ider: Intrinsic Dimension Estimation with R

by Hideitsu Hino

Abstract In many data analyses, the dimensionality of the observed data is high while its intrinsic dimension remains quite low. Estimating the intrinsic dimension of an observed dataset is an essential preliminary step for dimensionality reduction, manifold learning, and visualization. This paper introduces an R package, named **ider**, that implements eight intrinsic dimension estimation methods, including a recently proposed method based on second-order expansion of probability mass function and generalized linear model. The usage of each function in the package is explained with datasets generated using a function that is also included in the package.

Introduction

An assumption that the intrinsic dimension is low even when the apparent dimension is high—that the data distribution is constrained onto a low dimensional manifold—is the basis of many machine learning and data analysis methods, such as dimension reduction and visualization (Cook and Yin, 2001; Kokiopoulou and Saad, 2007). Without good estimates of the intrinsic dimension, dimensionality reduction is no more than a risky bet, insofar as one does not know to what extent the dimensionality can be reduced. We may overlook important information by projecting the original data on too small dimensional subspace. By analyzing high-dimensional data unnecessarily, computation resources and time can be wasted. When we use visualization techniques to gain insights about data, it is essential to understand whether the data at hand can be safely visualized at low dimensions, and to what extent the original information will be preserved via the visualization method. Several methods for intrinsic dimension estimation (IDE) have been proposed, and they can be roughly divided into two categories:

- projection-based methods, and
- distance-based methods.

The former category of IDE methods basically involves two steps. First, the given dataset is partitioned. Then, in each partition, principal component analysis (PCA) or another procedure for finding a dominant subspace is performed. This approach is generally easy to implement and suitable for exploratory data analysis (Fukunaga and Olsen, 1971; Verveer and Duin, 1995; Kambhatla and Leen, 1997; Bruske and Sommer, 1998). However, the estimated dimension is heavily influenced by how the data space is partitioned. Moreover, it is also unknown how the threshold for the eigenvalue obtained by PCA should be determined. This class of methods is useful for explanatory analysis with human interaction and trial-and-error iteration. However, it is unsuitable for plugging into a pipeline for automated data analysis, and we do not consider this sort of method in this paper.

The package **ider** implements various methods for estimating the intrinsic dimension from a set of observed data using a distance-based approach (Pettis et al., 1979; Grassberger and Procaccia, 1983; Kégl, 2002; Levina and Bickel, 2005; Hein and Audibert, 2005; Fan et al., 2009; Gupta and Huang, 2010; Eriksson and Crovella, 2012; Hino et al., 2017). The implemented algorithms work with either a data matrix or a distance matrix. There are a large number of distance-based IDE methods. Among them, methods based on the fractal dimension (Mandelbrot, 1977) are well studied in the fields of both mathematics and physics. The proposed package **ider** implements the following fractal dimension-based methods:

`corint`: the correlation integral (Grassberger and Procaccia, 1983)

`convU`: the kernel-version of the correlation integral (Hein and Audibert, 2005)

`packG`,`packT`: capacity dimension-based methods with packing number estimation (a greedy method (Kégl, 2002) and a tree-based method (Eriksson and Crovella, 2012))

`mada`: first-order local dimension estimation (Farahmand et al., 2007)

`side`: second-order local dimension estimation (Hino et al., 2017)

There are several other distance-based methods, such as one based on a maximum-likelihood estimate of the Poisson distribution (Levina and Bickel, 2005), which approximates the distance distribution from an inspection point to other points in a given dataset. This method is implemented in our package as a function `lbmle`. A similar but different approach utilizing the nearest-neighbor information has also been implemented as a function `nni` (Pettis et al., 1979).

The proposed package also provides a data-generating function `gendata` that generates several famous artificial datasets often used as benchmarks for IDE and manifold learning.

Fractal dimensions

In fractal analysis, the Euclidean concept of a dimension is replaced with the notion of a fractal dimension, which characterizes how the given shape or datasets occupy their ambient space. There are many different definitions of the fractal dimension, from both mathematical and physical perspectives. Well-known fractal dimensions include the correlation dimension and the capacity dimension. There are already some R packages for estimating the fractal dimension, such as **fractal**, **nonlinearTseries**, and **tseriesChaos**. In **fractal** and **nonlinearTseries**, the correlation dimension and its generalization estimators are implemented, and in **tseriesChaos**, the method of false nearest neighbors (Kennel et al., 1992) is implemented. These packages focus on estimates of the embedded dimension of a time series in order to characterize its chaotic property. To complement the above-mentioned packages, we implemented several fractal dimension estimators for vector-valued observations.

Global dimensions

Correlation dimension

For a set of observed data $\mathcal{D} = \{x_i\}_{i=1}^n$, the correlation integral is defined as

$$V_2(\varepsilon) = \lim_{n\to\infty} \frac{2}{n(n-1)} \sum_{i<j}^{n} I(\|x_i - x_j\| < \varepsilon) \tag{1}$$

using a sufficiently small $\varepsilon > 0$. In Eq. (1), $I(u)$ is the indicator function which returns one when the statement u is true and zero if the statement is false. The correlation integral $V_2(\varepsilon)$ is the ratio of pairs whose distance is below ε, and this number grows as a length for a one-dimensional object, as a surface for a two-dimensional object, as a volume for a three-dimensional object, and so forth. So, it is natural to assume that $V_2(\varepsilon)$ grows proportional to the intrinsic dimension, and the intrinsic dimension associated with $V_2(\varepsilon)$ is defined as the correlation dimension. To be precise, using the correlation integral, the correlation dimension is defined as

$$p_{cor} = \lim_{\varepsilon\to 0} \frac{\log V_2(\varepsilon)}{\log \varepsilon}. \tag{2}$$

Intuitively, the number of sample pairs with a distance smaller than ε should increase in proportion to ε^p, where p is the intrinsic dimension. The correlation dimension exploits this property, i.e., $V_2(\varepsilon) \propto \varepsilon^p$, to define the intrinsic dimension p_{cor}. Grassberger and Procaccia (1983) proposed the use of the empirical (finite sample) estimates $\hat{V}_2(\varepsilon_k) = \frac{2}{n(n-1)} \sum_{i<j}^n I(\|x_i - x_j\| < \varepsilon_k)$, $k = 1, 2$ of the correlation integral $V_2(\varepsilon)$ with two different radii, ε_1 and ε_2, in order to estimate the correlation dimension (2) as follows:

$$\hat{p}_{cor}(\varepsilon_1, \varepsilon_2) = \frac{\log \hat{V}_2(\varepsilon_2) - \log \hat{V}_2(\varepsilon_1)}{\log \varepsilon_2 - \log \varepsilon_1}. \tag{3}$$

