Information Sciences 328 (2016) 26–41

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Intrinsic dimension estimation: Advances and open problems

Francesco Camastra*, Antonino Staiano

Department of Science and Technology, University of Naples Parthenope, Centro Direzionale Isola C4 - 80143 Napoli, Italy

ARTICLE INFO

Article history:
Received 2 January 2014
Revised 1 July 2015
Accepted 8 August 2015
Available online 24 August 2015

Keywords:
Intrinsic dimension
Curse of dimensionality
Maximum likelihood
Correlation dimension
Dimensionality reduction

ABSTRACT

Dimensionality reduction methods are preprocessing techniques used for coping with high dimensionality. They have the aim of projecting the original data set of dimensionality N, without information loss, onto a lower M-dimensional submanifold. Since the value of M is unknown, techniques that allow knowing in advance the value of M, called intrinsic dimension (ID), are quite useful. The aim of the paper is to review state-of-the-art of the methods of ID estimation, underlining the recent advances and the open problems.

1. Introduction

Dimensionality reduction methods are preprocessing techniques used for coping with high dimensionality. Dimensionality reduction methods aim to project the original data set $\Omega \subset \mathbb{R}^N$, without information loss, onto a lower M-dimensional submanifold of $\mathbb{R}^N$. Since the value of M is unknown, techniques that provide the value of M in advance, are quite useful. Following Fukunaga [34], the minimum number of parameters required to account for the observed properties of data is the intrinsic (or effective) dimension of the data set. The estimation of the intrinsic dimension (ID) of a data set is a classical problem of pattern recognition and machine learning. The first algorithm of data dimensionality estimation, by Bennett, dates back to 1969 [4]. ID estimation is relevant in machine learning not only for dimensionality reduction methods but also for several other reasons. First, using more dimensions than necessary leads to several problems, such as an increase in the space required to store data, and a decrease in the algorithm speed, since it generally depends on data dimensionality. Besides, building reliable classifiers becomes harder and harder when the dimensionality grows (*curse of dimensionality*) [3]. To this purpose, we recall that the capacity (*VC-dimension*) [96] of the linear classifiers, that determines their generalization capability, depends on ID. Nearest neighbor searching algorithms can profit from a good ID estimate, since the complexity of search data structures (e.g., kd-trees and R-trees) increases exponentially with ID [15]. Finally, ID is relevant for some prototype-based clustering algorithms. For instance, in a trained Neural Gas, data density P and density of the neural gas weight vectors ρ are related by $\rho \propto P^{\mu}$, where μ, called *magnification factor* [16,98], depends on ID according to $\mu = \frac{ID}{ID+2}$. Although in the literature there are surveys on ID estimation [12,45], they are dated so that they cannot take into account recent advances in the field.

The aim of the paper is to make state-of-the-art of the methods of ID estimation, underlining the advances and the open problems. By extending the taxonomy proposed by Jain and Dubes [45], we group the algorithms for estimating ID in three disjoint categories, i.e., *local*, *global*, and *pointwise*. In the local category, there are algorithms that provide an ID estimation by using information contained in sample neighborhoods. The algorithms, belonging to the global category, make use of the whole

* Corresponding author. Tel.: +39- 3384447991.
E-mail addresses: camastra@ieee.org, francesco.camastra@uniparthenope.it (F. Camastra), antonino.staiano@uniparthenope.it (A. Staiano).

http://dx.doi.org/10.1016/j.ins.2015.08.029
0020-0255/

data set providing a unique and global ID estimate for the data set. Finally, in the pointwise category, there are the algorithms that can produce both a global ID estimate of the whole data set and local ID estimate of particular subsets of the data set. In the paper the most relevant algorithms for each category, underlining their weak points, will be presented. The paper is organized as follows: In Section 2 the Intrinsic Dimension is defined; Section 3 introduces the concept of ideal ID estimator, discussing the properties that it should have; Sections 4, 5, 6 describe global, local and pointwise methods, respectively; In Section 7 the main ID estimation methods are discussed under the ideal ID estimator framework; finally, in Section 8 open problems are analyzed and some conclusion are drawn.

2. Intrinsic dimension

In the Introduction we have informally introduced the concept of the intrinsic dimension saying that it is given by the number of parameters (or degrees of freedom) required to describe all data. A more formal definition of the intrinsic dimension is the following, due to [33]:

Definition 1. A data set $\Omega \subseteq \mathbb{R}^N$ is said to have *intrinsic dimension* (ID) equal to M if its elements lie entirely, without information loss, within a M-dimensional manifold of $\mathbb{R}^N$, where $M < N$.

Since most methods of the ID estimation are based on mathematical concepts, we briefly review the definition of dimension in the mathematical domain. The definition of dimension in mathematics is not univocal. The first mathematical definition of dimension (*Hausdorff dimension*) is due to Hausdorff [38]. The *Hausdorff dimension* m_H of a set Ω is defined by introducing the quantity $\Gamma_H^m(r) = \inf_{s_i} \sum_i (r_i)^m$, where the set Ω is covered by cells s_i with variable diameter r_i, and all diameters satisfy $r_i < r$. In other words, we look for that collection of covering sets s_i with diameters less than or equal to r which minimizes the sum and denote the minimized sum $\Gamma_H^m(r)$. The *m-dimensional Hausdorff measure* is defined as $\Gamma_H^m = \lim_{r\to 0} \Gamma_H^m(r)$. The m-dimensional Hausdorff measure generalizes the usual notion of the total length, area and volume of simple sets. Hausdorff proved that Γ_H^m, for every set Ω, is $+\infty$ if m is less than some critical value m_H and is 0 otherwise. The critical value m_H is called the *Hausdorff dimension* of the set.

A further definition of dimension, strictly related to Hausdorff's one, is the so-called *Information Dimension* [44]:

Definition 2. The information dimension $m_H(P)$, of a probability measure, P, is defined to be the smallest Hausdorff dimension of sets that have measure 1, i.e.,

$$m_H(P) = \inf_B \{m_H(B) : P(B) = 1\}. \tag{1}$$

Besides, it is possible to define a notion of dimension strictly related to a single data point, namely the so-called *pointwise dimension* [102]:

Definition 3. Let $B_r(\vec{x})$ be a closed ball of radius r and centre the data point $\vec{x} \in \mathbb{R}^N$, if P is a probability measure such that the limit

$$q = \lim_{r\to 0} \frac{\ln P(B_r(\vec{x}))}{\ln r} \tag{2}$$

exists, then the limit q is called the *pointwise* (or *local Hausdorff*) dimension.

3. Ideal ID estimator properties

Before describing the different methods for estimating ID, it is necessary to define criteria that allow comparing them each with other. To this purpose, extending Pestov's axiomatic approach [75], we say that an *Ideal* ID estimator should:

1. be *computational feasible*;
2. be *robust to the multiscaling*;
3. be *robust to the high dimensionality*;
4. have a *work envelope* (or *operative range*);
5. be accurate, i.e., give an ID estimate *close to the underlying manifold dimensionality* (*accuracy*).

The first requirement is motivated by applicative reasons. An estimator that is hard to implement or requires huge computational resources that cannot be used in real world problems. The robustness of an ID estimator w.r.t. the multiscaling is motivated by ID dependence on the data scale [51,72,100]. In order to show this, we consider a two-dimensional data set (e.g., a K-Möbius strip [40]), and we add a three-dimensional gaussian noise to it. The data set, obtained this way, has an ID equal to two at a coarse scale, since the two-dimensional set is dominant. But if the same data set is observed at fine scale, the noise becomes dominant and the data set ID is three. ID estimator robustness w.r.t. high dimensionality is desirable since such an estimator should provide a reliable estimate even if the data set ID is very high, as it can happen in bioinformatics and text categorization applications. The fourth requirement is borrowed by Virtual Reality [10] where the work envelope (or *operative range*) indicates the range where a sensor gives reliable measures. In our specific case the work envelope of an ID estimator is the minimum cardinality that a data set should have so that the estimator gets a reliable estimate. In this way, we view implicitly the ID estimator as a

sensor and assuming that most parameters used to characterize the sensor performance can be used to evaluate ID estimators. Finally, it should be desirable that the estimate provided by ID estimator is very close to the effective dimension of the manifold where the data set lies [75]. In the rest of paper, we will review the major ID estimators under this framework, underlining which requirements, in each estimator, are guaranteed.

4. Global methods

Global methods use the whole data set making implicitly the assumption that data lie on a unique manifold of a fixed dimensionality. Global methods can be grouped in five families: *projection*, *multidimensional scaling*, *fractal-based*, *multiscale*, and *other* methods, where all the methods that cannot be assigned to the first four categories are collected.

4.1. Projection methods

Projection methods search for the best subspace to project the data by minimizing the projection error.

Principal Component Analysis (*PCA*) [5,47,52] is the simplest projection method and projects the data along the directions of maximal variance. PCA, used for ID estimation, has the following steps:

1. Compute the N eigenvalues of the covariance matrix. Order them in a decreasing way, such that $\lambda_1 \geq \lambda_2, \cdots \geq \lambda_N$.
2. Normalize the eigenvalues dividing each eigenvalue by the largest one λ_1.
3. Choose a threshold value θ and compute the integer J such that $\lambda_J \geq \theta$ and $\lambda_{J+1} < \theta$. J is the ID estimate.

PCA is a poor ID estimator since, in most cases, it overestimates ID. If we consider a data set formed by data points lying on a curve (see Fig. 1), PCA provides an ID estimate equal to 2 instead of the correct value of 1. Therefore PCA satifies only the first Ideal ID requirement, i.e., it is easy to implement and computational feasible.

