



## Research



**Cite this article:** Grindrod P, Higham DJ, de Kergorlay H-L. 2024 Estimating network dimension when the spectrum struggles. *R. Soc. Open Sci.* **11**: 230898.  
<https://doi.org/10.1098/rsos.230898>

Received: 3 July 2023

Accepted: 18 March 2024

### Subject Category:

Mathematics

### Subject Areas:

graph theory, computational mathematics,  
applied mathematics

### Keywords:

spectral embedding, graph, eigenvector, box  
counting, nearest neighbour

### Author for correspondence:

Desmond J. Higham

e-mail: [d.j.higham@ed.ac.uk](mailto:d.j.higham@ed.ac.uk)

# Estimating network dimension when the spectrum struggles

Peter Grindrod<sup>1</sup>, Desmond J. Higham<sup>2</sup> and Henry-Louis  
de Kergorlay<sup>2</sup>

<sup>1</sup>Mathematical Institute, University of Oxford, OX2 6GG, UK

<sup>2</sup>School of Mathematics, University of Edinburgh, Edinburgh, EH9 3FD, UK

PG, 0000-0003-0924-7010; DJH, 0000-0002-6635-3461

What is the dimension of a network? Here, we view it as the smallest dimension of Euclidean space into which nodes can be embedded so that pairwise distances accurately reflect the connectivity structure. We show that a recently proposed and extremely efficient algorithm for data clouds, based on computing first- and second-nearest neighbour distances, can be used as the basis of an approach for estimating the dimension of a network with weighted edges. We also show how the algorithm can be extended to unweighted networks when combined with spectral embedding. We illustrate the advantages of this technique over the widely used approach of characterizing dimension by visually searching for a suitable gap in the spectrum of the Laplacian.

## 1. Motivation

Given a network, it is often desirable to embed the nodes into Euclidean space so that distances between nodes reflect the connection strengths. Such an embedding may form a preprocessing step for visualization, clustering or semi-supervised learning of labels [1]. Spectral techniques, based on eigenvectors of a suitable Laplacian, are commonly used for the projection. In this case, choosing the dimension of the embedding; that is, the number of eigenvectors used, is an important task. However, this task is difficult to formalize, with a widely accepted rule of thumb being ‘look for a gap in the spectrum’ [2]. In this work, we investigate the use of the recently proposed two nearest neighbours (twoNN), algorithm in [3] as a means to inform the choice of dimension.

The twoNN algorithm is designed to estimate the dimension of a cloud of data points, assuming the points are samples from a continuous manifold. It is extremely efficient, compared,

for example, with box counting techniques, requiring only the pairwise distances between all first- and second-nearest neighbours. In the case of a weighted (undirected) network for which edge weights can be used to define the desired distances, the algorithm is directly applicable. We find that it performs well on examples where a ground truth is available and where information from the Laplacian spectrum is at best ambiguous. In the case of unweighted networks, where edge weights are either present or absent, the algorithm can no longer be applied directly. However, we show that in examples where ground truth is available, useful estimates of the dimension can be recovered by spectrally embedding into successively higher dimensional Euclidean space and applying twoNN at each stage. In experiments where the ground truth is contaminated by noise we find that, unlike the Laplacian spectrum, twoNN is robust and informative. In a final experiment on real data, the twoNN estimate remains consistent when applied to a  $K$ -nearest neighbour binarization of the underlying weighted network.

## 2. Set-up

Suppose we are given a non-negative  $N \times N$  symmetric *dissimilarity* matrix  $M$ . We assume that there are  $N$  distinct underlying objects,  $\mathbf{x}_i$ , within some universal set  $S$ , and that the element  $M_{ij}$  for  $i \neq j$  measures the dissimilarity between objects  $\mathbf{x}_i$  and  $\mathbf{x}_j$ ; so a larger  $M_{ij}$  indicates that  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are more dissimilar. Ideally, elements of  $M$  should also satisfy a triangle relationship; that is,  $M_{ij} \leq M_{ik} + M_{kj}$  for all distinct  $i, j, k$ . This would hold by construction if elements of  $M$  were derived from a metric space,  $(S, \delta)$ , containing the underlying  $N$  objects; and  $M_{ij} = \delta(\mathbf{x}_i, \mathbf{x}_j)$ . However, a triangle relationship is not necessary in what follows.

In some contexts,  $M$  will be implied via a weighted graph or a *similarity* matrix: we have a symmetric  $W \in \mathbb{R}^{N \times N}$  such that each  $W_{ij} \geq 0$  denotes the weight or strength of the connection between the distinct objects  $\mathbf{x}_i$  and  $\mathbf{x}_j$ , with  $W_{ii} = 0$ . In this case, a larger  $W_{ij}$  indicates that  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are more similar.

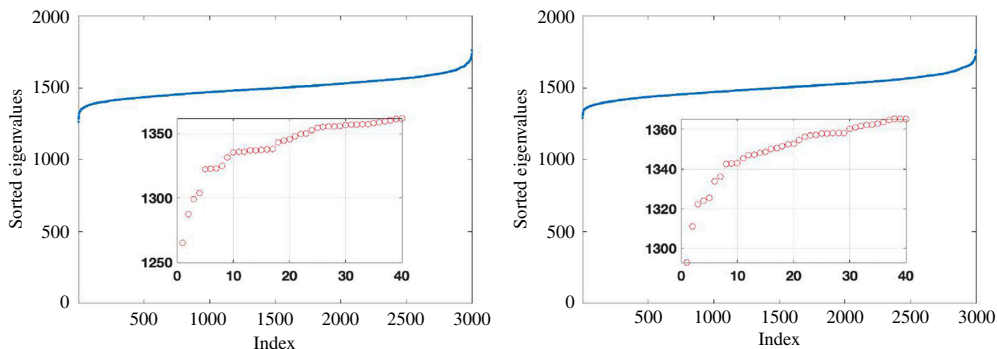
To convert between a dissimilarity matrix  $M$  and a similarity matrix  $W$ , we may use, for example,  $M_{ij} = 1/W_{ij}$ ,  $M_{ij} = 1 - W_{ij}$  or  $W_{ij} = \exp(-M_{ij}\sigma^2)$  for some  $\sigma$  [4].

Given a dissimilarity matrix,  $M$ , with no notion of  $S$ , we are often tasked with embedding  $M$  into some Euclidean space; that is, we wish to define a set of real *location* vectors,  $\{\mathbf{y}^{[i]}\}_{i=1}^N$  in  $\mathbb{R}^k$ , such that the Euclidean distances  $\|\mathbf{y}^{[i]} - \mathbf{y}^{[j]}\|$  are monotonic functions of the  $M_{ij}$ . Ideally, the embedding dimension,  $k$ , should reflect the inherent dimension  $d$  of the data. There are many ways to embed such matrices, or equivalently the weighted graphs, and these usually involve some spectral analysis of a matrix related to  $M$  or  $W$ :

- (i) [2] ‘we can see from the [Davis-Kahan] theorem that the larger this eigengap is, the closer the eigenvectors of the ideal case and the perturbed case are, and hence the better spectral clustering works.’ Also, ‘...the eigengap heuristic usually works well if the data contains very well pronounced clusters, but in ambiguous cases it also returns ambiguous results’;
- (ii) [5] ‘dimensionality reduction is achieved by projecting the original images onto the principal component directions and selecting the components corresponding to the leading eigenvectors’; and
- (iii) [6] ‘the traditional solution is to use the spectral gap of the eigenvalues to determine  $K$ ’.

