Mach Learn (2012) 89:37–65
DOI 10.1007/s10994-012-5294-7

Novel high intrinsic dimensionality estimators

A. Rozza · G. Lombardi · C. Ceruti · E. Casiraghi · P. Campadelli

Received: 29 October 2011 / Accepted: 1 May 2012 / Published online: 26 May 2012

Abstract Recently, a great deal of research work has been devoted to the development of algorithms to estimate the intrinsic dimensionality (id) of a given dataset, that is the minimum number of parameters needed to represent the data without information loss. id estimation is important for the following reasons: the capacity and the generalization capability of discriminant methods depend on it; id is a necessary information for any dimensionality reduction technique; in neural network design the number of hidden units in the encoding middle layer should be chosen according to the id of data; the id value is strongly related to the model order in a time series, that is crucial to obtain reliable time series predictions.

Although many estimation techniques have been proposed in the literature, most of them fail on noisy data, or compute underestimated values when the id is sufficiently high. In this paper, after reviewing some of the most important id estimators related to our work, we provide a theoretical motivation of the bias that causes the underestimation effect, and we present two id estimators based on the statistical properties of manifold neighborhoods, which have been developed in order to reduce this effect. We exhaustively evaluate the proposed techniques on synthetic and real datasets, by employing an objective evaluation measure to compare their performance with those achieved by state of the art algorithms; the results show that the proposed methods are promising, and produce reliable estimates also in the difficult case of datasets drawn from non-linearly embedded manifolds, characterized by high id.

Keywords Intrinsic dimensionality estimation · Dimensionality reduction · Manifold learning

1 Introduction

At the present, pattern recognition techniques applied to solve real life classification problems (such as face recognition, Abate et al. 2007, protein subcellular localization, Rozza et

Editors: Dimitrios Gunopulos, Donato Malerba, and Michalis Vazirgiannis.

A. Rozza (✉) · G. Lombardi · C. Ceruti · E. Casiraghi · P. Campadelli
Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano, Via Comelico 39-41, 20135 Milan, Italy
e-mail: rozza@dsi.unimi.it

al. 2010, and EEG classification, Rozza et al. 2010) must be able to deal with high dimensional feature vectors. Unfortunately, as it is formalized in Jollife (1961), due to the "curse of dimensionality" (Hughes effect) high dimensional data is difficult to work with for several reasons. At first, a high number of features can increase the noise, and hence the error; secondly, it is difficult to collect an amount of observations that is enough to obtain reliable classifiers since the theories at the basis of several classifiers become inconsistent when the number of observations is lower than, or comparable to, the data dimensionality (see Friedman 1989; Chen et al. 2000; Rozza et al. 2012); finally, as the amount of available features increases, the space needed to store the data becomes high, while the speed of the employed algorithms could be too low.

For all the above mentioned reasons, dimensionality reduction algorithms are often employed as the first preprocessing step of discriminating methods to improve classification performance. Obviously, dimensionality reduction is profitable when the reduced data are projected along the most characterizing dimensions. To identify them from the analysis of a given dataset $\boldsymbol{X}_N \equiv \{\boldsymbol{x}_i\}_{i=1}^N \subset \Re^D$, one useful information is the minimum number of parameters needed to represent the data without information loss, which is generally referred as *intrinsic dimensionality* (id) of the given dataset. To estimate the id of $\boldsymbol{X}_N$ the feature vectors are generally viewed as points constrained to lie on a low dimensional manifold $\boldsymbol{\mathcal{M}} \subseteq \Re^d$ embedded in a higher dimensional space $\Re^D$, where the dimensionality d of $\boldsymbol{\mathcal{M}}$ is the id value to be estimated. In more general terms, according to Fukunaga (1982), $\boldsymbol{X}_N$ is said to have id equal to $d \in \{1..D\}$ if its elements lie entirely within a d-dimensional subspace of $\Re^D$.

At the state of the art, a great deal of research work has been devoted to the development of id estimation algorithms because it is an important information required not only by dimensionality reduction techniques, but also by several applications, as summarized in the following.

At first, according to the statistical learning theory approach (Vapnik 1998) the capacity and the generalization capability of classifiers may depend on the id. More specifically, in the particular case of linear classifiers, where the data are drawn from a manifold embedded through an identical map, the Vapnik-Chervonenkis (VC) dimension of the separation hyperplane is $d+1$ where d is the id (see Vapnik 1998, pp. 156–158). Since the generalization error depends on the VC dimension, it follows that the generalization capability may depend on the id. Moreover, in Friedman et al. (2009) the authors mark that, in order to balance a classifier's generalization ability and its empirical error, the complexity of the classification model should also be related to the id of the available dataset. Besides, as mentioned above, the estimation of the id is a fundamental step of any dimensionality reduction technique; as an example, when using an autoassociative neural network (Kirby 1998) to perform a nonlinear feature extraction, the id can suggest a reasonable value for the number of hidden neurons. Moreover, as explained in Sect. 8.6.2 of Bishop (1995), in neural network design a reasonable number of hidden units in the encoding middle layer can be chosen by considering the dataset's id. This fact could be motivated by considering a network with a single hidden layer whose d hidden neurons have linear activation functions; in this case, it can be shown that the error function has a unique global minimum and that at this minimum the network performs a projection onto the subspace spanned by the first d principal components computed on the dataset. Finally, as it has been recently shown by Camastra et al. in Camastra and Vinciarelli (2002), Camastra and Filippone (2009), id estimation methods are used to evaluate the model order in a time series, that is crucial in order to make reliable time series predictions; this consideration is supported by the fact that the domain of attraction of a nonlinear dynamic system has a very complex geometric structure, and the studies on the geometry of the attraction domain are closely related to fractal geometry, and therefore to fractal dimension.

Unfortunately, even if a great deal of research work has been focused at the development of id estimators, and several interesting methods have been presented in the literature, to our knowledge only the method reported by Camastra and Vinciarelli (2002) has investigated the problem of high dimensional datasets characterized by a sufficiently high id (that is id ≥ 10) drawn from manifolds non-linearly embedded in higher dimensional spaces. This fact is also highlighted by the experiments reported in this paper showing that well-known state of the art techniques fail when dealing with non-linearly embedded manifolds characterized by high id.

However, we must note that interesting results have been proposed in Baraniuk and Wakin (2006), Hegde et al. (2007), Clarkson (2008), where the authors try to solve the problem of working with manifolds embedded in high dimensional spaces, where the too high number of dimensions D would make the problem untractable. More precisely, these works demonstrate that d-dimensional manifolds (of bounded curvature) can be projected onto random subspaces of (bounded) dimension by preserving geodesic distances, up to a small distortion. This immediately suggests that geometry-based id estimation techniques, as well as manifold learning algorithms, could be applied to the lower-dimensional, randomly projected version of the dataset. Considering that random linear projections are easily computed, these works greatly simplify the problem of high dimensional embedding spaces.

In Sect. 3.2 of Lombardi et al. (2011) we have proposed a family of id estimation methods, called "Minimum Neighbor Distance—Maximum Likelihood" ($\mathtt{MiND}_{\mathtt{ML}*}$) estimators, that exploit a maximum likelihood approach on the pdf related to the normalized nearest neighbor distances. The description of these algorithms is also reported in this paper since they are the first fundamental step of our research work aimed to the development of id estimators. Indeed, the critical evaluation of these approaches, by testing them on both synthetic and real datasets and by comparing their results to state of the art methods, have shown that, although promising results have been obtained, they are affected by a bias which produces underestimated id values when the dataset dimensionality becomes sufficiently high. Even if the comparison to state of the art algorithms proves that these methods are less affected by this underestimation effect, we have concentrated our efforts to understand its cause with the aim to avoid it. In Sect. 4 we show that this bias is caused by the fact that id estimators based on nearest neighbor distances are often founded on statistics derived by assuming that the amount of available data is unlimited; therefore, when a limited amount of samples are employed to estimate all the statistics which lay the foundations of these id estimators, unreliable results are obtained.

This considerations lead us to the development of other two id estimators, which are also described in this paper, called "Minimum Neighbor Distance—Kullback Leibler" ($\mathtt{MiND}_{\mathtt{KL}}$, Lombardi et al. 2011) estimator, and "Intrinsic Dimensionality Estimation Algorithm" (IDEA, Rozza et al. 2011), which are less affected by the bias, as it is shown by experiments on both synthetic and real datasets and by the comparison of the achieved results with those reported by state of the art algorithms. Note that in Sect. 7 we also propose an approach to reduce the time complexity of these two algorithms.

This paper is organized as follows: in Sect. 2 some of the most important state of the art id estimators are critically reviewed, highlighting both their advantages and drawbacks; in Sect. 3 the $\mathtt{MiND}_{\mathtt{ML}*}$ algorithms are described, while in Sect. 4 the cause of the bias is explained and formalized; Sect. 5 and Sect. 6 contain detailed descriptions of, respectively, $\mathtt{MiND}_{\mathtt{KL}}$ and IDEA estimators; in Sect. 7 approaches to reduce the time complexity of our algorithms are presented; in Sect. 8 experimental settings and results are reported; in Sect. 9 conclusions and future works are presented.

2 Related works

In this section we recall interesting id estimation methods which are related to our work. Note that a more detailed description of id estimators, up to year 2003, is also reported in the extensive survey of Camastra (2003).

The most cited example of id estimator is the Principal Component Analysis (PCA) (Jollife 1986), which is a well known technique that is often used as the first step of several machine learning methods to reduce the data dimensionality.

Given a dataset of observed points $\boldsymbol{p}_t \in \Re^D$, PCA computes their low dimensional representations $\boldsymbol{p}_x \in \Re^d$ by projecting the points $\boldsymbol{p}_t \in \Re^D$ on the d directions (also called principal components, PCs) of their maximum variance. More precisely, PCA computes the $D \times d$ projection matrix $\boldsymbol{W}$ (whose columns $\boldsymbol{w}_i$, $i = 1, \ldots, d$ are the PCs), so that the reduced points $\boldsymbol{p}_x$ are computed from the observed points as $\boldsymbol{p}_x = \boldsymbol{W}^T(\boldsymbol{p}_t - \bar{\boldsymbol{p}}_t)$ (being $\bar{\boldsymbol{p}}_t$ the mean of the observed points), while, given the reduced data, the reconstructed points are computed as $\tilde{\boldsymbol{p}}_t = \boldsymbol{W}\boldsymbol{p}_x + \bar{\boldsymbol{p}}_t$. The projection matrix $\boldsymbol{W}$ is computed so that the sum of square distances between the observed points $\boldsymbol{p}_t$ and the reconstructed ones $\tilde{\boldsymbol{p}}_t$ is minimized. To this aim, in Jollife (1986) it is proved that the PCs must be the eigenvectors corresponding to the highest eigenvalues of the data covariance matrix. Exploiting PCA, the intrinsic dimension d can be estimated by counting the number of retained PCs, that generally are the PCs whose corresponding (normalized) eigenvalue is higher than a threshold parameter. The main problem of using PCA for id estimation relies in the difficulty of choosing a proper value for the threshold.

To cope with this problem, in Fukunaga (1971) the author achieves more accurate results by applying a local PCA; this method works in small subregions of the dataset to estimate their local id. The id of the whole dataset is then determined by combining all the local ids. Unfortunately, the correct selection of the local regions and thresholds could be difficult, as shown by the empirical evaluations reported in Verveer and Duin (1995).

In Tipping and Bishop (1997), Tipping and Bishop noted that PCA and its variants are deterministic models lacking an associated probabilistic model for the observed data and a method for selecting the number of PCs to be retained, which is an estimate of the id. For this reason, in Tipping and Bishop (1997) they initially presented the Probabilistic PCA (PPCA), by reformulating PCA as the maximum likelihood solution of a specific latent variable model. More precisely, the authors consider a d-dimensional latent variable $\boldsymbol{x}$ (representing the reduced data) and set their prior distribution to be a zero mean Gaussian whose covariance matrix is a d-dimensional identity matrix (that is $\mathcal{N}(\boldsymbol{x}|\boldsymbol{0}, \boldsymbol{I}_d)$). The D-dimensional observed variable $\boldsymbol{t}$, which represents the observed data, is then defined as: $\boldsymbol{t} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{\mu} + \boldsymbol{\epsilon}$, that is a linear transformation of the latent variable $\boldsymbol{x}$ where $\boldsymbol{W}$ is a $D \times d$ parameter representing the projection matrix (containing the PCs), $\boldsymbol{\mu}$ is a D-dimensional vector, and $\boldsymbol{\epsilon}$ is a zero-mean Gaussian distributed vector with covariance $\sigma^2 \boldsymbol{I}_D$ representing noise. Therefore, the marginal distribution of the observed variable $\boldsymbol{t}$, which represents the probabilistic formulation of PCA, is a constrained Gaussian distribution governed by three parameters, which are $\boldsymbol{W}$, $\boldsymbol{\mu}$, and σ. The maximum likelihood solution for these parameters allows to project the observed dataset on the reduced d-dimensional space and therefore represents the solution computed by means of PPCA.

Although PPCA has been successfully applied to problems in data compression, density estimation and data visualization, and has been extended to both mixture and hierarchical mixture models, as noted in Bishop (1998), it still does not provide any mechanism for estimating the best value of the latent space dimensionality d, that is the id. For this reason, Bishop (1998) further extends the PPCA model by defining a Bayesian treatment of PCA (called Bayesian PCA or BPCA). To this aim the author introduces a prior distribution over the three parameters $\boldsymbol{W}$, $\boldsymbol{\mu}$, and σ, which allows to formulate the posterior over the dataset (thanks to the Bayes formula) and hence the predictive density by marginalizing over the three parameters. To automatically determine an effective dimensionality d for the latent variable $\boldsymbol{x}$, the author further introduces a "hierarchical" prior $p(\boldsymbol{W}|\boldsymbol{\alpha})$ over the parameter $\boldsymbol{W}$ that is governed by a q-dimensional vector of hyper-parameters $\boldsymbol{\alpha} = \alpha_1, \dots, \alpha_q$, where the value of q is usually set to its highest value, that is $q = D - 1$. This prior, which is motivated by the framework of "automatic relevance determination" (ARD) (MacKay 1995), is formulated as a product of q conditional Gaussian distributions, where the ith Gaussian depends only on α_i and $\boldsymbol{w}_i$ (where w_i is the ith column of the matrix $\boldsymbol{W}$, that is the ith PC), so that each α_i controls the inverse "relevance" of one PC. To make use of this model the authors exploit a local Gaussian approximation to estimate the posterior distribution of $\boldsymbol{W}$, which must be marginalized to solve the problem. Combining this procedure with maximum likelihood to determine the values of the α_i, the authors note that the vectors $\boldsymbol{w}_i$ for which there is insufficient support from the data will be driven to zero, with the corresponding $\alpha_i \longrightarrow \infty$, so that unused dimensions are switched off completely; the id is then defined as the number of PCs whose "relevance" value remains non-zero.