4.1.1. Probabilistic and Bayesian PCA

A further drawback in the use of PCA as ID estimator consists in determining an appropriate value for the threshold θ. In order to cope with this problem, Tipping and Bishop [93] formulated PCA, renamed it *Probabilistic PCA* (PPCA), as the maximum likelihood solution of a latent variable method. Assuming that M is the ID of the manifold where data lie, they considered an M-dimensional latent variable $\vec{u}$ with a zero mean prior distribution given by $\mathcal{N}(\vec{u}|\vec{0}, \mathbb{I}_M)$, where the covariance matrix is the M-dimensional identity matrix $\mathbb{I}_M$. The observed data point $\vec{x} \in \mathbb{R}^N$ is related to the latent variable $\vec{u}$ by the equation: $\vec{x} = W\vec{u} + \mu + \epsilon$, where W is a $N \times M$ projection matrix, formed by the principal components, $\mu \in \mathbb{R}^M$ is an appropriate parameter and ϵ is noise having a zero-mean gaussian distribution with covariance $\sigma^2 I_M$, where σ is a proper parameter. The distribution of the observed data set $\Omega = (\vec{x}_1, \ldots, \vec{x}_\ell)$ is a normal distribution defined by the values of the parameters W, μ, σ. Therefore the computation of probabilistic principal components is reduced to the solution of the maximum likelihood estimation problem of W, μ, σ. However PPCA cannot be used for ID estimation since it does not offer reliable strategies for estimating the latent space dimensionality that corresponds to the data set ID [6]. To cope with this drawback, Bishop proposed a PPCA extension, called *Bayesian PCA* (BPCA), where a prior distribution over the parameters W, σ, μ is introduced. Using the Bayes' Theorem [22] it can obtain the posterior probability over the data set Ω. To compute the latent dimensionality, i.e., ID data set, Bishop defines a hierarchical prior $p(W|\vec{\alpha})$ over W that is managed by a vector of hyperparameters $\vec{\alpha} \in \mathbb{R}^q$ where q is set to $N - 1$. The prior $p(W|\vec{\alpha})$, whose theoretical roots are in *Automatic Relevance Determination* (ARD) [65] framework, is defined as a product of q conditional Gaussian distributions. The ith Gaussian only depends on α_i and w_i, where the latter parameter denotes the ith column of the matrix W, namely the ith principal component. Hence, the inverse of each parameter α_i supervises the *relevance* of the respective ith principal component. To evaluate the matrix W, Bishop uses a local Gaussian approximation for estimating the

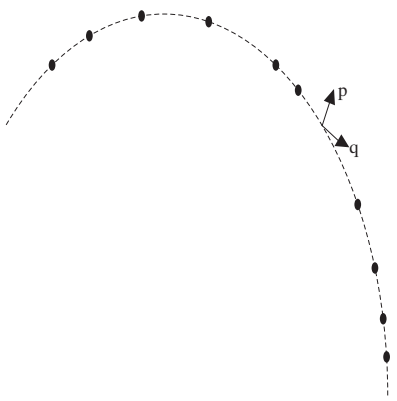

Fig. 1. The data set is formed by points lying on a curve. The data set ID is *1*. Neverthless PCA yields *two* non-null eigenvalues. The principal components are indicated by *p* and *q*.

posterior distribution of the W, that then it has to be marginalized to solve the problem. Whereas the parameters α_i are computed by the maximum likelihood principle. The data set ID is given by the number of principal components $\vec{w}_i$ whose related inverse relevance, $\frac{1}{\alpha_i}$, is not null. A weakness of BPCA consists in assuming the Gaussian distribution of data. This assumption may not be satisfied when data are represented by binary or integer values. To overcome this limitation, Li and Tao [59] proposed a BPCA extension, called *Simple Exponential Family PCA* (SePCA), that replaces the Gaussian distribution with the exponential family distribution. Therefore the exponential family distributions determine the likelihood function of the principal components. This framework allows connecting real-valued latent variables with observed data of any kind. It is worth mentioning that Bouveyron et al [7] recently proposed to apply the asymptotic consistency of the maximum likelihood criterion for determining the ID of a data set in the PPCA approach. In addition to the above-mentioned methods, two other techniques, that allow the automatic selection of the principal components, have to be quoted. The former is the *Sparse Principal Component Analysis* (SPCA)[103] that is based on the reformulation of PCA in terms of a regression optimization problem imposing the *lasso* [92] constraint on the regression parameters. In this way, the algorithm imposes the sparsity of the projection matrix *W* making the selection of the principal components possible. The main drawback of the approach lies in the manual tuning of the trade-off parameter λ that manages the influence of the lasso constraint in the regression optimization problem. The latter technique, called *Sparse Probability Principal Component Analysis* (SPPCA), reformulates SPCA using a probabilistic Bayesian approach [36]. In SPPCA the parameter λ is learnt by the maximum likelihood principle. We conclude this section underlining that the methods described above are linear and they tend to overestimate ID. Therefore, as PCA, they only guarantee the first Ideal ID requirement.

4.1.2. Nonlinear PCA

Nonlinear PCA was proposed to overcome PCA linear limitations. A widely used approach to get a Nonlinear PCA is the autoassociative approach [50]. Nonlinear PCA is performed by means of a five-layers neural network. The neural net has a typical bottleneck structure. The first (*input*) and the last (*output*) layer have the same number of neurons, while the remaining hidden layers have less neurons than the first and the last ones. The second, the third and the fourth layer are denoted as *mapping*, *bottleneck* and *demapping layer*, respectively. Mapping and demapping layers have usually the same number of neurons. The number of neurons of the bottleneck layer provides an ID estimate. The targets used to train Nonlinear PCA are simply the input vector themselves. Though autoassociative neural networks (ANNs) outperform linear PCA, as ID estimators, in some contexts, ANNs present some drawbacks, e.g., ANN projections onto curves and surfaces are not optimal [67].

4.2. Multidimensional scaling methods

Multidimensional Scaling (MDS) [20] methods are projection techniques that tend to preserve the distances among data. Hence data that are close in the original data set should be projected in such a way that their projections, in the new space (*output space*), are still close. To each projection is associated an index, usually defined *stress*, that measures the goodness of the projection. The best projection is the one whose stress is minimal. ID is determined in the following way. The minimum stress for projections of different dimensionalities is computed. Then a plot of the minimum stress versus dimensionality of the output space is performed. ID is the dimensionality value for which there is a knee or a flattening of the curve. Examples of MDS methods are Bennett's algorithm [4], that now has only historical interest, *MDSCAL* [56,79], *Sammon's mapping* [83]. For sake of space, we only describe the last one. Sammon proposed to minimize a stress measure $\mathcal{E}$, defined as follows:

$$\mathcal{E} = \left[\sum_{i<j} \frac{(\delta(\vec{x}_i, \vec{x}_j) - \Delta(\vec{x}_i, \vec{x}_j))^2}{\delta(\vec{x}_i, \vec{x}_j)} \right] \left[\sum_{i<j} \delta(\vec{x}_i, \vec{x}_j) \right]^{-1}, \tag{3}$$

where $\delta(\vec{x}_i, \vec{x}_j)$ is the distance between patterns $\vec{x}_i$ and $\vec{x}_j$ in the original data space and $\Delta(\vec{x}_i, \vec{x}_j)$ is the distance in the two- or three- dimensional output space. The stress is usually minimized by the gradient-descent algorithm.

An alternative approach to perform MDS, the *Curvilinear Component Analysis* (*CCA*) was proposed by Demartines and Herault [21]. CCA is a *self-organizing* neural network that performs two tasks. The former is the vector quantization of the data set, performed by *SOM* [53]. The latter is the nonlinear projection of the quantized vectors onto a lower dimensionality space, carried out by the minimization of a cost function that measures the goodness of the projection. We conclude the section by analyzing MDS methods w.r.t. Ideal ID requirements. The first requirement is guaranteed when the data set dimensionality is not large since the application of MDS techniques for ID estimation can become infeasible when data set dimensionality is high. MDS methods do not guarantee the other Ideal ID requirements even if they do not suffer from the linearity limitation of PCA methods.

4.3. Fractal-based methods

Fractal-based techniques are global methods that were originally proposed in physics to estimate the attractor dimension of nonlinear systems [23,49]. Unlike other global methods, they can provide a non-integer value as ID estimate. Since fractals are generally characterized by a non-integer dimensionality (e.g., Koch's curve dimension [68] is $\frac{\ln 4}{\ln 3}$), these methods are called *fractal*.

Since Hausdorff dimension is very hard to estimate [23,74] , fractal methods usually replace it with an upper bound, the *Box-Counting Dimension* [74].

4.3.1. Kégl's algorithm

Let $\Omega = \{\vec{x}_1, \ldots, \vec{x}_\ell\} \subseteq \mathbb{R}^N$ be a data set, we denote by $\nu(r)$ the number of the boxes (i.e., hypercubes) of size r required to cover Ω. It can be proved [74] that $\nu(r) \propto (\frac{1}{r})^M$, where M is the *dimension* of the set Ω. This motivates the following definition. The Box-Counting Dimension (or *Kolmogorov capacity*) M_B of the set Ω [74] is defined by

$$M_B = \lim_{r \to 0} \frac{\ln(\nu(r))}{\ln(\frac{1}{r})} \tag{4}$$

where the limit is assumed to exist.

Kégl's algorithm [51] is a fast algorithm for estimating the Box-Counting Dimension. Kégl's algorithm is based on the observation that $\nu(r)$ is equivalent to the cardinality of the maximum independent vertex set $MI(G_r)$ of the graph $G_r(V, E)$ with vertex set $V = \Omega$ and edge set $E = \{(\vec{x}_i, \vec{x}_j) \mid d(\vec{x}_i, \vec{x}_j) < r\}$. Kégl proposed to estimate $MI(G)$ using the following greedy approximation. Given a data set Ω, we start with an empty set $\mathcal{C}$. In an iteration over Ω, we add to $\mathcal{C}$ data points that are at distance of at least r from all elements of $\mathcal{C}$. The cardinality of $\mathcal{C}$, after every point in Ω has been visited, is the estimate of $\nu(r)$. The Box-Counting Dimension estimate is given by:

$$M_B = -\frac{\ln \nu(r_2) - \ln \nu(r_1)}{\ln r_2 - \ln r_1} \tag{5}$$

where r_2 and r_1 are values that can be set up heuristically. The complexity of Kégl's algorithm is given by $O(N\ell^2)$, where ℓ and N are the cardinality and the dimensionality of the data set, respectively. We have to mention that a Kégl algorithm extension was proposed by Raginsky and Lazebnik [78].

Finally, we pass to discuss Kégl's algorithm under Ideal ID framework. Kégl's algorithm does not take into account multiscaling. Moreover, the robustness w.r.t. high dimensionality seems not be guaranteed [13].