For unweighted graphs, where  $W$  is binary and symmetric, there has been much interest in combinatorics. The dimension of a graph is the smallest value of  $d$  for which its vertices may be embedded in  $\mathbb{R}^d$  such that the distances between the endpoints of each edge are equal to unity. Furthermore, in answer to a question of [7], any graph with less than  $\binom{d+2}{2}$  edges has a dimension at most  $d$ ; while the dimension of a graph with maximum degree  $d$  is at most  $d$  [8].

In this work, we are concerned with the practical question of how to compute a representative value for the network dimension when it is likely that the network information is noisy or incomplete.



**Figure 1.** Ordered eigenvalues of the Laplacian based on inverse pairwise Euclidean distances between data clouds of  $N = 3000$  points in  $\mathbb{R}^{25}$ . Two independent instances are shown. Interior plots focus on the first 40 non-zero eigenvalues. The spectrum gives no obvious argument for an embedding in dimension 25.

### 3. Spectral embedding

Given the similarity matrix  $W$ , it is common to embed the graph into a  $k$ -dimensional Euclidean space using a spectral method. The embedding may be regarded as an  $N \times k$  matrix:

$$G = [\mathbf{g}^{[1]}, \mathbf{g}^{[2]}, \dots, \mathbf{g}^{[k]}],$$

with orthonormal columns  $\mathbf{g}^{[j]} \in \mathbb{R}^N$ . The  $i$ th object  $\mathbf{x}_i$  is given coordinates according to the  $i$ th row of  $G$ ; that is:

$$\mathbf{y}^{[i]} = \begin{bmatrix} \mathbf{g}_i^{[1]} \\ \mathbf{g}_i^{[2]} \\ \vdots \\ \mathbf{g}_i^{[k]} \end{bmatrix} \in \mathbb{R}^k. \quad (3.1)$$

As described in [9], it is natural to specify  $G$  via

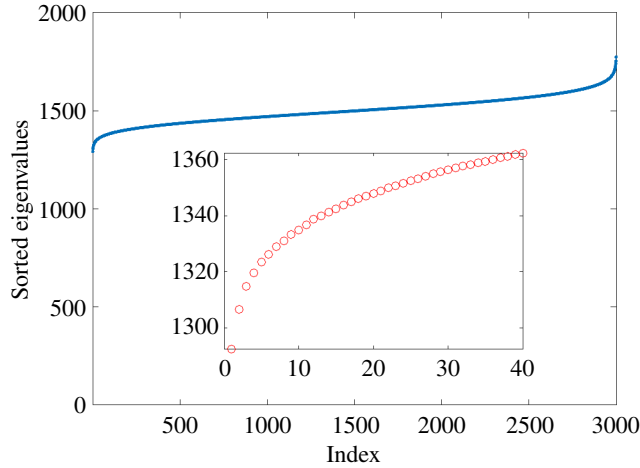
$$\arg \min_{\mathbf{g}^{[1]}, \mathbf{g}^{[2]}, \dots, \mathbf{g}^{[k]}} \sum_{i=1, j=1}^m W_{ij} \|\mathbf{g}^{[i]} - \mathbf{g}^{[j]}\|^2,$$

with  $G^T G = I$ . This leads to a solution where the columns  $\mathbf{g}^{[i]}$  are given by eigenvectors corresponding to the  $k$  lowest non-zero eigenvalues of the Laplacian matrix  $L = D - W$ . Here,  $D \in \mathbb{R}^{N \times N}$  is the diagonal matrix whose diagonal contains the row/column sums of  $W$ . We note that  $L$  is self-adjoint and has a zero eigenvalue with geometric multiplicity given by the number of connected components of the graph [2]. Other related methods are available [10]. This type of spectral embedding approach is closely related to principal component analysis and multidimensional scaling [11,12], and similar techniques are used for clustering [2,13,14], ranking [15,16], subgraph detection [17] and graph visualization [18].

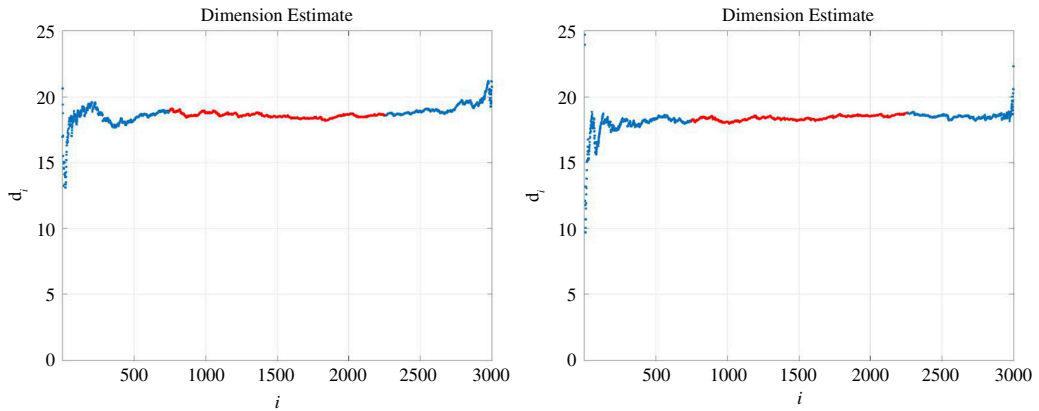
How should we best choose the embedding dimension,  $k$ ? Typically one might examine the spectrum of  $L$ , perhaps on a log scale, and search for an upward step, or *gap*, discarding the eigenvalues/eigenvectors to the right [2,6]. In many circumstances, though, no clear step may be apparent.

In essence, for a given choice of dimension  $k$ , spectral information from the Laplacian provides an optimal embedding, in a least squares sense. However, that spectral information does not always tell us how to choose the best  $k$ .

In figure 1, we show the Laplacian spectrum for two examples where  $N = 3000$ . Here, we sampled vectors  $\mathbf{x}_i \in \mathbb{R}^{25}$  with components chosen independently and uniformly at random in  $[0, 1]$ , and set  $W_{ij} = 1/\|\mathbf{x}_i - \mathbf{x}_j\|$ , for  $i \neq j$ . So, by design, we hope that the graph (or the matrix) will be embeddable in  $\mathbb{R}^{25}$ . Yet, this is not reflected by the spectrum shown in figure 1: an ‘eyeball’ search for a gap in the spectrum would not highlight a dimension of 25. Also, the gaps appear at different places in the two network samples. In figure 2, we show the average of the ordered eigenvalues over 100 independent samples, and no spectral gap is apparent around 25.



**Figure 2.** As in figure 1 with ordered eigenvalues averaged over 100 independent network instances.



**Figure 3.** TwoNN estimates for the embedding dimension. We show dimension estimates  $d_i$  in (3) versus the index  $i$ .

## 4. Two nearest neighbours and weighted graphs

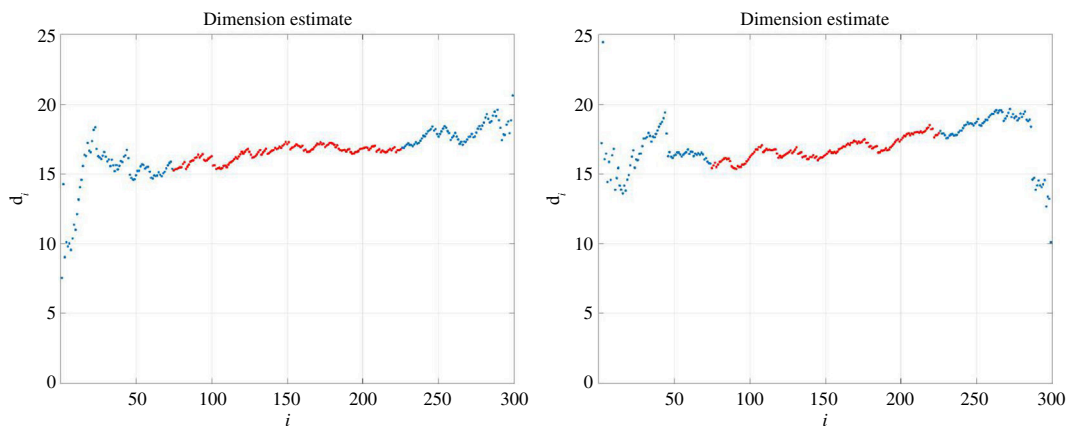
In this work, we investigate the approach of choosing the embedding dimension with the twoNN method, which was developed in [3] to estimate what those authors refer to as the intrinsic (fractal) dimension of sparse point clouds in very high-dimensional Euclidean spaces; this may be viewed as the dimension of an underlying continuous manifold from which the data points are sampled. We refer to [19,20] for further discussion of intrinsic dimension and connections to learning theory and topological data analysis. With twoNN, for all objects in the point cloud, one finds the distances to the nearest neighbour and to the second-nearest neighbour. The ratio,  $\mu > 1$ , of the latter distance to the former, calculated separately for all points, produces an empirical cumulative distribution, say  $F^{\text{emp}}(\mu)$ . This can be compared with the expression