Hein and Audibert (2005) proposed the use of a U-statistic with the form

$$\hat{V}_{2,h} = \frac{2}{n(n-1)} \sum_{i<j}^{n} \kappa_h(\|x_i - x_j\|^2) \tag{4}$$

using a kernel function κ_h with bandwidth h to count the number of samples, and replaced the correlation integral by $\hat{V}_{2,h}$. The convergence of this U-statistic with $n \to \infty$, by an argument similar to kernel bandwidth selection (Wand and Jones, 1994), requires that $h \to 0$ and $nh^p \to \infty$. These conditions are used in (Hein and Audibert, 2005) to derive a formula for estimating the global intrinsic dimension p.

In the **ider** package, the classical correlation dimension estimator proposed in Grassberger and Procaccia (1983) is performed using the function `corint` as follows.

```
> set.seed(123)
>  x <- gendata(DataName='SwissRoll',n=300)
> estcorint <- corint(x=x,k1=5,k2=10)
> print(estcorint)
> [1] 1.963088
```

where $k1$ and $k2$ respectively correspond to ε_1 and ε_2 in Eq. (3). Indeed, it is easy and safe to specify an integer k for the k-th nearest neighbor rather than the radius ε, because there is no guarantee that there is a data point in ε-ball in general. In the above example, we used the function `gendata` to generate the famous 'SwissRoll' data with an ambient dimension of three and an intrinsic dimension of two. As observed, the correlation integral method by Grassberger and Procaccia (1983) works well for this dataset. The kernel-based correlation dimension estimator is performed by using the function convU as follows:

```
> set.seed(123)
> x <- gendata(DataName='SwissRoll',n=300)
> estconvU <- convU(x=x,maxDim=5)
> print(estconvU)
> [1] 2
```

The method proposed by Hein and Audibert (2005) attempts to find the possible intrinsic dimension one-by-one up to maxDim. Consequently, the estimated dimension can only be a natural number. All IDE functions in **ider** support both vector-valued data matrices and distance matrices as the input data. This is useful in cases where we exclusively obtain a distance matrix, and in cases where the original data object cannot be represented by a finite and fixed dimensional vector. This is also useful when we treat very high-dimensional data, such that retaining its distance matrix saves memory storage. To indicate that the input is a distance matrix, we set the parameter DM to TRUE as follows:

```
> set.seed(123)
> x <- gendata(DataName='SwissRoll',n=300)
> estcorint <- corint(x=dist(x),DM=TRUE,k1=5,k2=10)
> print(estcorint)
> [1] 1.963088
```

The distance matrix can be either a matrix object or dist object.

Capacity dimension

Let $\mathcal{X}$ be a given metric space with distance metric $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$. The ε-covering number $N(\varepsilon)$ of a set $\mathcal{S} \subset \mathcal{X}$ is the minimum number of open balls $b(z;\varepsilon) = \{x \in \mathcal{X} | d(x,z) < \varepsilon\}$ whose union is a covering of $\mathcal{S}$. The capacity dimension (Hentschel and Procaccia, 1983) or box-counting dimension is defined by

$$p_{cap} = -\lim_{\varepsilon \to 0} \frac{\log N(\varepsilon)}{\log \varepsilon}. \tag{5}$$

The intuition behind the definition of capacity dimension is the following. Assuming a three-dimensional space divided in small cubic boxes with a fixed edge length ε, the box-counting dimension is related to the proportion of occupied boxes. For a growing one-dimensional object placed in this compartmentalized space, the number of occupied boxes grows proportionally to the object length. Similarly, for a growing two-dimensional object, the number of occupied boxes grows proportionally to the object surface, and for a growing three-dimensional object, the number grows proportionally to the volume. Considering the situation that the size of the object remains unchanged but the edge length ε of the boxes decreases justify the definition of p_{cap}.

The problem of estimating p_{cap} is reduced to the problem of estimating $N(\varepsilon)$, but finding the covering number is computationally intractable. Kégl (2002) proposed the replacement of the covering number $N(\varepsilon)$ with the ε-packing number $M(\varepsilon)$. Given a metric space $\mathcal{X}$ with distance d, a set $V \subset \mathcal{X}$ is said to be ε-separated if $d(x,y) \geq \varepsilon$ for all $x, y \in V, x \neq y$. The ε-packing number $M(\varepsilon)$ is defined by the maximum cardinality of an ε-separated subset of the data space $\mathcal{X}$, and it is known that the inequalities $N(\varepsilon) \leq M(\varepsilon) \leq N(\varepsilon/2)$ hold. Considering the fact that the capacity dimension is defined as the limit $\varepsilon \to 0$, the following holds:

$$p_{cap} = -\lim_{\varepsilon \to 0} \frac{\log M(\varepsilon)}{\log \varepsilon}. \tag{6}$$

The capacity dimension based on the packing number is estimated using the estimates of the packing number at two different radii, ε_1 and ε_2, as

$$\hat{p}_{cap} = -\frac{\log \hat{M}(\varepsilon_2) - \log \hat{M}(\varepsilon_1)}{\log \varepsilon_2 - \log \varepsilon_1}. \tag{7}$$

The ε-packing number $M(\varepsilon)$ has been estimated using a greedy algorithm (Kégl, 2002) and by using a hierarchical clustering algorithm (Eriksson and Crovella, 2012).

In the **ider** package, the capacity dimension estimation is based on the packing number with greedy approximation, and it is performed using the function pack as

```
> set.seed(123)
> x <- gendata(DataName='SwissRoll',n=300)
> estpackG <- pack(x=x,greedy=TRUE)  ## estimate the packing number by greedy method
> print(estpackG)
> [1] 2.289935
```

whereas the hierarchical clustering-based method is performed as

```
> estpackC <- pack(x=x,greedy=FALSE) ## estimate the packing number by cluttering
> print(estpackC)
> [1] 2.393657
```

Packing-number-based methods require two radii ε_1 and ε_2, which are specified by arguments k_1 and k_2, respectively. If one of these arguments is NULL, both can be determined by a 0.25 and 0.75 quantile of distance from all pairs of data points.