4.3.2. Grassberger–Procaccia algorithm

A good alternative to the Box-Counting Dimension, among many proposed [74,101] is the *Correlation Dimension* [35], defined as follows. If the *correlation integral* $C(r)$ is given by:

$$C(r) = \lim_{\ell \to \infty} \frac{2}{\ell(\ell-1)} \sum_{i=1}^{\ell} \sum_{j=i+1}^{\ell} I(\|\vec{x}_j - \vec{x}_i\| \leq r) \tag{6}$$

where I is an *indicator function* (i.e., it is 1 if condition holds, 0 otherwise), then the Correlation Dimension M_c of Ω is:

$$M_c = \lim_{r \to 0} \frac{\ln(C(r))}{\ln(r)} \tag{7}$$

It can be proved that the Correlation Dimension is a lower bound of the Box-Counting Dimension. The most popular method to estimate Correlation Dimension is the *Grassberger–Procaccia algorithm* [35]. This method consists in plotting $\ln(C_m(r))$ versus $\ln(r)$. The Correlation Dimension is the slope of the linear part of the curve (see Fig. 2b). The computational complexity of the Grassberger–Procaccia algorithm is $O(\ell^2 s)$ where ℓ is the cardinality of the data set and s is the number of different times that

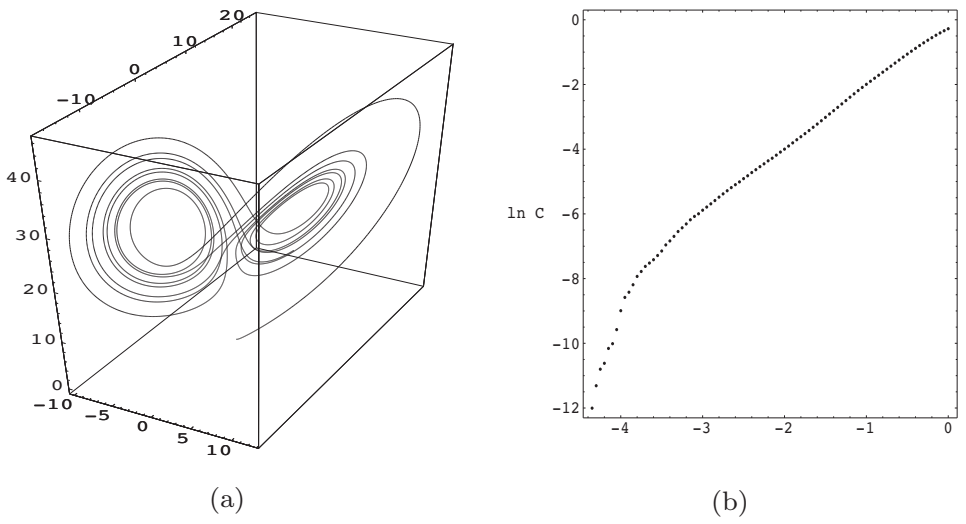

Fig. 2. (a) The attractor of the Lorenz system [64]. (b) The log–log plot on Data Set A. Data Set A is a real data time series generated by a Lorenz-like system, implemented by NH_3-FIR lasers.

the integral correlation is evaluated, respectively. However, there are efficient implementations of the Grassberger–Procaccia algorithm whose complexity does not depend on s. For these implementations, the computational complexity is $O(\ell^2)$.

Different approaches for estimating the Correlation Dimension were proposed [26,27,87]. Fan et al [27] proposed to estimate the Correlation Dimension by means of a polynomial fitting technique. In their approach, the Correlation Dimension is given by the degree of the best data fitting polynomial. This implies than the Fan et al's method always provides an integer value for the Correlation Dimension, unlike other methods that may yield non-integer values.

4.3.3. Takens' method

Takens [87] proposed a method, based on the *Maximum Likelihood* principle [22], that estimates the expectation value of Correlation Dimension. Let $Q = \{q_k \mid q_k < r\}$ be the set formed by the Euclidean distances (denoted by q_k), between data points of Ω, lower than the so-called *cut-off radius* r. Using the Maximum Likelihood principle Takens proved that the expectation value of the Correlation Dimension $\langle M_c \rangle$ is:

$$\langle M_c \rangle = -\left(\frac{1}{|Q|} \sum_{k=1}^{|Q|} q_k \right)^{-1} \tag{8}$$

where $|Q|$ denotes the cardinality of Q. Takens' method presents some drawbacks. Firstly, the cut-off radius can be set only by using some heuristics [90]. Besides, the method is optimal [89] only if the correlation integral $C(r)$ has the form $C(r) = \alpha r^D[1 + \beta r^2 + o(r^2)]$ where $\alpha, \beta \in \mathbb{R}^+$.

4.3.4. Work envelope of fractal-based methods

Differently from the other ID methods described before, fractal-based methods satisfy, in addition to the third one, the fourth Ideal ID requirement, i.e., they have a work envelope. They provide a lower bound that the cardinality of data set must fulfill in order to get an accurate ID estimate. Eckmann and Ruelle [24,85] proved that to get an accurate estimate of the dimension M, the data set cardinality ℓ has to satisfy the following inequality:

$$M < 2 \log_{10} \ell. \tag{9}$$

The inequality (9) shows that the number ℓ of data points required to estimate accurately the dimension of a M-dimensional set is at least $10^{\frac{M}{2}}$. Even for sets of moderate dimension this leads to huge values of ℓ.

To improve the reliability of the ID estimate when the cardinality ℓ does not fulfill the inequality (9), the *method of surrogate data* [91] was proposed. The method of surrogate data is based on *bootstrap* [25]. Given a data set Ω, the method consists in creating a new synthetic data set Ω', with larger cardinality, that has the same mean, variance and Fourier Spectrum of Ω. Although the cardinality of Ω' can be chosen arbitrarily, the method of surrogate data becomes infeasible when the dimensionality of the data set is high. For instance, a 50-dimensional data set to be estimated must have at least 10^{25} data points, on the basis of the inequality (9).

4.3.5. Camastra–Vinciarelli's algorithm

For the reasons described above, Fractal-based algorithms do not satisfy the third Ideal ID requirement, i.e., they do not provide reliable ID estimate when the cardinality of the data set is high. In order to cope with this problem, Camastra and Vinciarelli [11] proposed an algorithm to power Grassberger and Procaccia method (GP method) w.r.t. high dimensionality, evaluating empirically how much the GP method underestimates the dimensionality of a data set when the data set cardinality is unadequate to estimate ID properly. Let $\Omega = \{\vec{x}_1, \ldots, \vec{x}_\ell\} \subseteq \mathbb{R}^N$ be a data set, Camastra–Vinciarelli's algorithm has the following steps:

1. Create a set Ω', whose ID M is known, with the same cardinality ℓ of Ω. For instance, Ω' could be composed of ℓ data points randomly generated in a M-dimensional hypercube.
2. Measure the Correlation Dimension M_c of Ω' by the GP method.
3. Repeat the two previous steps for T different values of M, obtaining the set $\mathcal{C} = \{(M_i, M_c^i) : i = 1, 2, \ldots, T)$.
4. Perform a best fitting of data in $\mathcal{C}$ by a nonlinear method. A plot (*reference curve*) $\mathcal{P}$ of M_c versus M is generated. The reference curve allows inferring the value of M_c when M is known.
5. The Correlation Dimension M_c of Ω is computed by the GP method and, by using $\mathcal{P}$, the ID of Ω can be estimated.

The algorithm assumes that the curve $\mathcal{P}$ depends on ℓ and its dependence on Ω' sets are negligible. It is worth mentioning that Oganov and Valle [95] used Camastra–Vinciarelli's algorithm to estimate ID of high dimensional Crystal Fingerprint spaces [73].

4.4. Multiscale global methods

The aforementioned global methods do not satisfy the second Ideal ID requirement, i.e., they are not robust w.r.t. multiscaling. To the best of our knowledge, there are only two global ID methods that take into account the multiscaling problem. The former was proposed by Wang and Marron [100] who, on the basis of geometrical considerations, produced, instead of a single value, an ID estimation function, which by taking the scale parameter S as input, yields the respective ID estimate. However, the method

was only validated on low-dimensional synthetic data and was not applied on real data benchmarks. The latter is the *Hein–Audibert's algorithm*, that we analyze below. Hein and Audibert [40] proposed a generalization of the correlation integral (see Section 4.3.2), in terms of U-statistics [42], defined as follows:

$$U_{\ell,h}(K) = \frac{2}{\ell(\ell-1)} \sum_{i=1}^{\ell} \sum_{j=i+1}^{\ell} \frac{1}{h^M} K(\|\vec{x}_j - \vec{x}_i\|^2/h^2) \tag{10}$$

where $K(\cdot)$ is a generic kernel of bandwidth h, M is the dimensionality of the manifold where the data are assumed to lie and ℓ is the cardinality of a generic subsample of the data set. On the basis of the Hoeffding Theorem [43], in order to guarantee the convergence of the U-statistics the bandwidth h must fulfill $\ell h^M \to \infty$. Hein and Audibert used this property by fixing a convergence rate for each dimension, that means that h has been fixed as a function of the data set cardinality ℓ. Then Eq. (10) is computed for subsamples of different cardinalities of the data set Ω, where h varies according to the function that has been chosen. ID is determined by making the log–log plot between the U-statistic and h, for each subsample, and taking the smallest slope as the ID value. In detail, Hein–Audibert's algorithm has two steps. In the first step, the scale function $h_M(s)$ is fixed as function of the dimension M and the cardinality of the subset considered s. Hein and Audibert suggested that the scale function $h_M(s)$ should be given by:

$$h_M(s) = h_M(\ell) \left(\frac{\ell \log s}{s \log \ell} \right)^{\frac{1}{M}}$$

where ℓ is the cardinality of the whole data set Ω, whereas s is the cardinality of its subsample. The function $h_M(\ell)$ is given by $h_M(\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \delta(\vec{x}_i)$, where $\delta(\vec{x}_i)$ provides the distance, assumed Euclidean for simplicity, of the sample $\vec{x}_i$ to its nearest neighbor. In the second step the ID dimension is computed. First, the kernel $K(\cdot)$ must be fixed. To this purpose, Hein and Audibert suggested to use a kernel with compact support, for instance[1] $K(u) = (1-u)_+$. Then we consider subsamples of the data set Ω, of cardinality $[\lfloor \ell/5 \rfloor, \lfloor \ell/4 \rfloor, \lfloor \ell/3 \rfloor, \lfloor \ell/2 \rfloor, \ell]$. For each dimension $d \in [1, d_{max}]$, where d_{max} is chosen properly, the empirical estimate of $U_{\lfloor \ell/r \rfloor, h_M(\lfloor \ell/r \rfloor)}(K)$, $(r = 1, \ldots, 5)$, is computed. Finally, the best fitting, for each dimension d, of the points $[\log h_M(\lfloor \ell/r \rfloor), \log U_{\lfloor \ell/r \rfloor, h_M(\lfloor \ell/r \rfloor)}(K)]$, $(r = 1, \ldots, 5)$ is performed and the line slope is computed. The ID is obtained by picking the smallest value among the computed slopes. Hein–Audibert's algorithm addresses, even partially, the problem of satisfying the second ID requirement. Although the algorithm also fulfills the first Ideal ID requirement, it does not guarantee the other requirements since it does not have a work envelope and is not robust w.r.t. high dimensionality.