Although BPCA is a theoretically founded approach for estimating the id, it assumes that data are Gaussian distributed and it is therefore not suited for those types of practical observations that cannot be expressed as real-valued vectors, such as binary or integers values. To cope with this problem an interesting and recent approach has been presented in Li and Tao (2010), where the authors propose a Simple Exponential Family PCA (SePCA), which is a generalized family of probabilistic principal component analyzers. This algorithm essentially extends BPCA by substituting the Gaussian distributions with exponential family distributions. In other words, given the observed data, the exponential family distributions define the likelihood functions of the latent variables, that are the PCs and the low-dimensional representations. Such likelihood functions link real-valued latent variables and observations of any kind, such as integers or binary values.

Other two interesting works improving PCA are those reported in Zou et al. (2004), Guan and Dy (2009). More precisely, rather than using a given threshold as in PCA, in Zou et al. (2004) the authors force the sparsity of the projection matrix $\boldsymbol{W}$ to achieve an automatic selection of meaningful principal components; this method, called Sparse Principal Component Analysis (SPCA), is obtained by reformulating PCA as a regression optimization problem and imposing the lasso constraint on the regression coefficients. The main drawback of this approach is due to the fact that the weight of the constraint must be manually set. To overcome this limitation, in Guan and Dy (2009) the authors introduce a probabilistic Bayesian formulation of SPCA, using a different prior to achieve sparsity. In this way, the proposed Sparse Probability Principal Component Analysis (SPPCA), can automatically learn the hyperparameter related to the weight of the constraint of SPCA through a type II maximum likelihood.

At the state of the art the above mentioned PCA-based methods are generally classified as projection methods (Camastra 2003; Levina and Bickel 2005) since they search for the best subspace to project the data. Unfortunately, although all these methods have shown to be a valuable tool for exploratory data analysis, where the user might plot the eigenvalues and manually look for a clear-cut boundary, they cannot provide reliable estimates of intrinsic dimensions since they are too sensitive to noise and parameter settings (Levina and Bickel 2005).

Other id estimators, such as Locally Linear Embedding (LLE, Roweis and Saul 2000), Nearest Neighbor estimator (Pettis et al. 1979), and Tensor Voting Framework (TVF, Mordohai and Medioni 2010; Lombardi et al. 2009), which are generally classified as geometric methods (Levina and Bickel 2005), exploit the intrinsic geometry of the dataset and are most often based on fractal dimensions or nearest neighbor distances within the data. Most of these methods consider hyperspheres with sufficiently small radius r and centered on the points in the dataset, and they estimate some statistics related to the distances of neighboring points included into the hypersphere; these statistics are expressed as functions of the intrinsic dimensionality of the manifold from which the points have been randomly drawn.

Perhaps the most popular fractal dimension estimators is the Correlation Dimension (CD) algorithm (Grassberger and Procaccia 1983; Camastra and Vinciarelli 2002); it is based on the assumption that the volume of a d-dimensional set scales with its size r as r^d, which implies that also the number of samples covered by a hypersphere with radius r grows proportionally to r^d. Since the performance of the CD estimator is much affected by the choice of the scale r, in Hein (2005) the authors suggest an estimator (which we will be referred as Hein in the following) based on the asymptotics of a smoothed version of the CD estimate. Another interesting approach is proposed in Farahmand et al. (2007), where the author presented an algorithm to estimate the id of a manifold in a small neighborhood of a selected point, and they analyzed its finite-sample convergence properties.

Another well known fractal based approach, is the Packing Number technique (Kégl 2002) that exploits the r-packing number $M(r)$ of the dataset $\boldsymbol{X}_N \subset \boldsymbol{\mathcal{S}}$, where $\boldsymbol{\mathcal{S}}$ is a metric space with distance metric $\delta(\cdot,\cdot)$. More precisely, $\boldsymbol{X}_N$ is said to be r-separated if $\forall\, \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{X}_N, \boldsymbol{x} \neq \boldsymbol{y} \Rightarrow \delta(\boldsymbol{x}, \boldsymbol{y}) \geq r$, and $M(r)$ is the maximum cardinality of an r-separated subset of $\boldsymbol{X}_N$. Given this definition, the authors demonstrate that the id of $\boldsymbol{X}_N$ can be found by approximating the limit:

$$d = -\lim_{r\to 0} \frac{\log M(r)}{\log r} \quad \text{with } \hat{d} = -\frac{\log(M(r_2) - M(r_1))}{\log(r_2 - r_1)}$$

where $r_2 > r_1$ are two radiuses to be set as parameters.

Another interesting technique, based on the analysis of point neighborhoods, is the Maximum Likelihood Estimator (MLE) (Levina and Bickel 2005) that applies the principle of maximum likelihood to the distances between close neighbors, and derives the estimator by a Poisson process approximation. More precisely, calling k the number of neighbors, $\boldsymbol{x}_i$ the ith point, and $T_k(\boldsymbol{x}_i)$ the radius of the smallest sphere centered in $\boldsymbol{x}_i$ containing exactly k neighbors, the local intrinsic dimension is estimated as:

$$\hat{d}(\boldsymbol{x}_i) = \left(\frac{1}{k} \sum_{j=1}^{k} \log \frac{T_{k+1}(\boldsymbol{x}_i)}{T_j(\boldsymbol{x}_i)} \right)^{-1}$$

In Costa and Hero (2004) the authors propose an algorithm that exploits entropic graphs to estimate both the id of a manifold, and the intrinsic entropy of the manifold random samples. This technique is based on the observation that the length function of such graphs, that is the sum of arc weights on the minimal graph that spans all the points in the dataset, is strongly dependent on d. The authors test their method by adopting either the geodesic minimal spanning tree (GMST, Costa and Hero 2004), where the arc weights are the geodetic distances computed through the ISOMAP (Tenenbaum et al. 2000) algorithm, or the kNN-graph (kNNG, Costa and Hero 2004), where the arc weights are based on the Euclidean distances, thus requiring a lower computational cost.

We note that most of the neighborhood based estimators generally underestimate d when its value is sufficiently high, and to our knowledge the only work that addresses this problem has been proposed in Camastra and Vinciarelli (2002). In this work, Camastra et al.

propose an empirical correction procedure of the computed id based on the estimation of the error obtained on synthetically produced datasets of known dimensionality (hypercubes). More precisely, after generating T datasets characterized by incremental id values ($d_i = 1, \dots, T$), the authors apply the CD algorithm (Grassberger and Procaccia 1983) to compute the estimated id ($\hat{d}_i$) of each dataset. Fitting the points ($d_i, \hat{d}_i$) the authors obtain the "correction curve" that allows to correct the estimated value on datasets whose id is unknown.

Another problem affecting most of the id estimators is due to their high computational complexity that causes problems when datasets of high cardinality must be processed.

3 The MiND$_{\text{ML}*}$ id estimators

In this section we describe a family of id estimators, called "Minimum Neighbor Distance—Maximum Likelihood" (MiND$_{\text{ML}*}$) estimators, since they exploit a maximum likelihood approach on the pdf related to the normalized nearest neighbor distances. For clarity of presentation, in Sect. 3.1 we firstly present the basic theories of these estimators; according to the derived statistics, in Sect. 3.2 we describe a family of maximum likelihood id estimators and its variants (MiND$_{\text{ML}*}$).

3.1 Theoretical results

Considering a manifold $\mathcal{M} \equiv \Re^d$ embedded in a higher dimensional space $\Re^D$ through a locally isometric non-linear smooth map $\psi : \Re^d \to \Re^D$, to estimate the id of $\mathcal{M}$ we need to identify a "mathematical object" depending only on d that can be estimated by means of points drawn from the embedded manifold.

To theoretically face this problem, in this section we firstly analyse in detail a specific case concerning a simple manifold, and then we generalize the assumptions step by step.

At first, consider the manifold $\mathcal{M} \equiv \mathcal{B}_d(\mathbf{0}_d, 1) \subset \Re^d$, where $\mathcal{B}_d(\mathbf{0}_d, 1)$ is the unit hypersphere centered in the origin and uniformly sampled; and assume the embedding map ψ to be the identity map. Considering k points $\{z_i\}_{i=1}^k$ uniformly drawn from $\mathcal{B}_d(\mathbf{0}_d, 1)$, our aim is to estimate the id of $\mathcal{M}$ by means of these samples only. To face this problem, we exploit the concentration of norms that is dimensionality-dependent (Lee and Veleysen 2007). For this reason, we have chosen to identify the closed form of the pdf related to the minimum distance between the k points and the hypersphere center $\mathbf{0}_d$; furthermore, we show that this "mathematical object" could be used to estimate the id also on non-linearly embedded smooth manifolds whose points are drawn by means of a general smooth pdf.

To this aim, considering a generic point z_i, $i \in 1, \dots, k$, we denote with $p(r)$ the pdf for the event $\|z_i\| = r$ ($r \in [0, 1]$) where $\| \cdot \|$ is the L_2 norm operator, and with $P(\breve{r} < r)$ the probability for the event $\|z_i\| < r$. Being z_i uniformly drawn it is possible to evaluate $P(\breve{r} < r)$ by means of hypersphere volume ratios, that is by computing the volume, V_r, of a d-dimensional hypersphere of radius r, and normalizing it by the volume, V_1, of the unit d-dimensional hypersphere. V_r is computed as follows:

$$V_r = r^d \frac{\pi^{d/2}}{\Gamma(\frac{d}{2} + 1)} = r^d V_1$$

where $\Gamma(\cdot)$ is the Gamma function. This yields $P(\breve{r} < r) = \frac{V_r}{V_1} = r^d$; moreover, being $P(\breve{r} < r)$ the cumulative density function (cdf) related to the pdf $p(r)$, it is $p(r) = \partial(\frac{V_r}{V_1})/\partial r = \frac{1}{V_1} d r^{d-1}$.

We further note that the pdf $g(r;d,k)$ related to the event $\min_{i\in\{1,\dots,k\}}\|z_i\|=r$ (i.e. the pdf for the event "the minimum distance between the points $\{z_i\}_{i=1}^k$ and the hypersphere center $\mathbf{0}_d$ equals r") is proportional to the probability of drawing one point with distance r multiplied by that of drawing $k-1$ points with distance $\breve{r}>r$, that is:

$$\begin{aligned} g(r;d,k) \propto \breve{g}(r;d,k) &= p(r)\big(1-P(\breve{r}<r)\big)^{k-1} \\ &= \frac{\partial(\frac{V_r}{V_1})}{\partial r}\left(1-\frac{V_r}{V_1}\right)^{k-1} = \frac{1}{V_1}dr^{d-1}\big(1-r^d\big)^{k-1} \end{aligned}$$

Normalizing by $\int_0^1 \breve{g}(r;d,k)dr=(V_1k)^{-1}$ we finally get:

$$g(r;k,d)=\frac{\breve{g}(r;d,k)}{\int_0^1\breve{g}(r;d,k)dr}=kdr^{d-1}\big(1-r^d\big)^{k-1} \tag{1}$$

Notice that Eq. (1) holds only if we assume that the manifold is the unit radius hypersphere. Nevertheless, choosing a d-dimensional open ball $\mathcal{B}_d(\boldsymbol{c},\epsilon)$ with center $\boldsymbol{c}\in\mathcal{M}$ and radius $\epsilon>0$, as long as $\mathcal{M}$ is embedded in $\Re^D$ through a non-linear smooth map ψ that preserves distances in $\mathcal{B}_d$, and z is uniformly drawn from $\mathcal{B}_d$, the quantities $\frac{1}{\epsilon}\|\psi(\boldsymbol{c})-\psi(z)\|=\frac{1}{\epsilon}\|\boldsymbol{c}-z\|$ are distributed as the norms of points uniformly drawn from $\mathcal{B}_d(\mathbf{0}_d,1)$. This fact ensures that Eq. (1) holds in $\mathcal{B}_d(\boldsymbol{c},\epsilon)$ for $r=\frac{1}{\epsilon}\|\boldsymbol{c}-z\|$.

To further generalize our theoretical results, we consider a locally isometric smooth map $\psi:\mathcal{M}\to\Re^D$, and samples drawn from $\mathcal{M}\equiv\Re^d$ by means of a non-uniform smooth pdf $f:\mathcal{M}\to\Re^+$. Notice that, being ψ a local isometry, it induces a distance function $\delta_\psi(\cdot,\cdot)$ representing the metric on $\psi(\mathcal{M})$. Under these assumptions Eq. (1) does not represent the correct pdf of the distances. However, without loss of generality, we consider $\boldsymbol{c}=\mathbf{0}_d\in\Re^d$ and $\psi(\boldsymbol{c})=\mathbf{0}_D\in\Re^D$, and we show that any smooth pdf f is locally uniform where the probability is not zero. To this aim, assuming $f(\mathbf{0}_d)>0$ and $z\in\Re^d$, we denote with f_ϵ the pdf obtained by setting $f_\epsilon(z)=0$ when $\|z\|>1$, and $f_\epsilon(z)\propto f(\epsilon z)$ when $\|z\|\le 1$. More precisely, denoting with $\chi_{\mathcal{B}_d(\mathbf{0}_d,1)}$ the indicator function on the ball $\mathcal{B}_d(\mathbf{0}_d,1)$, we obtain:

$$f_\epsilon(z)=\frac{f(\epsilon z)\chi_{\mathcal{B}_d(\mathbf{0}_d,1)}(z)}{\int_{\boldsymbol{t}\in\mathcal{B}_d(\mathbf{0}_d,1)}f(\epsilon\boldsymbol{t})d\boldsymbol{t}} \tag{2}$$

Theorem 1 *Given* $\{\epsilon_i\}\to 0^+$, *Eq.* (2) *describes a sequence of* pdf *having the unit* d*-dimensional ball as support*; *such sequence converges uniformly to the uniform distribution* $\mathbf{B}_d$ *in the ball* $\mathcal{B}_d(\mathbf{0}_d,1)$.