Local dimensions

The former two fractal dimensions, viz., the correlation dimension and capacity dimension, are designed to estimate a global intrinsic dimension. Any global IDE method can be converted into a local method by running the global method on a neighborhood of a point. However, we introduce two inherently local fractal dimension estimators. The relationship between global and local fractal dimensions are shown in (Hino et al., 2017).

Let μ be an absolutely continuous probability measure on a metric space $\mathcal{X}$, and let the corresponding probability density function (pdf) be $f(x)$. Consider the problem of estimating the value of the pdf at a point $z \in \mathcal{X} \subseteq \mathbb{R}^p$ using a set of observations $\mathcal{D} = \{x_i\}_{i=1}^n$.

First-order method

Let the p-dimensional hyper-ball of radius ε centered at z be $b(z;\varepsilon) = \{x \in \mathbb{R}^p | d(z,x) < \varepsilon\}$. The probability mass of the ball $b(z;\varepsilon)$ is defined as

$$\Pr(X \in b(z;\varepsilon)) = \int_{x \in b(z;\varepsilon)} \mathrm{d}\mu(x) = \int_{x \in b(z;\varepsilon)} f(x)\mathrm{d}\nu(x),$$

where ν is the uniform measure in p-dimensional Euclidean space. We assume that for a sufficiently small radius $\varepsilon > 0$, the value of the pdf $f(z)$ is approximately a constant within the ball $b(z;\varepsilon)$. Under this assumption, using the Taylor series expansion of the probability mass, we obtain

$$\begin{aligned}\Pr(X \in b(z;\varepsilon)) &= \int_{x \in b(z;\varepsilon)} \left\{ f(z) + (x-z)^\top \nabla f(z) + O(\varepsilon^2) \right\} \mathrm{d}\nu(x) \\ &= |b(z;\varepsilon)| \left(f(z) + O(\varepsilon^2) \right) = c_p \varepsilon^p f(z) + O(\varepsilon^{p+2}),\end{aligned}$$

where $\int_{x \in b(z;\varepsilon)} \mathrm{d}\nu(x) = |b(z;\varepsilon)|$. The volume of a ball with a uniform measure is $|b(z;\varepsilon)| = c_p \varepsilon^p$, where $c_p = \pi^{p/2}/\Gamma(p/2+1)$, and $\Gamma(\,\cdot\,)$ is the gamma function. In this expansion, the integration is performed within the ε-ball; hence $x - z$ is of the same order as ε. The term with the first derivative of the density function vanishes owing to symmetry. When we fix the number of samples k falling within the ball $b(z;\varepsilon)$ instead of the radius ε, the radius ε is determined by the distance between the inspection point z and its k-th nearest neighbor. In this paper, ε_k denotes the radius determined by k. Inversely, when we fix the radius ε, the number of samples falling within the ε-ball centered at an inspection point is determined and denoted by k_ε. In (Farahmand et al., 2007), $\Pr(X \in b(z;\varepsilon))$ is approximated by the ratio of $b(z;\varepsilon)$ and the sample size n as follows:

$$\Pr(X \in b(z;\varepsilon)) \simeq \frac{k_\varepsilon}{n} \simeq c_p \varepsilon^p f(z). \tag{8}$$

Then, for different radii $\varepsilon_1, \varepsilon_2$, the logarithm of the above approximation formula derives the following:

$$\begin{aligned}\log \frac{k_{\varepsilon_1}}{n} &= \log c_p f(z) + p \log \varepsilon_1, \\ \log \frac{k_{\varepsilon_2}}{n} &= \log c_p f(z) + p \log \varepsilon_2.\end{aligned}$$

Solving this system of equations with respect to the dimension yields the estimate of the local fractal dimension:

$$\hat{p}_{mada} = \frac{\log k_{\varepsilon_2} - \log k_{\varepsilon_1}}{\log \varepsilon_2 - \log \varepsilon_1}. \tag{9}$$

The convergence rate of this estimator is independent of the ambient dimension, but it depends on the intrinsic dimension. Hence, $\hat{p}_{mada}$ is called the manifold adaptive dimension estimator in (Farahmand et al., 2007). This estimator is simple and easy to implement, and it provides a finite sample error bound. We explain the usage of this first-order local IDE method in package **ider**. To demonstrate the ability of a local estimate, we use a dataset "lbdl" (i.e., line-disc-ball-line), which comprises sub-datasets with one, two, and three dimensions embedded in three-dimensional space. Using this dataset, the example of the use of a first-order local IDE called mada is shown below:

```
> set.seed(123)
> tmp <- gendata(DataName='ldbl',n=300)
> x <- tmp$x
> estmada <- mada(x=x,local=TRUE)
> estmada[c(which(tmp$tDim==1)[1],which(tmp$tDim==2)[1],which(tmp$tDim==3)[1])]
> 1.113473 2.545525 2.207250
```

This sample code estimates the local intrinsic dimensions of every point in the dataset x, and shows the estimates at the points with true intrinsic dimensions of one, two, and three.

Second-order method

In (Hino et al., 2017), accurate local IDE methods based on a higher-order expansion of the probability mass function and Poisson regression modeling are proposed. By using the second-order Taylor series expansion for $\Pr(X \in b(z;\varepsilon))$, we obtain the following proposition:

Proposition 1 *The probability mass* $\Pr(X \in b(z;\varepsilon))$ *of the* ε*-ball centered at* z *is expressed in the form*

$$\Pr(X \in b(z;\varepsilon)) = c_p f(z)\varepsilon^p + \frac{p}{4(p/2+1)} c_p \mathrm{tr}\nabla^2 f(z)\varepsilon^{p+2} + O(\varepsilon^{p+4}).$$

The proof for this is detailed in (Hino et al., 2017). However, there is no need to know the exact form of the second-order expansion. By approximating the probability mass $\Pr(X \in b(z;\varepsilon))$ empirically with the ratio k_ε/n, i.e., the ratio of the number of samples falling into the ε-ball to the whole sample size, we obtain the following relationship:

$$\frac{k_\varepsilon}{n} = c_p f(z)\varepsilon^p + \frac{p}{4(p/2+1)} c_p \mathrm{tr}\nabla^2 f(z)\varepsilon^{p+2}$$

by ignoring the higher-order term with respect to ε. Furthermore, by multiplying both sides of each equation by n, and letting the coefficients of ε^p and ε^{p+2} be β_1 and β_2, respectively, we obtain

$$k_\varepsilon = \beta_1\varepsilon^p + \beta_2\varepsilon^{p+2}. \tag{10}$$

To estimate the intrinsic dimension using the second-order Taylor series expansion, we fit a generalized linear model (GLM; (Dobson, 2002)) to Eq. (10), which expresses the counting nature of the left-hand side of the equation.