4.5. Other global methods

In this group, we collect the global methods that do not belong to above-mentioned categories. Costa-Hero's algorithm [19], Lin-Zha's simplex-based method [61], IDEA [81,82] and DANCo [14] algorithms belong to this category. All these algorithms do not satisfy the second Ideal ID requirement, since they do not take into account the multiscaling problem. Costa-Hero's algorithm assumes that data set $\Omega = (\vec{x}_1, \ldots, \vec{x}_\ell) \subseteq \mathbb{R}^N$ lie on a smooth compact M-dimensional manifold. Costa-Hero's algorithm builds a Euclidean neighborhood graph $\mathcal{G}$ over the data set Ω where each pattern $\vec{x}$ is represented by a vertex of $\mathcal{G}$. Each vertex is connected to the vertices, by an edge e, whose weight $w(e)$ is given by the Euclidean distance between the patterns representing the two vertices. Then, the Minimum Spanning Tree (MST) is built on the graph $\mathcal{G}$ by Kruskal's [55] (or Prim's [77]) algorithm. We denote by $L_\gamma^{\mathcal{M}}(\Omega)$, the so-called *Geodetic Minimum Spanning Tree Length* (GMSTL), defined as $L_\gamma^{\mathbb{R}^N}(\Omega) = \min_{T \in ST} \sum_{e \in T} w(e)^\gamma$, where ST is the set of spanning trees built on the graph G and $\gamma \in (0, N)$ is the so-called *edge exponent* (or *power-weighting constant*). Starting from an important result in geometric probability, due to Beardwood et al. [1], Costa and Hero derived an equation that connects GMSTL to ID. If we define $\mathcal{L}_\ell = \log L_\gamma^{\mathcal{M}}(\Omega)$, then the following equation holds:

$$\mathcal{L}_\ell = a \log \ell + b + \epsilon_\ell \tag{11}$$

where $a = \frac{ID-\gamma}{\gamma}$, b is a parameter related to the *intrinsic entropy* of the manifold (not described here for sake of brevity) and ϵ_ℓ is an error residual that goes to zero, with probability 1, as $\ell \to \infty$. We generate a collection of Ω_i data sets, each with a different cardinality ℓ_i, from Ω by bootstrap, and we compute for each Ω_i the corresponding $\mathcal{L}_{\ell_i}$. Then we plot $\mathcal{L}_{\ell_i}$ versus ℓ_i and we evaluate the estimates of a and b, indicated by $\hat{a}, \hat{b}$, respectively, by a Linear Least Squares method. Finally, if we fix to 1 the exponent γ, as suggested by Costa and Hero, the ID estimate, denoted by $\hat{ID}$, is given by:

$$\hat{ID} = \frac{1}{1-\hat{a}}. \tag{12}$$

Lin-Zha's method assumes that the data set lie on a *Riemannian manifold*[2] and it estimates ID by means of *simplicial reconstruction* [31] of the Riemannian manifold. The manifold ID is given by the maximal dimension of its simplices. Hence the method provides only integer values for ID estimate. Lin and Zha tested their algorithm on synthetical data sets and low-dimensional real data. Although efficient algorithms were proposed [31], however, the simplex reconstruction starting from data samples remains a difficult problem. Therefore the reliability of simplex-based ID estimators is an open problem.

[1] $K(u)_+ = u$ if $u \geq 0$, otherwise $K(u)_+ = 0$.

[2] A *Riemannian manifold* is a differentiable manifold $\mathcal{M}$ endowed with a smooth inner product (*Riemannian metric*) $g(u, v)$ on each tangent space $T_p\mathcal{M}$ [57].

IDEA, acronym of Intrinsic Dimension Estimation Algorithm, is based on the following observation. An M-dimensional vector $\vec{z}$ randomly sampled from an M-dimensional hypersphere according to the uniform probability density function, can be generated by drawing a point $\vec{\hat{z}}$ from a normal distribution $\mathcal{N}(0,1)$ and by scaling its norm (see [30]), i.e., $\vec{z} = \frac{u^{\frac{1}{M}}}{\|\vec{\hat{z}}\|}\vec{\hat{z}}$, where u is a random sample drawn from the uniform distribution $\mathcal{U}(0,1)$. Since u is uniformly distributed, the quantities $1-u^{\frac{1}{M}}$ are distributed according to the beta probability density function with expectation $\mathbb{E}[1-u^{\frac{1}{M}}] = \frac{1}{1+M}$. Therefore Rozza et al assume that $\mathbb{E}[1-\|\vec{z}\|] = \frac{1}{1+M}$ and derive the following relation:

$$M = \frac{\mathbb{E}[\|z\|]}{1-\mathbb{E}[\|z\|]}. \tag{13}$$

IDEA algorithm is as follows. Given the data set $\Omega = (\vec{x}_1, \ldots, \vec{x}_\ell) \subseteq \mathbb{R}^N$, for each point $\vec{x}_i \in \Omega$, the set $X_{k+1}(\vec{x}_i) = (\vec{u}_1, \ldots, \vec{u}_{k+1})$ of its $k+1$ nearest neighbors is computed. Let $\vec{x'} \in \hat{X}_{k+1}(\vec{x}_i)$ be the farthest point from $\vec{x}_i$ and denote by $X_k = X_{k+1}\backslash\vec{x'}$ the set of the neighbors of $\vec{x}_i$ without $\vec{x'}$. Since almost surely $\|\vec{x}-\vec{x}_i\| < \|\vec{x'}-\vec{x}_i\| \ \ \forall \vec{x} \in X_k$, points in X_k can be viewed as drawn from the hypersphere $\mathcal{B}(\vec{x}_i, \|\vec{x'}-\vec{x}_i\|)$. Therefore, to compute the ID of Ω, the expectation of distances, denoted by m, is estimated as follows:

$$m \simeq \sum_{\vec{x}\in X_k} \frac{\|\vec{x}-\vec{x}_i\|}{\|\vec{x'}-\vec{x}_i\|} \tag{14}$$

The equation (14) is justified since the limit of its right side, for $k \to \infty$, is m [82]. Hence, ID of Ω, denoted by M, is given by:

$$M = \frac{\frac{1}{N\ell}\sum_{i=1}^{\ell}\sum_{\vec{x}\in X_k}\frac{\|\vec{x}-\vec{x}_i\|}{\|\vec{x'}-\vec{x}_i\|}}{1-\frac{1}{N\ell}\sum_{i=1}^{\ell}\sum_{\vec{x}\in X_k}\frac{\|\vec{x}-\vec{x}_i\|}{\|\vec{x'}-\vec{x}_i\|}} \tag{15}$$

The computational complexity of the IDEA algorithm is $O(N\ell^2)$. IDEA does not have a work envelope, i.e., it does not provide a lower bound on the data set cardinality to get a reliable ID estimate. Nevertheless, Rozza et al tried to make robust IDEA w.r.t. high dimensionality, by proposing the following empirical procedure. Given the data set $\Omega = (\vec{x}_1, \ldots, \vec{x}_\ell) \subseteq \mathbb{R}^N$, it generates T subsets Ω_r of cardinality ℓ_r, from Ω and it estimates their dimensionality, denoted by M_r, by means of IDEA. Then it plots $\log(\Omega_r)$ versus M_r and it does the best fitting of the empirical function $M_r \simeq a_0 - \frac{a_1}{\log_2(\frac{\ell_r}{a_2}+a_3)}$, where a_0, a_1, a_2, a_3 are parameters that have to be estimated by a nonlinear least square algorithm. If $a_1 > 0$ then the ID value of Ω is a_0; otherwise, the ID value is provided by the IDEA algorithm applied to the whole data set Ω. The computational complexity of the empirical procedure is $O(TN\ell^2)$.

Recently, the same authors of IDEA have proposed a further ID estimator, DANCo [14], based on the angle and norm concentration effects, that they are observed when the data dimensionality tends to infinity.

We pass to describe DANCo. Given the data set $\Omega = (\vec{x}_1, \ldots, \vec{x}_\ell) \subseteq \mathbb{R}^N$, for each $\vec{x}_i \in \Omega$, the set, denoted by X_{k+1}, of $(k+1)$ nearest neighbors is computed. The farthest neighbors of $\vec{x}_i$ is indicated with $\hat{\vec{x}}_i$. The function $\rho(\vec{x}_i)$ is defined as follows:

$$\rho(\vec{x}_i) = \min_{\vec{x}_j \in X_{k+1}} \frac{\|\vec{x}_i - \vec{x}_j\|}{\|\vec{x}_i - \hat{\vec{x}}_i\|} \tag{16}$$

The maximum likelihood ID estimate, denoted by $\hat{M}_{ML}$ is obtained, as suggested in [63], by solving numerically the optimization problem:

$$\hat{M}_{ML} = \arg_{1\le d\le N} \max \ell \log kd + (d-1)\sum_{\vec{x}_i\in\Omega}[\rho(\vec{x}_i) + (k-1)\log(1-\rho^d(\vec{x}_i))] \tag{17}$$

To this purpose, the authors suggest to use the constrained optimization method proposed in [17].

Then, for each point $\vec{x}_i \in \Omega$, the k nearest neighbors are computed, and successively centered by subtracting the point $\vec{x}_i$, denoting by X_k^i the formed set, namely $X_k^i = \{\vec{x}_j - \vec{x}_i : \forall \vec{x}_i \in X_k(\vec{x}_i)\}$, where $X_k(\vec{x}_i)$ indicates the set of k neighbors of $\vec{x}_i$. Successively, the function

$$\theta(\vec{x_w}, \vec{x_j}) = \arccos\frac{\vec{x_w}\cdot\vec{x_j}}{\|\vec{x_w}\|\,\|\vec{x_j}\|} \tag{18}$$

is used to compute the angles of all the possible couples of vectors in X_k^i. Therefore, for each neighborhood of a data point $\vec{x}_i$ a vector $\vec{\theta}_i$ is computed, whose components are $\theta(\vec{x_w}, \vec{x_j}) : \forall \vec{x_w}, \vec{x_j} \in X_k^i$. Each component of $\vec{\theta}_i$ follows the *Von-Mises Fisher distribution* [69], the parameters of this distribution, ν and τ, are estimated by using the maximum likelihood approach, denoting with $\bar{\nu}$ and $\bar{\tau}$, their respective means. The statistics computed on Ω is compared with synthetic data sets, whose ID is known. To this purpose, for each dimensionality d, between 1 and N a set of ℓ points from the unit d-dimensional hypersphere are extracted and the consequent Maximum Likelihood estimate M_{ML}^d is computed. Then the parameters $\bar{\nu}_d$ and $\bar{\tau}_d$ of Von Mises–Fisher distribution are computed. Finally, DANCo computes the ID of Ω minimizing:

$$ID = \arg_{d\in\{1,N\}} \max KL_d + KL_{\nu\tau} \tag{19}$$

with

$$KL_d = H_k \frac{M_{ML}^d}{M_{ML}} - 1 - H_{k-1} - \log \frac{M_{ML}^d}{M_{ML}} - (k-1) \sum_{i=0}^{k} (-1)^i \binom{k}{i} \Psi\left(1 + \frac{iM_{ML}}{M_{ML}^d}\right)$$

where $\Psi(\cdot)$ is the digamma function and $H_k = \sum_{i=1}^{k} \frac{1}{i}$;

whereas

$$KL_{v\tau} = \log \frac{I_0(\bar{\tau}_d)}{I_0(\bar{\tau})} + \frac{I_1(\bar{\tau}) - I_1(-\bar{\tau})}{2I_0(\bar{\tau})} (\bar{\tau} - \bar{\tau}_d \cos(\bar{\nu}_d - \bar{\nu})) \tag{20}$$

where I_1 and I_0 denote the modified Bessel function of first kind with order 1 and 0, respectively [99].

