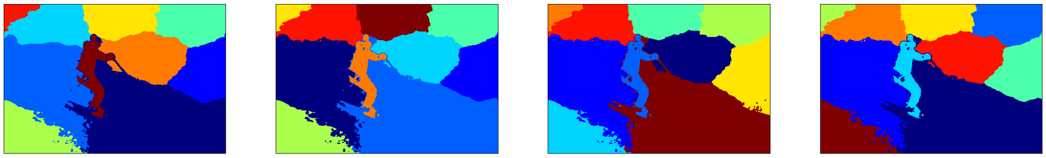
(a) Original Image (b) SKLearn GK (c) Our Algorithm



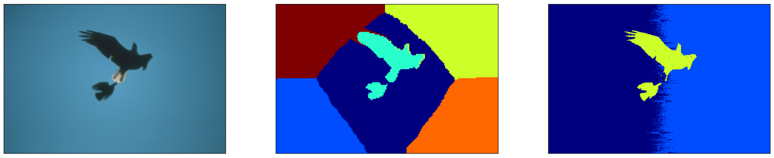
(d) SKLearn k-NN (e) FAISS Exact (f) FAISS HNSW (g) FAISS IVF



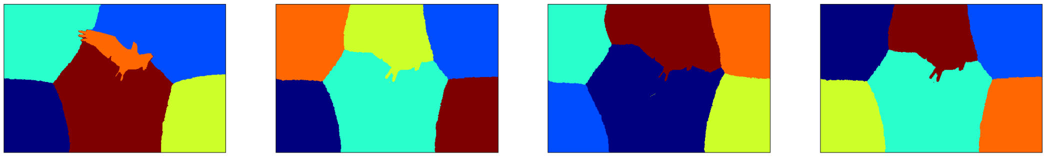
(h) Original Image (i) SKLearn GK (j) Our Algorithm



(k) SKLearn k-NN (l) FAISS Exact (m) FAISS HNSW (n) FAISS IVF

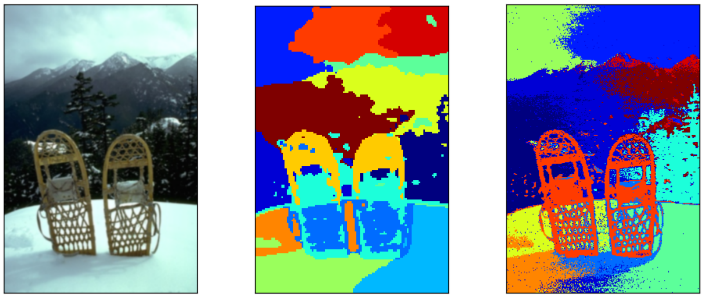


(o) Original Image (p) SKLearn GK (q) Our Algorithm

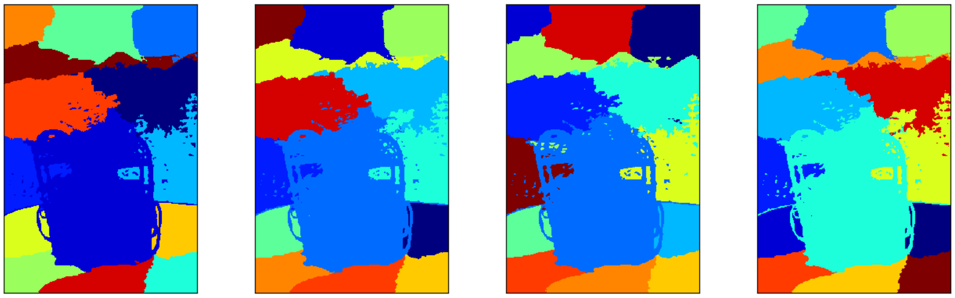


(r) SKLearn k-NN (s) FAISS Exact (t) FAISS HNSW (u) FAISS IVF

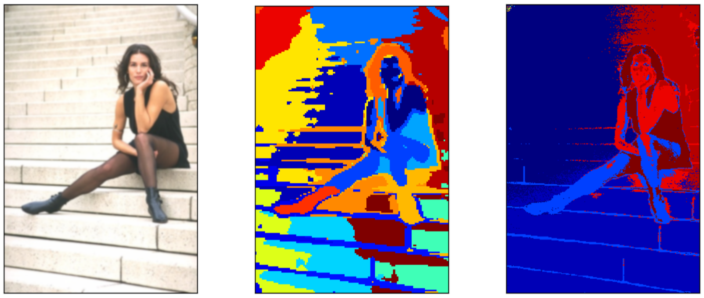
Figure 1: Further examples of the performance of the spectral clustering algorithms for image segmentation.



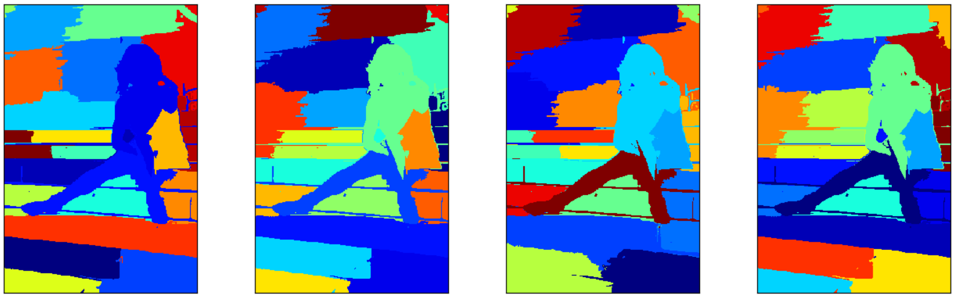
(a) Original Image (b) SKLEARN GK (c) OUR ALGORITHM



(d) SKLEARN k-NN (e) FAISS EXACT (f) FAISS HNSW (g) FAISS IVF



(h) Original Image (i) SKLEARN GK (j) OUR ALGORITHM



(k) SKLEARN k-NN (l) FAISS EXACT (m) FAISS HNSW (n) FAISS IVF

Figure 2: Further examples of the performance of the spectral clustering algorithms for image segmentation.