$$F(\mu) = (1 - \mu^{-d}), \quad (4.1)$$

that the authors derived under the assumption of local uniform sampling density from a space of dimension  $d$ . Comparing the empirical and exact cumulative distributions allows us to estimate the dimension  $d$ . If we let  $\mu_i$  denote the second-to-first-nearest neighbour distance ratio for node  $i$ , and let  $\sigma$  denote a permutation vector such that  $\mu_{\sigma(i)}$  are in ascending order, then the empirical cumulative distribution has  $F^{\text{emp}}(\mu_{\sigma(i)}) = i/N$ , and the points  $[\log \mu_{\sigma(i)}, -\log(1 - F^{\text{emp}}(\mu_{\sigma(i)}))]$  should lie on a line of slope  $d$ . So, the points

$$d_i = -\frac{\log(1 - i/N)}{\log \mu_{\sigma(i)}}, \quad (4.2)$$

are estimates for the dimension  $d$ .



**Figure 4.** As for figure 3, but with the number of data points reduced from  $N = 3000$  to  $N = 300$ .

When we are presented with data in the form of a dissimilarity matrix  $M$ , we may infer pairwise distances directly: the twoNN distances of an object simply correspond to the two smallest off-diagonal elements of the corresponding row of  $M$ . In figure 3, we illustrate the method on the two datasets used in figure 1. Here, we set  $M_{ij} = 1/W_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ . The plots show  $d_i$  in (Equation 4.2) against  $i$ . As discussed in [3], for small  $i$ , we expect sampling errors to dominate, with too little information in play; whereas if  $i$  is too large then we would see the lack of very large-scale distance differences affect the estimate (since the distances are globally bounded). So, a sweetspot is desirable, where the estimate for  $d$  is relatively static. In figure 3, we highlight in red the portion of the curve where  $N/4 \leq i \leq 3N/4$ , which leads to stable estimates, and throughout this work, we use the mean of  $d_i$  over this range as our overall estimate of  $d$ . From figure 3, we obtain estimates of 18.6 and 18.4. This approach appears much more definitive than a visual search for a spectral gap. (Averaging over the estimates for the 100 network instances used in figure 2 produced a value of 18.5.)

We emphasize that in figures 1 and 3, the  $N = 3000$  data points  $\mathbf{x}_i$  were generated within the cube  $[0, 1]^{25}$ . Yet  $2^{25} = 33,554,432$ , so this set of points will hardly get close to any of the extremities. Hence, this example is challenging—high-dimensional spaces are very lonely places. Moreover, 25 should be considered as a hard upper bound for any estimate of  $d$ , with the distance/dissimilarity matrix likely to be more consistent with a smaller dimension.

From a theoretical perspective, the following conditions are known to be sufficient for a discrete graph Laplacian to converge spectrally to the Laplace–Beltrami operator over an underlying sampling manifold, for example [21]:

$$\begin{cases} \eta(0) > 0 \text{ and } \eta \text{ is continuous at } 0, \\ \eta \text{ is non-increasing,} \\ \int_{\mathbb{R}^d} \eta(x) |\langle x, e_1 \rangle|^2 dx < \infty, \end{cases} \quad (4.3)$$

where  $\eta$  is the radially symmetric kernel function such that  $W_{ij} =: \eta(\mathbf{x}_i - \mathbf{x}_j)$  for all  $i, j \in \{0, \dots, N\}$ , and where  $\eta$  is the radial profile, or the shape, of  $\eta$ , that is,  $\eta(x) = \eta(|x|)$  for all  $x \in \mathbb{R}^d$ . However, in a typical practical setting, such conditions cannot be validated, in which case there is no guarantee that the spectral embedding is consistent. Following on from the arguments above, we regard the estimate of  $d \approx 18.5$  from figure 3 provided by twoNN as acceptable, even though the conditions in equation (4.3) for  $W$  are not satisfied. Moreover, figure 4 shows an experiment where  $N$  is reduced to 300. Here, twoNN still produces consistent estimates of 16.5 and 16.8.

We also note that the algorithm is insensitive to smooth rescaling of the distance measure, in the sense that for any differentiable function  $f$  such that  $f(0) = 0$  and  $f'(0) \neq 0$ , and small distances  $r_1$  and  $r_2$ , we have

$$\frac{f(r_2)}{f(r_1)} \approx \frac{f(0) + r_2 f'(0)}{f(0) + r_1 f'(0)} = \frac{r_2}{r_1}.$$

We, therefore, suggest that the direct application of the twoNN method from [3] on a weighted network provides reliable information for choosing a spectral embedding dimension  $k$ , and at the very least may be regarded as a backup procedure or sanity check for the widely adopted approach of

visually inspecting the spectrum of the Laplacian. In the remainder of the manuscript, we focus on the more challenging case of an unweighted network, where the method is not directly applicable.

## 5. Two two nearest neighbours and unweighted graphs

With an unweighted graph, it remains natural to seek a node embedding such that nearby nodes are connected, and distant nodes are unconnected. For example, we may postulate that the connectivity structure in the graph arose from some sort of (unobserved) binarization mechanism, such as  $K$ -nearest neighbour (KNN) or radius-based thresholding (geometric). For an unweighted graph, the notion of first- and second-nearest neighbour is not immediately applicable. Hence, we advocate an indirect approach where spectral embedding is used as an intermediate step. Here, we look for the largest  $s$  such that after spectrally embedding into dimension  $s$  the twoNN algorithm also delivers an estimate close to  $s$  for the dimension. Intuitively, if we spectrally embed into a dimension that is unnecessarily small, then twoNN will reproduce this dimension, whereas if we spectrally embed into a dimension that is unnecessarily large, then twoNN will find the appropriate, smaller dimension.

To be concrete, in algorithm 1 we outline the steps involved in spectral embedding followed by the application of twoNN. Our overall approach is then to apply algorithm 1 for  $s = 2, 3, \dots$  and observe how the twoNN estimate  $d^*$  compares with the embedding dimension  $s$ , stopping when  $d^*$  plateaus as a function of  $s$ .

In the next subsection, we test this approach as follows. First, we create a ground truth by starting with a node sampling. We then binarize the pairwise distance information using a KNN construction. We consider two different sampling settings; first from the standard normal distribution on  $\mathbb{R}^d$  and second using components that are uniform on  $[0, 1]$ .