According to the authors, the method seems to be quite robust to the increase of dimensionality. However, DANCo shares with IDEA the same limitations, namely it does not take into account the multiscaling problem and it does not provide an operative range for the algorithm.

5. Local methods

Local methods are algorithms that provide an ID estimation using the information contained in sample neighborhoods. In this case, data do not lie on a unique manifold of constant dimensionality but on multiple manifolds of different dimensionalities. Since a unique ID estimate for the whole data is clearly not meaningful, it prefers to provide an ID estimate for each small subset of data, assuming that it lies on a manifold of constant dimensionality. Properly, local methods estimate the *topological dimension* [41] of data manifold. Topological dimension is the basis dimension of the local linear approximation of the manifold where data lie, i.e., the tangent space. For example, if the data set lies on a M-dimensional manifold, then it has a M-dimensional tangent space at every point in the set. For instance, a sphere has a two-dimensional tangent space at every point and it may be viewed as a two-dimensional manifold. Since the sphere ID is three, the topological dimension is an ID lower bound. The topological dimension is often referred to as the *local dimension*. For this reason methods that estimate the topological dimension are called local. Local ID estimation methods are Fukunaga-Olsen's [32], local MDS, local Multiscale, *nearest-neighbor algorithms*. Examples of nearest-neighbor's algorithms are Pettis et al's [76], Trunk's [94] and Verveer-Duin's [97] ones, whose descriptions are omitted since they have, in practice, only historical interest. We have to mention that recently a novel nearest-neighbor's algorithm [60] has been proposed that we do not describe for the sake of brevity.

5.1. Fukunaga–Olsen's algorithm

Fukunaga–Olsen's algorithm is based on the observation that for data embedded in a linear subspace, the dimension is equal to the number of non-zero eigenvalues of the covariance matrix. Fukunaga and Olsen proposed that the data set ID can be computed by dividing the data set in small regions (*Voronoi tesselation* of data space), assuming that in each region (*Voronoi set*) the surface, where data lie, is approximately linear. Fukunaga-Olsen's algorithm is as follows. First, Voronoi tesselation is performed by means of a clustering algorithm, e.g., K-Means, and the eigenvalues of the local covariance matrix in each Voronoi set are computed. Then eigenvalues are normalized by dividing them by the largest eigenvalue. ID is defined as the number of normalized eigenvalues that are *significant*, i.e., larger than a threshold θ, fixed in a heuristic way. This is one of the weak points of the algorithm since it is not possible to fix a threshold value θ good for every problem.

Bruske and Sommer [9] proposed to improve Fukunaga-Olsen's algorithm using Topology Representing Network (TRN) [70] to perform the Voronoi tesselation of data space. Bruske-Sommer's algorithm is as follows. An optimal topology preserving map $\mathcal{T}$ by TRN is computed. Then, for each neuron $i \in \mathcal{T}$, a PCA is performed on the set Q_i consisting of the differences between the neuron i and all of its m_i closest neurons in $\mathcal{T}$. Bruske-Sommer's algorithm shares with Fukunaga-Olsen's one the same limitations. Since none of the eigenvalues of the covariance matrix is null due to noise, it is necessary to use heuristic thresholds in order to decide whether an eigenvalue is significant or not. It is appropriate to remark that the problem of fixing a threshold in Fukunaga-Olsen's and Bruske-Sommer's algorithms could be solved by performing PCA with Bayesian techniques, see Section 4.1.1, that compute automatically the significant eigenvalues. Recently, alternative approaches for the local ID estimation have been proposed by Fan et al [28] and Johnsson et al [46]. Fan et al' s method is based on minimal cover approximation. The *set cover* is defined as follows. Given a data set $\Omega = (\vec{x}_1, \ldots, \vec{x}_\ell) \subseteq \mathbb{R}^N$, and a collection $\mathcal{O} = \{\Omega_1, \ldots, \Omega_N\}$ of subsets of Ω. The *set cover* is the minimum sub-collection of $\mathcal{O}$ that covers all data points. However, the set cover problem is NP-hard and therefore its solution becomes infeasible for large data sets. Fan et al's algorithm has two steps. In the former step they compute an approximation of the set cover of the data set and then they estimate locally ID in each subset of set cover by PCA.

Johnson et al's method works on local data set. The method considers simplices with one vertex in the centroid and the other vertices in data points. Such simplices are used to estimate the expected value of what that authors call the *simplex skewness measure*. The simplex skewness measure is defined as the volume of a simplex constructed by the volume it would have if the edges incident to the centroid vertex were orthogonal. As a consequence of the so-called concentration phenomenon, the simplex skewness measure tends to 1 as the dimension increases. The value of ID is estimated from the simplex skewness measure.

5.2. Local MDS methods

Just as global MDS methods (see Section 4.2), local MDS methods are projection techniques that tend to preserve, as much as possible, the distances among data. In local MDS, as in global ones, to each projection is associated an index or a cost that measures the goodness of the projection. Unlike MDS methods, where the whole data set is considered, local MDS methods work only on a small subset of data. Examples of Local MDS methods are *ISOMAP* [88], *Local Linear Embedding* (*LLE*) [80] and *Laplacian Eigenmaps* [2]. The method for estimating ID is the same of global MDS and it has the following steps:

1. Compute several MDS projections considering different dimensionality for the output space.
2. Pick the MDS projection with the best index or the minimum cost.
3. ID is given by the dimensionality of the output space of the MDS projection selected.

In this section we just describe ISOMAP since both LLE and Laplacian Eigenmaps require as input the data set ID, and hence they cannot be used to estimate ID.

5.2.1. ISOMAP

ISOMAP [88], acronym of *Isometric feature mapping*, is one of the most popular MDS local methods. Let $\Omega = \{\vec{x}_1, \ldots, \vec{x}_\ell\} \in \mathcal{M} \subset \mathbb{R}^N$ be a data set formed by data drawn by the manifold $\mathcal{M}$. Before describing ISOMAP, we recall some basic notions of topology. A *homeomorphism* is a continuous function whose inverse is a continuous function, too. An M-dimensional manifold $\mathcal{M}$ is a set that is *locally homeomorphic* with $\mathbb{R}^M$, i.e., $\forall m \in \mathcal{M}$ an open neighborhood N_m around m and a homeomorphism $h : N_m \to \mathbb{R}^M$ exist. The neighborhood and the homeomorphism are called *coordinate patch* and *coordinate chart*, respectively. Having said that, ISOMAP aims at finding a *coordinate chart* that allows projecting the data set in $\mathbb{R}^M$. ISOMAP assumes that an *isometric chart* exists, i.e., a chart that preserves the distances between the data points. Therefore if two data points $\vec{x}_i, \vec{x}_j \in \mathcal{M}$ have *geodetic distance* $\Delta_{\mathcal{M}}(\vec{x}_i, \vec{x}_j)$, i.e., the distance along the manifold, then there is a chart $h : \mathcal{M} \to \mathbb{R}^M$ such that $\|h(\vec{x}_i) - h(\vec{x}_j)\| = \Delta_{\mathcal{M}}(\vec{x}_i, \vec{x}_j)$. Besides, ISOMAP assumes that the manifold $\mathcal{M}$ is smooth enough so that the geodetic distance between close points can be approximated by a line. ISOMAP uses the usual Euclidean distance between points to compute the geodetic distance between close points. On the contrary, Euclidean distance is not a good estimate of the geodetic distance between not close points, since the linear approximation becomes more and more inaccurate as the distance between data points increases. To compute the geodetic distance ISOMAP builds a *neighborhood graph* in the following way. ISOMAP computes, for each data point $\vec{x}$, the set of its neighbors $U(\vec{x})$ which can be composed in two different ways, yielding two ISOMAP versions, *K-ISOMAP* and *ϵ-ISOMAP*. In the former the set of neighbors is formed by its K nearest neighbors, in the latter it is formed by all data points whose distance is lower than ϵ. After the computation of the set of neighbors for each data point, ISOMAP builds a labeled graph $\mathcal{G}$ over the data set Ω where each data point is represented by a vertex of $\mathcal{G}$. Besides, each vertex, corresponding to a given data point $\vec{x}$, is connected to the vertices, corresponding to data points belonging to the set of its neighbors $U(\vec{x})$, by a weighted edge. The weight of the edge is given by the Euclidean distances between the data points representing the two vertices. Then ISOMAP evaluates the geodetic distance $\Delta_{\mathcal{M}}(\vec{x}_i, \vec{x}_j)$ between all data points of Ω computing the shortest-path between the corresponding vertices on the graph $\mathcal{G}$ by the *Dijkstra's* algorithm [18]. At the end of this step, ISOMAP produces a matrix $\Delta_{\mathcal{M}}$ whose element $\Delta_{\mathcal{M}}(i, j)$ is given by the geodetic distance between the data points $\vec{x}_i$ and $\vec{x}_j$, i.e., $\Delta_{\mathcal{M}}(i, j) = \Delta_{\mathcal{M}}(\vec{x}_i, \vec{x}_j)$. The final step of ISOMAP consists in applying a global MDS algorithm, e.g., Sammon's mapping, costructing an embedding of the data in an M-dimensional space which preserves as much as possible the geometry of the manifold. ISOMAP can be summarized in the following steps:

1. Take as input the data set Ω and the parameter K (or ϵ).
2. Compute the set of neighbors for each data point.
3. Build the neighborhood graph.
4. Compute the shortest path graph given the neighborhood graph.
5. Make an M-dimensional embedding by means of a MDS algorithm.

The parameter K (or ϵ) controls the size of the neighborhood and is crucial [88] since the ISOMAP performance are strongly influenced by the size of neighborhood. Techniques [84] for the automatic tuning of the parameter K are available. ISOMAP guarantees theoretically the fidelity of the manifold reconstruction under given assumptions. If the manifold is compact, sampled everywhere, isometrically embedded in $\mathbb{R}^M$ and the parameter space, i.e., the image of the chart, is convex then ISOMAP can reconstruct the manifold. The method for estimating ID by ISOMAP is the following.

1. Compute several ISOMAP with different dimensionality for the output space.
2. The ID estimate is given by the dimensionality of the output space of the ISOMAP with the minimum cost.

Finally, we discuss ISOMAP w.r.t. Ideal ID requirements. The first requirement is guaranteed when the data set dimensionality is not very large since the application of ISOMAP to ID estimation can become infeasibile when data set dimensionality is huge. ISOMAP does not guarantee the other Ideal ID requirements even if it does not suffer from the linearity limitation of PCA-based methods.

5.3. Multiscale local methods

In this section we review the unique local method, to the best of our knowledge, that address the multiscaling problems, i.e., the Brand's method [8] and the *Little-Jung-Maggioni's algorithm* [62].