Proof Evaluating the limit for $\epsilon\to 0^+$ of the distance between f_ϵ and $\mathbf{B}_d$ in the supremum norm we get:

$$\begin{aligned} \lim_{\epsilon\to 0^+}\big\|f_\epsilon(z)-\mathbf{B}_d(z)\big\|_{\sup} &= \lim_{\epsilon\to 0^+}\left\|\frac{f(\epsilon z)\chi_{\mathcal{B}_d(\mathbf{0}_d,1)}}{\int_{\mathcal{B}_d(\mathbf{0}_d,1)}f(\epsilon\boldsymbol{t})d\boldsymbol{t}}-\frac{\chi_{\mathcal{B}_d(\mathbf{0}_d,1)}}{\int_{\mathcal{B}_d(\mathbf{0}_d,1)}d\boldsymbol{t}}\right\|_{\sup} \\ \{\text{just notation}\} &= \lim_{\epsilon\to 0^+}\left\|\frac{f(\epsilon z)}{\int_{\mathcal{B}_d(\mathbf{0}_d,1)}f(\epsilon\boldsymbol{t})d\boldsymbol{t}}-\frac{1}{\int_{\mathcal{B}_d(\mathbf{0}_d,1)}d\boldsymbol{t}}\right\|_{\sup\mathcal{B}_d(\mathbf{0}_d,1)} \\ \left\{\text{setting } V=\int_{\mathcal{B}_d(\mathbf{0}_d,1)}d\boldsymbol{t}\right\} &= \lim_{\epsilon\to 0^+}\left\|\frac{Vf(\epsilon z)-\int_{\mathcal{B}_d(\mathbf{0}_d,1)}f(\epsilon\boldsymbol{t})d\boldsymbol{t}}{V\int_{\mathcal{B}_d(\mathbf{0}_d,1)}f(\epsilon\boldsymbol{t})d\boldsymbol{t}}\right\|_{\sup\mathcal{B}_d(\mathbf{0}_d,1)} \end{aligned}$$

$$\left\{0 < \lim_{\epsilon\to 0^+} V \int_{\mathcal{B}_d(\mathbf{0}_d,1)} f(\epsilon \boldsymbol{t})d\boldsymbol{t} < \infty\right\} = \lim_{\epsilon\to 0^+} \left\| V f(\epsilon \boldsymbol{z}) - \int_{\mathcal{B}_d(\mathbf{0}_d,1)} f(\epsilon \boldsymbol{t})d\boldsymbol{t} \right\|_{\sup \mathcal{B}_d(\mathbf{0}_d,1)}$$

Defining:

$$min(\epsilon) = \min_{\mathcal{B}_d(\mathbf{0}_d,1)} f(\epsilon \boldsymbol{z}) \qquad max(\epsilon) = \max_{\mathcal{B}_d(\mathbf{0}_d,1)} f(\epsilon \boldsymbol{z})$$

and noting that $min(\epsilon) > 0$ definitely since $f(\mathbf{0}_d) > 0$, we have:

$$V \cdot min(\epsilon) \le V f(\epsilon \boldsymbol{z}) \le V \cdot max(\epsilon)$$

$$V \cdot min(\epsilon) \le \int_{\mathcal{B}_d(\mathbf{0},1)} f(\epsilon \boldsymbol{t})d\boldsymbol{t} \le V \cdot max(\epsilon)$$

thus their difference is bounded by $V(max(\epsilon) - min(\epsilon)) \xrightarrow[\epsilon\to 0^+]{} 0^+$. □

Theorem 1 proves that the convergence of f_ϵ to $\mathbf{B}_d$ is uniform, so that when $\epsilon \to 0^+$ the pdf related to the geodetic distances $\frac{1}{\epsilon}\delta_\psi(\psi(\boldsymbol{c}), \psi(\boldsymbol{z})) = \frac{1}{\epsilon}\|\boldsymbol{c} - \boldsymbol{z}\|$ converges to the pdf g reported in Eq. (1).

Finally, we would like to note that the embedding map ψ is used only in the theoretical argumentation reported above and it should not be chosen, since neither a map nor a kernel function are required by our algorithms (see Sects. 3.2, 6, and 5).

3.2 Maximum likelihood approaches

In this section we show how the theoretical results presented in Sect. 3.1 can be exploited to estimate the id of a given dataset.

More precisely, we consider a manifold $\mathcal{M} \equiv \Re^d$ embedded in a higher dimensional space $\Re^D$ through a locally isometric non-linear smooth map $\psi : \mathcal{M} \to \Re^D$, and a sample set $\boldsymbol{X}_N = \{\boldsymbol{x}_i\}_{i=1}^N = \{\psi(\boldsymbol{z}_i)\}_{i=1}^N \subset \Re^D$, where $\boldsymbol{z}_i$ are independent identically distributed points drawn from $\mathcal{M}$ according to a non-uniform smooth pdf $f : \mathcal{M} \to \Re^+$.

To estimate the id of this dataset, for each point $\boldsymbol{x}_i \in \boldsymbol{X}_N$ we find the set of $k+1$ $(1 \le k \le N-1)$ nearest neighbors $\bar{\boldsymbol{X}}_{k+1} = \bar{\boldsymbol{X}}_{k+1}(\boldsymbol{x}_i) = \{\boldsymbol{x}_j\}_{j=1}^{k+1} \subset \boldsymbol{X}_N$. Calling $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_{k+1}(\boldsymbol{x}_i) \in \bar{\boldsymbol{X}}_{k+1}$ the most distant point from $\boldsymbol{x}_i$, we calculate the distance between $\boldsymbol{x}_i$ and the nearest neighbor in $\bar{\boldsymbol{X}}_{k+1}$ and we normalize it by means of the distance between $\boldsymbol{x}_i$ and $\hat{\boldsymbol{x}}$. More precisely, we have:

$$\rho(\boldsymbol{x}_i) = \min_{\boldsymbol{x}_j \in \bar{\boldsymbol{X}}_{k+1}} \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\|\boldsymbol{x}_i - \hat{\boldsymbol{x}}\|} \tag{3}$$

Theorem 4 in Costa and Hero (2005) ensures that geodetic distances in the infinitesimal ball converge to Euclidean distances with probability 1; moreover, recalling the result reported in Theorem 1, it is possible to notice that, for $\boldsymbol{x}_i \neq \hat{\boldsymbol{x}}$, the quantities $\rho(\boldsymbol{x}_i)$ are samples drawn from the pdf reported in Eq. (1), where the parameter k is known and the parameter d must be estimated. A simple approach for the estimation of d is the maximization of the log-likelihood function:

$$\begin{aligned} ll(d) &= \sum_{\boldsymbol{x}_i \in \boldsymbol{X}_N} \log g(\boldsymbol{x}_i; k, d) \\ &= N \log k + N \log d + (d-1) \sum_{\boldsymbol{x}_i \in \boldsymbol{X}_N} \log \rho(\boldsymbol{x}_i) + (k-1) \sum_{\boldsymbol{x}_i \in \boldsymbol{X}_N} \log\big(1 - \rho^d(\boldsymbol{x}_i)\big) \end{aligned} \tag{4}$$

To select an integer value in $\hat{d} \in \{1..D\}$ as the estimated id, it suffices to evaluate $\hat{d} = \arg\max_{d\in\{1..D\}} ll(d)$. We call this estimator $\mathtt{MiND}_{\mathtt{MLi}}$; its time complexity is $O(D^2N^2)$

On the other hand, since a real value is often required as a fractal id estimation, we also developed a variant of the $\mathtt{MiND}_{\mathtt{MLi}}$ algorithms, called $\mathtt{MiND}_{\mathtt{MLk}}$ algorithms, that finds the maximal value of $ll(d)$ in $[1, D]$. To this aim, we compute the first derivative of $ll(d)$ and we determine the solutions of $\frac{\partial ll}{\partial d} = 0$, thus obtaining:

$$\frac{N}{d} + \sum_{\boldsymbol{x}_i \in \boldsymbol{X}_N} \left(\log \rho(\boldsymbol{x}_i) - (k-1) \frac{\rho^d(\boldsymbol{x}_i) \log \rho(\boldsymbol{x}_i)}{1-\rho^d(\boldsymbol{x}_i)} \right) = 0 \tag{5}$$

We recall that the well-known MLE technique adopts a similar derivation since it extracts distance information from all the first k nearest neighbors. We note that, in the particular case $k = 1$, the solution of Eq. (5) is:

$$\hat{d} = -\left(\frac{1}{N} \sum_{\boldsymbol{x}_i \in \boldsymbol{X}_N} \log \rho(\boldsymbol{x}_i) \right)^{-1} \tag{6}$$

that is exactly the MLE estimator proposed in MacKay and Ghahramani (2005) when $k = 1$; this estimator is $\mathtt{MiND}_{\mathtt{ML1}}$ and its time complexity is $O(DN^2)$.

For $k > 1$ we numerically solve the following optimization problem:

$$\hat{d} = \underset{0<d\le D}{\arg\max}\, ll(d) \tag{7}$$

To solve this maximization problem we employed the constrained optimization method proposed in Coleman and Li (1996) with the initial (integer) value $d_0 = \arg\max_{d\in\{1..D\}} ll(d)$, thus obtaining a fractal dimensionality estimation.

Notice that the second derivative of Eq. (4) is:

$$\frac{\partial^2 ll}{\partial d^2} = -\frac{N}{d^2} - (k-1) \sum_{\boldsymbol{x}_i \in \boldsymbol{X}_N} \frac{\rho^d(\boldsymbol{x}_i) \log^2 \rho(\boldsymbol{x}_i)}{(1-\rho^d(\boldsymbol{x}_i))^2} \tag{8}$$

Considering that $k \ge 1$, $0 < \rho(\boldsymbol{x}_i) < 1$, and $d > 0$, $\frac{\partial^2 ll}{\partial d^2}$ is always negative, so that the optimization problem reported in Eq. (7) is convex in the domain of interest.

The time complexity of the $\mathtt{MiND}_{\mathtt{MLk}}$ algorithms is $O(D^2N^2)$.

4 Considerations about estimators based on kNN normalized distances

Though promising results have been reported in Lombardi et al. (2011), the experimental results reported by the authors on synthetic and real datasets of known id showed that all the $\mathtt{MiND}_{\mathtt{ML}*}$ algorithms are affected by a bias whose effect is to produce underestimated id values. Unfortunately, when the id value is low the underestimation effect can be ignored, but when the value of the id increases the gap between the underestimated id value and the real one grows dramatically.

This bias is due to the fact that the theoretical results reported in Sect. 3.1, which assume that an unlimited amount of samples are available, need to be translated into practice by employing the available high dimensional observations to compute approximated values of all the used statistics. Unfortunately, being the number of available samples limited, the computed approximations are unreliable estimates.

We would like to further note that this problem affects not only the MiND$_{\mathrm{ML}*}$ algorithms but also most of the geometric id estimators whose underlying theory is based on the analysis of nearest neighbor distances, since all these techniques are based on the assumption that nearest neighbors have the same behavior of points uniformly drawn from a d-dimensional hypersphere.

In this section we show that this basic assumption, on which the MiND$_{\mathrm{ML}*}$ algorithms are founded, cease to be true when the value of the id becomes sufficiently high (e.g. when the id is equal or higher than 10) and the number of samples is limited, thus producing unacceptable underestimations.

To this aim, we recall that in Sect. 3.1 we consider k points that are supposed to be uniformly drawn from a d-dimensional hypersphere whose center is in the origin $\mathbf{0}_d$, and we show that the pdf related to the normalized nearest neighbor distances is depending on the id value d. More precisely, given a generic point z_i ($i \in \{1, \dots, k\}$) uniformly drawn from the unit d-dimensional hypersphere centered in $\mathbf{0}_d$, in Sect. 3.1 we show that the probability for the event $\|z_i\| < r$ is $P(\breve{r} < r) = \frac{V_r}{V_1} = r^d$.

When these statistics must be applied by exploiting the samples in the dataset $\boldsymbol{X}_N$, the MiND$_{\mathrm{ML}*}$ algorithms (see Sect. 3.2) iteratively select each sample point $\boldsymbol{x}_i \in \boldsymbol{X}_N$ as an hypersphere center, and consider its k-nearest neighbors as samples uniformly drawn from the unit hypersphere centered in $\boldsymbol{x}_i$. Theorem 1 and Theorem 4 in Costa and Hero (2005) guarantee that this assumption is true only when $k \to \infty$, $N \to \infty$, $\frac{k}{N} \to 0$, and these requirements are often translated into practice by several id estimators by requiring both the dataset cardinality and the parameter k of nearest neighbors to be considered to be sufficiently high. However, as we will show in the following, the value of k required to get an acceptable approximation grows exponentially with the dimensionality d.

To formally show this fact, considering a sample $\boldsymbol{x}_i \in \boldsymbol{X}_N$ and its k-nearest neighbors, we can prove that $\boldsymbol{x}_i$ has a very low probability to be located close to the center of the hypersphere from which its k-nearest neighbors are supposed to be uniformly drawn. To this aim we firstly recall that, considering the point $\boldsymbol{x}_i$ and the normalized distances between $\boldsymbol{x}_i$ and its k nearest neighbors, if $N \to \infty$, $k \to \infty$ and $\frac{k}{N} \to 0$, then Theorem 1 applies and the pdf associated to the computed distances converges to that associated to the distances of points uniformly drawn from the unit hypersphere. At this point we need to evaluate the speed of this convergence by computing the probability of $\boldsymbol{x}_i$ to be at a given distance from the center of the hypersphere given the number of its k nearest neighbors.