Let the intrinsic dimension at the inspection point z be p $(p = 1, \ldots, \mathrm{maxDim})$, where maxDim is the pre-determined upper limit of the intrinsic dimension. We express realizations of a vector-valued random variable $x_{\varepsilon,p} \in \mathbb{R}^2$, which is composed of ε^p and ε^{p+2}, where ε is the distance from the inspection point, by

$$x_{\varepsilon,p} \in \left\{ \begin{pmatrix} \varepsilon_1^p \\ \varepsilon_1^{p+2} \end{pmatrix}, \begin{pmatrix} \varepsilon_2^p \\ \varepsilon_2^{p+2} \end{pmatrix}, \ldots \right\}. \tag{11}$$

We also introduce realizations of a random variable $y_\varepsilon = k_\varepsilon \in \{1, 2, \ldots\}$, which is the number of samples included in the ball $b(z;\varepsilon)$. Specifically, we consider a pair of random variables (Y, X_p) and fix a radius ε corresponding to a *trial* that results in realizations $(y_\varepsilon, x_{\varepsilon,p})$. Because the realization y_ε is the number of samples within the ε-ball, and assuming that the number of observation n is sufficiently large, we assume that the error structure of Y is a Poisson distribution. Then, we can formulate the relationship between the distance from the inspection point ε and the number of samples falling within the ε-ball using a generalized linear model with a Poisson error structure and linear link function as follows:

$$E[y] = x^\top \beta. \tag{12}$$

A set of m different radii is denoted as $\mathcal{E}$, i.e., $\{\varepsilon_1, \ldots, \varepsilon_m\} \in \mathcal{E}$. We maximize the log-likelihood of the Poisson distribution with the observation $\{(y_\varepsilon, x_{\varepsilon,p})\}_{\varepsilon\in\mathcal{E}}$ with respect to the coefficient vector $\beta \in \mathbb{R}^2$. In this work, we simply consider the m-th nearest neighbor with $m = \min\{\lceil n/5 \rceil, 100\}$, and let the Euclidean distance from the inspection point z to its m-th nearest point $x_{(m)}$ be $d(z, x_{(m)})$. Then, we uniformly sample m radii $\mathcal{E} = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m\}$ from a uniform distribution in $[0, d(z, x_{(m)})]$. Let the observation vector and design matrix, which are composed of realizations of Y and X, be

$$\begin{aligned} \boldsymbol{y} &= (y_{\varepsilon_1}, y_{\varepsilon_2}, \ldots, y_{\varepsilon_m})^\top \in \mathbb{R}^m && (13)\\ \boldsymbol{X}_p &= (x_{\varepsilon_1,p}, x_{\varepsilon_2,p}, \ldots, x_{\varepsilon_m,p})^\top \\ &= \begin{pmatrix} \varepsilon_1^p & \varepsilon_2^p & \ldots & \varepsilon_m^p \\ \varepsilon_1^{p+2} & \varepsilon_2^{p+2} & \ldots & \varepsilon_m^{p+2} \end{pmatrix}^\top \in \mathbb{R}^{m\times 2}. && (14) \end{aligned}$$

We consider a generalized linear model (Dobson, 2002) with the linear predictor $\boldsymbol{X}_p\beta$ and identity link

$$E[\boldsymbol{y}] = \boldsymbol{X}_p\beta, \tag{15}$$

and the log-likelihood function of the Poisson distribution:

$$L(\{\boldsymbol{y}, \boldsymbol{X}_p\}; \beta) = \log \prod_{\varepsilon\in\mathcal{E}} \frac{e^{-x_{\varepsilon,p}^\top \beta} (x_{\varepsilon,p}^\top \beta)^{y_\varepsilon}}{y_\varepsilon!}. \tag{16}$$

By assuming that the intrinsic dimension is p, the optimal IDE is estimated on the basis of the goodness of fit of the data to the regression model (10). We use the log-likelihood (16) to measure this goodness of fit. Note that the number of parameters is always two, even when we change the assumed IDE p; hence, the problem of over-fitting by maximizing the likelihood is avoided in our setting.

In the package **ider**, two IDE algorithms are implemented based on the maximization of Eq. (16). The first method simply assumes distinct intrinsic dimensions and maximizes the log-likelihood (16) with respect to β with fixed p. Let the ambient dimension or maximum possible dimension of the observation be maxDim. We assume that the intrinsic dimension is $p = 1, 2, \ldots, \text{maxDim}$, and for every p, we fit the regression model (10) by maximizing the log-likelihood (16) with respect to β, and employ the dimension p that maximizes the likelihood:

$$\hat{p}_{s1} = \mathop{\arg\max}_{p\in\{1,\ldots,\text{maxDim}\}} \max_{\beta\in\mathbb{R}^2} L(\{\boldsymbol{y}, \boldsymbol{X}_p\}; \beta). \tag{17}$$

The second method treats the log-likelihood (16) as a function of both the regression coefficients $\beta \in \mathbb{R}^2$ and the intrinsic dimension $p \in \mathbb{R}_+$. Given a set of observations, we can maximize the log-likelihood function with respect to (p, β_1, β_2). Because it is difficult to obtain a closed-form solution for the maximizer of the likelihood (16), we numerically maximize the likelihood to obtain the estimate as

$$\hat{p}_{s2} = \mathop{\arg\max}_{p\in\mathbb{R}_+} \max_{\beta\in\mathbb{R}^2} L(\{\boldsymbol{y}, \boldsymbol{X}_p\}; \beta) \tag{18}$$

by using the quasi-Newton (BFGS) method. The initial point for the variables (p, β_1, β_2) is set to the estimate obtained using the first method explained above.