5.3.1. Brand's method

Brand's method is a local method that can estimate, by an heuristic strategy, the data set ID. Let $\Omega = (\vec{x}_1, \ldots, \vec{x}_\ell) \subseteq \mathbb{R}^N$ be a data set, the method assumes that the data points of Ω are samples of an M-dimensional manifold. We search for a mapping $\mathcal{R} : \Omega \to \mathcal{Y} = (\vec{y}_1, \ldots, \vec{y}_\ell) \subseteq \mathbb{R}^M$ and its inverse $\mathcal{R}^{-1} : \mathcal{Y} \to \Omega$, such that local relations between close points are maintained. The map $\mathcal{R}$ guarantees that parallel lines in $\mathbb{R}^N$ are mapped onto continuous smooth non-intersecting curves in $\mathbb{R}^M$ in order to assure that linear operations on $\mathcal{Y}$ can be replaced by similar operations on Ω. Having said that, we consider a ball of radius r centered on a data point and containing $n(r)$ data points. The number $n(r)$ grows as r^M at the locally linear scale. The number is inflated at a smaller scale by noise, at a larger scale by curvature due to the manifold nonlinearity. In order to estimate the scale r we study how the r-ball grows when data points are added, tracking $c(r) = \frac{\log r}{\log n(r)}$. At the locally linear scale, $c(r)$ has a maximum, whose value is $\frac{1}{M}$, since data points are distributed only in the directions of the manifold's local tangent space. Therefore the maximum of $c(r)$ provides an estimate of both the scale and the local dimensionality of the manifold. Brand's method was tested only on data sets, mainly synthetic, of low ID, two or three, at most. Regarding to Ideal ID requirements, even if Brand's method addresses the problem of multiscaling, it does not guarantee most other requirements. Brand's method does not have a work envelope and robustness w.r.t. high dimensionality. Finally, since Brand's method estimates the dimensionality of the local tangent space, the ID estimate, yielded by the method, could not be close to the underlying manifold dimensionality, when the manifold is nonlinear.

5.3.2. Little-Jung-Maggioni's algorithm

Little-Jung-Maggioni's algorithm is based on *Multiscale Singular Vector Decomposition* (MSVD) [48]. Given a data set $\Omega = (x_1, \ldots, x_\ell) \subseteq \mathbb{R}^N$, represented under the form of a matrix $\ell \times N$, the *singular value decomposition*(SVD) decompones the matrix as $\Omega = U\Sigma V^T$, where U and V are appropriate matrices, denoting by V^T the transpose of V, and Σ a $N \times N$ diagonal matrix. We denote by $\Sigma_{ii} = \sigma_i$ its generic ith positive diagonal value. Little-Jung-Maggioni's algorithm has the following steps:

1. Compute, for each scale parameter r the SVD and denote by $SV^{(r)}$ the corresponding diagonal values $(\sigma^{(r)})_{i=1}^n$ obtained, thus performing the so-called multiscale singular value decomposition.
2. Estimate the *noise size* of data ϵ, that is obtained from the $SV^{(r)}$ since they do not grow as r increases.
3. Split the $SV^{(r)}$ in two sets: *non-noise SVs*, that are $> \epsilon$, and *noise SVs*, that are $\leq \epsilon$.
4. Identify a range of scales r where the noise SVs are small compared with other SV. For the range of scale identified, the ID estimate is provided by the number of non-scale SV.

SVD is strictly related to PCA. We recall that the ith diagonal value of Σ matrix is the square of the respective ith eigenvalue of PCA. Therefore, as PCA, SVD tends to overestimate ID. For this reason, Little-Jung-Maggioni's algorithm can provide only local reliable Multiscale ID estimates. It is necessary to remark that Little-Jung-Maggioni's algorithm provides no methods to identify the regions where to perform locally Multiscale SVD. Little-Jung-Maggioni's algorithm has a work envelope. The cardinality ℓ of the data set Ω required by the algorithm to get a reliable ID estimate is $\ell = O(M)$, where M is the ID of the data set.

6. Pointwise methods

In this category there are the algorithms that can produce both a global ID estimate of the whole data set and local pointwise ID estimate of each pattern of the data set. Unlike local methods, where the term local refers to the topological dimension, in pointwise methods local means that the dimension is estimated for the neighborhood of each data sample, thus providing an estimate of pointwise dimension (see Section 2). The global ID estimate is given by the mean of pointwise dimension of all patterns of data set.

Examples of pointwise methods are Farahmand et al's [29], Mordohai-Medioni's [71], He et al's [39] and Levina-Bickel's algorithms [58]. Farahmand et al's algorithm estimates ID locally around the data points using a nearest-neighbor method and it also provides a global ID estimate of the whole data set, averaging the local ID estimates.

Mordohai–Medioni's algorithm estimates geometric relationships among data by tensor voting. Then the ID at each data point is provided by the maximum gap in the eigenvalues of the tensor. ID global estimate can also be obtained averaging ID of each single data point.

He et al's method is based on manifold sampling assumption. If the manifold is densely sampled then the number of samples that are in an M-dimensional ball is given by the product between the volume and the ball density. The volume of the ball is derived by computing the radius of the ball by the graph distance that approximates the geodetic distance on manifold. He et al's method provides an ID estimation for each data point and the global ID estimate is yielded as in the above-mentioned algorithms, i.e., averaging all ID local estimates.

6.1. Levina–Bickel's algorithm

Levina–Bickel's algorithm derives the maximum likelihood estimator of the intrinsic dimensionality M from a data set $\Omega = (\vec{x}_1, \ldots, \vec{x}_\ell) \subseteq \mathbb{R}^N$. The data set Ω represents an embedding of a lower-dimensional sample, i.e., $\vec{x}_i = g(Y_i)$, where Y_i are sampled from an unknown smooth density f on $\mathbb{R}^M$, with $M \leq N$ and g is a smooth mapping. The last assumption guarantees that close data in $\mathbb{R}^M$ are mapped to close neighbors in the embedding. Having said that, we fix a data point $\vec{x} \in \mathbb{R}^N$ assuming that $f(\vec{x})$

is constant in a sphere $S_{\vec{x}}(r)$ centered in $\vec{x}$ of radius r and we view Ω as a homogeneous Poisson process in $S_{\vec{x}}(r)$. Given the inhomogeneous process $\{P(t,\vec{x}), 0 \leq t \leq r\}$

$$P(t,\vec{x}) = \sum_{i=1}^{\ell} I(\vec{x}_i \in S_{\vec{x}}(t)), \tag{21}$$

which counts the data whose distance from $\vec{x}$ is less than t. If we approximate it by a Poisson process and we neglect the dependence on $\vec{x}$, the rate $\lambda(t)$ of the process $P(t)$ is given by:

$$\lambda(t) = f(\vec{x})V(M)Mt^{M-1}, \tag{22}$$

where $V(M)$ is the volume of a M-dimensional unit hypersphere.

Eq. (22) is justified by the Poisson process properties since the surface area of the sphere $S_{\vec{x}}(t)$ is $\frac{d}{dt}[V(M)t^M] = V(M)Mt^{M-1}$. If we define $\theta = log f(\vec{x})$, the log-likelihood of the process $P(t)$ [86] is:

$$L(M,\theta) = \int_0^r log\lambda(t)dP(t) - \int_0^r \lambda(t)dt. \tag{23}$$

The equation describes an exponential family for which a maximum likelihood estimator exists with probability that tends to 1 as the number of samples ℓ tends to infinity. The maximum likelihood estimator is unique and must satisfy the following equations:

$$\frac{\partial L}{\partial \theta} = \int_0^r dP(t) - \int_0^r \lambda(t)dt = P(r) - e^{\theta}V(M)r^M = 0. \tag{24}$$

$$\frac{\partial L}{\partial M} = \left(\frac{1}{M} + \frac{V'(M)}{V(M)}\right)P(r) + \int_0^r log\ t\ dP(t) - e^{\theta}V(M)r^M\left(log\ r + \frac{V'(M)}{V(M)}\right) = 0. \tag{25}$$

If we plug Eq. (24) into Eq. (25), we obtain the maximum likelihood estimate for the dimensionality M:

$$\hat{M}_r(\vec{x}) = \left[\frac{1}{P(r,\vec{x})}\sum_{j=1}^{P(r,\vec{x})} log\frac{r}{T_j(\vec{x})}\right]^{-1}, \tag{26}$$

where $T_j(\vec{x})$ denotes the Euclidean distance between $\vec{x}$ and its jth nearest neighbor. Levina and Bickel suggested to fix the number of the neighbors k rather than the radius of the sphere r. Therefore the pointwise ID estimate becomes:

$$\hat{M}_k(\vec{x}) = \left[\frac{1}{k-1}\sum_{j=1}^{k-1} log\frac{T_k(\vec{x})}{T_j(\vec{x})}\right]^{-1}. \tag{27}$$

The global estimate of ID is obtained by averaging on all data points of the data set Ω, namely:

$$\hat{M}_k = \frac{1}{\ell}\sum_{i=1}^{\ell}\hat{M}_k(\vec{x}_i). \tag{28}$$

The dimension estimate depends on the value of k. Levina and Bickel suggest to average over a range of values of $k = k_1, \ldots, k_2$ obtaining the final global estimate of ID, i.e.,

$$\hat{M} = \frac{1}{k_2 - k_1 + 1}\sum_{k=k_1}^{k=k_2}\hat{M}_k. \tag{29}$$

David Mac Kay and Zoubin Ghamarani, in an unpublished comment [66], criticized Levina and Bickel' s procedure of the global ID estimation. Instead, they proposed to average the inverse of the estimators $\hat{M}_k(\vec{x}_i)$. In this way, Eq. (28) must be replaced with:

$$\hat{M}_k = \frac{\ell(k-1)}{\sum_{i=1}^{\ell}\sum_{j=1}^{k-1} log\frac{T_k(\vec{x}_i)}{T_j(\vec{x}_i)}}. \tag{30}$$

Using the same Levina and Bickel's approach, the final ID estimate has to be obtained by averaging $\hat{M}_k$ over a range of values of $k = k_1, \ldots, k_2$, obtaining the final ID estimate expressed by Eq. (29). A variant of Levina–Bickel's algorithm was recently proposed by Gupta and Huang [37]. Regarding the computational complexity, the Levina–Bickel's algorithm requires a sorting algorithm (e.g., mergesort), whose complexity is $O(\ell \log \ell)$, where ℓ denotes the cardinality of the data set. Hence the computational complexity for estimating $\hat{M}_k$ is $O(k\ell^2 \log \ell)$, where k denotes the number of the neighbors that have to be considered. Besides, Levina and Bickel suggested to consider an average estimate repeating the estimate M_k s times, where s is the difference between the maximum and the minimum value that k can assume, i.e., k_2 and k_1, respectively. Therefore the overall computational complexity of the Levina–Bickel's algorithm is $O(k_2\ s\ell^2 \log \ell)$. Finally, we move on to discuss the Levina–Bickel's algorithm under Ideal ID framework. Levina–Bickel's algorithm does not take into account multiscaling and it does not have a work envelope. Moreover, the robustness w.r.t. high dimensionality is not guaranteed.