**input** : Similarity (adjacency) **unweighted** matrix,  $W \in \mathbb{R}^{N \times N}$ ; trial embedding dimension,  $s$

**output**: twoNN dimension estimate,  $d^*$

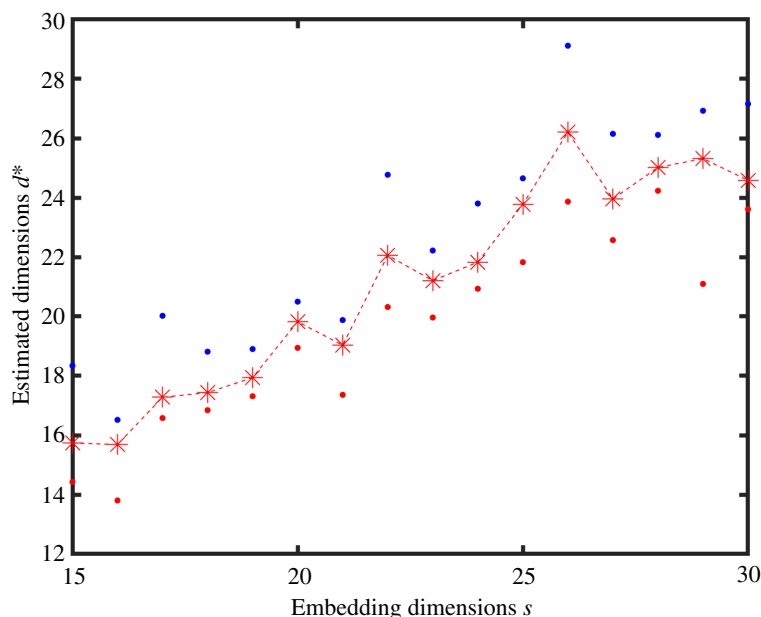
- 1  $\Delta \leftarrow$  graph Laplacian obtained from  $W$ ;
- 2  $g^{[1]}, \dots, g^{[s]} \leftarrow s$  eigenvectors associated with the  $s$  first non-trivial eigenvalues of  $\Delta$ ;
- 3  $y^{[1]}, \dots, y^{[N]} \leftarrow N$  embedding locations from (3.1);
- 4 **for**  $i = 1 : N$  **do**
- 5      $r_i^1, r_i^2 \leftarrow$  1st and 2nd nearest neighbour distances for  $y^{[i]}$ ;
- 6      $\mu_i \leftarrow \frac{r_i^2}{r_i^1}$ ;
- 7 **end**
- 8  $\sigma \leftarrow$  permutation such that  $\mu \circ \sigma$  is ordered in ascending order;
- 9  $d_1, d_2, \dots, d_N \leftarrow$  dimension estimates from (4.2);
- 10  $d^* \leftarrow$  mean of  $d_i$  over  $N/4 \leq i \leq 3N/4$ ;

The case of binarization via a geometric graph construction is considered in §5.2.

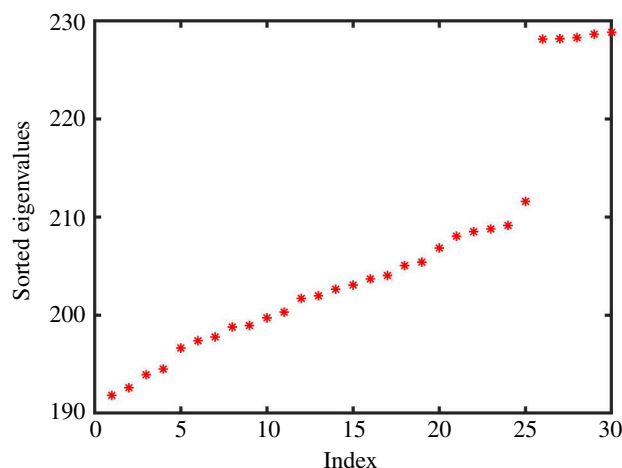
### 5.1. Results for $K$ -nearest neighbour construction

Given  $K$ , let  $A \in \mathbb{R}^{N \times N}$  be the adjacency matrix of a KNN graph constructed as follows. Associate  $\mathbf{x}^{[i]}$  with a vector in  $\mathbb{R}^d$  with independently chosen entries and record an edge between every distinct pair of nodes if one of them is among the KNN of the other. There exist alternative constructions for building a KNN graph, for example [22], all yielding a symmetric affinity matrix, and we expect the computational results and choices of parameters to be equivalent for these alternative constructions.

It is known [23,24] that unless  $K = \Omega(\log N)$ , the KNN graph (any construction) is almost surely not connected for  $N$  sufficiently large. It is also known [25] that one must choose  $K = \omega(\log N)$  in order for the discrete graph Laplacian to converge spectrally to the underlying continuous Laplace–Beltrami operator, whose spectrum characterizes the geometry of the underlying sampling domain. If this



**Figure 5.** Results from algorithm 1 with dimension  $s$  varying from 15 to 30 on the horizontal axis. Here, we sampled  $N = 3000$  points from a Gaussian in  $\mathbb{R}^{25}$  and used a  $k$ NN construction to produce an unweighted graph. The red dots, blue dots and asterisks show the minimum, maximum and mean  $d_i$  value for each  $s$ .



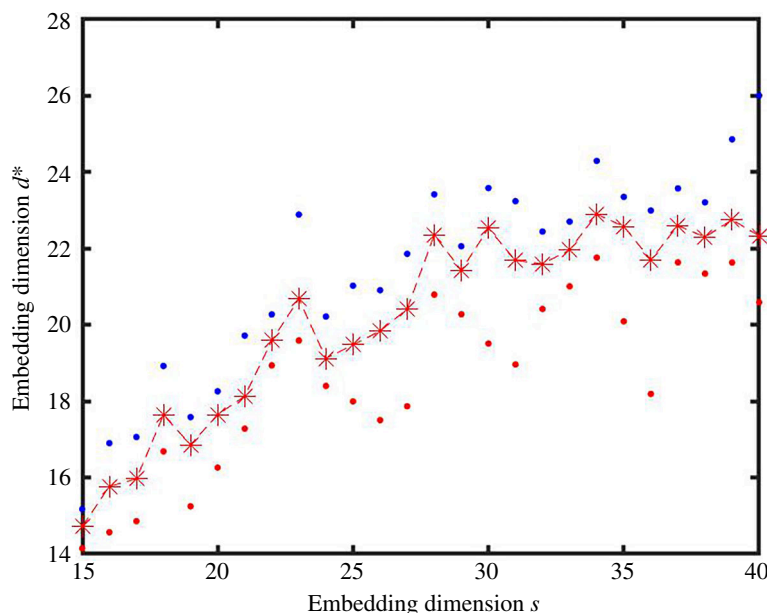
**Figure 6.** First 30 ordered non-zero eigenvalues for the Laplacian of an unweighted  $k$ NN graph from the experiment in figure 5, showing a spectral jump at dimension  $d = 25$ .

condition is not satisfied, the eigenvectors of the discrete Laplacian are not guaranteed to converge to the eigenfunctions of the associated Laplace–Beltrami operator; hence the spectral embedding in equation (3.1) is not guaranteed to be accurate. Based on this observation, it makes sense to choose  $K = \omega(\log N)$ . In our experiments, we fix  $d = 25$ ,  $N = 3000$ , and choose  $K := \lceil 30 \log N \rceil$ . The factor 30 is chosen somewhat arbitrarily as a value that is significantly larger than  $\log N$ .