In the **ider** package, a second-order local IDE with discrete dimensions is performed using the function `side` (Second-order Intrinsic Dimension Estimator) as follows:

```
> set.seed(123)
> tmp <- gendata(DataName='ldbl', n=300)
> x <- tmp$x
> idx <- c(sample(which(tmp$tDim==1)[1:10],3), sample(which(tmp$tDim==2)[1:30],3))
> estside <- side(x=x[1:100,], local=TRUE,method='disc')
> print(estside[idx]) ## estimated discrete local intrinsic dimensions by side
[1] 1 1 1 3 1 2
```

An example of the same, using the method 'cont' is as follows:

```
> estside <- side(x=x[1:100,], local=TRUE,method='cont')
> print(estside[idx]) ## estimated continuous local intrinsic dimensions by side
[1] 1.020254 1.338089 1.000000 2.126269 3.360426 2.074643
```

It is seen that the obtained estimates are not natural numbers.

The local dimension estimate is easily aggregated to a global estimate by taking an average, median, or voting of local estimates, and this is realized when we set the argument `local = TRUE` in mada or side. The functions mada and side have an argument comb to specify how the local estimates are combined. When comb='average', the local estimates are averaged as a global IDE. Likewise, when comb='median', the median of the local estimates is adopted, and when comb='vote', the voting of local estimates is adopted as a global IDE. Note that the combination method vote should be used only with method='disc'.

```
> set.seed(123)
> x <- gendata(DataName='SwissRoll',n=300)
> estmada <- mada(x=x, local=FALSE, comb='median')
> estmada
[1] 1.754866
> estside <- side(x=x, local=FALSE, comb='median', method='disc')
> estside
[1] 2
```

Other distance-based approaches

The package **ider** supports two other distance-based dimension-estimation methods, namely lbmle and nni.

Maximum likelihood estimation

Levina and Bickel (2005) derived the maximum likelihood estimator of the dimension p from i.i.d. observations $\mathcal{D} = \{x_i\}_{i=1}^n$. Let f be a pdf of the data points smoothly embedded in a p-dimensional space, i.e., a space with intrinsic dimension, and assume that when a point x is fixed, the value of pdf $f(x)$ is constant in a small ball $b(x;\varepsilon)$. Consider the point process

$$\{N(t,x),\ 0 \le t \le \varepsilon\}, \quad N(t,x) = \sum_{i=1}^{n} \mathbf{1}\{x_i \in b(x;t)\}, \tag{19}$$

which counts the observations within distance t from the inspection point x. This point process is approximated using a homogeneous Poisson process, with rate

$$\lambda(t) = c_p f(x) p t^{p-1}. \tag{20}$$

The log-likelihood of the observed process $N(t)$ is written as

$$L(p, \log f(x)) = \int_0^{\varepsilon} \log \lambda(t) \mathrm{d}N(t) - \int_0^{\varepsilon} \lambda(t) \mathrm{d}t. \tag{21}$$

Solving the likelihood equation, the maximum likelihood estimate of the intrinsic dimension around x is

$$\hat{p}_\varepsilon(x) = \left\{ \frac{1}{N(\varepsilon,x)} \sum_{j=1}^{N(\varepsilon,x)} \log \frac{\varepsilon}{\varepsilon_j(x)} \right\}^{-1}, \tag{22}$$

or, more conveniently in practice,

$$\hat{p}_k(x) = \left\{ \frac{1}{k-1} \sum_{j=1}^{k-1} \log \frac{\varepsilon_k(x)}{\varepsilon_j(x)} \right\}^{-1}, \tag{23}$$

where $\varepsilon_j(x)$ denotes the distance between the inspection point x to its j-th nearest point. Then, choosing two indices, k_1 and k_2, the maximum likelihood estimate $\hat{p}_{ml}$ of the intrinsic dimension is obtained as follows:

$$\hat{p}_{ml} = \frac{1}{k_2 - k_1 + 1} \sum_{k=k_1}^{k_2} \hat{q}_k, \qquad \hat{q}_k = \frac{1}{n} \sum_{i=1}^{n} \hat{p}_k(x_i). \tag{24}$$

In the **ider** package, the maximum likelihood estimation is obtained by using the function lbmle:

```
> set.seed(123)
> x <- gendata(DataName='SwissRoll',n=300)
> estmle <- lbmle(x=x, k1=3,k2=5, BC=FALSE)
> print(estmle)
[1] 3.174426
```

It was pointed out by MacKay and Ghahramani that the above MLE contains a certain bias [1]. With the function `lbmle`, however, we can calculate the bias-corrected estimate by setting the argument `BC`, which stands for "bias-correction", to `TRUE`:

```
> set.seed(123)
> x <- gendata(DataName='SwissRoll',n=300)
> estmle <- lbmle(x=x, k1=3,k2=5,BC=TRUE)
> print(estmle)
[1] 2.032756
```

Near-neighbor information

Pettis et al. (1979) proposed an IDE method based on the analysis of the distribution of distances from one point to its nearest neighbors. Pettis et al. (1979) derived that the distribution of the distance from a point x to its k-th nearest neighbor $\epsilon_{k,x}$ is, based on the Poisson approximation, given by the following probability density function

$$f_{k,x}(\epsilon_{k,x}) = nf(x)c_p \frac{\{nf(x)c_p\}^{k-1}}{\Gamma(k)} \exp(-nf(x)c_p\epsilon_{k,x}^p). \tag{25}$$

The expected value of the sample average of distance to the k-th nearest neighbor over the given dataset is

$$\mathbb{E}[\bar{\epsilon}_k] = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[\epsilon_{k,x_i}] = \frac{1}{G_{k,p}} k^{1/p} A_n, \tag{26}$$

where

$$G_{k,p} = \frac{k^{1/p}\Gamma(k)}{\Gamma(k+1/p)}, \quad A_n = \frac{1}{n}\sum_{i=1}^{n}\{nf(x_i)c_p\}^{-1/p}. \tag{27}$$

Note that A_n is sample-dependent but independent of k. Let $\hat{p}_0$ be the first rough estimate of the intrinsic dimension. Taking logarithm of eq. (26) yields

$$\log G_{k,\hat{p}_0} + \log \bar{\epsilon}_k = \frac{1}{p}\log k + \log A_n, \tag{28}$$

where $\mathbb{E}[\epsilon_k]$ is replaced with the sample average $\bar{\epsilon}_k$. From k_1 to k_2, we calculate the left hand side of eq. (28) for each k, and treat them as the realizations of the *response variable*. Linear regression of $\log k,\ k \in [k_1, k_2]$ on those response variable yields the updated estimate $\hat{p}_1$ of the intrinsic dimension. Replacing $\hat{p}_0$ in $G_{k,p}$ with the updated $\hat{p}_1$ and repeat the procedure until the gap between the new and the old estimates $\hat{p}$ is smaller than certain threshold.