To this aim, calling $h(\tilde{k}; r, d)$ the probability that the $\tilde{k}$th sampled point is the first point drawn at a distance $\breve{r} < r$ from the hypersphere center, we get the following pdf:

$$h(\tilde{k}; r, d) = \left(1 - r^d\right)^{\tilde{k}-1} r^d$$

A first insight is provided by the consideration that $h(\tilde{k}; r, d)$ is an exponential probability function depending on the parameter d. This means that, having fixed the value of $\tilde{k}$ and r, as d grows the probability to get the $\tilde{k}$th sample near the center decreases. Similarly, having fixed the value of $\tilde{k}$ and d, as r becomes smaller the probability to get the $\tilde{k}$th sample at a distance $\breve{r} < r$ from the center decreases. A further consideration is raised by the observation that the expectation of $h(\tilde{k}; r, d)$ is $(\frac{1}{r})^d$; this highlights the fact that, on average, the number $\tilde{k}$ of neighbors required to finally get the $\tilde{k}$th point at distance $\breve{r} < r$ from the hypersphere center grows exponentially with d.

Moreover, we note that the cumulative distribution related to $h(\tilde{k}; r, d)$, that is the probability to draw $\tilde{k}$ samples such that one of them is a point at distance $\check{r} < r$ from the hypersphere center, is:

$$H(\tilde{k}; r, d) = \sum_{i=0}^{\tilde{k}} h(i; r, d) = 1 - \left(1 - r^d\right)^{\tilde{k}} \tag{9}$$

Exploiting Eq. (9), and fixing the values of r and $\tilde{k}$ (that is $r = 0.1$ and $\tilde{k} = 30$), and increasing the id value (that is $d = \{2, 5, 10, 50\}$) we see that the value of $H(30; 0.1, d)$ becomes lower and lower:

$$\begin{aligned} H(30; 0.1, 2) &\approx 2.603e{-}01 \\ H(30; 0.1, 5) &\approx 2.9996e{-}004 \\ H(30; 0.1, 10) &\approx 3.0000e{-}009 \\ H(30; 0.1, 50) &\approx 0 \end{aligned}$$

On the other hand, the value of H increases when the values of $\tilde{k}$ and d are fixed, and the value of r is increased. This means that, as the id increases, all the sampled points are far from the center,[1] which essentially means that there is no point that could be considered as the center of the hypersphere from which its $k = \tilde{k} - 1$ nearest neighbors are supposed to be uniformly drawn.

To further support our conjecture, solving Eq. (9) to compute the number $\tilde{k}$ of nearest neighbors required to sample a point in $\Re^d$ at a distance $\check{r} < r$ with probability H; we obtain:

$$\tilde{k}(r, H, d) = \frac{\log(1 - H)}{\log(1 - r^d)}$$

Evaluating this function with fixed values of r and H (that is $r = 0.1$, $H = 0.9$), and increasing the id value ($d = \{2, 5, 10\}$), we obtain increasingly high values of the required number $\tilde{k}$ of nearest neighbors:

$$\begin{aligned} \tilde{k}(0.1, 0.9, 2) &\approx 229 \\ \tilde{k}(0.1, 0.9, 5) &\approx 230257 \\ \tilde{k}(0.1, 0.9, 10) &\approx 23025849023 \end{aligned} \tag{10}$$

These results show that, given a sample point $\boldsymbol{x}_i$, it can be assumed to be the center of the hypersphere from which its k-nearest neighbors are supposed to be uniformly drawn only when the id value is low and the available number of nearest neighbors is sufficiently high.

The discussion reported in this section shows that geometric id estimators based on the hypothesis that the normalized KNN distances resemble the distances between nearest neighbors uniformly sampled from the unit hypersphere, have a well founded theory but lack a proper statistical model. According to the results reported in this section, we believe that an id estimator exploiting the normalized KNN distances should adopt a different probability

[1] This consideration is similar to that described by the "edge effect", and reported in Verveer and Duin (1995). More precisely, the authors prove that the fraction between the points on (or close to) the edge of the manifold, and the other points (inside the manifold) increases in probability when the dimensionality increases.

distribution that, to our knowledge, has not been formalized yet. For these reasons, in Sect. 5 we investigate a novel id estimation approach, which estimates the missing pdf by means of simulation, and then compare it with the pdf obtained by the data under analysis. Note that, in Sect. 6 we propose a different id estimator that reduces the underestimation effect by means of an asymptotic correction technique.

5 $\mathtt{MiND}_{\mathtt{KL}}$: a pdf comparison approach

In Sect. 3.2 we have presented maximum likelihood estimators for the parameter d (id) in the pdf reported in Eq. (1). Notice that, once k is fixed, Eq. (1) represents a finite family of D pdfs for all the parameter values $1 \le d \le D$. Exploiting both this fact, and trying to avoid the biasing which affects the $\mathtt{MiND}_{\mathtt{ML*}}$ algorithms (see Sect. 4), we propose another approach for the estimation of the missing parameter d based on the comparison between the D theoretical pdfs and a density function estimated by means of the given data.

Consider $\mathcal{M}$ to be a d-dimensional hypersphere embedded in the Euclidean space $\Re^D$; moreover, denote with $\hat{g}(r;k)$ an estimation of $g(r;k,d)$ computed by solely using the sample data points and therefore independent from d. The estimate $\hat{d}$ is computed by choosing the dimensionality which minimizes the Kullback-Leibler divergence between g and $\hat{g}$:

$$\hat{d} = \underset{1\le d\le D}{\arg\min} \int_0^1 \hat{g}(r;k)\log\left(\frac{\hat{g}(r;k)}{g(r;k,d)}\right)dr \tag{11}$$

The function $\hat{g}$ can be obtained by means of a set of sample data points as a parametric model; nevertheless, as shown in Eckmann and Ruelle (1992) and in Sect. 4, the number of sample points required to perform dimensionality estimation grows exponentially with the value of the id. For this reason, when the dimensionality is sufficiently high, the number of sample points practically available is insufficient to compute an acceptable estimation.

In Sect. 2 we recall that to our knowledge only one approach has been proposed in literature to address this problem (Camastra and Vinciarelli 2002).

In our work, to reduce the bias between the analytical pdf g and the estimated one $\hat{g}$, for each value $1 \le d \le D$ we learn a test pdf $\check{g}_d(r;k)$ by means of points uniformly drawn from the d-dimensional unit hypersphere; moreover, to best resemble the point density of the given dataset, we draw exactly N points per dimensionality. Finally, we numerically estimate the Kullback-Leibler divergence by means of the estimates $\hat{g}$ and $\check{g}_d$.

More precisely, consider a manifold $\mathcal{M} \equiv \Re^d$ embedded in a higher dimensional space $\Re^D$ through a locally isometric non-linear smooth map $\psi : \mathcal{M} \to \Re^D$. Given a sample set $\boldsymbol{X}_N = \{\boldsymbol{x}_i\}_{i=1}^N = \{\psi(\boldsymbol{z}_i)\}_{i=1}^N \subset \Re^D$ where $\boldsymbol{z}_i$ are independent identically distributed points drawn from $\mathcal{M}$ according to a non-uniform smooth pdf $f : \mathcal{M} \to \Re^+$, we compute a vector of normalized distances $\hat{\boldsymbol{r}} = \{\hat{r}_i\}_{i=1}^N = \{\rho(\boldsymbol{x}_i)\}_{i=1}^N$ by means of Eq. (3). Moreover, for each dimensionality $d \in \{1..D\}$ we uniformly draw a set of N points $\boldsymbol{Y}_{Nd} = \{\boldsymbol{y}_i\}_{i=1}^N$ from the unit d-dimensional hypersphere,[2] and we similarly compute a vector of normalized distances $\check{\boldsymbol{r}}_d = \{\check{r}_{id}\}_{i=1}^N = \{\rho(\boldsymbol{y}_i)\}_{i=1}^N$.

[2] Notice that, a d-dimensional vector randomly sampled from a d dimensional hypersphere according to the uniform pdf, can be generated by drawing a point $\bar{\boldsymbol{y}}$ from a standard normal distribution $\mathcal{N}(\cdot|\boldsymbol{0}_d, 1)$ and by scaling its norm, as we describe in detail in Eq. (16) of the following Sect. 6.

Given a set of N values $r_{i=1}^N \subset [0,1]$ distributed according to a generic pdf p, in Wang et al. (2006) the following pdf estimator $\hat{p}(r)$ is proposed:

$$\hat{p}(r) = \frac{N^{-1}}{2\rho(r)} \tag{12}$$

where $\rho(r)$ is the distance between r and its nearest neighbor. In our problem, considering a distance $\hat{r}_i \in \hat{\boldsymbol{r}}$, the pdf estimates $\hat{g}$ and $\check{g}_d$ can be computed as follows:

$$\hat{g}(\hat{r}_i; k) = \frac{1/(N-1)}{2\hat{\rho}(\hat{r}_i)}, \qquad \check{g}_d(\hat{r}_i; k) = \frac{1/N}{2\check{\rho}_d(\hat{r}_i)} \tag{13}$$

where $\hat{\rho}(\hat{r}_i)$ and $\check{\rho}_d(\hat{r}_i)$ are the distances between $\hat{r}_i$ and its first neighbor in $\hat{\boldsymbol{r}}$ and in $\check{\boldsymbol{r}}_d$ respectively.

In Wang et al. (2006) a Kullback-Leibler divergence estimator based on the nearest neighbor search is proposed; moreover, the authors show that their method is more effective than partitioning-based techniques, especially when the number of samples is limited. Employing this estimator between $\hat{g}$ and $\check{g}_d$ we obtain:

$$\begin{aligned} \hat{KL}(\hat{g}, \check{g}_d) &= \frac{1}{N}\sum_{i=1}^{N} \log \frac{\hat{g}(\hat{r}_i; k)}{\check{g}_d(\hat{r}_i; k)} = \frac{1}{N}\sum_{i=1}^{N} \log \frac{\frac{1/(N-1)}{2\hat{\rho}(\hat{r}_i)}}{\frac{1/N}{2\check{\rho}_d(\hat{r}_i)}} \\ &= \log \frac{N}{N-1} + \frac{1}{N}\sum_{i=1}^{N} \log \frac{\check{\rho}(\hat{r}_i)}{\hat{\rho}_d(\hat{r}_i)} \end{aligned} \tag{14}$$

Employing Eq. (14), the estimated id value ($\hat{d}$) is computed as follows:

$$\hat{d} = \underset{d \in \{1..D\}}{\arg\min} \left(\log \frac{N}{N-1} + \frac{1}{N}\sum_{i=1}^{N} \log \frac{\check{\rho}(\hat{r}_i)}{\hat{\rho}_d(\hat{r}_i)} \right) \tag{15}$$

We call this estimator MiND$_{\mathrm{KL}}$; its time complexity is $O(D^2N^2)$.

Due to Theorem 1, Theorem 4 in Costa and Hero (2005), and considering that the employed Kullback-Leibler divergence estimator is consistent (see Wang et al. 2006), Eq. (15) represents a consistent estimator for the intrinsic dimensionality of the manifold $\boldsymbol{\mathcal{M}}$.

For the sake of clarity, in Appendix the pseudocode of this algorithm is reported.

6 **IDEA**

In this section we present a novel id estimator, called "Intrinsic Dimensionality Estimation Algorithm" (IDEA); to recover from the underestimation effect due to the bias described in Sect. 4, IDEA exploits a correction technique, described in Sect. 6.3, which is based on asymptotic estimation. In Sect. 6.1 we report the theories which lay the foundation of the algorithm described in Sect. 6.2.

6.1 IDEA theoretical results

To report the basic theoretical results that lead us to the development of IDEA, we firstly consider the more specific case of a manifold $\boldsymbol{\mathcal{M}} \equiv \boldsymbol{\mathcal{B}}_d(\mathbf{0}_d, 1)$, where $\boldsymbol{\mathcal{B}}_d(\mathbf{0}_d, 1)$ is a d-dimensional centered open ball with unitary radius, embedded in $\Re^D$ through the identity map ψ.

To estimate the dimensionality d of $\mathcal{B}_d(\mathbf{0}_d, 1)$ we need to identify a measurable characteristic of the hypersphere depending only on d. To this aim, we consider that a d dimensional vector randomly sampled from a d dimensional hypersphere according to the uniform probability density function (pdf), can be generated by drawing a point $\hat{z}$ from a standard normal distribution $\mathcal{N}(\cdot|\mathbf{0}, 1)$ and by scaling its norm (see Sect. 3.29 of Fishman 1996):

$$z = \frac{u^{\frac{1}{d}}}{\|\hat{z}\|}\hat{z}, \quad \hat{z} \sim \mathcal{N}(\cdot|\mathbf{0}, 1) \tag{16}$$

where u is a random sample drawn from the uniform distribution $U(0, 1)$.

Being u uniformly distributed, the quantities $1 - u^{1/d}$ are distributed according to the beta pdf $\beta_{1,d}$ with expectation $\mathbb{E}_{u\sim U(0,1)}[1 - u^{1/d}] = \frac{1}{1+d}$. Therefore, the intrinsic dimensionality of the hypersphere is computed as:

$$\mathbb{E}_{z\sim\mathbf{B}_d}\big[1 - \|z\|\big] = \mathbb{E}_{z\sim\mathbf{B}_d}\big[1 - u^{\frac{1}{d}}\big] = \frac{1}{1+d} \quad \Rightarrow \quad d = \frac{\mathbb{E}_{z\sim\mathbf{B}_d}[\|z\|]}{1 - \mathbb{E}_{z\sim\mathbf{B}_d}[\|z\|]} \tag{17}$$

where $\mathbf{B}_d$ is the uniform pdf in the unit d-dimensional sphere. Notice that, embedding the hypersphere in a higher dimensional space $\Re^D$ by means of a map ψ that applies only a rotation, does not change this result.

To extend this method to more general cases, we now consider points uniformly drawn from a d-dimensional manifold $\mathcal{M} \equiv \Re^d$ embedded in $\Re^D$ through a smooth map $\psi : \mathcal{M} \to \Re^D$.