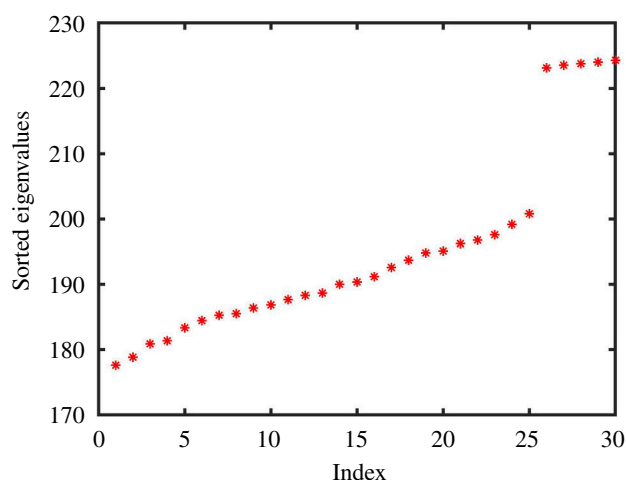
Figure 5 shows the estimation of the intrinsic dimension from algorithm 1, for each embedding dimension in  $\{15, 16, \dots, 30\}$ , in the case where we sample  $N = 3000$  points from a Gaussian in  $\mathbb{R}^{25}$ . We see that the slope increases and stabilizes around 25. Figure 6 shows that the corresponding graph Laplacian has a spectral jump at  $d = 25$ . In the next section, we will show that this spectral information degrades in the presence of noise.

We obtain similar pictures in the case where we sample the data point components uniformly in  $[0, 1]$ , as indicated in figures 7 and 8. Here, algorithm 1 stabilizes at around 22.

We refer to §5.3 for a discussion of the assumptions under which twoNN was originally derived.



**Figure 7.** As for figure 5 with components sampled uniformly in  $[0, 1]$  and embedding dimension between 15 and 40.



**Figure 8.** As for figure 6 with components sampled uniformly in  $[0, 1]$ .

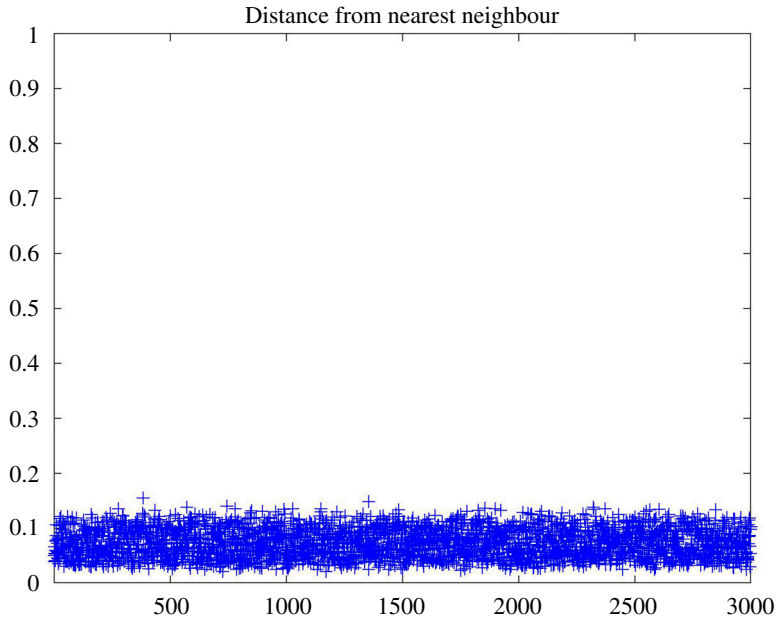
## 5.2. Geometric graph constructions and the curse of dimensionality

For random geometric graphs where edges are inserted when the pairwise distance is below a radius  $r > 0$ , there are well-known results on connectivity; for example, [26,27] provide asymptotic conditions. However, choosing  $r$  suitably in practice remains challenging, as we are typically in a non-asymptotic regime and becomes unfeasible as the intrinsic dimension of the data becomes large. This is an example of what is commonly referred to as the curse of dimensionality. We illustrate the effect with the following geometric construction, which helps to explain what may go wrong when the dimension is large.

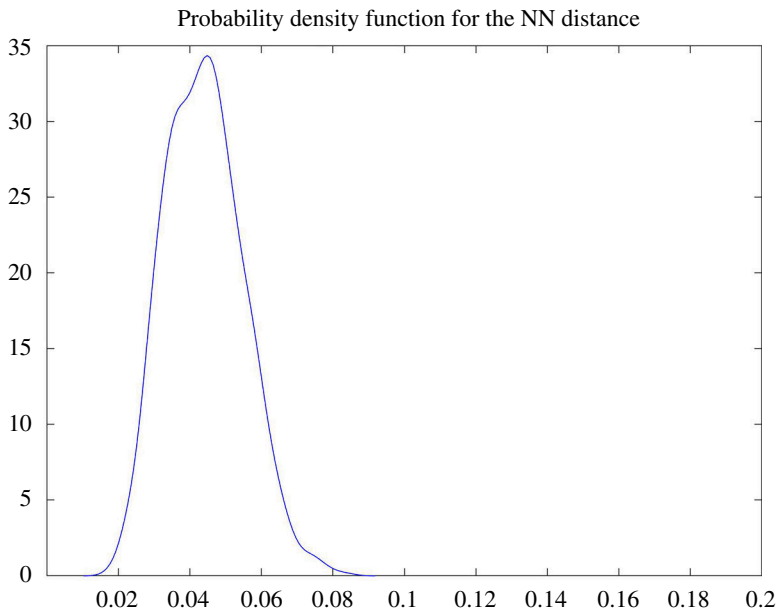
Given the unit cube  $C$  in  $\mathbb{R}^d$ , consider the  $d$ -dimensional rectangle  $R$  given by  $[0, 1] \times [r, 1-r]^{d-1}$ . Let  $e_1, \dots, e_d$  be orthonormal vectors in the  $d$  directions spanned by the axes of  $\mathbb{R}^d$ . The rectangle  $R$  has exactly two faces per direction ( $(d-1)$ -dimensional faces), and we can assume, without loss of generality, that the interval  $[0, 1]$  of  $R$  is along the  $e_1$  direction, so that the two faces in the  $e_1$  direction are squares contained in two faces of the unit cube. These faces can be written as

$$\{(\gamma_1, \dots, \gamma_d) \mid \gamma_1 = \epsilon, \forall k \in \{2, \dots, d\}, \gamma_k \in [r, 1-r]\},$$

where  $\epsilon \in \{0, 1\}$ . Letting  $k \in \{2, \dots, d\}$ , the two faces associated with the  $e_k$  direction can be written as



**Figure 9.** Nearest neighbour distance for each of the 3000 points uniformly sampled from the unit cube in  $\mathbb{R}^{25}$ , spectrally embedded in  $\mathbb{R}^{30}$ , obtained from a  $\kappa$ NN construction. We observe that the nearest neighbour distances concentrate tightly around an expected distance. This suggests that the embedded points are close to being uniformly distributed, in which case their nearest neighbour distances would follow an approximately normal distribution.



**Figure 10.** Sampling density of the nearest distance neighbours from the spectral embedding of dimension 30, from points sampled uniformly from the unit cube in  $\mathbb{R}^{25}$ , obtained from a  $\kappa$ NN construction as in figure 9. The horizontal axis indicates the value of the nearest neighbour distance, and the vertical axis indicates the number of points whose nearest neighbour distance satisfies this value. The normal-like distribution suggests that the points in the spectral embedding are approximately uniformly distributed.

$$\{(\gamma_1, \dots, \gamma_d) \mid \gamma_1 \in [0, 1], \gamma_k = \gamma, \forall j \notin \{1, k\}, \gamma_j \in [r, 1-r]\},$$

where  $\gamma \in [r, 1-r]$  is the constant value of the  $k$ th coordinate of every element of a given face in the  $e_k$  direction.