The estimator is implemented as a function `nni` in **ider** and used as follows:

```
> set.seed(123)
> x <- gendata(DataName='SwissRoll',n=300)
> estnni <- nni(x=x)
> print(estnni)
[1] 2.147266
```

The function `nni` has parameters $k1$ and $k2$, which are the same in `lbmle`. This method is based on an iterative estimate of IDE, and the function `nni` has a parameter `eps` to specify the threshold for stopping the iteration, which is set at 0.01 by default.

Data-generating function

The **ider** package is equipped with a data-generating function `gendata`. It can generate nine different artificial datasets, which are manifolds of dimension p embedded in ambient space of dimension $(\geq p)$. The dataset is specified by setting the argument `DataName` to one of the following:

`SwissRoll` SwissRoll data, a 2D manifold in 3D space.

`NDSwissRoll` Non-deformable SwissRoll data, a 2D manifold in 3D space.

`Moebius` Moebius strip, a 2D manifold in 3D space.

`SphericalShell` Spherical Shell, p-dimensional manifold in $(p+1)$-dimensional space.

[1] http://www.inference.phy.cam.ac.uk/mackay/dimension/

`Sinusoidal` Sinusoidal data, a 1D manifold in 3D space.

`Spiral` Spiral-shaped data, a 1D manifold in 2D space.

`Cylinder` Cylinder-shaped data, a 2D manifold in 3D space.

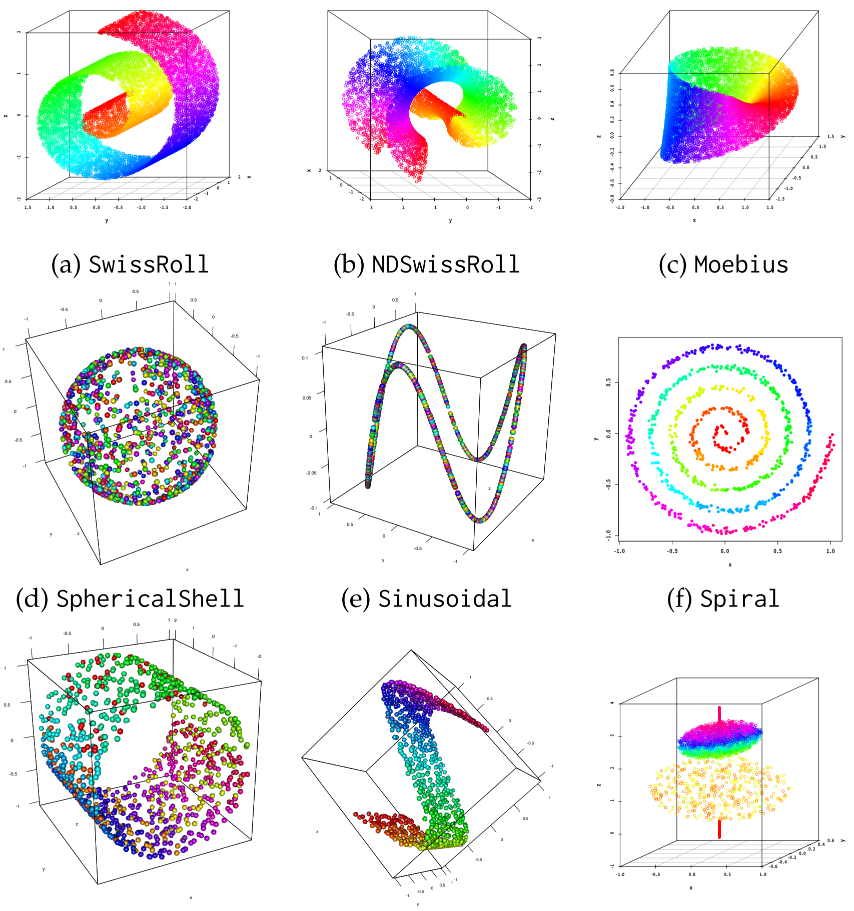
`SShape` S-shaped data, a 2D manifold in 3D space.

`ldbl` Four subspaces, line - disc - filled ball - line, in this order, along the *z*-axis, embedded in 3D space.

The final dataset `lbdl` is used to see the ability of local dimension estimations. The dataset comprises four sub-manifolds: line-shape (1D), disc (2D), filled ball (3D), and line-shape again, and these four sub-manifolds are concatenated in this order.

The parameter `n` of the function `gendata` specifies the number of samples in a dataset. All but the `SphericalShell` dataset have fixed ambient and intrinsic dimensions. For the `SphericalShell` dataset, an arbitrary integer can be set as the ambient dimension by setting the argument `p`.

The realizations of each dataset are shown in Fig. 1.



(a) `SwissRoll` (b) `NDSwissRoll` (c) `Moebius`

(d) `SphericalShell` (e) `Sinusoidal` (f) `Spiral`

(g) `Cylinder` (h) `SShape` (i) `ldbl`

Figure 1: Samples of datasets generated by `gendata`.

Example: estimating the degree of freedom of hand motion

The aim of this paper is to introduce the package **ider** and explain the standard usage of its implemented functions. Exhaustive experiments comparing IDE methods in various settings can be found in (Hino et al., 2017). In this paper, as a simple example of the application of the IDE methods to realistic problems, we consider estimating the intrinsic dimension of a set of images. We used the CMU Hand Rotation dataset[2], which was also used in (Kégl, 2002) and (Levina and Bickel, 2005). Examples of the hand images are shown in Fig. 2. The original CMU Hand Rotation dataset is composed of 481 images

[2] http://vasc.ri.cmu.edu/idb/html/motion/hand/index.html

Figure 2: Example images in the Hand Rotation dataset.

of 512 × 480 pixels. In the package ider, the distance matrix of these images is included:

```
> data(handD)
> str(handD)
Class 'dist'  atomic [1:115440] 4.96 8.27 8.33 8.31 8.12 ...
  ..- attr(*, "Labels")= chr [1:481] "dimg" "dimg" "dimg" "dimg" ...
  ..- attr(*, "Size")= int 481
  ..- attr(*, "call")= language as.dist.default(m = handD)
  ..- attr(*, "Diag")= logi FALSE
  ..- attr(*, "Upper")= logi FALSE
 > dim(as.matrix(handD))
 [1] 481 481
```