Under these assumptions the point norms may be not distributed as $u^{\frac{1}{d}}$; however, being ψ a smooth map, close neighbors of $\mathcal{M}$ are mapped to close neighbors of $\Re^D$. Moreover, choosing a d-dimensional open ball $\mathcal{B}_d(\boldsymbol{c}, \epsilon)$ with center $\boldsymbol{c} \in \mathcal{M}$ and radius $\epsilon > 0$, as long as ψ preserves distances in $\mathcal{B}_d$, then for z uniformly drawn from $\mathcal{B}_d$, the distances $\frac{1}{\epsilon}\|\psi(\boldsymbol{c}) - \psi(z)\| = \frac{1}{\epsilon}\|\boldsymbol{c} - z\|$ are distributed as $u^{\frac{1}{d}}$, so that the result reported in Eq. (17) is still valid and we obtain:

$$d = \frac{\mathbb{E}_{z\sim\mathbf{B}_d(\boldsymbol{c},\epsilon)}[\frac{1}{\epsilon}\|\psi(\boldsymbol{c}) - \psi(z)\|]}{1 - \mathbb{E}_{z\sim\mathbf{B}_d(\boldsymbol{c},\epsilon)}[\frac{1}{\epsilon}\|\psi(\boldsymbol{c}) - \psi(z)\|]} \tag{18}$$

where, $\mathbf{B}_d(\boldsymbol{c}, \epsilon)$ is the uniform distribution in the ball $\mathcal{B}_d(\boldsymbol{c}, \epsilon)$.

To further generalize our theoretical results, we consider a locally isometric smooth map $\psi : \mathcal{M} \to \Re^D$, and samples drawn from $\mathcal{M} \equiv \Re^d$ by means of a non-uniform smooth pdf $f : \mathcal{M} \to \Re^+$. Notice that, being ψ a local isometry, it induces a distance function $\delta_\psi(\cdot, \cdot)$ representing the metric on $\psi(\mathcal{M})$. Under these assumptions Eqs. (17), (18) do not hold.

However, without loss of generality, we consider $\boldsymbol{c} = \mathbf{0}_d \in \Re^d$ and $\psi(\boldsymbol{c}) = \mathbf{0}_D \in \Re^D$, and we employ Theorem 1 to show that any smooth pdf f is locally uniform where the probability is not zero. To this aim, assuming $f(\mathbf{0}_d) > 0$ and $z \in \Re^d$, we denote with f_ϵ the pdf obtained by setting $f_\epsilon(z) = 0$ when $\|z\| > 1$, and $f_\epsilon(z) \propto f(\epsilon z)$ when $\|z\| \le 1$. More precisely, denoting with $\chi_{\mathcal{B}_d(\mathbf{0}_d,1)}$ the indicator function on the ball $\mathcal{B}_d(\mathbf{0}_d, 1)$, we obtain:

$$f_\epsilon(z) = \frac{f(\epsilon z)\chi_{\mathcal{B}_d(\mathbf{0}_d,1)}(z)}{\int_{\boldsymbol{t}\in\mathcal{B}_d(\mathbf{0}_d,1)} f(\epsilon \boldsymbol{t})d\boldsymbol{t}} \tag{19}$$

Theorem 1 proves that the convergence of f_ϵ to $\mathbf{B}_d$ is uniform, so that in the limit ($\epsilon \to 0^+$) Eq. (17) holds both for d-dimensional nonlinear manifolds embedded in $\Re^D$, and for points drawn by means of a non-uniform density function f. More precisely, for the smoothness and for the local isometry of ψ:

$$\mathbb{E}_{z\sim f_\epsilon}\left[\delta_\psi\left(\psi(z), \psi(\mathbf{0}_d)\right)\right] = \mathbb{E}_{z\sim f_\epsilon}\left[\|z\|\right] \xrightarrow[\epsilon\to 0^+]{} \mathbb{E}_{z\sim \mathbf{B}_d}\left[\|z\|\right] = m \tag{20}$$

6.2 The base algorithm

In this section we describe how the theoretical results reported in Sect. 6.1 can be applied to develop an `id` estimator.

To this aim, we consider a d-dimensional manifold $\boldsymbol{\mathcal{M}} \equiv \Re^d$ non-linearly embedded in $\Re^D$ through a smooth locally isometric map $\psi : \boldsymbol{\mathcal{M}} \to \Re^D$, and a given sample set $\boldsymbol{X}_N = \{\boldsymbol{x}_i\}_{i=1}^N = \{\psi(z_i)\}_{i=1}^N \subset \Re^D$, where $z_i \in \Re^d$ are independent identically distributed points drawn from $\boldsymbol{\mathcal{M}}$ according to a smooth `pdf` $f : \boldsymbol{\mathcal{M}} \to \Re^+$. To estimate the intrinsic dimensionality of $\boldsymbol{\mathcal{M}}$ by means of the points in the set $\boldsymbol{X}_N$, according to the theoretical results reported in Sect. 6.1 we must estimate the expectation of distances $\frac{1}{\epsilon}\delta_\psi(\psi(\boldsymbol{c}), \boldsymbol{x})$ for infinitesimal balls $\boldsymbol{\mathcal{B}}_D(\psi(\boldsymbol{c}), \epsilon)$ with $\boldsymbol{c} \in \boldsymbol{\mathcal{M}}$. To this aim, for each point $\boldsymbol{x}_i \in \boldsymbol{X}_N$ we find the set of $k+1$ $(1 \le k \le N-1)$ nearest neighbors $\hat{\boldsymbol{X}}_{k+1}^N = \hat{\boldsymbol{X}}_{k+1}^N(\boldsymbol{x}_i) = \{\boldsymbol{x}_j\}_{j=1}^{k+1} \subset \boldsymbol{X}_N$. Call $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_{k+1}^N(\boldsymbol{x}_i) \in \hat{\boldsymbol{X}}_{k+1}^N$ the most distant point from $\boldsymbol{x}_i$, and denote $\boldsymbol{X}_k^N = \boldsymbol{X}_k^N(\boldsymbol{x}_i) = \hat{\boldsymbol{X}}_{k+1}^N \backslash \{\hat{\boldsymbol{x}}\}$. Notice that, when $\boldsymbol{x}_i$ is fixed, almost surely (a.s.) we have $\|\boldsymbol{x} - \boldsymbol{x}_i\| < \|\hat{\boldsymbol{x}} - \boldsymbol{x}_i\|$ $\forall \boldsymbol{x} \in \boldsymbol{X}_k^N$; therefore, we can consider points in $\boldsymbol{X}_k^N$ as drawn from the open ball $\boldsymbol{\mathcal{B}}_D(\boldsymbol{x}_i, \|\hat{\boldsymbol{x}} - \boldsymbol{x}_i\|)$. Exploiting this fact, in order to estimate the intrinsic dimension d of $\boldsymbol{\mathcal{M}}$, we estimate the expectation of distances as follows:

$$m \simeq \frac{1}{k} \sum_{\boldsymbol{x}\in \boldsymbol{X}_k^N} \frac{\|\boldsymbol{x}_i - \boldsymbol{x}\|}{\|\hat{\boldsymbol{x}} - \boldsymbol{x}_i\|}$$

Note that m depends only upon the intrinsic dimensionality d of $\boldsymbol{\mathcal{M}}$ and does not depend on the chosen center $\boldsymbol{x}_i$.

Corollary 1 *Given two sequences $\{k_j\}$ and $\{N_j\}$ such that for $j \to +\infty$:*

$$k_j \to +\infty, \qquad N_j \to +\infty, \qquad \frac{k_j}{N_j} \to 0 \tag{21}$$

We have the limit:

$$\lim_{j\to+\infty} \frac{1}{k_j} \sum_{\boldsymbol{x}\in \boldsymbol{X}_{k_j}^{N_j}} \frac{\|\boldsymbol{x}_i - \boldsymbol{x}\|}{\|\hat{\boldsymbol{x}} - \boldsymbol{x}_i\|} = m \quad a.s. \tag{22}$$

Proof Considering the sequences $\{k_j\}$ and $\{N_j\}$, the conditions reported in Eq. (21) ensure that $\epsilon = \|\hat{\boldsymbol{x}} - \boldsymbol{x}_i\| \to 0^+$ when $j \to +\infty$.[3] Theorem 4 in Costa and Hero (2005) ensures that geodetic distances in the infinitesimal ball converge to Euclidean distances with probability 1; furthermore, the sample mean is an unbiased estimator for the expectation (law of

[3] See proof of Theorem 4 in Costa and Hero (2005) where k must be substituted by $o(n)$.

large numbers); moreover, Theorem 1 guarantees that the underlying pdf converges to the uniform one. Considering all these facts, we obtain:

$$
\begin{aligned}
\lim_{j\to+\infty}\frac{1}{k_j}\sum_{\boldsymbol{x}\in\boldsymbol{X}_{k_j}^{N_j}}\frac{\|\boldsymbol{x}_i-\boldsymbol{x}\|}{\|\hat{\boldsymbol{x}}-\boldsymbol{x}_i\|} &= \lim_{j\to+\infty}\frac{1}{k_j\epsilon}\sum_{\boldsymbol{x}\in\boldsymbol{X}_{k_j}^{N_j}}\delta_\psi(\boldsymbol{x}_i,\boldsymbol{x})+o(\epsilon)\\
&= \lim_{\epsilon\to0^+}\mathbb{E}_{z\sim f_\epsilon}\left[\frac{1}{\epsilon}\delta_\psi\big(\psi(z),\psi(\mathbf{0}_d)\big)\right]\\
&= \lim_{\epsilon\to0^+}\mathbb{E}_{z\sim\mathbf{B}_d(\mathbf{0},\epsilon)}\left[\frac{1}{\epsilon}\delta_\psi\big(\psi(z),\psi(\mathbf{0}_d)\big)\right]\\
&= \lim_{\epsilon\to0^+}\frac{1}{\epsilon}\,\epsilon\,\mathbb{E}_{z\sim\mathbf{B}_d(\mathbf{0},1)}\big[\delta_\psi\big(\psi(z),\psi(\mathbf{0}_d)\big)\big]\\
&= \mathbb{E}_{z\sim\mathbf{B}_d(\mathbf{0},1)}\big[\|z\|\big]=m
\end{aligned}
\tag{23}
$$

□

By employing Eq. (17) and Corollary 1, we get a consistent estimator $\hat{d}$ for the intrinsic dimensionality d of $\boldsymbol{\mathcal{M}}$ as follows:

$$
\begin{aligned}
m \simeq \hat{m} &= \frac{1}{Nk}\sum_{i=1}^{N}\sum_{\boldsymbol{x}\in\boldsymbol{X}_k^N}\frac{\|\boldsymbol{x}_i-\boldsymbol{x}\|}{\|\hat{\boldsymbol{x}}-\boldsymbol{x}_i\|}\\
d &= \frac{m}{1-m}\simeq\frac{\hat{m}}{1-\hat{m}}=\hat{d}
\end{aligned}
\tag{24}
$$

The time complexity of IDEA is $O(DN^2)$. For the sake of clarity, in Appendix the pseudocode of this algorithm is reported.

6.3 Asymptotic correction

Although the algorithm described in Sect. 6.2 proposes a consistent estimator of the id, when this value is sufficiently high the number of sample points becomes insufficient to compute an acceptable estimation. This is due both the bias described in Sect. 4 and to the fact that, as shown in Eckmann and Ruelle (1992), the number of sample points required to perform id estimation with acceptable results, grows exponentially with the value of the id.

To reduce the effect due to this problem, in this section we propose a method that allows to study the asymptotic behavior described by the available data. To this aim, we adopt a Monte Carlo approach performing R runs of the algorithm reported in Sect. 6.2. We extract from the given dataset $\boldsymbol{X}_N$ random subsets $\boldsymbol{\mathcal{R}}_{r=1}^R$ with different cardinalities $\boldsymbol{R}_{r=1}^R$. The cardinalities $\boldsymbol{R}_r$ are randomly generated by means of the binomial distribution $Binom(N,p)$, where the value of p spans a fixed range.[4] The intrinsic dimensionality, estimated during each run, becomes a sample from a "trend curve"; moreover, for each subsample the parameter k_r, that is the number of nearest neighbors to be considered, is set to $k_r=\lceil k\sqrt{p}\rceil$.

[4] In our tests $p\in\{0.1,\dots,0.9\}$.

This choice is performed to emulate a sequence $\{k_r\}$ such that $k_r \to +\infty$, $\boldsymbol{R}_r \to +\infty$, and $\frac{k_r}{\boldsymbol{R}_r} \to 0$, thus fulfilling the conditions reported in Eq. (21).

We noticed that, when the base algorithm proposed in Sect. 6.2 underestimates the intrinsic dimensionality, its application to point subsets $\boldsymbol{\mathcal{R}}_{r=1}^{R}$ with increasing cardinality produces increasing estimations of the intrinsic dimension $\hat{d} = \hat{d}(\boldsymbol{R}_r)$. As demonstrated in Sect. 6.2, these estimates converge to the real intrinsic dimensionality for $j \to +\infty$ (see conditions reported in Eq. (21)). Our assumption, based on this empirical observation, is that the function $\hat{d}(N)$ has a horizontal asymptote. Therefore, we fit the pairs $(\log(\boldsymbol{R}_r), \hat{d}(\boldsymbol{R}_r))$ by means of the parametric function[5] g described below:

$$\hat{d}(\boldsymbol{R}_r) \simeq g(\boldsymbol{R}_r) = a_0 - \frac{a_1}{\log_2(\frac{\boldsymbol{R}_r}{a_2} + a_3)} \tag{25}$$

where $\{a_i\}_{i=0}^{3}$ are fitting parameters controlling translation and scaling on both axes; their values are computed by a non-linear least squares fitting algorithm. Notice that, since $\lim_{\boldsymbol{R}_r \to +\infty} g(\boldsymbol{R}_r) = a_0$ then the asymptote of Eq. (25) is $\hat{d} = a_0$. Moreover, the derivate $g' = \frac{\partial g(\boldsymbol{R}_r)}{\partial \boldsymbol{R}_r}$ shows that the parameter a_1 controls the increasing/decreasing behavior of the function g. For these reasons, when the estimated parameter $a_1 > 0$ (increasing function), we use the parameter a_0 as the final estimate for d; otherwise, we use the estimation obtained by the base algorithm applied to the whole dataset.

To obtain a stable estimation of the intrinsic dimension we execute the asymptotic correction algorithm $T = 20$ times and we average the obtained results. When IDEA employs this asymptotic correction the time complexity is $O(TDN^2)$.