For a direction  $k \in \{2, \dots, d\}$  and a face in the  $e_k$  direction given by the choice of  $\gamma \in [r, 1-r]$ , define the  $d$ -dimensional rectangle:

$$R_{k,\gamma} := \{(\gamma_1, \dots, \gamma_d) \mid \gamma_1 \in [0, 1], \gamma_k \in I_\gamma, \forall j \notin \{1, k\}, \gamma_j \in [1/3, 2/3]\},$$

where

$$I_\gamma := \begin{cases} [1-r, 1], & \text{if } \gamma = 1-r, \\ [0, r], & \text{if } \gamma = r. \end{cases}$$

Each such rectangle ‘closes the gap between two adjacent corners of the unit cube’, such that the set obtained, after removing from the unit cube  $R$  and every  $R_{k,\gamma}$ ,  $k \in \{2, \dots, d\}$ ,  $\gamma \in \{r, 1-r\}$ , consists of  $2^{d-1}$  connected components. Furthermore, the volume of the union of these sets is

$$V(R \cup \cup_{k,\gamma} R_{k,\gamma}) = 2(d-1)(1/3)^{d-2}r + (1-2r)^{d-1},$$

which gets very small very quickly as the dimension  $d$  increases, even for small values of  $r$ . For instance, for  $d = 10$  and  $r = 0.1$ , this volume is 0.135, for  $d = 15$ , it is 0.0440, and for  $d = 25$ , it is 0.0047.

Thus, unless the dimension of the dataset is small, most of the sampled points will be equi-distributed among the  $2^{d-1}$  connected components of the unit cube after having removed  $R \cup \cup_{k,\gamma} R_{k,\gamma}$ , which as a union do not form a connected set. It is not possible to create a connected geometric graph from such a sample, unless we choose  $r > 0$  trivially to be so large that its size is comparable to the side width of the unit cube, in which case, the generated graph will be close to being complete. This issue does not arise with the KNN construction, since the number of neighbours is fixed and determined by the choice of  $K$ , which allows us to choose  $K$  such that the graph is both connected and sufficiently sparse, even in high dimensions.

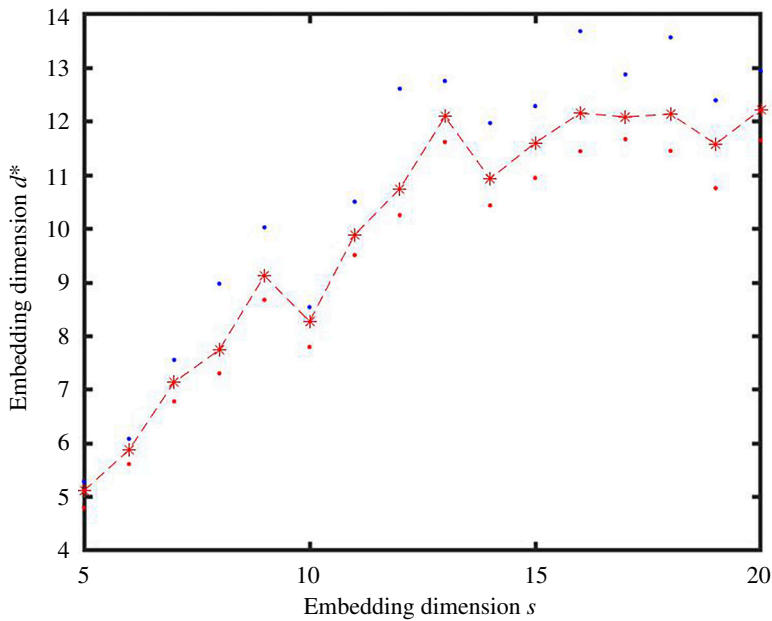
This thought experiment also indicates that, in our context, it is unreasonable to assume that a high-dimensional unweighted graph arose from a random geometric graph construction—sparsity and connectivity are unlikely to hold simultaneously. A KNN construction is thus a more realistic binarization mechanism.

### 5.3. Observations on non-uniform sampling densities and $K$ -nearest neighbour constructions

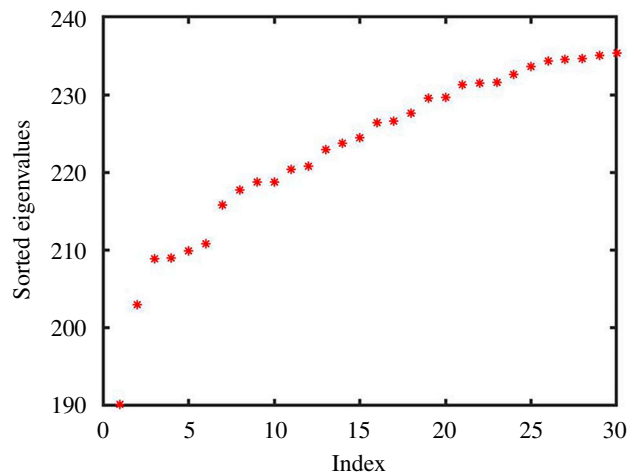
Neither the uniform distribution on the unit cube nor the Gaussian distribution on  $\mathbb{R}^d$  is uniform in the underlying metric space (here,  $\mathbb{R}^d$  with the Euclidean distance in both cases). Hence, the sampling density will not be approximately constant in the neighbourhood of every sampled point, a condition that was assumed in the derivation of the twoNN algorithm in [3]. In the case of the uniform distribution on the unit cube, the sampling density fails to be approximately constant around points close to the boundary, while in the case of a Gaussian distribution, the sampling density is never approximately constant, in any neighbourhood. However, the KNN graph construction allows us to cancel to some extent the negative effect of the non-uniformity of the sampling densities, as observed by the accurate estimations of the intrinsic dimension of the data in §5.1. By analogy with a geometric graph, where points are connected if they are at a distance less than a fixed bandwidth parameter  $r > 0$ , a KNN graph can be thought of as connecting points if they are at a distance less than a varying bandwidth parameter, inversely proportional to the values taken by the density in a given neighbourhood of those points. Every point is connected to roughly the same number of neighbours, mimicking a geometric graph if the points had been sampled from a uniform density, cancelling the presence of irregularities on the sampling domain, such as boundaries. We can thus expect the spectrally embedded points in twoNN to be approximately uniformly distributed in the case of a KNN construction, even if the points were sampled from a non-uniform density on the original domain.

We test this hypothesis by looking at the distribution of the embedded points  $\{\mathbf{y}^{[i]} | i \in \{1, \dots, n\}\}$  in  $\mathbb{R}^s$ ,  $s > d$ , in the case where the points are uniformly sampled from the unit cube in  $[0, 1]^d$ . In the figures below, we choose  $d = 25$  and  $s = 30$ .

If the sampling density is uniform, then the nearest neighbour distances of the sampled points should concentrate highly around their expected value, forming an inverse exponential distribution centred around the expected nearest distance. In figure 9, we plot and compare the nearest distances for all embedded points. We observe that the nearest neighbour distances are indeed highly concentrated around their expected value, which does suggest that the embedded points in the image space must be approximately uniformly distributed. In figure 10, the approximate uniform distribution of the embedded points is confirmed by the observation that the nearest neighbour distances concentrate around their expected value following a Gaussian-like distribution.



**Figure 11.** Results from algorithm 1 with dimension  $s$  varying from 5 to 20 on the horizontal axis. Here, we sampled  $N = 17000$  points from the unit cube  $[0, 1]^{15}$ , used a  $k$ NN construction to produce an unweighted graph and then flipped each edge/missing edge with independent probability  $p = 0.01$ . The red dots, blue dots and asterisks show the minimum, maximum and mean  $d_i$  value for each  $s$ .