Because the object handD is a distance matrix, when we apply IDE methods to this data, we must set the argument DM to TRUE:

```
>lbmle(x=handD,DM=TRUE,k1=3,k2=5,BC=TRUE,p=NULL)
[1] 4.433915
>corint(x=handD,DM=TRUE,k1=3,k2=10)
[1] 2.529079
> pack(x=handD,DM=TRUE,greedy=TRUE)
[1] 3.314233
> pack(x=handD,DM=TRUE,greedy=FALSE)
[1] 2.122698
> nni(handD,DM=TRUE)
[1] 2.646178
> side(x=handD,DM=TRUE,local=FALSE,method='disc',comb='median')
[1] 3
> side(x=handD,DM=TRUE,local=FALSE,method='cont',comb='median')
[1] 2
```

These results suggest that even the extrinsic dimension of the image is very high ($512 \times 480 = 245760$), whereas the intrinsic dimension is quite low, and there are only a few between 2 to 4.5.

Computational cost

We measured computational costs of dimension estimation methods for SwissRoll dataset, where the number of observations n were varied from 500 to 3000 by 500. Datasets of size n are repeatedly generated 100 times. The experiments are performed on an iMac with Mac OS X, 2.6GHz Intel Core i7 processor, with 16GB memory. The results are shown in Fig. 3 by boxplots. It is seen that the computational costs of packT and side grow almost quadratically. Among various methods, lbmle and nni are computationally very efficient. When we apply packT or side to large scale dataset, it is advised to subsample the original dataset.

Summary and future directions

In this paper, we introduced an R package, ider, that implements several IDE algorithms for vector-valued observation or distance matrices. Different IDE methods capture different aspects of the data distribution in low-dimensional subspace, and the estimated dimension varies. Our goal in developing ider is to provide a systematic method for selecting or comparing different methods for analyzing a given dataset. In practice, it is not expected that there is a ground-truth intrinsic dimension. In such cases, one possible approach is to apply all of the IDE methods provided in ider. In doing so, we can

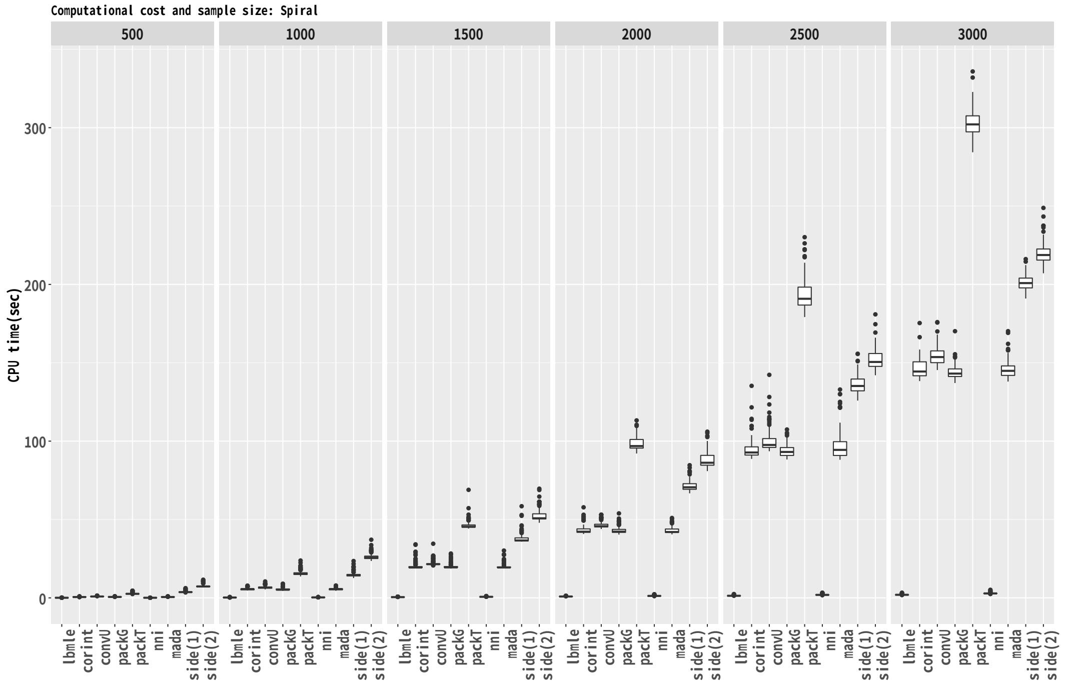


Figure 3: Computational costs and number of observations.

check whether the estimates agree with one another. If a consensus on the estimate cannot be reached, the reason why will be of interest, and this can help to characterize the nature of the data distribution.

It is worth noting that there is another approach to IDE based on the minimum spanning tree. In (Costa et al., 2004), a method for estimating the f-divergence (Ali and Silvey, 1966; Csiszár and Shields, 2004) based on the total edge length of the minimum spanning tree of the observed objects has been proposed, and the authors used this method to estimate the intrinsic dimension (Costa and Hero, 2006; Carter et al., 2010). This kind of IDE method, and some of the projection-based methods, shall be included in a future version of the package **ider**. Furthermore, as we included representative fractal dimension estimation methods, the related methods based on them are not covered in the current version of **ider**. For example, the correlation dimension by Grassberger and Procaccia (1983) is known to have finite sample bias, and bias correction method based on simulated data is proposed in (Camastra and Vinciarelli, 2002). Also, the maximum likelihood and other approaches for correlation dimension estimation are summarized in (Camastra, 2003). We will update the package with implementations of these variations of the methods already included in current **ider**.

Finally, another important future work is improving computational efficiency. IDE methods based on packing number and second-order expansion of probability mass function are not computational efficient. Implementation using **Rcpp** or parallelization would effective means for realizing the fast computation, and we will now working on parallel implementation.

Acknowledge

The authors would like to express their special thanks to the editor, the associate editor and anonymous reviewers whose comments led to valuable improvements of the manuscript. Part of this research was supported by JSPS KAKENHI Grant Numbers 16H02842, 25120011, 16K16108, and 17H01793.