7. Evaluation of ID estimation methods

In this section we discuss the properties of the main ID estimation methods, described before in the paper. To this purpose, we remark, as discussed in Section 3, that the accuracy criterion cannot be used as the unique parameter for assessing the merit of an ID estimation method. Therefore, we analyze the main ID estimation methods under the Ideal ID estimation framework, introduced before. The results are summarized in Table 1, where the last four requirements are reported, since the first one, namely the *computational feasible* criterion, is fulfilled by all techniques quoted in the table. We state in advance that the accuracy is evaluated for low-dimensional data sets, since the performance on the high dimensional data sets is evaluated by a properly defined parameter (i.e., high dimensionality robustness). In the table we do not include PCA-based methods [6,7,36,93,103] whose accuracy is not guaranteed since, being linear methods, they generally overestimate ID (e.g., Table 2). For the same reason, we have not included in the table, Fukunaga–Olsen's algorithm and its variants [9,28], based on local PCA, that have the same limitations of PCA-based techniques. Finally, Nearest-Neighbors methods [76,94,97] are not included since do they not satisfy any requirement.

Having said that, we move on to discuss what is described in the Table 1.

As shown in the Table 1, only Hein-Audibert's [40], Brand's [8] and Little-Jung-Maggioni's [62] algorithms take into account the problem of multiscaling. At the same time, there are very few algorithms (i.e., Grassberger-Procaccia plus Camastra-Vinciarelli's correction [11], IDEA [81,82] and DANCo [14]) that try to cope with the high dimensionality challange. Finally, only fractal-based (e.g., Kégl's , Grassberger-Procaccia's algorithms) and Little-Jung-Maggioni's algorithms have the operative range, namely they indicate the minimum cardinality that a data set must have in order to get a reliable ID estimate.

As it can be observed, there is no ID estimation method that satisfies all Ideal ID criteria. For this reason, it is not possible to assess what is the best ID estimation method. As a practical advice, the best strategy for estimating the unknown ID of a data set is to use an ensemble of ID estimators that have similar qualities and then to combine their ID estimates. For instance, if we want to estimate the ID of a high dimensional data set, a good strategy is to form an ensemble of ID estimators that have high dimensionality robustness and then to average their estimates.

Finally, we briefly summarize what are the research trends in ID Estimation methods.

Table 1
ID estimation methods evaluated under the Ideal ID estimation framework. The acronyms GP, GP+CV, JSF, LJM stand for Grassberger–Procaccia, Grassberger–Procaccia plus Camastra-Vinciarelli's correction, Johnson-Soneson-Fontes and Little-Jung-Maggioni, respectively.

ID estimation methods	Multiscaling robustness	High dimensionality robustness	Operative range	Accuracy
Kégl			√	√
GP			√	√
GP+CV		√	√	√
Hein-Audibert	√			√
Costa-Hero				√
IDEA		√		√
DANCo		√		√
JSF				√
Brand	√			√
LJM	√		√	
Levina-Bickel				√

Table 2
ID of Santa-Fe benchmark D estimated by main ID estimation methods. Santa Fe D benchmark, formed by 1000 real data points, has ID equal to 9. The acronyms GP, GP+CV stand for Grassberger–Procaccia, Grassberger–Procaccia plus Camastra-Vinciarelli's correction, respectively.

ID Estimation methods	ID estimate
BPCA [14]	18.00
GP	7.54
GP+CV	8.84
Hein-Audibert [14]	6.00
IDEA [14]	7.26
DANCo [14]	8.19
Levina-Bickel [14]	7.16

7.1. Research trends in ID estimation methods

In the last years the research on ID estimation methods has been developed along the following main three lines. The first one yields ID estimators that make use of the so-called norm and angle concentration effect (see Section 5.1), namely norms and angles between vectors tend to assume the same value as the dimensionality increases. DANCo and JSF algorithms, are based on norm and angle concentration effect, although the former is global and the latter is local.

The second line lies in the attempt of developing ID estimators that are robust w.r.t. high dimensionality. This line has been pursued by a single research group that has developed some algorithms (e.g., IDEA, DANCo) that would seem to guarantee an ID robustness.

The last one consists in developing local methods that try to overcome the limitations of Fukunaga–Olsen's algorithm [28,46]. In reality, only the second method seems to have achieved the goal.

Finally, we have to mention that an ID estimator, that does not require the computation of distances between data points, has been recently proposed [54].

8. Conclusions

In the paper we have reviewed the ID estimation methods underlining their advances. We have defined the properties that an ideal ID estimator should have, discussing the ID estimation methods surveyed according to this framework. Nevertheless, some problems remain open. As remarked previously, ID depends on the scale of data. Although some ID estimation methods [8,40,62] tried to take into account, even if partially, of the data scale, a reliable multiscale ID estimator is not available, yet.

The other open problems are related to the robustness of ID estimators w.r.t. the curse of dimensionality. About this topic, there are two issues that remain to be fully addressed. The former issue is the work envelope, i.e., each ID estimation method should provide a lower bound on the cardinality in order to guarantee an accurate ID estimation. To the best of our knowledge, the lower bound is available only for fractal-based methods and Little-Jung-Maggioni's algorithm, whereas the other ID methods ignored the topic. The latter issue is the lack of the robustness of ID estimators w.r.t. high dimensionality. Although empirical solutions [11,14,82] were proposed, the construction of a robust ID estimator w.r.t. high dimensionality remains one of the challanges of the research in machine learning and pattern recognition.

Acknowledgments

Firstly, the authors are indebted to the anonymous reviewers for their valuable comments. The authors would like to thank Frances Mary Donegan for the proofreading of the paper.

References

[1] J. Beardwood, J.H. Halton, Hammersley, The shortest path through many points, Proc. Camb. Philo. Soc. 55 (1959) 299–327.
[2] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (2003) 1373–1396.
[3] R. Bellman, Adaptive Control Processes: A Guided Tour, Princeton University Press, 1961.
[4] R.S. Bennett, The intrinsic dimensionality of signal collections, IEEE Trans. Inf. Theory 15 (1969) 517–525.
[5] C. Bishop, Neural Networks for Pattern Recognition, Cambridge University Press, 1995.
[6] C.M. Bishop, Bayesian pca, in: Advances in Neural Information Processing Systems, MIT Press, 1998, pp. 382–388.
[7] C. Bouveyron, G. Celeux, S. Girard, Intrinsic dimension estimation by maximum likelihood in isotropic probabilistic pca, Pattern Recogn. Lett. 32 (2011) 1706–1713.
[8] M. Brand, Charting a manifold, in: Advances in Neural Information Processing, MIT Press, 2003, pp. 961–968.
[9] J. Bruske, G. Sommer, Intrinsic dimensionality estimation with optimally topology preserving maps, IEEE Trans. Pattern Anal. Mach. Intell. 20 (5) (1998) 572–575.
[10] G. Burdea, P. Coiffet, Virtual Reality Technology, John-Wiley & Sons, New York, 2003.
[11] F. Camastra, A. Vinciarelli, Estimating the intrinsic dimension of data with a fractal-based method, IEEE Trans. Pattern Anal. Mach. Intell. 24 (10) (2002) 1404–1407.
[12] F. Camastra, Data dimensionality estimation methods: a survey, Pattern Recogn. 36 (12) (2003) 2945–2954.
[13] F. Camastra, M. Filippone, A comparative evaluation of nonlinear dynamics methods for time series prediction, Neural Comput. Appl. 18 (8) (2009) 1021–1029.
[14] C. Ceruti, S. Bassis, A. Rozza, G. Lombardi, E. Casiraghi, P. Campadelli, Danco: an intrinsic dimensionality estimator exploiting angle and norm concentration, Pattern Recogn. 47 (2014) 2569–2581.
[15] E. Chavez, G. Navarro, R. Baeza-Yates, J.L. Marroquin, Searching in metric spaces, ACM Comput. Surv. 33 (3) (2001) 273–321.
[16] J. Claussen, T. Villmann, Magnification control in winner relaxing neural gas, Neurocomputing 63 (1) (2005) 125–137.
[17] T.F. Coleman, Y. Li, An interior, trust region approach for nonlinear minimization subject to bounds, SIAM J. Optim. 6 (1996) 418–445.
[18] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, MIT Press, 1990.
[19] J. Costa, A. Hero, Geodesic entropic graphs for dimension and entropy estimation in manifold learning, IEEE Trans. Signal Process. 52 (8) (2004) 2210–2221.
[20] T.F. Cox, M.A.A. Cox, Multidimensional Scaling, CRC Press, 2010.
[21] P. Demartines, J. Herault, Curvilinear component analysis: a self-organizing neural network for nonlinear mapping in cluster analysis, IEEE Trans. Neural Netw. 8 (1) (1997) 148–154.
[22] R. Duda, P. Hart, D. Stork, Pattern Classification, John-Wiley & Sons, New York, 2001.
[23] J.P. Eckmann, D. Ruelle, Ergodic theory of chaos and strange attractors, Rev. Modern Phys. 57 (1985) 617–659.
[24] J.P. Eckmann, D. Ruelle, Fundamental limitations for estimating dimensions and lyapounov exponents in dynamical systems, Physica D-56 (1992) 185–187.
[25] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, Chapman and Hall, 1993.
[26] J. Einbeck, Z. Kalantana, Intrinsic dimensionality estimation for high-dimensional data sets: new approaches for the computation of correlation dimension, J. Emerg. Technol. Web Intell. 5 (2) (2013) 91–97.
[27] M. Fan, H. Qiao, B. Zhang, Intrinsic dimension estimation of manifolds by incising balls, Pattern Recogn. 42 (2009) 780–787.