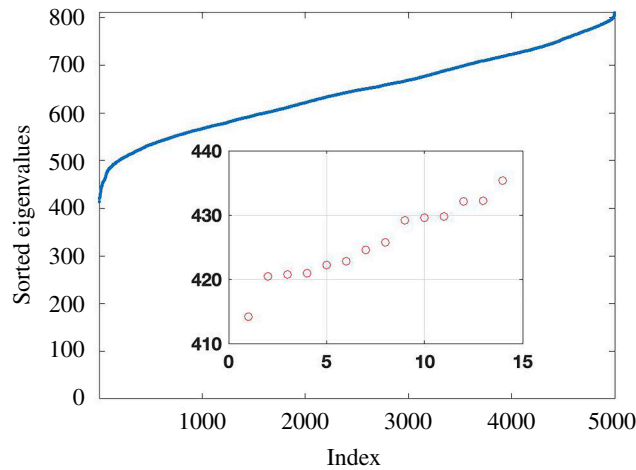


**Figure 12.** First 30 ordered eigenvalues for the Laplacian of an unweighted  $k$ NN graph from the experiment in figure 11 based on data points from  $[0, 1]^{15}$ .

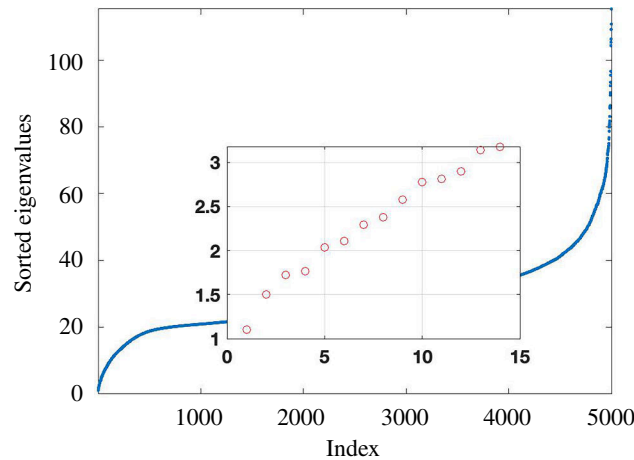
## 6. Tests on noisy data

The experiments in §5 assume that the similarity matrix is recorded with perfect information. In that idealized setting, we found that the twoNN algorithm did not provide a better method to infer the intrinsic dimension of the data than a simple spectral gap reading. In this section, we test a more realistic scenario where the information is recorded with noise.

For every entry of the upper triangular part of the binary adjacency matrix  $A$ , we change independently the value from 0 to 1 or from 1 to 0 with probability  $p$ , and otherwise leave it unchanged. To keep the matrix  $A$  symmetric, the upper triangular part of  $A$  determines the remaining entries of  $A$ . In other words, each edge and each missing edge are flipped independently with probability  $p$ . In our experiment, we used intrinsic dimension  $d = 15$  and sampled  $N = 17000$  vectors uniformly at random from the unit cube  $[0, 1]^d$ . We generated  $A$  from a  $k$ NN construction, picking  $K = \lceil 30 \log(N) \rceil$  as before, and introduced noise to  $A$  as described above with parameter  $p = 0.01$ .



**Figure 13.** Ordered non-zero eigenvalues of the Laplacian based on inverse pairwise Euclidean distances between  $N = 5000$  handwritten digits from 0 to 9. Interior plot zooms in on the smallest eigenvalues.



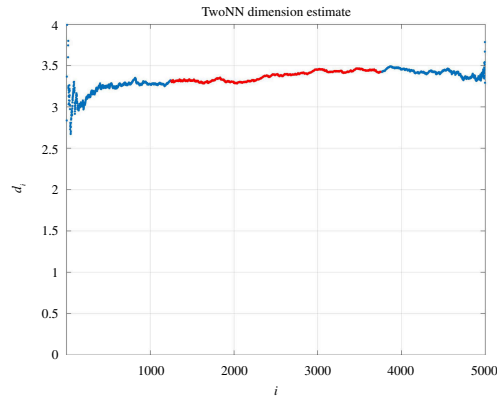
**Figure 14.** Ordered non-zero eigenvalues of the Laplacian based on  $K = 20$  nearest neighbour graph from  $N = 5000$  handwritten digits from 0 to 9. Interior plot zooms in on the smallest eigenvalues.

With this introduction of noise, algorithm 1 stabilizes to an estimate of the intrinsic dimension at around 12, as shown in figure 11. In figure 12, we see, however, that the spectrum of the Laplacian does not allow a definitive answer, with noticeable jumps at 1, 2, 6, 10, 12, 15, 18 and 20. Intuitively, the twoNN construction should be less sensitive to the presence of a small percentage of missing or spurious edges, since these perturbations will not affect the majority of the first or second pairwise distances.

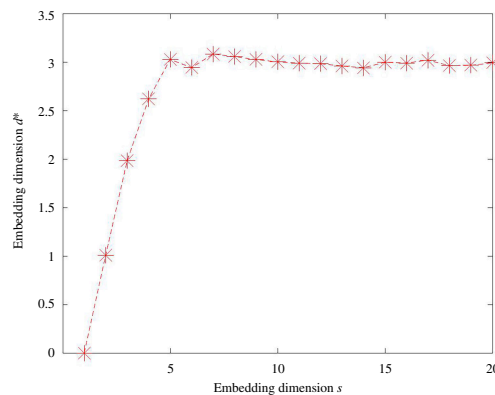
## 7. A test on real data

We finish with a test on MNIST image data [28]. Here, each data point in  $\mathbb{R}^{784}$  represents the  $28 \times 28$  greyscale pixel values of a handwritten digit, from 0 to 9. We uploaded 5000 images using `digit-Train4DArrayData` in MATLAB [29], where each image has been arbitrarily rotated. The Laplacian spectrum for the similarity matrix based on the reciprocal of Euclidean distance is shown in figure 13. We then constructed the binarized  $K = 20$  nearest neighbour graph. The corresponding spectrum is shown in figure 14. We see that neither spectrum plot shows a definitive spectral gap nor the two plots are consistent.

Applying twoNN to the Euclidean distance data gave a dimension estimate of  $d^* = 3.37$ . The individual  $d_i$  values in equation (4.2) are shown in figure 15. For the  $K = 20$  nearest neighbour version of the data, figure 16 shows the results from algorithm 1 as the embedding dimension  $s$  varies from 5 to



**Figure 15.** TwoNN estimates for the embedding dimension of the MNIST data. We show dimension estimates  $d_i$  in (3) versus the index  $i$ .



**Figure 16.** Results from algorithm 1 with dimension  $s$  varying from 5 to 20 on the horizontal axis. Here, we used the same nearest neighbour graph as in figure 14.

20. We see that on this binarized data, it is possible to use twoNN in a way that gives an estimate that is consistent with the result on the original pairwise distance data.

We emphasize that the aim of this test was not to produce a definitive result for the underlying dimension of MNIST. This issue has been tackled in a number of works, including [30,31], with answers that depend on the way that the concept of dimension is introduced. We aimed to test instead whether algorithm 1 allows twoNN to remain consistent under KNN binarization. We note, however, that the MNIST data can be embedded into three-dimensional space in a visually pleasing manner.<sup>1</sup>

## 8. Conclusions

The twoNN algorithm in [3] gives a computationally efficient way to estimate the dimension of a data cloud, assuming the points are samples from a continuous manifold. The algorithm requires only first- and second-nearest neighbour distances; for a weighted (undirected, not necessarily connected) network this information is immediately available, and we found that the algorithm performed well on examples where a ground truth is available, and where information from the Laplacian spectrum was not useful. For unweighted networks, where the given ‘distance’ data only take two possible values, the algorithm can no longer be applied directly. However, we showed that plausible estimates of the dimension can be recovered by spectrally embedding into successively higher-dimensional Euclidean space and applying twoNN at each stage. We also found that this approach was more robust to noise than the direct use of the Laplacian spectrum, and more consistent under KNN binarization.

<sup>1</sup><https://colah.github.io/posts/2014-10-Visualizing-MNIST/>.