Bibliography

S. M. Ali and S. D. Silvey. A General Class of Coefficients of Divergence of One Distribution from Another. *Journal of the Royal Statistical Society B*, 28(1):131–142, 1966. URL `http://www.jstor.org/stable/2984279`. [p11]

J. Bruske and G. Sommer. Intrinsic dimensionality estimation with optimally topology preserving maps. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(5):572–575, 1998. ISSN 0162-8828. URL `https://doi.org/10.1109/34.682189`. [p1]

F. Camastra. Data dimensionality estimation methods: a survey. *Pattern Recognition*, 36(12):2945–2954, 2003. URL https://doi.org/10.1016/s0031-3203(03)00176-6. [p11]

F. Camastra and A. Vinciarelli. Estimating the intrinsic dimension of data with a fractal-based method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(10):1404–1407, 2002. ISSN 0162-8828. URL https://doi.org/10.1109/tpami.2002.1039212. [p11]

K. M. Carter, R. Raich, and A. O. H. III. On local intrinsic dimension estimation and its applications. *IEEE Transactions on Signal Processing*, 58(2):650–663, 2010. ISSN 1053-587X. URL https://doi.org/10.1109/tsp.2009.2031722. [p11]

R. D. Cook and X. Yin. Dimension reduction and visualization in discriminant analysis (with discussion). *Australian and New Zealand Journal of Statistics*, 43(2):147–199, 2001. URL https://doi.org/10.1111/1467-842x.00164. [p1]

J. A. Costa and A. O. Hero. *Determining Intrinsic Dimension and Entropy of High-Dimensional Shape Spaces*, chapter 8, pages 231–252. Birkhäuser Boston, Boston, MA, 2006. ISBN 978-0-8176-4481-9. URL https://doi.org/10.1007/0-8176-4481-4_9. [p11]

J. A. Costa, A. O. Hero, and III. Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Transactions on Signal Processing*, 52:2210–2221, 2004. URL https://doi.org/10.1109/tsp.2004.831130. [p11]

I. Csiszár and P. C. Shields. Information theory and statistics: A tutorial. *Commun. Inf. Theory*, 1(4): 417–528, 2004. ISSN 1567-2190. URL https://doi.org/10.1561/0100000004. [p11]

A. J. Dobson. *An Introduction to Generalized Linear Models*. Chapman & Hall/CRC, Boca Raton :, 2nd ed. edition, 2002. URL https://doi.org/10.4135/9781412983273. [p5, 6]

B. Eriksson and M. Crovella. Estimating intrinsic dimension via clustering. In *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, pages 760–763, 2012. URL https://doi.org/10.1109/ssp.2012.6319815. [p1, 3]

M. Fan, H. Qiao, and B. Zhang. Intrinsic dimension estimation of manifolds by incising balls. *Pattern Recogn.*, 42(5):780–787, 2009. ISSN 0031-3203. URL https://doi.org/10.1016/j.patcog.2008.09.016. [p1]

A. M. Farahmand, C. Szepesvári, and J.-Y. Audibert. Manifold-adaptive dimension estimation. In *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, pages 265–272, 2007. URL https://doi.org/10.1145/1273496.1273530. [p1, 4, 5]

K. Fukunaga and D. R. Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Trans. Comput.*, 20(2):176–183, 1971. ISSN 0018-9340. URL https://doi.org/10.1109/t-c.1971.223208. [p1]

P. Grassberger and I. Procaccia. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9(1–2):189–208, 1983. ISSN 0167-2789. URL https://doi.org/10.1016/0167-2789(83)90298-1. [p1, 2, 3, 11]

M. D. Gupta and T. S. Huang. Regularized maximum likelihood for intrinsic dimension estimation. In *UAI 2010, Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, July 8-11, 2010*, pages 220–227, 2010. [p1]

M. Hein and J.-Y. Audibert. Intrinsic dimensionality estimation of submanifolds in R^d. In *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, pages 289–296, 2005. URL https://doi.org/10.1145/1102351.1102388. [p1, 2, 3]

H. G. E. Hentschel and I. Procaccia. The infinite number of generalized dimensions of fractals and strange attractors. *Physica D: Nonlinear Phenomena*, 8(3):435–444, 1983. ISSN 0167-2789. URL https://doi.org/10.1016/0167-2789(83)90235-x. [p3]

H. Hino, J. Fujiki, S. Akaho, and N. Murata. Local intrinsic dimension estimation by generalized linear modeling. *Neural Computation*, 29(7):1838–1878, 2017. URL https://doi.org/10.1162/neco_a_00969. [p1, 4, 5, 9]

N. Kambhatla and T. K. Leen. Dimension reduction by local principal component analysis. *Neural Computation*, 9(7):1493–1516, 1997. URL https://doi.org/10.1162/neco.1997.9.7.1493. [p1]

M. B. Kennel, R. Brown, and H. D. I. Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Phys. Rev. A*, 45:3403–3411, 1992. URL https://doi.org/10.1103/physreva.45.3403. [p2]

E. Kokiopoulou and Y. Saad. Orthogonal Neighborhood Preserving Projections: A projection-based dimensionality reduction technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2143–2156, 2007. URL https://doi.org/10.1109/tpami.2007.1131. [p1]

B. Kégl. Intrinsic dimension estimation using packing numbers. In *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*, pages 681–688, 2002. [p1, 3, 9]

E. Levina and P. J. Bickel. Maximum likelihood estimation of intrinsic dimension. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 777–784. MIT Press, Cambridge, MA, 2005. [p1, 7, 9]

B. B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman and Company, 1977. URL https://doi.org/10.1119/1.13295. [p1]

K. W. Pettis, T. A. Bailey, A. K. Jain, and R. C. Dubes. An intrinsic dimensionality estimator from near-neighbor information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-1 (1):25–37, 1979. ISSN 0162-8828. URL https://doi.org/10.1109/tpami.1979.4766873. [p1, 8]

P. J. Verveer and R. P. W. Duin. An evaluation of intrinsic dimensionality estimators. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(1):81–86, 1995. ISSN 0162-8828. URL https://doi.org/10.1109/34.368147. [p1]

M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman & Hall/CRC, 1994. ISBN 0412552701. [p2]

Hideitsu Hino
Graduate School of Systems and Information Engineering, University of Tsukuba
1–1–1 Tennoudai, Tsukuba, Ibaraki, 305–8573
Japan
hideitsu.hino@gmail.com