[28] M. Fan, X. Zhang, S. Chen, H. Bao, S. Maybank, Dimension estimation of image manifolds by minimal cover approximation, Neurocomputing 105 (2013) 19–29.
[29] A.M. Farahmand, C. Szepesvari, J.-Y. Audibert, Manifold-adaptive dimension estimation, in: Proceedings of the 24th International Conference on Machine Learning, 2007, pp. 265–272.
[30] G.S. Fishman, Monte Carlo: Concepts, Algorithms, and Applications, Springer-Verlag, 1996.
[31] D. Freedman, Efficient simplicial reconstructions of manifolds from their samples, IEEE Trans. Pattern Anal. Mach. Intell. 24 (10) (2002) 1349–1357.
[32] K. Fukunaga, D. Olsen, An algorithm for finding intrinsic dimensionality of data, IEEE Trans. Comput. C-20 (2) (1971) 176–183.
[33] K. Fukunaga, Intrinsic dimensionality extraction, in: Classification, Pattern Recognition and Reduction of Dimensionality, Handbook of Statistics, North Holland, Amsterdam, 1982, pp. 347–360.
[34] K. Fukunaga, Introduction to Statistical Pattern Recognition, Morgan Kaufman, 1990.
[35] P. Grassberger, I. Procaccia, Measuring the strangeness of strange attractors, Physica D9 (1983) 189–208.
[36] Y. Guan, J.G. Dy, Sparse probabilistic principal component analysis, J. Mach. Learn. Res. - Proc. Track 5 (2009) 185–192.
[37] M.D. Gupta, T.S. Huang, Regularized maximum likelihood for intrinsic dimension estimation, in: Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI2010), 2010, pp. 220–227.
[38] F. Hausdorff, Dimension und äusseres mass, Math. Ann. 79 (157) (1918).
[39] J. He, L. Ding, L. Jiang, Z. Li, Q. Hu, Intrinsic dimensionality estimation based on manifold assumption, J. Vis. Commun. Image Represent. 25 (2014) 740–747.
[40] M. Hein, J.-Y. Audibert, Intrinsic dimensionality estimation of submanifolds in $\mathbb{R}^d$, in: ICML' 05 Proceedings of the 22*nd* international conference on Machine Learning, 2005, pp. 289–296.
[41] A. Heyting, H. Freudenthal, Collected Works of L.E.J Brouwer, North Holland Elsevier, 1975.
[42] W. Hoeffding, A class of statistics with asymptotically normal distributions, Ann. Stat. 19 (1948) 293–325.
[43] W. Hoeffding, Probability inequalities for sums of bounded random variables, Journal of American Statistical Association 58 (1963) 13–30.
[44] V. Isham, Statistical aspects of chaos: a review, in: Networks and Chaos-Statistical and Probabilistic Aspects, Chapman & Hall, 1993, pp. 124–200.
[45] A. Jain, R. Dubes, Algorithms for Clustering Data, Prentice Hall, Englewood Cliffs, 1988.
[46] K. Johnsson, C. Soneson, M. Fontes, Low bias local intrinsic dimension estimator from expected simplex skewness, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (1) (2015) 196–202.
[47] I.T. Jollife, Principal Component Analysis, Springer-Verlag, 1986.
[48] P.W. Jones, Rectifiable sets and the traveling salesman problem, Inventiones Mathematicae 102 (1990) 1–15.
[49] D. Kaplan, L. Glass, Understanding Nonlinear Dynamics, Springer-Verlag, 1995.
[50] J. Karhunen, J. Joutsensalo, Representations and separation of signals using nonlinear pca type learning, Neural Networks 7 (1) (1994) 113–127.
[51] B. Kégl, Intrinsic dimension estimation using packing numbers, in: Advances in Neural Information Processing, MIT Press, 2003, pp. 681–688.
[52] M. Kirby, Geometric Data Analysis: An Empirical Approach to Dimensionality Reduction and the Study of Patterns, John Wiley and Sons, 2001.
[53] T. Kohonen, Self-Organizing Map, Springer-Verlag, 1995.
[54] M. Kleindessner, U. von Luxburg, Dimensionality estimation without distances, in: Proceedings of the 18th International Conference on Atificial Intelligence and Statistics (AISTATS), 2015, p. toappear.
[55] J.B. Kruskal, On the shortest spanning subtree of a graph and the travelling salesman problem, Proceedings of the American Mathematical Society 7 (1956) 48–50.
[56] J.B. Kruskal, Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, Psychometrika 29 (1964) 1–27.
[57] J.M. Lee, Riemannian Manifolds: An Introduction to Curvature, Springer-Verlag, 1997.
[58] E. Levina, P. Bickel, Maximum likelihood estimation of intrinsic dimension, in: Advances in Neural Information Processing, MIT Press, 2005, pp. 777–784.
[59] J. Li, D. Tao, Simple exponential family pca, in: Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS), 2010, pp. 453–460.
[60] L. Liao, Y. Zhang, S.J. Maybank, Z. Liu, Intrinsic dimension estimation via nearest constrained subspace classifier, Pattern Recognition 47 (3) (2014) 1485–1493.
[61] T. Lin, H. Zha, Riemannian manifold learning, Ieee Transactions on Pattern Analysis and Machine Intelligence 30 (5) (2008) 796–809.
[62] A. Little, Y.-M. Jung, M. Maggioni, Multiscale estimation of intrinsic dimensionality of a data set, in: Manifold Learning and its Applications: papers from the AAAI Fall Symposium, IEEE, 2009, pp. 26–33.
[63] G. Lombardi, A. Rozza, C. Ceruti, E. Casiraghi, P. Campadelli, Minimum neighbor distance estimators of intrinsic dimension, in: Machine Learning and Knowledge Discovery in Databases, Springer, 2011, pp. 374–389.
[64] E.N. Lorenz, Deterministic non-periodic flow, Journal of Atmospheric Science 20 (1963) 130–141.
[65] D.J.C. Mac Kay, Probable networks and plausible prediction - a review of practical bayesian methods for supervised neural networks, Network: Computation in Neural Systems 6 (3) (1995) 469–505.
[66] D. MacKay, Z. Ghamarani, Comments on 'maximum likelihood estimation of intrinsic dimension by e.levina and m.bickel'., 2005. University of Cambridge, http://inference.phy.cam.uc.uk/mackay/dimension.
[67] E.C. Malthouse, Limitations of nonlinear pca as performed with generic neural networks, IEEE Transaction on Neural Networks 9 (1) (1998) 165–173.
[68] B. Mandelbrot, Fractals: Form, Chance and Dimension, Freeman, 1977.
[69] V.M. Mardia, P. Jupp, Directional Statistics and Probability, John Wiley & Sons, 2009.
[70] T. Martinetz, K. Schulten, Topology representing networks, Neural Networks 3 (1994) 507–522.
[71] P. Mordohai, G. Medioni, Dimensionality estimation, manifold learning and function approximation using tensor voting, Journal of Machine Learning Research 11 (2010) 410–450.
[72] D. Nikolic, V. Moca, W. Singer, R. Muresan, Properties of multivariate data investigated by fractal dimensionality, Journal of Neuroscience Methods 172 (2008) 27–33.
[73] A. Oganov, M. Valle, How to quantify energy landscapes of solids, The Journal of Chemical Physics 130 (2009) 104504.
[74] E. Ott, Chaos in Dynamical Systems, Cambridge University Press, Cambridge, 1988.
[75] V. Pestov, An axiomatic approach to intrinsic dimension of a dataset, Neural Networks 21 (2008) 204–213.
[76] K. Pettis, T. Bailey, T. Jain, R. Dubes, An intrinsic dimensionality estimator from near-neighbor information, IEEE Transaction on Pattern Analysis and Machine Intelligence 1 (1) (1979) 25–37.
[77] R.C. Prim, Shortest connection networks and some generalizations, Bell System Technical Journal 36 (1957) 1389–1401.
[78] M. Raginsky, S. Lazebnik, Estimation of intrinsic dimensionality using high-rate vector quantization, in: Advances in Neural Information Processing, MIT Press, 2006, pp. 1105–1112.
[79] A.K. Romney, R.N. Shepard, S.B. Nerlove, Multidimensionaling Scaling, vol. I, Theory, Seminar Press, 1972.
[80] S. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (12) (2000) 2323–2326.
[81] A. Rozza, G. Lombardi, M. Rosa, E. Casiraghi, P. Campadelli, Idea: Intrinsic dimension estimation algorithm, in: Image Analysis and Processing- ICIAP 2011, Springer, 2011, pp. 433–442.
[82] A. Rozza, G. Lombardi, C. Ceruti, E. Casiraghi, P. Campadelli, Novel high intrinsic dimensionality estimators, Machine Learning 89 (1-2) (2012) 37–65.
[83] J.W.J. Sammon, A nonlinear mapping for data structure analysis, IEEE Transaction on Computers C-18 (1969) 401–409.
[84] O. Samko, A.D. Marshall, P. Rosin, Selection of the optimal parameter value for the isomap algorithm, Pattern Recognition Letters 27 (9) (2006) 968–979.
[85] R. Smith, Optimal estimation of fractal dimension, in: Nonlinear Modeling and Forecasting, Addison Wesley, New York, 1992, pp. 115–135.
[86] D. Snyder, Random Point Processes, Wiley, New York, 1975.
[87] F. Takens, On the numerical determination of the dimension of an attractor, in: Dynamical Systems and Bifurcations, Proceedings Groningen 1984, Addison Wesley, New York, 1985, pp. 99–106.

[88] J.B. Tanenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (12) (2000) 2319–2323.
[89] J. Theiler, Lacunarity in a best estimator of fractal dimension, Physics Letters A133 (1988) 195–200.
[90] J. Theiler, Statistical precision of dimension estimators, Physical Review A41 (1990) 3038–3051.
[91] J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, J.D. Farmer, Testing for nonlinearity in time series: the method for surrogate date, Physica D58 (1992) 77–94.
[92] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society, SeriesB 58 (1996) 267–288.
[93] M.E. Tipping, C.M. Bishop, Probabilistic principal component analysis, Journal of the Royal Statistical Society Series B (61, Part 3) (1997) 611–622.
[94] G.V. Trunk, Statistical estimation of the intrinsic dimensionality of a noisy signal collection, IEEE Transaction on Computers 25 (1976) 165–171.
[95] M. Valle, A. Oganov, Crystal fingerprint space- a novel paradigm for studying crystal-structure sets, Acta Crystallographica Section A A66 (2010) 507–517.
[96] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, 1998.
[97] P.J. Verveer, R. Duin, An evaluation of intrinsic dimensionality estimators, IEEE Transaction on Pattern Analysis and Machine Intelligence 17 (1) (1995) 81–86.
[98] T. Villmann, J.C. Claussen, Magnification control in self-organizing maps and neural gas, Neural Computation 18 (2) (2000) 446–469.
[99] A.P.V.N. Vo, S. Oraintara, T.T. Nguyen, Statistical image modeling using von mises distribution in the complex directional wavelet domain, in: Proceedings of ISCAS 2008, 2008, pp. 2885–2888.
[100] X. Wang, J. Marron, Intrinsic dimension estimation of manifolds by incising balls, Electronic Journal of Statistics 2 (2008) 127–148.
[101] X. Yang, S. Michea, H. Zha, Conical dimension as an intrinsic dimension estimator and its applications, in: Proceedings of the 7th SIAM International Conference on Data Mining, 2007, pp. 169–179.
[102] L.-S. Young, Dimensions, entropy and lyapunov exponents, Ergodic Theory and Dinamical Systems 2 (1) (1918) 109–124.
[103] H. Zou, T. Hastie, R. Tibshirani, Sparse principal component analysis, Journal of Computational and Graphical Statistics 15 (2004) 262–286.