7 Fast KNN approximation

To reduce the time complexity of the proposed algorithms we should optimize the method employed to build the KNN graph, since the KNN graph construction technique by brute-force has time complexity $O(DN^2)$, thus representing the most computationally expensive part of our methods. To this aim, some interesting approaches have been proposed in literature, including two methods proposed by Paredes et al. (2006), where the authors presented a KNN graph construction for general metric spaces, whose empirical time complexity is low. Unfortunately, both the proposed methods require a global data structure and are therefore difficult to be parallelized across machines. Other two efficient methods for the Euclidean metric space, have been recently developed, which are based on space filling curves (Connor and Kumar 2010) and recursive data partitioning (Chen et al. 2009).

The last approach is particularly suited for our goal, since it allows to reduce the time complexity according to the value of a parameter t. More precisely, Chen et al. propose two divide and conquer methods (called KNN$_{glue}$ and KNN$_{overlap}$) for computing an approximate KNN graph that has time complexity $O(DN^t)$. The exponent $t \in (1, 2)$ is an increasing function of an internal parameter α which governs the size of the common region in the divide step. Experiments proposed by the authors show that a high quality graph can usually be obtained with small overlaps, that is, for small values of t.

These algorithms are structured as follows: the divide step uses an inexpensive Lanczos procedure to perform recursive spectral bisection, and then, after each conquer step, an

[5] The choice of using 2 as the log base does not affect the results, being the change of base just a change of scale in the y axis.

additional refinement is performed to improve the accuracy of the graph. Note that a hash table is continuously updated to avoid repeating distance calculations during the divide and conquer process.

The strong difference between the two algorithms is in the divide step; indeed, while $\mathtt{KNN}_{glue}$ splits the set into three subsets, $\mathtt{KNN}_{overlap}$ divides the set into two subsets.

To evaluate how these novel KNN constructions could affect the quality of our estimators, we have performed specific tests by substituting the brute-force KNN graph construction with these approaches (see Table 5).

8 Algorithm evaluation

In this section we describe the datasets employed in our experiments (see Sect. 8.1), we summarize the adopted experimental settings (see Sect. 8.2), and we report the results achieved by the proposed algorithms comparing them to those obtained by six state of the art id estimators (see Sect. 8.3).

8.1 Dataset description

To evaluate our algorithms, we have performed experiments on 17 synthetic and 6 real datasets (see Table 1) described in the following.

To generate 15 synthetic datasets we have employed the tool proposed in Hein (2005), and we have extended it to produce the $\mathcal{M}_{14}$ and the $\mathcal{M}_{15}$ datasets by drawing points from non-linearly embedded manifolds characterized by high id. More precisely, to generate $\mathcal{M}_{14}$ we have proceeded as follows:

- we sample 2500 points in $\Re^{18}$, whose elements have been uniformly drawn in the range $[0, 1]$, and we have stored them in a matrix $\boldsymbol{X}_N \in \Re^{2500\times 18}$;
- we multiply each element of $\boldsymbol{X}_N$ $(\boldsymbol{X}_N(i, j))$ by $\sin(\cos(2\pi \boldsymbol{X}_N(i, j)))$, thus obtaining a matrix $\boldsymbol{D}' \in \Re^{2500\times 18}$;
- we multiply each element of $\boldsymbol{X}_N$ by $\cos(\sin(2\pi \boldsymbol{X}_N(ij)))$, thus obtaining another matrix $\boldsymbol{D}'' \in \Re^{2500\times 18}$;
- we append $\boldsymbol{D}'$ and $\boldsymbol{D}''$ to generate a matrix $\boldsymbol{D}''' \in \Re^{2500\times 36}$;
- we duplicate $\boldsymbol{D}'''$ and we append the two matrices to finally generate the $\mathcal{M}_{14}$ dataset containing 2500 points in $\Re^{72}$.

Notice that, the id of this non-linearly embedded manifold is 18. The manifold $\mathcal{M}_{15}$ is similarly generated employing the same number of uniformly sampled points in $\Re^{24}$ to generate a dataset whose id is 24.

The real datasets employed are: the ISOMAP face database (Tenenbaum et al. 2000), the MNIST database (LeCun et al. 1998), the Santa Fe (Pineda and Sommerer 1994) dataset, the Isolet dataset (Frank and Asuncion 2010), the DSVC1 time series (Camastra and Filippone 2009), and the Paris14e Parc Montsouris time series (Camastra and Filippone 2009).

The ISOMAP face database consists in 698 gray-level images of size 64 × 64 depicting the face of a sculpture. This dataset has three degrees of freedom: two for the pose and one for the lighting direction.

The MNIST database consists in 70000 gray-level images of size 28 × 28 of hand-written digits; in our tests we used the 6742 training points representing the digit 1. The id of this database is not actually known, but some works (Hein 2005; Costa and Hero 2005) have

Table 1 Brief description of the 17 synthetic and 6 real datasets, where d is the `id` and D is the embedding space dimension. In the name of the synthetic datasets, the number in the subscript refers to the dataset name used by the generator proposed in Hein (2005)

Dataset	Name	d	D	Description
Synthetic	$\mathcal{M}_1$	10	11	Uniformly sampled sphere linearly embedded.
	$\mathcal{M}_2$	3	5	Affine space.
	$\mathcal{M}_3$	4	6	Concentrated figure, confusable with a 3d one.
	$\mathcal{M}_4$	4	8	Non-linear manifold.
	$\mathcal{M}_5$	2	3	2-d Helix.
	$\mathcal{M}_6$	6	36	Non-linear manifold.
	$\mathcal{M}_7$	2	3	Swiss-Roll.
	$\mathcal{M}_8$	12	72	Non-linear manifold.
	$\mathcal{M}_9$	20	20	Affine space.
	$\mathcal{M}_{10a}$	10	11	Uniformly sampled hypercube.
	$\mathcal{M}_{10b}$	17	18	Uniformly sampled hypercube.
	$\mathcal{M}_{10c}$	24	25	Uniformly sampled hypercube.
	$\mathcal{M}_{11}$	2	3	Möebius band 10-times twisted.
	$\mathcal{M}_{12}$	20	20	Isotropic multivariate Gaussian.
	$\mathcal{M}_{13}$	1	13	Curve.
	$\mathcal{M}_{14}$	18	72	Non-linear manifold.
	$\mathcal{M}_{15}$	24	96	Non-linear manifold.
Real	$\mathcal{M}_{\texttt{Faces}}$	3	4096	`ISOMAP` face dataset.
	$\mathcal{M}_{\texttt{MNIST1}}$	8–11	784	`MNIST` database (digit 1).
	$\mathcal{M}_{\texttt{SantaFe}}$	9	50	`Santa Fe` dataset (version $D2$).
	$\mathcal{M}_{\texttt{Isolet}}$	16–22	617	Spoken letter of the alphabet.
	$\mathcal{M}_{\texttt{DSVC1}}$	2.26	20	Real time series of a Chua's circuit.
	$\mathcal{M}_{\texttt{Paris14e}}$	4–6	20	Real time series of temperatures.

proposed similar estimations for the different digits; considering digit 1, the proposed `id` values are in the range $\{8..11\}$.

The version $D2$ of the `Santa Fe` dataset is a synthetic time series of 50000 one-dimensional points; it was generated by a simulation of particle motion, and it has nine degrees of freedom. In order to estimate the attractor dimension of this time series, we used the method of delays described in Ott (1993), which generates D-dimensional vectors by collecting D values from the original dataset; by choosing $D = 50$ we obtained a dataset containing 1000 points in $\Re^{50}$.

The `Isolet` dataset has been generated as follows: 150 subjects spoke the name of each letter of the alphabet twice, thus producing 52 training examples from each speaker. The speakers are grouped into sets of 30 speakers each, and are referred to as *isolet*1, *isolet*2, *isolet*3, *isolet*4, and *isolet*5, for a total of 7797 samples. The `id` of this dataset is not actually known, but a study reported in Kivimäki et al. (2010) has proposed that the correct estimation could be in the range $\{16..22\}$.

The `DSVC1` is a real data time series, formed by 5000 samples, measured from a hardware realization of Chua's circuit (Chua et al. 1985). We used the method of delays choosing $D = 20$, and we obtained a dataset containing 250 points in $\Re^{20}$. The `id` of the dataset is $\sim$2.26 as reported in Camastra and Filippone (2009).

The Paris14e Parc Montsouris is a real data time series formed by the daily average temperatures, expressed in tenth of Celsius degrees, in Paris. This series covers the period form January 1958 to December 2001, and contains 15700 samples. We used the method of delays by choosing $D = 20$, and we obtained a dataset containing 785 points in $\Re^{20}$ whose id is in the range $\{4..6\}$ as reported in Camastra and Filippone (2009).

8.2 Experimental setting

To objectively assess our methods, we compared them with six well-known id estimators: PCA, SPPCA kNNG, CD, MLE, Hein, and BPCA. For kNNG, MLE, Hein, and BPCA we used the authors' implementation,[6] while for the other algorithms we employed the version provided by the dimensionality reduction toolbox.[7]

To generate the synthetic datasets we adopted the modified generator described in Hein (2005) creating 20 instances of each dataset reported in Table 1, each of which is composed by 2500 randomly sampled points. To obtain an unbiased estimation, for each technique we averaged the results achieved on the 20 instances. To execute multiple tests on $\mathcal{M}_{\mathtt{MNIST1}}$ and Isolet we extracted 5 random subsets containing 2500 points each, and we averaged the achieved results.

In Table 2 the employed configuration parameters are summarized. To relax the dependency of kNNG algorithm from the selection of the value of its parameter k, we performed multiple runs with $k_1 \le k \le k_2$ (see Table 2) and we averaged the achieved results.

8.3 Experimental results

In this section the results achieved on both real and synthetic datasets are reported.

In Table 3 the results achieved on the synthetic datasets are summarized. As can be noticed, all the algorithms but PCA and SPPCA achieve good results for datasets with low id ($d < 10$), whilst the PCA and SPPCA methods obtains highly overestimated values when dealing with non-linearly embedded manifolds.

Another consideration raised by the observation of the results achieved on linearly embedded manifolds with high id, is that all the techniques but PCA, $\mathtt{MiND}_{\mathtt{KL}}$, and IDEA strongly underestimate the id.

Furthermore, when the id is high and the manifold is embedded through a non-linear map ($\mathcal{M}_{14}$ and $\mathcal{M}_{15}$) only $\mathtt{MiND}_{\mathtt{KL}}$, and IDEA obtain stable estimations. These facts confirm that these two techniques are the only estimators that guarantee good estimation with both linear and non-linear embeddings and with both high and low id, obtaining the most reliable estimations, which are always comparable with the best ones.

In the last row of Table 3 the Mean Percentage Error (MPE) indicator is reported; for each algorithm this value is computed as the mean of the percentage errors obtained on each dataset:

$$\mathtt{MPE} = \frac{100}{\#\mathcal{M}} \sum_{\mathcal{M}} \frac{|\hat{d}_{\mathcal{M}} - d_{\mathcal{M}}|}{d_{\mathcal{M}}} \tag{26}$$

[6] http://www.eecs.umich.edu/~hero/IntrinsicDim/, http://www.stat.lsa.umich.edu/~elevina/mledim.m, http://www.ml.uni-saarland.de/code.shtml, http://research.microsoft.com/en-us/um/cambridge/projects/infernet/blogs/bayesianpca.aspx.

[7] http://cseweb.ucsd.edu/~lvdmaaten/dr/download.php.

Table 2 Parameter settings for the different estimators: k represents the number of neighbors, γ the edge weighting factor for kNN, M the number of Least Square (LS) runs, N the number of resampling trials per LS iteration, α and π represent the parameters (shape and rate) of the Gamma prior distributions, describing the hyperparameters and the observation noise model of BPCA, μ contains the mean and the precision of the Gaussian prior distribution describing the bias inserted in the inference of BPCA

Dataset	Method	Parameters
Synthetic	PCA	$\mathit{Threshold} = 0.025$
	SPPCA	*None*
	CD	*None*
	MLE	$k_1 = 6\ k_2 = 20$
	kNNG$_1$	$k_1 = 6$, $k_2 = 20$, $\gamma = 1$, $M = 1$, $N = 10$
	kNNG$_2$	$k_1 = 6$, $k_2 = 20$, $\gamma = 1$, $M = 10$, $N = 1$
	BPCA	$\mathit{iters} = 500$, $\alpha = (2.0, 2.0)$ $\pi = (2.0, 2.0)$ $\mu = (0.0, 0.01)$
	MiND$_{\mathtt{MLk}}$	$k = 10$
	MiND$_{\mathtt{MLi}}$	$k = 10$
	MiND$_{\mathtt{KL}}$	$k = 10$
	IDEA	$k = 10$
Real	PCA	$\mathit{Threshold} = 0.0025$
	SPPCA	*None*
	CD	*None*
	MLE	$k_1 = 3\ k_2 = 8$
	kNNG$_1$	$k_1 = 3$, $k_2 = 8$, $\gamma = 1$, $M = 1$, $N = 10$
	kNNG$_2$	$k_1 = 3$, $k_2 = 8$, $\gamma = 1$, $M = 10$, $N = 1$
	BPCA	$\mathit{iters} = 2000$, $\alpha = (2.0, 2.0)$ $\pi = (2.0, 2.0)$ $\mu = (0.0, 0.01)$
	MiND$_{\mathtt{MLk}}$	$k = 5$
	MiND$_{\mathtt{MLi}}$	$k = 5$
	MiND$_{\mathtt{KL}}$	$k = 5$
	IDEA	$k = 5$

where $d_{\mathcal{M}}$ is the real id, $\hat{d}_{\mathcal{M}}$ is the estimated one, and $\#\mathcal{M}$ is the number of tested manifolds (see Table 1). Notice that IDEA and MiND$_{\mathtt{KL}}$ obtain the minimum MPE, confirming that these estimators achieve the best average estimation results on synthetic datasets.