Overall, these results highlight and extend the usefulness of twoNN in the context of network analysis.

**Ethics.** This work did not require ethical approval from a human subject or animal welfare committee.

**Data accessibility.** Only public domain data was used. Computer codes to reproduce all experiments are available at <https://www.maths.ed.ac.uk/~dhigham/algfiles.html>.

**Declaration of AI use.** We have not used AI-assisted technologies in creating this article.

**Authors' contributions.** P.G.: conceptualization, formal analysis, methodology, writing—original draft, writing—review and editing; D.J.H.: conceptualization, formal analysis, methodology, software, writing—original draft, writing—review and editing; H.-L.d.K.: conceptualization, formal analysis, methodology, software, writing—original draft, writing—review and editing.

All authors gave final approval for publication and agreed to be held accountable for the work performed therein.

**Conflict of interest declaration.** We declare we have no competing interests.

**Funding.** D.J.H. was supported by the Engineering and Physical Sciences Research Council under grants EP/P020720/1 and EP/V015605/1.

**Acknowledgements.** We thank Clive Bowman for comments on this work.

## References

- Bertozzi AL, Luo X, Stuart AM, Zygalakis KC. 2018 Uncertainty quantification in graph-based classification of high dimensional data. *SIAM-ASA J. Uncertain. Quantif.* **6**, 568–595. (doi:10.1137/17M1134214)
- von Luxburg U. 2007 A tutorial on spectral clustering. *Stat. Comput.* **17**, 395–416. (doi:10.1007/s11222-007-9033-z)
- Facco E, d'Errico M, Rodriguez A, Laio A. 2017 Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Sci. Rep.* **7**, 12140. (doi:10.1038/s41598-017-11873-y)
- de Kergorlay HL, Higham DJ. 2022 Consistency of anchor-based spectral clustering. *Inf. Inference* **11**, 801–822. (doi:10.1093/imaia/iaab023)
- Luo B, C. Wilson R, Hancock ER. 2003 Spectral embedding of graphs. *Pattern Recognition* **36**, 2213–2230. (doi:10.1016/S0031-3203(03)00084-0)
- Cheng X, Mishne G. 2020 Spectral embedding norm: looking deep into the spectrum of the graph Laplacian. *SIAM J. Imaging Sci.* **13**, 1015–1048. (doi:10.1137/18m1283160)
- Erdős P, Simonovits M. 1980 On the chromatic number of geometric graphs. *Ars Combin.* **9**, 229–246.
- Frankl N, Kupavskii A, Swanepoel KJ. 2020 Embedding graphs in Euclidean space. *J. Comb. Theory Ser. A* **171**, 105146. (doi:10.1016/j.jcta.2019.105146)
- Belkin M, Niyogi P. 2008 Towards a theoretical foundation for Laplacian-based manifold methods. *J. Comp. Syst. Sci.* **74**, 1289–1308. (doi:10.1016/j.jcss.2007.08.006)
- Makarov I, Kiselev D, Nikitinsky N, Subelj L. 2021 Survey on graph embeddings and their applications to machine learning problems on graphs. *PeerJ. Comput. Sci.* **7**, e357. (doi:10.7717/peerj-cs.357)
- Borg I, Groenen P. 2005 *Modern multidimensional scaling: theory and applications*, 2nd edn. New York, NY: Springer Verlag.
- Jolliffe IT, Cadima J. 2016 Principal component analysis: a review and recent developments. *Phil. Trans. A. Math. Phys. Eng. Sci.* **374**, 20150202. (doi:10.1098/rsta.2015.0202)
- Chodrow P, Eikmeier N, Haddock J. 2023 Nonbacktracking spectral clustering of nonuniform hypergraphs. *SIAM J. Math. Data Sci.* **5**, 251–279. (doi:10.1137/22M1494713)
- Kazimer J, Domenico MD, Mucha PJ, Taylor D. 2022 Ranking edges by their impact on the spectral complexity of information diffusion over networks. *ArXiv*. (doi:arXiv:2210.15148)
- d'Aspremont A, Cucuringu M, Tyagi H. 2021 Ranking and synchronization from pairwise measurements via SVD. *J. Mach. Learn. Res.* **22**, 866–928. (doi:1906.02746)
- Shur D, Huang Y, Gleich DF. 2022 A flexible PageRank-based graph embedding framework closely related to spectral eigenvector embeddings. *J. Appl. Comput. Topol.* **37**, 2207. (doi:10.1007/s41468-023-00129-6)
- Boyd ZM, Fraiman N, Marzuola JL, Mucha PJ, Osting B. 2024 An escape time formulation for subgraph detection and partitioning of directed graphs. *SIAM J. Matrix Anal.* **45**, 685–711. (doi:10.1137/23M155379)
- Koren Y. 2005 Drawing graphs by eigenvectors: theory and practice. *Comput. Math. Appl.* **49**, 1867–1888. (doi:10.1016/j.camwa.2004.08.015)
- Birdal T, Lou A, Guibas LJ, Simsekli U. 2021 Intrinsic dimension, persistent Homology and generalization in neural networks. In *Advances in neural information processing systems*, vol. 34 (eds M Ranzato, A Beygelzimer, Y Dauphin, PS Liang, JW Vaughan), pp. 6776–6789. Curran Associates, Inc.
- Cui Z, Grindrod P. 2023 Mappings, dimensionality and reversing out of deep neural networks. *IMA J. Appl. Math.* hxad019. (doi:10.1093/imamat/hxad019)

21. García Trillos N, Slepčev D. 2018 A variational approach to the consistency of spectral clustering. *Appl. Comput. Harmon. Anal.* **45**, 239–281. (doi:10.1016/j.acha.2016.09.003)
22. Maier M, Hein M, von Luxburg U. 2009 Optimal construction of k-nearest-neighbor graphs for identifying noisy clusters. *Theor. Comput. Sci* **410**, 1749–1764. (doi:10.1016/j.tcs.2009.01.009)
23. Balister P, Bollobás B, Sarkar A, Walters M. 2005 Connectivity of random k-nearest-neighbor graphs. *Adv. Appl. Probab* **37**, 1–24. (doi:10.1239/aap/1113402397)
24. Xue F, Kumar PR. 2004 The number of neighbors needed for connectivity of wireless networks. *Wireless Networks* **10**, 169–181. (doi:10.1023/B:WINE.0000013081.09837.c0)
25. Calder J, García Trillos N. 2022 Improved spectral convergence rates for graph laplacians on  $\epsilon$ -graphs and k-NN graphs. *Appl. Comput. Harmon. Anal.* **60**, 123–175. (doi:10.1016/j.acha.2022.02.004)
26. Penrose MD. 1997 The longest edge of the random minimal spanning tree. *Ann. Appl. Probab.* **7**, 340–361. (doi:10.1214/aoap/1034625335)
27. Penrose MD. 2003 *Random geometric graphs. Oxford studies in probability*. Oxford, UK: Oxford University Press. See <https://academic.oup.com/book/9064>.
28. LeCun Y, Cortes C, Burges CJC. 2010 The MNIST database of handwritten digits <http://yann.lecun.com/exdb/mnist/>
29. MATLAB. 2022 Version: 9.13.0.2080170 (R2022B). Natick, MA: The Mathworks Inc.
30. Hein M, Audibert J. 2005 Intrinsic dimensionality estimation of submanifolds in  $\mathbb{R}^d$ . In *Proc. of the 22nd Int. Conf. on Machine Learning*, pp. 289–296 New York, NY: Association for Computing Machinery.
31. Pope P, Zhu C, Abdelkader A, Goldblum M, Goldstein T. 2021 The intrinsic dimension of images and its impact on learning. In *9th Int. Conf. on Learning Representations*. <https://openreview.net/>.