In Table 4 the results achieved on real datasets have been summarized. Notice that, also in the case of noisy real datasets, MiND$_{\mathtt{KL}}$ and IDEA have obtained either the best approximation of the id, or estimates always comparable with those computed by the best performing techniques. These results confirm that IDEA and MiND$_{\mathtt{KL}}$ are really promising. Besides, the MPE[8] objectively shows that our techniques achieve the best average estimation precision also on real datasets.

Another performed experiment is aimed to test the robustness of our algorithms with respect to the choice of the parameter k. To achieve this goal, we reproduced the experiments proposed for the MLE algorithm in Fig. 1(a) of Levina and Bickel (2005) employing MiND$_{\mathtt{KL}}$, MiND$_{\mathtt{MLk}}$, and IDEA and we averaged the curves obtained in 10 runs. In these tests the adopted datasets are composed by points drawn from the standard Gaussian pdf in $\mathfrak{R}^5$.

[8] When the value of the id is in a range we compute the MPE considering the mean value of the range as $d_{\mathcal{M}}$.

Table 3 Results achieved on the synthetic datasets. The last row contains the MPE indicator that allows to perform a direct comparison among methods. The best approximations are highlighted in bold case

Dataset	d	PCA	SPPCA	BPCA	kNNG$_1$	kNNG$_2$	CD	MLE	Hein	MiND$_{\mathrm{MLk}}$	MiND$_{\mathrm{MLi}}$	MiND$_{\mathrm{KL}}$	IDEA
$\mathcal{M}_{13}$	*1*	4.00	3.00	5.70	0.97	1.07	1.14	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.02
$\mathcal{M}_{5}$	*2*	3.00	3.00	**2.00**	1.96	2.06	1.98	1.97	**2.00**	1.99	**2.00**	**2.00**	**2.00**
$\mathcal{M}_{7}$	*2*	3.00	3.00	**2.00**	1.97	2.09	1.93	1.96	**2.00**	1.94	**2.00**	**2.00**	2.07
$\mathcal{M}_{11}$	*2*	3.00	3.00	1.55	1.95	2.03	2.19	2.21	**2.00**	1.99	**2.00**	**2.00**	1.98
$\mathcal{M}_{2}$	*3*	**3.00**	**3.00**	**3.00**	2.95	3.03	2.88	2.88	**3.00**	2.92	**3.00**	**3.00**	3.03
$\mathcal{M}_{3}$	*4*	**4.00**	**4.00**	**4.00**	3.75	3.82	3.23	3.83	**4.00**	3.84	**4.00**	**4.00**	4.01
$\mathcal{M}_{4}$	*4*	8.00	8.00	4.25	4.05	4.76	3.88	3.95	**4.00**	3.89	**4.00**	4.15	3.93
$\mathcal{M}_{6}$	*6*	12.00	12.00	12.00	6.46	11.24	5.91	6.39	5.95	6.15	**6.00**	6.50	6.33
$\mathcal{M}_{1}$	*10*	11.00	**10.00**	5.45	9.16	9.89	9.12	9.10	9.45	9.34	9.04	10.30	10.41
$\mathcal{M}_{10a}$	*10*	**10.00**	**10.00**	5.20	8.62	10.21	8.09	8.26	8.90	8.42	8.50	9.85	9.93
$\mathcal{M}_{8}$	*12*	24.00	24.00	24.00	13.87	16.61	11.29	13.68	**12.00**	13.49	13.30	16.50	14.49
$\mathcal{M}_{10b}$	*17*	**17.00**	**17.00**	9.46	13.69	15.38	12.30	12.87	13.85	13.23	13.00	16.25	16.07
$\mathcal{M}_{14}$	*18*	36.00	36.00	36.00	**17.58**	5.01	11.60	15.95	14.00	14.39	14.15	18.60	17.30
$\mathcal{M}_{9}$	*20*	**20.00**	**20.00**	13.55	15.25	10.59	13.75	14.64	15.50	14.95	14.75	19.15	18.51
$\mathcal{M}_{12}$	*20*	**20.00**	**20.00**	13.70	16.40	24.89	11.26	15.82	15.00	16.19	15.75	19.35	21.20
$\mathcal{M}_{10c}$	*24*	**24.00**	**24.00**	13.3	17.67	21.42	15.58	16.96	17.95	17.53	17.25	22.55	23.93
$\mathcal{M}_{15}$	*24*	48.00	48.00	48.00	19.66	22.80	14.03	19.83	17.00	18.03	18.00	25.30	**22.90**
MPE		56.47	50.00	67.35	10.09	19.90	17.90	11.79	9.40	11.12	10.64	4.75	**4.01**

Table 4 Results achieved on the real datasets by the employed approaches. The last column contains the MPE indicator that allows to perform a direct comparison among methods. The best approximations are highlighted in bold case

	$\mathcal{M}_{\mathrm{DSVC1}}$	$\mathcal{M}_{\mathrm{Faces}}$	$\mathcal{M}_{\mathrm{Paris14e}}$	$\mathcal{M}_{\mathrm{Santa\ Fe}}$	$\mathcal{M}_{\mathrm{MNIST1}}$	$\mathcal{M}_{\mathrm{Isolet}}$	MPE
PCA	7.00	21.00	12.00	18.00	11.80	47.20	203.72
SPPCA	4.00	5.00	10.00	19.00	9.00	45.00	82.81
BPCA	6.00	4.00	10.00	18.00	11.00	**19.00**	69.10
kNNG$_1$	1.77	3.60	7.80	7.28	10.37	6.50	31.95
kNNG$_2$	1.86	4.32	13.52	7.43	9.58	8.32	51.09
CD	1.92	3.37	4.91	4.39	6.96	3.65	31.32
MLE	2.03	4.05	5.96	7.16	10.29	15.78	18.34
Hein	3.00	**3.00**	9.00	6.00	8.00	3.00	41.01
MiND$_{\mathrm{MLk}}$	2.51	3.59	3.71	6.78	10.02	16.67	16.48
MiND$_{\mathrm{MLi}}$	3.00	4.00	4.00	7.00	**9.45**	17.25	19.67
MiND$_{\mathrm{KL}}$	2.50	3.90	**5.00**	**7.60**	11.00	20.00	12.87
IDEA	**2.14**	3.73	5.14	7.26	11.06	18.77	**11.56**
d	*2.26*	*3*	*4–6*	*9*	*8–11*	*16–22*	

We repeated the test for datasets with cardinalities $N \in \{200, 500, 1000, 2000\}$, and varying the parameter k in the range $\{5..100\}$. As shown in Fig. 1, MiND$_{\mathrm{KL}}$ (left top) demonstrates to be robust to the choice of its parameter k, whilst MiND$_{\mathrm{MLk}}$ (right top) shows a behavior

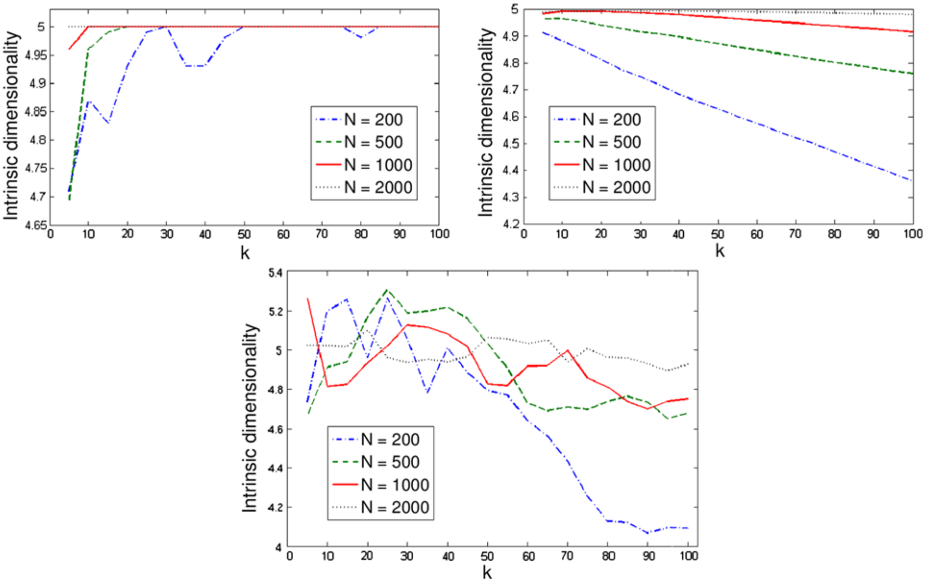

Fig. 1 Behavior of $\mathtt{MiND}_{\mathtt{KL}}$ (*left top*), $\mathtt{MiND}_{\mathtt{MLk}}$ (*right top*) and IDEA (*bottom*) applied to points drawn from a 5-dimensional standard normal distribution. In this test $N \in \{200, 500, 1000, 2000\}$ and $k \in \{5..100\}$

comparable to that of MLE (see Fig. 1(a) in Levina and Bickel 2005). Moreover, as shown in Fig. 1 (bottom), IDEA demonstrates to be resistant to the selection of k when a sufficient number of points is available.

The last experiment we performed, is aimed to test $\mathtt{MiND}_{\mathtt{KL}}$ and IDEA, on 5 different synthetic datasets ($\mathcal{M}_6$, $\mathcal{M}_{10c}$, $\mathcal{M}_{11}$, $\mathcal{M}_{13}$, and $\mathcal{M}_{15}$),[9] by substituting the brute-force KNN graph construction with those proposed in Sect. 7 ($\mathtt{KNN}_{glue}$ and $\mathtt{KNN}_{overlap}$). Notice that, employing these methods and setting the value of the parameter t in the range $\{0.1..0.4\}$, we obtain a graph construction whose time complexity is between $O(DN^{1.1})$ and $O(DN^{1.4})$. To obtain a fair comparison, we employed the same experimental settings described in Sect. 8.2.

The results reported in Table 5 show that the usage of the two different approximations of the KNN graph construction does not strongly affect the quality of the estimations produced by our methods (especially in the case of $\mathtt{MiND}_{\mathtt{KL}}$); therefore, we can employ these algorithms to strongly improve our algorithms' efficiency. Moreover, in our test it is possible to notice that $\mathtt{KNN}_{glue}$ converges faster than $\mathtt{KNN}_{overlap}$ to the estimation achieved employing the brute-force KNN graph construction. This is a further advantage since, despite the two techniques have the same time complexity, the experiments proposed in Chen et al. (2009) show that $\mathtt{KNN}_{glue}$ is more efficient than $\mathtt{KNN}_{overlap}$.

Concluding, the results achieved on both real and synthetic datasets have confirmed the quality of the proposed methods; more specifically, $\mathtt{MiND}_{\mathtt{KL}}$ and IDEA have proved to be the best estimators since they are robust to the setting of their unique parameter, they obtain the smallest MPE (see Tables 3 and 4), and they achieve good approximations both with high and low id, linearly and non-linearly embedded manifolds, and noisy or non-noisy data. Furthermore, the experiments reported in Table 5 show that it is possible to reduce the

[9] We have chosen these datasets because they cover a wide scenario of id values.

Table 5 Results achieved on 5 synthetic datasets by employing the two KNN graph construction methods proposed in Chen et al. (2009). Brute-F. indicates the brute-force method to build the KNN graph. In bold the closer results with respect to the Brute-F.

	KNN	t	$\mathcal{M}_{13}$	$\mathcal{M}_{11}$	$\mathcal{M}_{6}$	$\mathcal{M}_{10c}$	$\mathcal{M}_{15}$
MiND$_{KL}$	Glue	*0.1*	**1.00**	**2.00**	6.85	21.65	28.30
		0.2	**1.00**	**2.00**	6.70	22.45	28.05
		0.3	**1.00**	**2.00**	6.65	22.50	27.00
		0.4	**1.00**	**2.00**	**6.50**	**22.55**	**25.30**
	Overlap	*0.1*	**1.00**	**2.00**	6.95	22.15	27.80
		0.2	**1.00**	**2.00**	6.80	22.20	27.65
		0.3	**1.00**	**2.00**	6.70	22.40	27.55
		0.4	**1.00**	**2.00**	6.55	22.45	27.00
	Brute-F.		*1.00*	*2.00*	*6.50*	*22.55*	*25.30*
IDEA	Glue	*0.1*	0.98	2.10	6.41	24.95	18.87
		0.2	0.99	2.07	6.34	24.40	19.67
		0.3	0.99	2.05	6.33	23.80	21.55
		0.4	1.01	**2.00**	**6.33**	**23.90**	**22.70**
	Overlap	*0.1*	1.10	2.10	6.19	24.63	18.49
		0.2	1.05	**2.00**	6.22	24.43	19.88
		0.3	**1.02**	**2.00**	6.30	24.54	21.40
		0.4	**1.02**	**2.00**	6.35	23.97	22.48
	Brute-F.		*1.02*	*1.98*	*6.33*	*23.93*	*22.90*

time complexity of our algorithms without strongly affecting the quality of the computed estimations.

9 Conclusions and future works

In this work we focus our attention on two id estimators: MiND$_{KL}$ and IDEA. For each point in the dataset, MiND$_{KL}$ exploits the pdf related to the normalized distance of its nearest neighbors. More precisely, this estimator compares the pdf estimated on the available dataset with those estimated by employing random samples uniformly drawn from unitary hyperspheres with dimensionality in $\{1..D\}$. On the other hand, IDEA is a consistent local intrinsic dimensionality estimator that exploits the statistical properties of manifold neighborhoods, offering an asymptotic correction that reduces the underestimate behavior affecting most state of the art id estimators.

We tested our algorithms on synthetic and real datasets comparing them with six well-known id estimators. The achieved results and the Mean Percentage Error indicator objectively show that our techniques are promising. Furthermore, our experiments demonstrate that MiND$_{KL}$ and IDEA are the most robust estimators, since they deal with both low and high id, and they manage both linearly and non-linearly embedded manifolds, computing either the best estimates or values that are strongly comparable to the best ones. Furthermore, their performances are not strongly affected by the choice of their unique parameter k. Finally, we have shown that it is possible to reduce the time complexity of these algorithms without strongly affecting the quality of the computed estimations by substituting the brute-force KNN graph construction with the Fast KNN methods proposed in Chen et al. (2009).

In future works, we want to investigate more deeply the bias described in Sect. 4; indeed, our aim is to formalize (or to approximate) the `pdf` related to the `KNN` normalized distance model so that it will be possible to estimate the intrinsic dimensionality parameters by means of maximum likelihood performed on the right density function. Moreover, to further formally evaluate the effectiveness of our theoretical approach, we would like to identify a bound for the finite sample error.

Appendix: Algorithms implementation

In this appendix the pseudocode of `MiND`$_{\mathrm{KL}}$ and `IDEA` are reported. Algorithm 1 reports the pseudocode of `MiND`$_{\mathrm{KL}}$, where $NN(\boldsymbol{X}_N, \boldsymbol{x})$ is the procedure that returns the nearest neighbor of $\boldsymbol{x}$ in $\boldsymbol{X}_N$. In Algorithm 2 the pseudocode of `IDEA` is reported, where $kNN(\boldsymbol{X}_N, \boldsymbol{x}, k)$ is the procedure that employs a k-nearest neighbor search returning the set of the k nearest neighbors of $\boldsymbol{x}$ in $\boldsymbol{X}_N$.

Algorithm 1: Pseudocode for the `MiND`$_{\mathrm{KL}}$ algorithm

```
Input:
  $\boldsymbol{X}_N$:  The dataset points $\{\boldsymbol{x}_i\}_{i=1}^N$.
  $k$:  The kNN parameter.
Output:
  $\hat{d}$:  The estimated intrinsic dimensionality.
{Compute for each point the normalized radii}
for i:=1 to N do begin
  $\bar{\boldsymbol{X}}_{k+1} = kNN(\boldsymbol{X}_N, \boldsymbol{x}_i, k)$;  {Finding the $k$ neighbors of $\boldsymbol{x}_i$ in $\boldsymbol{X}_N$.}
  $\hat{r}_i = \rho(\boldsymbol{x}_i) = \min_{\boldsymbol{x}_j \in \bar{\boldsymbol{X}}_{k+1}} \|\boldsymbol{x}_i - \boldsymbol{x}_j\| / \max_{\hat{\boldsymbol{x}} \in \bar{\boldsymbol{X}}_{k+1}} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}\|$;
  {Computing the distance between $\hat{r}_i$ and the NN}
  $\hat{\rho}(\hat{r}_i) = |\hat{r}_i - NN(\{\hat{r}_j\}_{j \neq i}, \hat{r}_i)|$;
end
{Estimate the Kullback Leibler divergences}
for d:=1 to D do begin
  {Uniformly sampling from the unit ball}
  $\boldsymbol{Y}_{Nd} = \{\boldsymbol{y}_i = \bar{\boldsymbol{y}} u^{1/d} / \|\bar{\boldsymbol{y}}\|; \bar{\boldsymbol{y}} \sim \mathcal{N}(\cdot|\boldsymbol{0}_d, 1), u \sim U(0,1)\}_{i=1}^N$;
  {Compute for each point the normalized radii}
  for i:=1 to N do begin
    $\bar{\boldsymbol{Y}}_{k+1} = kNN(\boldsymbol{Y}_{Nd}, \boldsymbol{y}_i, k)$;
    $\check{r}_i = \rho(\boldsymbol{y}_i) = \min_{\boldsymbol{y}_j \in \bar{\boldsymbol{Y}}_{k+1}} \|\boldsymbol{y}_i - \boldsymbol{y}_j\| / \max_{\hat{\boldsymbol{y}} \in \bar{\boldsymbol{Y}}_{k+1}} \|\boldsymbol{y}_i - \hat{\boldsymbol{y}}\|$;
  end
  {Computing the distances $\check{\rho}_d(\hat{r}_i)$}
  for i:=1 to N do begin
    {Computing the distance between $\check{r}_i$ and the NN}
    $\check{\rho}_d(\hat{r}_i) = |\check{r}_i - NN(\{\check{r}_j\}_{j=1}^N, \hat{r}_i)|$;
  end
end
{Estimating the intrinsic dimensionality}
$\hat{d} = \arg\min_{d \in \{1..D\}} \left(\log \frac{N}{N-1} + \frac{1}{N} \sum_{i=1}^N \log \frac{\check{\rho}(\hat{r}_i)}{\hat{\rho}_d(\hat{r}_i)}\right)$
```

Algorithm 2: Pseudocode for the base IDEA algorithm

```
Input:
    $\boldsymbol{X}_N$:  The dataset points $\{\boldsymbol{x}_i\}_{i=1}^{N}$.
    $k$:    The kNN parameter.

Output:

    $\hat{d}$:    The estimated intrinsic dimensionality.

{Compute the average normalized distance}
$\hat{m} = 0$;

for i:=1 to N do begin

    $\bar{\boldsymbol{X}}_{k+1} = kNN(\boldsymbol{X}_N, \boldsymbol{x}_i, k)$; {Finding the $k$ neighbors of $\boldsymbol{x}_i$.}
    $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_{k+1}(\boldsymbol{x}_i) \in \bar{\boldsymbol{X}}_{k+1}$; {The most distant point from $\boldsymbol{x}_i$.}
    $\hat{m} = \hat{m} + \frac{1}{Nk} \sum_{\boldsymbol{x} \in \boldsymbol{X}_k} \frac{\|\boldsymbol{x}_i - \boldsymbol{x}\|}{\|\hat{\boldsymbol{x}} - \boldsymbol{x}_i\|}$;

end

{Evaluate the intrinsic dimension $\hat{d}$}

$\hat{d} = \frac{\hat{m}}{1-\hat{m}}$;
```

References

Abate, A. F., Nappi, M., Riccio, D., & Sabatino, G. (2007). 2d and 3d face recognition: a survey. *Pattern Recognition Letters*, *28*, 1885–1906.

Baraniuk, R., & Wakin, R. (2006). Random projections of smooth manifolds. In *Foundations of computational mathematics*.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford: Oxford University Press.

Bishop, C. M. (1998). Bayesian PCA. *Advances in Neural Information Processing Systems*, *11*, 382–388.

Camastra, F. (2003). Data dimensionality estimation methods: a survey. *Pattern Recognition*, *36*(12), 2945–2954.

Camastra, F., & Filippone, M. (2009). A comparative evaluation of nonlinear dynamics methods for time series prediction. *Neural Computing & Applications*, *18*(8), 1021–1029.

Camastra, F., & Vinciarelli, A. (2002). Estimating the intrinsic dimension of data with a fractal-based method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 1404–1407.

Chen, L., Liao, H., Ko, M., Lin, J., & Yu, G. (2000). A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, *30*, 1713–1726.

Chen, J., Fang, H. R., & Saad, Y. (2009). Fast approximate kNN graph construction for high dimensional data via recursive Lanczos bisection. *Journal of Machine Learning Research*, *10*, 1989–2012.

Chua, L., Komuro, M., & Matsumoto, T. (1985). The double scroll. *IEEE Transactions on Circuits and Systems*, *32*, 797–818.

Clarkson, K. L. (2008). Tighter bounds for random projections of manifolds. In M. Teillaud (Ed.), *Symposium on computational geometry* (pp. 39–48). New York: ACM.

Coleman, T. F., & Li, Y. (1996). An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, *6*, 418–445.

Connor, M., & Kumar, P. (2010). Fast construction of k-nearest neighbor graphs for point clouds. *IEEE Transactions on Visualization and Computer Graphics*, *16*(4), 599–608.

Costa, J. A., & Hero, A. O. (2004). Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Transactions on Signal Processing*, *52*(8), 2210–2221.

Costa, J. A., & Hero, A. O. (2004). Learning intrinsic dimension and entropy of high-dimensional shape spaces. In *Proceedings of European signal processing conference (EUSIPCO)*.

Costa, J. A., & Hero, A. O. (2005). Learning intrinsic dimension and entropy of shapes. In H. Krim & T. Yezzi (Eds.), *Statistics and analysis of shapes*. Basel: Birkhäuser.

Eckmann, J. P., & Ruelle, D. (1992). Fundamental limitations for estimating dimensions and Lyapunov exponents in dynamical systems. *Physica D: Nonlinear Phenomena*, *56*(2–3), 185–187.

Farahmand, A. M., Szepesvari, C., & Audibert, J. Y. (2007). Manifold-adaptive dimension estimation. In *Proceedings of international conference on machine learning (ICML)*.

Fishman, G. S. (1996). *Monte Carlo: concepts, algorithms, and applications*. *Springer series in operations research*. New York: Springer.

Frank, A., & Asuncion, A. (2010). UCI machine learning repository.

Friedman, J. H. (1989). Regularized discriminant analysis. *Journal of the American Statistical Association*, *84*, 165–175.

Friedman, J. H., Hastie, T., & Tibshirani, R. (2009). *The elements of statistical learning—data mining, inference and prediction*. Berlin: Springer.

Fukunaga, K. (1971). An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers*, *20*, 176–183.

Fukunaga, K. (1982). Intrinsic dimensionality extraction. In P. R. Krishnaiah & L. N. Kanal (Eds.), *Classification, pattern recognition and reduction of dimensionality*. Amsterdam: North-Holland.

Grassberger, P., & Procaccia, I. (1983). Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, *9*, 189–208.

Guan, Y., & Dy, J. G. (2009). Sparse probabilistic principal component analysis. *Journal of Machine Learning Research—Proceedings Track*, *5*, 185–192.

Hegde, C., Wakin, M. B., & Baraniuk, R. G. (2007). Random projections for manifold learning. In *Advances in neural information processing systems*.

Hein, M. (2005). Intrinsic dimensionality estimation of submanifolds in Euclidean space. In *Proceedings of international conference on machine learning (ICML)* (pp. 289–296).

Jollife, I. T. (1961). *Adaptive control processes: a guided tour*. Princeton: Princeton University Press.

Jollife, I. T. (1986). *Principal component analysis*. *Springer series in statistics*. New York: Springer.

Kégl, B. (2002). Intrinsic dimension estimation using packing numbers. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Proceedings of neural information processing systems (NIPS)* (pp. 681–688). Cambridge: MIT Press.

Kirby, M. (1998). *Geometric data analysis: an empirical approach to dimensionality reduction and the study of patterns*. New York: Wiley.

Kivimäki, I., Lagus, K., Nieminen, I., Väyrynen, J., & Honkela, T. (2010). Using correlation dimension for analysing text data. In *Proceedings of the 20th international conference on Artificial neural networks: Part I (ICANN2010)* (pp. 368–373). Berlin: Springer.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Lee, J. A., & Veleysen, M. (2007). *Nonlinear dimensionality reduction*. *Springer series in information science and statistics*. Berlin: Springer.

Levina, E., & Bickel, P. J. (2005). Maximum likelihood estimation of intrinsic dimension. *Advances in Neural Information Processing Systems*, *17*(1), 777–784.

Li, J., & Tao, D. (2010). Simple exponential family PCA. In *Proceedings of international conference on artificial intelligence and statistics (AISTATS)* (pp. 453–460).

Lombardi, G., Casiraghi, E., & Campadelli, P. (2009). The neighbors voting algorithm and its applications. In *Studies in computational intelligence: Vol.* 245. *Applications of supervised and unsupervised ensemble methods (SUEMA)* (pp. 151–173).

Lombardi, G., Rozza, A., Ceruti, C., Casiraghi, E., & Campadelli, P. (2011). Minimum neighbor distance estimators of intrinsic dimension. In *Proceedings of European conference on machine learning (ECML)* (Vol. 6912, pp. 374–389).

MacKay, D. J. C. (1995). Probable networks and plausible predictions—a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, *6*(3), 469–505.

MacKay, D. J. C., & Ghahramani, Z. (2005). Comments on maximum likelihood estimation of intrinsic dimension by Elizaveta Levina and Peter Bickel. http://www.inference.phy.cam.ac.uk/mackay/dimension/.

Mordohai, P., & Medioni, G. (2010). Dimensionality estimation, manifold learning and function approximation using tensor voting. *Journal of Machine Learning Research*, *11*, 411–450.

Ott, E. (1993). *Chaos in dynamical systems*. Cambridge: Cambridge University Press.

Paredes, R., Chávez, E., Figueroa, K., & Navarro, G. (2006). Practical construction of k-nearest neighbor graphs in metric spaces. In C. Àlvarez & M. J. Serna (Eds.), *Lecture notes in computer science: Vol.* 4007. *WEA* (pp. 85–97). Berlin: Springer.

Pettis, K., Bailey, T., Jain, A., & Dubes, R. (1979). An intrinsic dimensionality estimator from near-neighbor information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *1*(1), 25–37.

Pineda, F. J., & Sommerer, J. C. (1994). Estimating generalized dimensions and choosing time delays: a fast algorithm. In *Time series prediction. Forecasting the future and understanding the past* (pp. 367–385).

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*, 2323–2326.

Rozza, A., Lombardi, G., Re, M., Casiraghi, E., & Valentini, G. (2010). DDAG K-TIPCAC: an ensemble method for protein subcellular localization. In *Proceedings of European conference on machine learning PKDD—(SUEMA) workshop* (pp. 75–84).

Rozza, A., Lombardi, G., Rosa, M., & Casiraghi, E. (2010). O-IPCAC and its application to EEG classification. In *Proceedings of workshop on applications of pattern analysis (WAPA)* (pp. 4–11).

Rozza, A., Lombardi, G., Casiraghi, E., & Campadelli, P. (2012). Novel Fisher discriminant classifiers. *Pattern Recognition* (in press).

Rozza, A., Lombardi, G., Rosa, M., Casiraghi, E., & Campadelli, P. (2011). IDEA: intrinsic dimension estimation algorithm. In *Proceedings of international conference on image analysis and processing (ICIAP)* (Vol. 6978, pp. 433–442).

Tenenbaum, J. B., Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*, 2319–2323.

Tipping, M. E., & Bishop, C. M. (1997). Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B*, *61*(Part 3), 611–622.

Vapnik, V. (1998). *Statistical learning theory*. New York: Wiley.

Verveer, P. J., & Duin, R. P. W. (1995). An evaluation of intrinsic dimensionality estimators. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*, 81–86.

Wang, Q., Kulkarni, S. R., & Verdú, S. (2006). A nearest-neighbor approach to estimating divergence between continuous random vector. In *IEEE international symposium on information theory (ISIT2006)* (pp. 242–246).

Zou, H., Hastie, T., & Tibshirani, R. (2004). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, *15*, 262–286.